

GPU-Accelerated Lattice Boltzmann Method for Complex Free Surface Flows

李, 旭暉

<https://doi.org/10.15017/1785434>

出版情報：九州大学, 2016, 博士（工学）, 課程博士
バージョン：
権利関係：全文ファイル公表済



GPU-Accelerated Lattice Boltzmann Method for Complex Free Surface Flows

GPU を用いた格子ボルツマン法による複雑自由表面流れの数値解析

Thesis by

李 旭暉

Li Xuhui

リー シュフィ

submitted to

the Faculty of Interdisciplinary Graduate School of Engineering Sciences of

Kyushu University

in partial fulfillment of the requirements for

the degree of Doctor of Philosophy in

Earth System Science and Technology

August 2016

In memory of my father

Abstract

In the historical development of the fluid dynamics in ocean engineering applications, three branches have been going ahead together: experimental fluid dynamics (EFD), potential flow theory and the computational fluid dynamics (CFD). In the EFD, the ship model and platform model experiments can provide reliable and accurate results for the different stage design, parameters optimization and risk assessment while the huge expenditure, long period and the scale-effect prevent it from convenient and low-cost execution. The inviscid potential flow theory based on the boundary element method (BEM) using the complex Green function or Rankine source on the body surface, free surface with some far-field boundary condition plays pivotal role in the linear physical problems and some nonlinear physical problems but cannot be capable of most of the full nonlinear problems. The traditional CFD tools such as finite difference method (FDM), finite volume method (FVM) have fulfilled remarkable achievements in the ocean engineering applications. Unfortunately, due to the time-consuming property of the traditional CFD tool, researchers have been making efforts to develop an efficient alternative CFD tool. The well-suited and intrinsic parallelism of the lattice Boltzmann method (LBM) and the coming of the era of GPU parallel computation make it possible to develop an extremely efficient and even real-time numerical simulation tool.

As a young CFD branch, LBM has not yet practically applied into complicated free surface problems in the ocean engineering field. The work described in the present thesis is devoted to investigate the potential of the lattice Boltzmann method in the ocean engineering application. It is well known that wave-body interaction, a typical ocean engineering problem, such as ship and platform motion in waves, usually involves moving boundary issue and complicated free surface phenomena. The present work is devoted to handle these two topics. For the first issue, the difficulty is the spurious pressure oscillation in the treatment of moving fluid-solid boundary, which is systematically investigated in this study. To be first, the fundamental collision model in the lattice Boltzmann method causes intrinsic pressure fluctuation. At the meantime, it is found that the re-initialization algorithm for the fresh fluid

node that just emerged from the solid region remarkably affects the oscillation of the force exerted on the moving body immersed in the fluid domain. Moreover, the force evaluation approach (such as various momentum exchange methods, stress integration method, immersed boundary method) is also relevant to the pressure calculation accuracy and the oscillation of the force on the moving body. To improve the accuracy of force evaluation of the moving boundary, a bi-quadratic ghost fluid immersed boundary method has been proposed. To suppress the pressure oscillation issue, a new refilling scheme for the fresh fluid node has been developed. For the second topic, two approaches have been explored to model the violent free surface flow: single-phase free surface model and the high-density ratio multiphase model. To develop a stable, accurate and efficient free surface capture algorithm is a key challenge for both of them. In the present study, a VOF (volume-of-fluid) type algorithm is applied to capture the free surface in the single-phase free surface model which ignores the gas phase and models the atmosphere pressure as dynamic boundary condition on the free surface. The classical engineering applications such as dam breaking, dam breaking with obstacle, water entry and exit of circular cylinder with constant speed, water entry of circular cylinder with free falling have been successfully simulated. Meanwhile, the multiphase lattice Boltzmann models which are capable of high-density ratio and high Reynold number have been also investigated. The mass conservative phase-field interface tracking equation has been applied to treat the interfacial dynamics. The state-of-art mass-conservative multiphase lattice Boltzmann model developed by Abbas & Martin & Lee has been applied in this thesis. The bubble dynamics including single bubble behavior, bubble-bubble interactions and vortex ring bubble have been investigated deeply. To efficiently implement the numerical code of different LB solvers, the fashionable GPU accelerated technique has been realized and equipped to the present programs. In order to achieve the large-scale three-dimensional numerical simulation, the multi-GPU technique has been also developed due to the limited device memory on single GPU card. The accelerated program has fulfilled the amazing speed-up ratio of two orders of magnitude, which dramatically decreases the simulation time and improve the research efficiency.

Keywords: Lattice Boltzmann model; GPU programming; Moving boundary; Free surface flow; Water entry and exit; Multiphase flow; Bubble dynamics

Acknowledgments

This dissertation would not be possible without the help and contributions of many kind-hearted people who are gratefully acknowledged here. I would like to give earnest thanks and best wishes to them.

First and foremost, I would like to warmly acknowledge the financial support of China Scholarship Council (CSC) for my PhD research and the guidance of my life abroad.

I am honorable to express my deepest gratitude to my dedicated supervisor, Professor Changhong Hu, for his great guidance, inspirations, and encouragements in my doctoral research work. Professor Hu has extensive and fruitful experience and attainment in Computational Fluid Dynamics, and I feel it a great honor and pleasure to do research with him. I am also grateful for his support and encouragement for me to attend various conferences that broaden my research horizon. Without his kindly support, this dissertation could not have been finished.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Yuji Ohya and Prof. Shigeo Yoshida.

I would like to express my great appreciation to Prof. Luo Lishi of Old Dominion University, who is an outstanding and distinguished scholar in the field of lattice Boltzmann method. When I visited his laboratory, he was always pushing me to understand the basic theory and physical concepts clearly. Especially he gave me many explanations about the multiple-relaxation-time collision model and the multiphase lattice Boltzmann model. At the meantime, many thanks go to Dr. Weidong Li and Dr. Wei Li for the useful discussions and the happy time we enjoyed when I stayed in USA.

I would like to express my great thanks to Professor Takayuki Aoki, Dr. Onodera, Dr. Un-Hong Wong of Tokyo Institute of Technology. When I visited their research group, they gave me so much valuable guidance and shared me so much experience in GPU computing. Without their kindly help, I would not be able to finish the GPU code of lattice Boltzmann solver smoothly.

I would like to express my great thanks to Dr. Christian F. Janßen, who is a very handsome and promising young scholar I got to know in the 12th International Conference for

Mesosopic Methods in Engineering and Science. He gave me various ideas and tricks in terms of the free surface flow solver through many emails. I would like to express great thanks to Dr. Fakhari Abbas of University of Notre Dame, who gave me kind help in the multiphase lattice Boltzmann model. I would like to express my great thanks to Prof. Chang Shu and Dr. Yan Wang of National University of Singapore for the valuable discussions on interfacial dynamics.

Also I am grateful for continuous discussions on single-phase free surface model and the GPU programming and the constructive advice on the publication to Dr. Fei Jiang, who has been doing related research on the lattice Boltzmann method for many years. Sincere thanks to all members in Professor Hu's group. Dr. Makoto Sueyoshi is always nice to give me some advice on the post-analysis of the numerical results and many supports in dealing with the computer hardware troubles. Ms. Masako Yoshizu and Ms. Sakura Kawasaki are always kind-hearted to help me prepare the documents in doctoral course and internship. Special thanks go to Dr. Kangping Liao, Dr. Cheng Liu, Dr. Yingyi Liu for the valuable discussions on my research and the wonderful life we enjoyed. Thanks go to Mr. Tarek, who is a humorous man to cheer me up when I was down. Thanks go to Ms. Omori, who helped me to take care of my plant when I was absent of the laboratory. Thanks go to Mr. Nik and Mr. Mohamond for their valuable discussions on my work and job hunting. Thanks go to Mr. Guoling Li for the enjoyable time we spend together during the lonely and tough research life in Japan.

I would like to express my great thanks to Prof. Wenyang Duan and Dr. Guodong Xu of Harbin Engineering University for the discussions on research philosophy, attitude and water entry problem in chapter 5.

At last but not the least, I would like to thank my mother and sister for their understandings, encouragements and forever love.

This dissertation is dedicated to all those who have contributed to make this thesis possible and better

Contents

Abstract.....	I
Acknowledgments.....	IV
Contents	VI
Chapter 1 Introduction.....	1
1.1 Background and motivation	1
1.2 GPU implementation of lattice Boltzmann solver.....	3
1.3 Moving boundary	4
1.4 Single-phase free surface flow	7
1.5 Multiphase lattice Boltzmann model	8
1.6 Outline of the thesis.....	10
Chapter 2 Basic theory of lattice Boltzmann equation.....	14
2.1 Boltzmann Equation	14
2.2 Lattice Boltzmann Equation	16
2.3 Body force	23
2.4 Initial Condition	23
2.5 Boundary Condition	25
2.5.1 Periodic boundary condition.....	25
2.5.2 Symmetric boundary condition	26
2.5.3 Open boundary condition	26
2.5.4 Half-way bounce back scheme	27
2.5.5 Non-equilibrium extrapolation scheme	28
2.5.6 Zou-He boundary condition	30
2.5.7 Immersed boundary method	31
2.6 Turbulence Model	35
2.7 Unit Transformation	36
2.8 Benchmark Validation	40
2.8.1 2D Cavity Flow	40

2.8.2 2D Taylor-Green Vortex	43
2.8.3 Eccentric circular cylinder in a channel	46
2.9 Concluding Remarks	47
Chapter 3 GPU implementation of the lattice Boltzmann model	48
3.1 Introduction	48
3.2 Overview of the CUDA programming platform.....	48
3.2.1 Overview of the hardware	48
3.2.2 Programming in CUDA.....	51
3.3 CUDA implementation in the lattice Boltzmann solver	55
3.3.1 Corresponding relationship between threads and the fluid nodes	55
3.3.2 Implementation examples.....	61
3.4 Multi-GPU implementation of the lattice Boltzmann solver.....	66
3.4.1 Peer-to-Peer technique.....	66
3.4.2 CUDA with MPI.....	67
3.5 Concluding Remarks	69
Chapter 4 Moving boundary in the fluid-solid interactions	71
4.1 Numerical schemes for moving boundaries	71
4.1.1 Extrapolation scheme by Guo et al.(2002)	71
4.1.2 Interpolation bounce-back schemes.....	72
4.1.3 Ghost fluid method.....	73
4.2 Refilling of the fresh node.....	76
4.3 Force evaluation methods.....	78
4.4 Numerical simulation	80
4.4.1 Eccentric cylinder	80
4.4.2 Cylindrical Couette Flow	83
4.4.3 Impulsively started cylinder in a channel	86
4.4.4 Oscillation cylinder in a channel	93
4.4.5 Particle suspension	97
4.5 Concluding Remarks	103
Chapter 5 Free Surface Model and Its Applications	105
5.1 Free surface flow with floating bodies	105

5.1.1 Treatment of the free surface	105
5.1.2 Conversions of the liquid-gas-solid system.....	110
5.2 Numerical benchmarks.....	111
5.2.1 Dam breaking	111
5.2.2 Dam breaking with obstacle	113
5.3 Water entry with constant speed.....	115
5.3.1 Convergence Test	116
5.3.2 Numerical simulation vs the experiment result	120
5.4 Water exit with constant speed	122
5.5 Water entry with free fall.....	124
5.5.1 Neutrally buoyant free fall cylinder.....	124
5.5.2 Half-buoyant free fall cylinder	126
5.6 Concluding Remarks	128
Chapter 6 Multiphase Lattice Boltzmann Model and Its Applications	130
6.1 Theory of multiphase models	130
6.1.1 He-Cheng-Zhang two-phase model (1999)	130
6.1.2 Lee-Lin two phase model (2005).....	134
6.1.3 Lee-Liu two-phase model (2010)	136
6.1.4 Abbas-Martin-Lee two-phase model (2016).....	139
6.2 Numerical verification.....	144
6.2.1 Circular interface in a shear flow	144
6.2.2 Deformation of a circular interface	145
6.2.3 2D Rayleigh-Taylor instability	146
6.2.4 Drop splashing on a thin film	147
6.2.5 3D Laplace Law	149
6.3. Bubble dynamics	151
6.3.1 Single bubble	151
6.3.2 Bubble interaction	153
6.3.3 Motion of Bubble Ring.....	160
6.4 Concluding Remarks	162
Chapter 7 Conclusions.....	163

Appendix A	166
From lattice Boltzmann equation to Navier-Stokes Equation	166
Appendix B:	172
Matrix M for D2Q9 MRT model	172
Matrix M for D3Q19 MRT model	172
Appendix C	173
Derivation of the Body Force	173
Bibliography	180

Chapter 1 Introduction

1.1 Background and motivation

In the historical development of fluid dynamics, three branches have been going ahead: experimental fluid dynamics (EFD), potential flow and the computational fluid dynamics (CFD). Each one is of indispensable significance for the insight of the underlying mechanism of fluid phenomena. In the field of naval architecture and ocean engineering, the incompressible free surface flow is of crucial importance to be resolved, as shown in Fig. 1.1.



Fig.1.1 Free surface flow: left: wind turbine on the floating platform; right: wave-making of the ship (From the internet)

To solve the wave propagation, wave impact, wave-making resistance, wave induced added resistance, propulsion of propeller, green water, sea-keeping performance, maneuvering flexibility, parameter roll etc. in the ocean engineering, the ship model and platform model experiments can provide reliable and accurate results for the different stage design, parameters optimization and risk assessment. However, the huge expenditure, long period and the scale-effect prevent it from convenient and low-cost execution. The inviscid potential flow theory based on the boundary element method (BEM) using the complex Green function or Rankine source on the body surface and free surface with some far-field boundary condition plays pivotal role in the linear physical problems and some nonlinear physical problems in the ocean engineering applications. Unfortunately, for most full nonlinear problems, free surface breaking and merge, stability of the broken vessel, cavitation of the propeller or water entry, frictional resistance, resistance reduced by the bubble, turbulence flow etc., they are beyond the capability of potential theory. Computational fluid dynamics, as an interdisciplinary concerning

fluid dynamics, mathematics and computer science, is a powerful and versatile tool. With the development of the hardware of the modern computer, the high resolution and large scale numerical simulation has been becoming more and more convenient, efficient and available in the desktop. The finite difference method (FDM) (C. Hu et al. 2004, C.Hu et al. 2006, C.Hu et al. 2009), finite volume method (FVM) (R. Eymard, R. Herbin, 2000), finite element method (FEM) (Rainald Löhner et al.,1999), smooth particle hydrodynamics (SPH) (Lucy L.B., 1977) and moving particle simulation (MPS) (Koshizuka S. et al., 1996) have fulfilled remarkable achievements in the ocean engineering applications. However, attributing to the iterative solution of the pressure Poisson equation and the irregular memory access, the numerical simulations of these CFD tools are extremely time-consuming even the supercomputer is utilized. Therefore, to develop an efficient, robust, stable and accurate CFD tool is an impending task for the scholars in the CFD field.

In the past two decades, lattice Boltzmann method, which is a special discrete form of the continuous Boltzmann equation, has been growing to a mature and versatile numerical branch. The lattice Boltzmann method has attracted increasing popularity in simulating various complex fluid flows because of its implementation simplicity, excellent execution efficiency and intrinsic parallelism. The troublesome nonlinear advection term in the Navier-Stokes equation is extreme easily executed by the perfect linear stream operation in the lattice Boltzmann method, which is accurate to the machine precision. The collision term that is identical to the diffusion term in the traditional CFD braches is completely local, which is intrinsically for the parallel computation. Moreover, the lattice Boltzmann method is a strict second-order solver for both temporal and spatial convergence and free from the pressure Poisson equation. However, the lattice Boltzmann method has not achieved considerable success in the ocean engineering applications for some difficulties. To be first, the high density and high viscosity ratio is a non-trivial issue for the lattice Boltzmann model, especially the numerical stability; Second, the high Reynolds number flow, i.e. the turbulence flow, which is a challenge for the lattice Boltzmann model even the multiple-relaxation-time collision model is performed; Third, the accuracy of the pressure field is a rigorous issue for this transient flow solver, especially at the high Reynolds flow regime and the violent physical problems; Fourth, in the gas-liquid-solid three phase flow, the re-initialization of the fresh fluid nodes emerged

from the solid region is intractable if the concise and accurate interpolation bounce back scheme is applied in the complex moving boundary problems; Fifth, the GPU implementation of the lattice Boltzmann solver needs abundant knowledge of the hardware of modern computer and programming skills. In this thesis, many efforts have been made to overcome these difficulties. In order to make full use of the intrinsic parallelism of the lattice Boltzmann solver, the multi-GPU technique has been developed to accelerate the numerical simulation. The accuracy and the pressure oscillation in the complex moving fluid-solid boundary have been deeply investigated and enhanced schemes have been proposed to suppress the pressure oscillation and improve the accuracy of the force evaluation. To be capable of the high density and high viscosity ratio free surface flow, two approaches have been applied in this thesis. One is the single-phase free surface flow model that neglects the gas phase but models the atmosphere pressure as a dynamic boundary condition. The Volume-Of-Fluid interface capture algorithm is performed to handle the free surface. The other approach is the high density ratio multiphase lattice Boltzmann model that is not completely mature for the high density ratio and high Reynolds number but is more promising than the first one.

The research motivation in this thesis is to investigate the potential of the lattice Boltzmann method in the ocean engineering applications and the possibility of the extreme efficient alternative CFD tool based on the lattice Boltzmann method.

1.2 GPU implementation of lattice Boltzmann solver

During the past decade, the applications of the General Purpose Graphics Processor Units (GPU) in high performance computation have fulfilled explosive achievements. With the features of low power consumption, high memory-bandwidth and intensive computational capability, GPU technique has been widely applied in both the industry field and the scientific calculation, such as oil and gas exploration, finance analysis, medical imaging, genetic analysis, computational fluid dynamics. Benefit from the application programming interface which is so-called Compute Unified Device Architecture (CUDA) released by NVIDIA, the parallel programming has become extremely convenient and direct for the programmers to perform general purpose calculation. In the community of computational fluid dynamics, GPU acceleration technique has been quickly applied in the molecular dynamics, smooth particle

hydrodynamics, lattice Boltzmann method and magnetohydrodynamics.

In the GPU implementation of the lattice Boltzmann method, there exist many great pioneer works to be addressed. Jonas Tölke (2007) obtained more than one order in performance gain compared to standard CPUs for the lattice Boltzmann implementation of porous medium. Peter Bailey et al. (2009) introduced a space-efficient storage method which reduces GPU RAM requirements by 50% at a slight detriment to performance based on the D3Q19 lattice Boltzmann model. Christian Janssen et al. (2011) conducted the GPU implementation of the free surface flow simulations with a VOF based algorithm. Christian Obrecht et al. (2011) proposed a novel data transfer schemes in global memory access to improve the frequent misaligned memory accesses in the propagation step of the lattice Boltzmann method. Wang Xian & Aoki Takayuki (2011) investigated the multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster. Naoyuki Onodera et al. (2013) conducted the large-eddy simulation of fluid-structure interaction using lattice Boltzmann method on multi-GPU clusters. Naoyuki Onodera & Takayuki Aoki (2014) conducted large-eddy simulation of turbulent flow around a car body using lattice Boltzmann method on the TSUBAME supercomputer. Naoyuki Onodera and Takayuki Aoki (2014) conducted peta-scale large-eddy simulation using lattice Boltzmann method on the TSUBAME supercomputer. Naoyuki Onodera et al. (2015) conducted the GPU implementation of lattice Boltzmann method for free-surface flow.

In our GPU implementation of the lattice Boltzmann solver, the advanced GPU direct-MPI hybrid framework is firstly applied to the moving boundary lattice Boltzmann solver (Chapter 4), the single-phase free surface lattice Boltzmann solver (Chapter 5) and the multiphase lattice Boltzmann model (Chapter 6).

1.3 Moving boundary

The simple Cartesian mesh and the absence of solving the pressure Poisson equation make it a powerful and efficient tool in dealing with moving boundaries and fluid-structure interaction problems. The most widely used model for boundary treatment is the bounce-back (BB) method (e.g. the shell model by Ladd et al.,1994, the ALD (Aidun, Lu and Ding) by Aidun et al.,1998). During the past two decades, in order to overcome the uncertainty in boundary

capturing of the above midway bounce-back schemes, several interpolation or extrapolation based curved boundary schemes (O. Filippova and D. Hanel , 1998, Mei et al.,1999, M. Bouzidi et al.,2001, Mei et al., 2002, Z. Guo et al., 2002, P. Lallemand & Luo LS, 2003) have been proposed in which the accurate position of the boundary can be captured so that second-order accuracy has been achieved. These schemes treat the flow values on the fluid adjacent solid nodes by interpolation/extrapolation along the lattice-link direction regardless of the normal direction of the boundary.

Because of the common property by using Cartesian grid, the immersed boundary (IB) method which was firstly proposed by Peskin (1972) to investigate the biological fluid problems has been naturally incorporated into the lattice Boltzmann method. The LBM-IBs can be basically divided into two categories: diffused interface IBs and sharp interface IBs. Feng et al. (2004) constructed the first LBM-IB, in which the free parameter in the feedback forcing scheme introduced artificial arbitrariness. Subsequently, Feng et al. (2005) developed a direct-forcing LBM-IB using the Navier-Stokes equation to calculate the boundary force, which spoiled the merits of LBM. Niu et al. (2006) developed an explicit diffused interface scheme with the momentum exchange method calculating the boundary force, whereas streamlines penetrate the boundary easily. Alexandre Dupuis et al. (2008) analysed diffused interface IB with direct-forcing and interpolated-forcing techniques. Several correction methods have been proposed as well, such as velocity-correction methods to guarantee the non-penetrating boundary condition, as Shu et al. (2007) and Wu et al.(2009) and the split-forcing iterative method, as Kang et al. (2011). Le and Zhang (2009) demonstrated the existence of boundary slip in traditional diffused interface IBs through both theoretical derivation and numerical simulation with single-relaxation-time (SRT) model that revealed severe deviation between the interpolated velocity and the desired velocity on the boundary, especially for large relaxation parameter τ . Inspired by the work of Le et al.(2009), Lu et al.(2012) extended their work to the multiple-relaxation-time (MRT) LBM model with specific relationship between these relaxation time parameters and reduced the numerical boundary slip remarkably. Takeshi Seta et al.(2014) utilized the two-relaxation-time (TRT) LBM model to eliminate the boundary slip, and demonstrated that the error of the boundary velocity is essentially caused by the smoothing technique with the δ function used in the interpolation

process from flow field velocity to boundary velocity and by the distribution of boundary force to flow field. In the above mentioned diffused interface immersed boundary methods, it is not necessary to refill the fresh nodes which just emerge from solid region into fluid region, however, all of them require extra correction to guarantee the non-slip boundary condition.

Another category is the sharp interface immersed boundary, which is a robust and effective method to satisfy the Dirichlet and Neumann boundary conditions for velocity and pressure. Kang et al. (2011) developed a direct-forcing LBM-IB in which linear and bilinear interpolation are used to obtain the flow values on the ghost nodes (solid nodes with at least one neighboring fluid node). Tiwari et al. (2012) proposed another sharp interface LBM-IB scheme with bilinear interpolation to obtain the flow values on the ghost fluid nodes (BGFM). Although the non-slip boundary condition can be strictly guaranteed in the sharp interface IBs, the pressure-oscillation issue emerges. To overcome this problematical issue, Li Chen et al. (2013) combined the partially saturated computational (PSC) cell method of Noble et al. (1998) with the bilinear ghost fluid method of Tiwari et al. (2012). However, the inaccuracy of PSC scheme has also been inherited into this scheme. In the present work, the accuracies of different interpolation/extrapolation approaches of both bounce-back type schemes and sharp-interface IBMs are firstly compared, the intrinsic reasons of pressure oscillation are then deeply investigated.

For either classic bounce-back schemes or sharp interface LBM-IBs, the reconstruction treatment of particle distribution function (PDF) on the ghost nodes is important. The former methods interpolate the flow values along the grid line, whereas the latter methods interpolate the flow values usually through the mirror point of ghost node along the local normal vector direction of boundary. Inspired by the work of Yu-Heng Tseng et al.(2003) and Cheng Liu et al.(2014), in this study a bi-quadratic ghost fluid scheme is introduced to LBM as a high-order sharp interface LBM-IB (QGFM). Meanwhile, Guo's second-order extrapolation scheme (Guo et al., 2002), interpolation bounce-back scheme (M. Bouzidi et al.,2001, P. Lallemand & Luo LS, 2003) and Tiwari et al.'s (2012) bilinear ghost fluid scheme will be assessed to investigate their spatial accuracies. Refilling routine for the fresh nodes emerged from the solid phase is significant as well for the treatment of moving boundaries. The equilibrium refilling method and second-order extrapolation refilling method have been proposed by P. Lallemand and L.S.

Luo (2003). A. Caiazzo (2008) applied the asymptotic analysis to investigate this problem and concluded that there should be a high-order velocity gradient term in the refilling of distribution functions. S. Krithivasan et al.(2014) applied Grad's type distribution function (1949) to refill the fresh nodes, which concerns complex pressure tensor term. In the present work, a bilinear interpolation method is applied to obtain the non-equilibrium PDF as well as the pressure and velocity that are used to calculate the equilibrium PDF as a compromise method for accuracy and complexity. With regard to the force evaluation, the Galilean invariant momentum exchange method proposed by B. Wen et al.(2014) which is one of the various corrected momentum exchange methods (A. Caiazzo and M. Junk,2008, Eric Lorenz et al.,2009, J.R.Clausen & C.K.Aidun,2009, G.F.Zhou,2011, B. Wen et al.,2012, Y. Chen et al.,2013, B.Wen et al.,2014), stress integration method proposed by Inamuro et al.(2000) have been compared to assess their accuracies and effects of suppressing pressure oscillation in this study. Meanwhile, it has also been found that the collision models have remarkable effects on suppressing the pressure oscillation in fluid-solid interaction issue.

1.4 Single-phase free surface flow

In the past decade, some scholars have been making efforts in developing single-phase lattice Boltzmann model to capture the interface in the field of metal foam and free surface flow. I. Ginzburg & K. Steiner (2003) developed a generalized lattice Boltzmann model to simulate free surface with an anti-diffusion algorithm to maintain the sharp interface. The unknown distribution functions on the interfacial node are reconstructed based on the first-order Chapman-Enskog analysis. N. Thurey & U. Rude (2004) investigate the level set method coupled with the lattice Boltzmann equation. C. Korner et al. (2005) proposed a lattice Boltzmann model for free surface flow for modelling foaming. In their model, the fluid dynamic boundary conditions are approximated by using the mass and momentum fluxed across the interface while a similar procedure is applied to fulfill the diffusion boundary condition. N.Thurey et al. (2005) used adaptive parameterization to optimize and stabilize the free surface flow. Christian Janssen & Manfred Krafczyk (2010) proposed a non-uniform block-structured mesh based free surface lattice Boltzmann model. In their model, the additional advection equation for the volume-of-fluid fill level is discretized by a classical finite volume method and a piecewise linear interface reconstruction is conducted. Christian

Janssen et al.(2013) presented some extensive validations and applications of free surface flow problems by using the hybrid volume-of-fluid model. Zhuangming Zhao et al.(2013) proposed a new method based on the combination of the lattice Boltzmann equation and the kinematic boundary condition to simulate the viscous free surface wave. In their work, the standing wave and Stokes wave of small amplitude are successfully simulated. Simon Bogner & Ulrich Rude (2013) developed a liquid-gas-solid lattice Boltzmann model to simulate the floating bodies on the free surface. Unfortunately, their model is too rough for the practical applications for the following reasons: 1) the fluid-solid interface is not accurately captured and the stair-case approximation and half-way bounce-back scheme in the particle flow is applied; 2) the refilling scheme for the fresh nodes emerged from the obstacle is extremely rough, in which the pressure oscillation is considerably remarkable; 3) the force evaluation scheme exerted on the obstacle has been demonstrated to be incorrect (B. Wen, 2014); 4) the collision model used in their model is single-relaxation-time model, which has also been demonstrated that the pressure noise is too excessive (Luo et al., 2010), while the multiple-relaxation-time collision model can effectively suppress the pressure oscillation due to the dissipation effect of the energy mode.

In the present work, an efficient and more advanced lattice Boltzmann solver overcoming the drawbacks in the solver of Simon Bogner et al. (2013) is proposed to resolve the wave-body interaction issues in ocean engineering, such as wave impact on structure, water entry and exit.

1.5 Multiphase lattice Boltzmann model

Multiphase flows are ubiquitous in nature and of great significance for both scientific and engineering applications (G. Thömmes et al.,2009, C.G.Koh,2013, Luca Scarbolo et al.,2013). The numerical simulation of the multiphase/multi-component fluid flows has been a challenging problem in Computational Fluid Dynamics (CFD). Of various kinds of CFD branches, the approaches to deal with the interface can be divided into two categories: interface capturing method such as the widely used Volume-Of-Fluid (VOF) method and interface tracking method such as the front-tracking method. In terms of the interface tracking method, the grid line should be aligned with the interface or the interface is explicitly tracked by a set of

markers which are transported by the flow. Most of the interface-tracking methods define a thin sharp interface between different phases, which needs complicated calculation of interface curvature to incorporate the surface tension. On the other hand, the interface capturing methods apply a scalar field to implicitly identify the interface. The distance signed method such as Level-Set method and the volume fraction function based VOF are two typical interface capturing methods.

For the various lattice Boltzmann multiphase/multi-component models, they can be classified into four representative models. (a) Color-Gradient model (CG): this model is proposed by Gunstensen et al. (1991), in which the intermolecular force is calculated by the local color gradient associated with the density difference of the binary fluids. The recolor step was improved remarkably by Latva-Kokko & Rothman (2005) to reduce the spurious current across the interface and the lattice pinning effect. This model is hard to handle the high-density ratio multiphase flow and the recoloring step is a little bit time-consuming; (b) Shan-Chen model (SC): By introducing the inter-particle interaction forces, the model developed by Shan and Chen (1993) can automatically generate the interface by modifying the surface tension related terms in the collision operators using the inter-particle potentials. Although it cannot satisfy the local momentum conservation law (Shan & Chen 1993), the SC LB model has attracted much popularity due to its overall accuracy and ease for implementation; (c) Free energy model: M. Swift et al. (1995) proposed a thermodynamically consistent multiphase/multi-component lattice Boltzmann model from the theory of free energy, however, this model suffers from the non-Galilean invariance for some extra terms. Inamuro et al. (2004) developed an improved two-phase model through solving the Poisson equation to handle the pressure field. This model is capable of high-density ratio as high as 1000. However, the computation load of Poisson equation solver based on LBE is extremely heavy that hampers its widely applications; (d) Kinetic theory based discrete Boltzmann equation: The pioneer works of this kind of model were conducted by Luo et al. (1998), He & Shan & Doolen (1998), He & Chen & Zhang (1999). He-Chen-Zhang model is popularly used in the engineering applications in various fields for its versatile properties. The mass and momentum equations are transformed into the pressure and momentum equations. A reconstructed particle distribution function is applied to narrow the numerical error of the phase-field gradient across the interface. However,

Lee & Lin (2003) pointed out that the pressure evolution equation in this model is an ideal gas pressure evolution LBE and is demonstrated to be only valid for ideal gas flow in the incompressible limit. Lee & Lin (2005) developed a high-density ratio two-phase model through the stable discretization schemes of the intermolecular force and reasonable pressure updating scheme. However, it involves the switch of discretization scheme of intermolecular force at different stage of the LBE evolution process. The reconstructed LBEs have been developed by Lee & Liu (2010) and only the potential form is used to deal with the intermolecular force. However, this model involves the finite difference discretization of at least three macroscopic variables and the fourth-order derivatives of the phase-field. Abbas & Martin & Lee (2016) proposed a mass-conserving two-phase model with dynamic grid refinement technique. In this model, a conservative phase-field model has been applied and the cumbersome finite difference treatments have been avoided due to the continuum surface tension form.

In the present work, Abbas-Martin-Lee model is applied to simulate the bubble dynamics, including the single bubble rising, bubble-bubble interaction, bubble flow and vortex ring bubble.

1.6 Outline of the thesis

Many efforts have been made in this thesis to investigate the potential of the lattice Boltzmann method in the ocean engineering applications. To achieve this research goal, the theoretical study, scheme development, parallel computation and large-scale numerical simulations are conducted in the present work. The basic collision model including single-relaxation-time collision model, the multiple-relaxation-time collision model and the two-relaxation-time model are applied in the present study. The advanced external force scheme is applied in the lattice Boltzmann simulations of the moving boundary, free surface flow. The initial condition and various boundary conditions are utilized through the whole thesis. In order to handle the high-Reynolds number flow in the complicated free surface flow, the LES model has been incorporated in the present lattice Boltzmann model. To make the corresponding relationship between the physical system and the lattice system clear, the unit transformation is elaborated through a numerical example. For the large-scale

Chapter 1 Introduction

three-dimensional numerical simulations, the lattice Boltzmann solvers are accelerated by the multi-GPU technique, which is demonstrated to be two-orders of magnitude speed-up ratio compared with the series code. The lattice Boltzmann solver, as a transient explicit flow solver, the pressure fluctuation is considerable, especially the spurious pressure oscillation. The underlying mechanism of the pressure oscillation has been investigated deeply. The possible sources including collision models, moving boundary schemes, refilling schemes for re-initialization of the fresh nodes emerged from the solid region to fluid domain, the force evaluation schemes have been examined in detail. Based on the previous GPU acceleration technique and the mechanism analysis of the pressure oscillation, the violent free surface flows have been simulated successfully. The extreme violent dam-breaking, water entry and water exit physical phenomena have been simulated and compared with the experiment results and the results of FDM solver. To break through the limitation of the single-phase free surface flow model, such as the cavitation in the water entry simulation, the multiphase lattice Boltzmann model has been developed. At the first stage, the bubble dynamics including single bubble, bubble-bubble interactions, bubble flow and the vertex ring bubble has been simulated successfully and compared with the experiment results and other CFD tool. All these numerical studies demonstrated that our developed GPU accelerated LBM solver is versatile and powerful in terms of simulating the physical problems in ocean engineering and has great potential capability in modeling violent free surface flows. The research project has achieved the initial success in terms of the application of lattice Boltzmann solver into the ocean engineering.

The framework of this thesis is organized as followings:

In chapter 1, the background of the present research is described. The state of arts in the GPU implementation of LBE solver, the moving boundary, single-phase free surface flow and the high-density ratio multiphase lattice Boltzmann solver have been reviewed in detail. Finally, the motivation, achievement and outline of this thesis are summarized.

In Chapter 2, the basic theoretical framework is elaborated. Firstly, the underlying theoretical relationship of the Boltzmann equation and the lattice Boltzmann equation has been stated. Then the body force scheme in the lattice Boltzmann equation has been described. To be followed, the initial condition and various boundary conditions including the periodic boundary

condition, symmetric boundary condition, open boundary condition, half-way non-slip and free slip boundary condition, non-equilibrium extrapolation scheme, Zou-He boundary condition, immersed boundary method are depicted carefully. Subsequently, the widely-used turbulence model—large eddy simulation has been introduced in the present lattice Boltzmann equation to handle the high-Reynolds number flow. Furthermore, the unit transformation among physical system, non-dimensionless system and lattice system are explained by one numerical example. Finally, some benchmarks including the cavity flow, Taylor-Green vortex, eccentric circular cylinder in the channel have been presented.

In chapter 3, the GPU implementation of the lattice Boltzmann solver has been characterized from the viewpoints of both hardware and software. To be first, the overview of the CUDA programming platform including the hardware description of the GPU card, the CUDA programming rule has been reviewed. Then the GPU implementation particulars of the lattice Boltzmann solver have been depicted, including the shared memory usage, atomic operations and the stream concurrent execution in the boundary treatment. To be capable of the large-scale numerical simulation, the advanced GPU direct-MPI hybrid framework that utilizes both the peer-to-peer memory access and MPI communication has been developed.

In chapter 4, the accuracy and pressure oscillation of the moving fluid-solid boundary has been investigated deeply. A bi-quadratic ghost fluid immersed boundary method has been proposed for more accurate boundary treatment than the bi-linear ghost fluid immersed boundary method. A new refilling scheme for the fresh nodes that are emerged from the solid region to the fluid domain has also been proposed to suppress the pressure oscillation around the fluid-solid interface. The ghost fluid immersed boundary lattice Boltzmann methods have been compared with the conventional interpolation or extrapolation bounce-back schemes. The accuracy and the pressure oscillation feature in moving boundary problems for different collision models, refilling techniques and force evaluation methods have also been investigated and discussed. These investigations laid a solid foundation for the subsequent work.

In chapter 5, the ocean engineering applications of the single-phase free surface model have been stated. The technique developed in chapter 4 and the selected advanced approaches are applied in the present enhanced free surface solver. The dam-breaking, dam-breaking with

Chapter 1 Introduction

obstacle, water entry with constant speed, water exit with constant speed, water entry with free falling have been simulated successfully.

In chapter 6, the multiphase lattice Boltzmann models and their applications have been described. The thoroughgoing analysis of He-Chen-Zhang model (1999), Lee-Lin model (2005), Lee-Liu model (2010) and Abbas-Martin-Lee (2016) have been conducted to reveal their merits and demerits. The Abbas-Martin-Lee multiphase model is finally selected in the present work. This model has been verified by the benchmarks of circular interface in a shear flow, deformation of a circular interface, Rayleigh-Taylor instability, the drop splashing on a thin film and the three-dimensional Laplace law. Subsequently, the bubble dynamics has been investigated systematically. The single bubble rising, the co-axial coalescence of two bubbles, oblique coalescence of two bubbles, bubble flow and vortex ring bubble have been simulated successfully.

Conclusions and future prospects of this study are summarized in chapter 7.

Chapter 2 Basic theory of lattice Boltzmann equation

2.1 Boltzmann Equation

In any macroscopic system, as each molecular moves and rotates according to some basic laws of mechanics, the macroscopic variables can be obtained from the specific motions of a huge amount of molecular. If the probability of the molecular at some state is resolved, instead of the specific behavior of every molecular, the macroscopic variables can be also obtained through the statistic approach, which is the basic idea of the Boltzmann equation. Boltzmann equation is the evolution equation to describe the non-equilibrium distribution function in statistic mechanics based on the following three hypotheses: 1) Only the two-body collision is considered; 2) Molecular chaos hypothesis that assumes that the velocity of each molecular does not affect each other before collision; 3) The external force does not affect the local behavior of molecular collision. Based on the gas kinetic theory, the Boltzmann equation can be derived based on hard-ball model with elastic collision. The most significant concept is the velocity distribution function that is the statistic description of the gases. It assumes that the number of molecules at the position $\mathbf{x}$ and the velocity ξ is $f(\mathbf{x}, \xi, t)d\xi d\mathbf{x}$. The molecule density with the velocity ξ is $f(\mathbf{x}, \xi, t)d\xi$. Without the collision behavior in dt , any molecule will stream from the position $\mathbf{x}$ to $\mathbf{x} + d\mathbf{x}$ with velocity varied to $\xi + \mathbf{a}dt$. Therefore, at the time $t + dt$, the molecule quantity in the previous molecule block will keep the same but stream to the position $\mathbf{x} + d\mathbf{x}$ with velocity $\xi + \mathbf{a}dt$. It has

$$f(\mathbf{x} + \xi dt, \xi + \mathbf{a}dt, t) = f(\mathbf{x}, \xi, t) \quad (2-1)$$

If the Taylor expansion is applied to Eq.(2-1), then

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla_{\mathbf{x}} f + \mathbf{a} \cdot \nabla_{\xi} f = 0 \quad (2-2)$$

The contribution due to the stream process for molecular changing rate can be written as

$$\left(\frac{\partial f}{\partial t}\right)_{stream} = -\xi \cdot \nabla_{\mathbf{x}} f - \mathbf{a} \cdot \nabla_{\xi} f \quad (2-3)$$

Chapter 2 Basic theory of lattice Boltzmann equation

Except the augment of the distribution function due to the motion, the increment of the distribution functions due to the molecule collision should be also included:

$$\frac{\partial f}{\partial t} = \left(\frac{\partial f}{\partial t}\right)_{stream} + \left(\frac{\partial f}{\partial t}\right)_{collision} \quad (2-4)$$

Then we can have

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla_x f + \mathbf{a} \cdot \nabla_\xi f = \left(\frac{\partial f}{\partial t}\right)_{collision} \quad (2-5)$$

The collision term on the right hand-side is usually written as $\Omega(f)$. Then the general form of Eq.(2-5) is

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla_x f + \mathbf{a} \cdot \nabla_\xi f = \Omega(f) \quad (2-6)$$

The collision term on the right-hand side is a complicated nonlinear term

$$\Omega(f) = \int [f' f'_1 - f f_1] B(\theta, |\mathbf{V}|) d\theta d\varepsilon d\xi_1 \quad (2-7)$$

which is derived based on the hard-ball collision assumption. f' and f'_1 are post-collision quantities while f and f_1 are pre-collision quantities. The collision term is also related to the colliding angle and relative velocity.

The bridge between the microscopic and macroscopic variables is the momentum of the distribution functions. The density, velocity, energy and stress tensor can be obtained through the momentum of the distribution functions as:

$$\rho(\mathbf{x}, t) = m \int f(\mathbf{x}, \xi, t) d\xi \quad (2-8)$$

$$u(\mathbf{x}, t) = \frac{m}{\rho} \int \xi f(\mathbf{x}, \xi, t) d\xi \quad (2-9)$$

$$\varepsilon = \frac{m}{\rho} \int \frac{1}{2} (\xi - \mathbf{u})^2 f(\mathbf{x}, \xi, t) d\xi \quad (2-10)$$

$$P_{ij} = m \int (\xi_i - \mathbf{u}_i)(\xi_j - \mathbf{u}_j) f(\mathbf{x}, \xi, t) d\xi \quad (2-11)$$

where m is the mass of molecule.

2.2 Lattice Boltzmann Equation

As mentioned by He & Luo (1997), the lattice Boltzmann equation is a special discretized form of the continuous Boltzmann equation. Intrinsically, the lattice Boltzmann equation is the finite difference form of the continuous Boltzmann equation. The discretization will be conducted in the temporal, spatial and velocity space. The spatial discretization can be performed on the background meshes, such as Cartesian mesh and body fitting mesh. A certain number of velocities are chosen to describe the statistical behavior of the continuous velocity space. A finite velocity set $\{\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_N\}$ can be used in the derivation. At the meantime, the continuous distribution function can be discretized as $\{f_0, f_1, \dots, f_N\}$. To be simplicity, the Bhatnagar-Gross-Krook approximation of the complicated nonlinear collision term is applied to finish the derivation. The discrete form of the Boltzmann equation can thus be written as

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla f_\alpha = -\frac{1}{\tau_0} (f_\alpha - f_\alpha^{eq}) + F_\alpha \quad (2-12)$$

where $f_\alpha = f_\alpha(\mathbf{x}, \mathbf{e}_\alpha, t)$ is the discrete form of the distribution function and $F_\alpha = -(a \cdot \nabla_\xi f)_\alpha$ is the discrete force form. The Maxwell equilibrium distribution function can be applied

$$\begin{aligned} f_\alpha^{eq} &= \rho (2\pi RT)^{-D/2} \exp\left[-\frac{(\mathbf{e}_\alpha - \mathbf{u})^2}{2RT}\right] \\ &= \rho \omega_\alpha \left[1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{RT} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2R^2 T^2} - \frac{u^2}{2RT}\right] + O(u^3) \end{aligned} \quad (2-13)$$

where $\omega_\alpha = (2\pi RT)^{-D/2} \exp(-\frac{e_\alpha^2}{2RT})$ is the weighting coefficients that vary with different discrete model.

Integrating Eq.(2-12) along the characteristic line, the advection term on the left-side can be transformed into a perfect shift form. The trapezoidal rule is applied to obtain the collision term and force term on right-hand side. The lattice Boltzmann equation can be obtained as

Chapter 2 Basic theory of lattice Boltzmann equation

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta_t, t + \delta_t) - f_\alpha(\mathbf{x}, t) = -\frac{1}{\tau} [f_\alpha(\mathbf{x}, t) - f_\alpha^{eq}(\mathbf{x}, t)] + \delta_t F_\alpha(\mathbf{x}, t) \quad (2-14)$$

where $\tau = \tau_0 / \delta_t$ is the non-dimensionless relaxation time. Since the integration is conducted along the characteristic line, the time-marching accuracy is achieved to be second-order. Thus, the LBE solver is a second-order approximation solver of the NS equation.

To establish complete and rigorous lattice Boltzmann model, a certain of constraint conditions and discrete model should be chosen. Taking the D2Q9 (two dimensional and nine discrete velocity components) lattice model (Qian et al. 1992) for example, the constraints are as followings:

$$\sum_{\alpha=1}^4 e_{\alpha i} = 0, \sum_{\alpha=5}^8 e_{\alpha i} = 0 \quad (2-15a)$$

$$\sum_{\alpha=1}^4 e_{\alpha i} e_{\alpha j} = 2c^2 \delta_{ij}, \sum_{\alpha=5}^8 e_{\alpha i} e_{\alpha j} = 4c^2 \delta_{ij} \quad (2-15b)$$

$$\sum_{\alpha=1}^4 e_{\alpha i} e_{\alpha j} e_{\alpha k} = 0, \sum_{\alpha=5}^8 e_{\alpha i} e_{\alpha j} e_{\alpha k} = 0 \quad (2-15c)$$

$$\sum_{\alpha=1}^4 e_{\alpha i} e_{\alpha j} e_{\alpha k} e_{\alpha l} = 2c^4 \delta_{ijkl}, \sum_{\alpha=5}^8 e_{\alpha i} e_{\alpha j} e_{\alpha k} e_{\alpha l} = 4c^4 \Delta_{ijkl} - 8c^4 \delta_{ijkl} \quad (2-15d)$$

where $\Delta_{ijkl} = (\delta_{ij} \delta_{kl} + \delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk})$

The configuration of the discrete velocities in the above constraints is written as

$$E = \begin{Bmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{Bmatrix} \quad (2-16)$$

The weighting coefficients for D2Q9 model is

$$\omega_\alpha = \{4/9, 1/9, 1/9, 1/9, 1/9, 1/36, 1/36, 1/36, 1/36\} \quad (2-17)$$

The macroscopic variables are evaluated by the momentum of the discrete distribution function

$$\sum_\alpha f_\alpha^{(eq)} = \rho \quad (2-18a)$$

$$\sum_{\alpha} \tilde{e}_{\alpha} f_{\alpha}^{(eq)} = \rho \vec{u} \quad (2-18b)$$

$$\sum_{\alpha} e_{\alpha i} e_{\alpha j} f_{\alpha}^{(eq)} = \rho u_i u_j + p \delta_{ij} \quad (2-18c)$$

Similarly, for the three dimensional simulations, the D3Q19 model is applied. The velocity sets are as following

$$E = \begin{Bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{Bmatrix} \quad (2-19)$$

The geometry configurations of D2Q9 and D3Q19 models are shown in Fig.2.1 and Fig.2.2.

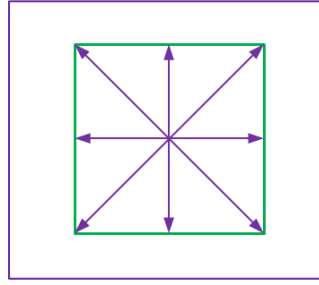


Fig.2.1 D2Q9 model

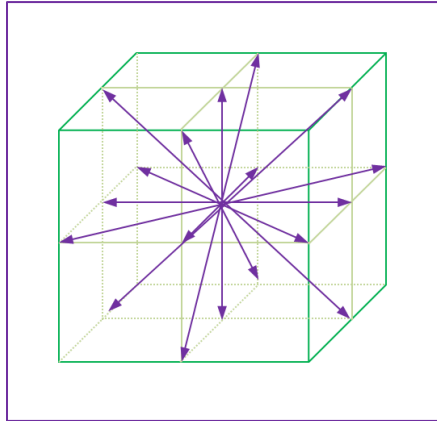


Fig.2.2 D3Q19 model

By the multi-scaling expansion technique, Chapman-Enskog expansion, the macroscopic NS equation can be recovered from the lattice Boltzmann equation Eq.(2-14), which can be found in Appendix A.

In the subsequent context, we would like to describe different approximations of the

Chapter 2 Basic theory of lattice Boltzmann equation

complicated collision term. In the historical development of the lattice Boltzmann model, there exist the LBGK model (Bhatnagar et al. 1954) or the so-called Single-Relaxation-Time model (SRT), Multiple-Relaxation-Time model (MRT) (D. d’Humières 1992, P. Lallemand & L.-S. Luo 2000, D. d’Humières 2002) and Two-Relaxation-Time model (TRT) (I. Ginzburg, 2003, 2008).

LBGK collision model

As aforementioned in the above context, the $\Omega(f)$ in Eq.(2-6) should be simplified. The following concise form was suggested to model the collision process by Bhatnagar et al. (1954):

$$\Omega(f) = -\frac{1}{\tau} [f_\alpha(\mathbf{x}, t) - f_\alpha^{eq}(\mathbf{x}, t)] \quad (2-20)$$

in which the relaxation rate is related to the viscosity by the relationship $\tau = 3\nu + 0.5$. The Maxwell distribution function is performed to calculate the equilibrium distribution functions

$$f_\alpha^{eq} = \rho \omega_\alpha \left[1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right] \quad (2-21)$$

The underlying philosophy of this approximation is that the collision process relaxes the particle distribution functions linearly towards their equilibrium states which are defined by the Maxwell distribution function. The practical numerical simulation demonstrates that the instability issue will occur when the relaxation time is close to 0.5. This indicates that the LB equation is not capable of the inviscid flow, in which the relaxation time should be identical to 0.5. Meanwhile, due to the single relaxation rate for all the physical values in the flow field, the rationality of this model is suspicious. In the free surface flow problems of chapter 4, numerical simulations demonstrate that the pressure noise in the flow field cannot be effectively dissipated in the LBGK model.

Multiple-Relaxation-Time collision model

In contrary to the SRT model, the MRT model provides a general collision approach to relax the different physical variables with different relaxation rates. This general formulation offers the maximum number of degrees of freedom for modelling the linearized collision

operator and forms the basis of the multiple-relaxation-time (MRT) model. One can try to assemble the collision matrix directly as it is done by Higuera et al.(1989). The collision process is conducted in the momentum space:

$$\Omega(f) = -\mathbf{M}^{-1}\Lambda(\mathbf{m} - \mathbf{m}^{eq}) \quad (2-22)$$

The matrix $\mathbf{M}$ is the orthogonal matrix based on the discrete velocity set, which can be found in Appendix B. $\mathbf{m} = \mathbf{M}f$ is the transformed moments of the distribution function. Λ is the diagonal relaxation matrix

$$\Lambda = \text{diag}(s_0, s_1, \dots, s_8) \quad (2-23)$$

For the D2Q9 model, one typically chooses the following moments which span the moment space:

$$m = (\rho, e, \varepsilon, j_x, q_x, j_y, q_y, p_{xx}, p_{xy}) \quad (2-24)$$

- Density (zeroth-order): ρ
- Energy (second-order): e
- Energy square (fourth-order): ε
- Momentum in x and y direction (first-order): j_x, j_y
- Heat flux in x and y direction (third-order): q_x, q_y
- Diagonal and off-diagonal stress (second-order): p_{xx}, p_{xy}

The density and momentum are conserved values, it is not necessary to relax them. The equilibrium values of the non-conserved moments in Eq.(2-24) can be chosen at will provided that the symmetry of the problem is respected

$$e^{eq} = -2\rho + 3(j_x^2 + j_y^2) \quad (2-25a)$$

$$\varepsilon^{eq} = \rho - 3(j_x^2 + j_y^2) \quad (2-25b)$$

$$q_x^{eq} = -j_x \quad (2-25c)$$

$$q_y^{eq} = -j_y \quad (2-25d)$$

$$p_{xx}^{eq} = j_x^2 - j_y^2 \quad (2-25e)$$

Chapter 2 Basic theory of lattice Boltzmann equation

$$p_{xy}^{eq} = j_x j_y \quad (2-25f)$$

The selection of the relaxation times in Λ can be referred to Luo et al. (2010). In the present work, the following configuration is usually performed

$$\Lambda = \text{diag}(0, 1.64, 1.54, 0, 1.9, 0, 1.9, s_v, s_v) \quad (2-26)$$

where s_v is determined by the shear viscosity. As mentioned in Luo et al. (2010), the relaxation rate of energy mode will reduce the pressure fluctuation remarkably, which indicates that the bulk viscosity effectively dissipates the pressure noise in the flow field, especially at the high Reynolds regime.

For the D3Q19 model, the matrix $\mathbf{M}$ can be also found in the Appendix B. The moments of the distribution functions are expressed as

$$\mathbf{m} = (\rho, e, \varepsilon, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{zx}, m_x, m_y, m_z) \quad (2-27)$$

The diagonal matrix is as followings

$$\Lambda = \text{diag}(0, s_1, s_2, 0, s_4, 0, s_4, s_9, s_{10}, s_9, s_{10}, s_{13}, s_{13}, s_{13}, s_{16}, s_{16}, s_{16}) \quad (2-28)$$

with $s_9 = s_{13}$.

The equilibrium moment for the non-conserved moments can be written as

$$e^{eq} = -11\rho + 19(j_x^2 + j_y^2 + j_z^2) \quad (2-29a)$$

$$\varepsilon^{eq} = \omega_\varepsilon \rho + \omega_{\varepsilon j} \mathbf{j} \cdot \mathbf{j} \quad (2-29b)$$

$$q_x^{eq} = -\frac{2}{3} j_x, q_y^{eq} = -\frac{2}{3} j_y, q_z^{eq} = -\frac{2}{3} j_z \quad (2-29c)$$

$$p_{xx}^{eq} = \frac{1}{3} [2j_x^2 - (j_y^2 + j_z^2)] \quad (2-29d)$$

$$p_{ww}^{eq} = [j_y^2 - j_z^2] \quad (2-29e)$$

$$p_{xy}^{eq} = j_x j_y, p_{yz}^{eq} = j_y j_z, p_{xz}^{eq} = j_x j_z \quad (2-29f)$$

$$\pi_{xx}^{eq} = \omega_{xx} p_{xx}^{eq} \quad (2-29g)$$

$$\pi_{\omega\omega}^{eq} = \omega_{xx} p_{ww}^{eq} \quad (2-29h)$$

$$m_x^{eq} = m_y^{eq} = m_z^{eq} = 0 \quad (2-29i)$$

The free parameters are set as $\omega_\varepsilon = 0, \omega_{\varepsilon j} = -475/63, \omega_{xx} = 0$, which is recommended by P. Lalleman & Luo (2000). The shear viscosity is given as

$$\nu = \frac{1}{3} \left(\frac{1}{s_9} - \frac{1}{2} \right) = \frac{1}{3} \left(\frac{1}{s_{13}} - \frac{1}{2} \right) \quad (2-30)$$

The relaxation parameters can be referred to d'Humières et al (2002).

Two-Relaxation-Time collision model

The philosophy of TRT model is to relax the even-order moments with the relaxation rate s_+ and the odd-order ones with s_- . For the D2Q9 model, e, ε and p_{xx}, p_{xy} are the even-order moments. In the D3Q19 model, q_x, q_y, q_z and m_x, m_y, m_z are the odd-order moments. The discrete evolution equation solved by the LBM with the TRT collision model is expressed as

$$\begin{aligned} f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta_t, t + \delta_t) - f_\alpha(\mathbf{x}, t) = & -s_+ [f_\alpha^+(\mathbf{x}, t) - f_\alpha^{+(eq)}(\mathbf{x}, t)] \\ & -s_- [f_\alpha^-(\mathbf{x}, t) - f_\alpha^{-(eq)}(\mathbf{x}, t)] + \delta_t F_\alpha(\mathbf{x}, t) \end{aligned} \quad (2-31)$$

where the even distribution functions and the odd distribution functions are generated with the following way

$$f_\alpha^+ = \frac{f_\alpha + f_{\bar{\alpha}}}{2}, \quad f_\alpha^- = \frac{f_\alpha - f_{\bar{\alpha}}}{2} \quad (2-32)$$

$$f_\alpha^{+(eq)} = \frac{f_\alpha^{eq} + f_{\bar{\alpha}}^{eq}}{2}, \quad f_\alpha^{-(eq)} = \frac{f_\alpha^{eq} - f_{\bar{\alpha}}^{eq}}{2} \quad (2-33)$$

$\bar{\alpha}$ is the opposite direction of α . The relaxation rate for the even-order and odd-order are as followings, respectively

$$s_+ = \frac{1}{\tau} = \frac{2}{6\nu + 1} \quad (2-34)$$

$$s_- = 8 \frac{2 - s_+}{8 - s_+} = \frac{16\nu}{8\nu + 1} \quad (2-35)$$

2.3 Body force

In terms of the external force model, it should be addressed that it is extremely direct in the traditional CFD branches while it is non-trivial in the lattice Boltzmann equation. In the historical development of the external force in the LBE, there exist various external force schemes such as Buick-Greated model (2000), He-Shan-Doolen (1998) model, the second-order moment model proposed by Luo (1998), modified second-order moment model improved by Ladd & Verberg (2001). However, the discrete effects of the force term in these models are not treated well, especially for the unsteady external force. Guo et al. (2002) proposed a general external force model to carefully deal with the discrete lattice effect and count the contributions of the body force to the momentum flux. In the present work, this force scheme is applied in the LBE simulation. The detailed derivation can be found in Appendix C.

In the SRT model, the following external force term can be added to the LBE directly

$$F_\alpha = (1 - \frac{1}{2\tau}) w_\alpha [\frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^4} \mathbf{e}_\alpha] \cdot \mathbf{F} \quad (2-36)$$

For the MRT model, the following form is applied

$$\mathbf{F} = \mathbf{M}^{-1} (\mathbf{I} - \frac{1}{2} \mathbf{S}) \mathbf{M} \bar{\mathbf{F}} \quad (2-37)$$

in which $\mathbf{F} = (F_0, F_1, \dots, F_{b-1})^T$ and $\bar{\mathbf{F}} = (\bar{F}_0, \bar{F}_1, \dots, \bar{F}_{b-1})^T$

And

$$\bar{F}_\alpha = w_\alpha [\frac{\mathbf{e}_\alpha \cdot \mathbf{F}}{c_s^2} + \frac{\mathbf{u} \mathbf{F} \otimes (\mathbf{e}_\alpha \mathbf{e}_\alpha - c_s^2 \mathbf{I})}{c_s^4}] \quad (2-38)$$

At the meantime, the velocity evaluation should be corrected as

$$\rho \mathbf{u} = \sum_\alpha \mathbf{e}_\alpha f_\alpha + \frac{\delta_t}{2} \mathbf{F} \quad (2-39)$$

2.4 Initial Condition

The initial condition is of significance in the practical applications of lattice Boltzmann

model. For the steady or quasi-steady state flow, the effect of the initial condition is not so obvious. Thus the equilibrium distribution functions can be used to initialize the distribution functions for SRT or MRT model, respectively

$$f_i(\mathbf{x}, t=0) = f_i^{eq}(\rho_0, \mathbf{u}_0) \quad (2-40)$$

$$\bar{f}_i(\mathbf{x}, t=0) = M^{-1} m_i^{eq}(\rho_0, \mathbf{u}_0) \quad (2-41)$$

Nevertheless, the above initial conditions are insufficient and inaccurate for the unsteady flows or the nonlinear flows that are sensitive to the initial condition such as turbulence flow and multiphase flow. The error of the initial variables usually exists throughout the simulation. Skordos et al. (1993) has ever proposed a non-equilibrium distribution function correction method to specify the initial condition. However, the given density or pressure distribution should be provided but usually the initial condition only imposes the velocity field. Mei et al. (2006) proposed an iterative approach to initialize the pressure field with the known velocity field through solving the pressure Poisson equation. This consistent initialization scheme is accomplished by replacing the original LBE scheme with three conserved quantities by the corresponding diffusion-advection equation with the only conserved quantity – density. After the pressure field is initialized to be zero and the velocity field to be the given values, the iterative process is applied as followings to obtain the consistent pressure field and distribution functions.

1) Collision in the momentum space:

$$\bar{m}_k'(\mathbf{x}, t) = \bar{m}_k(\mathbf{x}, t) - s_k [\bar{m}_k(\mathbf{x}, t) - \bar{m}_k^{eq}(\bar{\rho}, \mathbf{u}_0)] \quad (2-42)$$

In this step, the velocity is not evaluated by the distribution functions but fixed to be the given velocity and the relaxation time is identical to one that relaxes the momentum towards the state prescribed by $\mathbf{j}^{(eq)} = \mathbf{j}_0 = \mathbf{u}_0$.

2) Advection in the velocity space

$$\bar{f}_k'(\mathbf{x}, t) = M^{-1} \bar{m}_k'(\mathbf{x}, t) \quad (2-43)$$

$$\bar{f}_k(\mathbf{x} + \mathbf{e}_k \delta t, t + \delta t) = \bar{f}_k'(\mathbf{x}, t) \quad (2-44)$$

3) Update density

$$\bar{\rho}(\mathbf{x}, t + \delta t) = \sum_k \bar{f}_k(\mathbf{x}, t + \delta t) \quad (2-45)$$

4) Repeat the process until the density approaches to a steady state. After the iteration process, the initial distribution functions and the density (pressure) are set as

$$f_k(\mathbf{x}, t = 0) = \bar{f}_k(\mathbf{x}, t + \delta t) \quad (2-46)$$

$$\rho(\mathbf{x}, t = 0) = \bar{\rho}(\mathbf{x}, t = 0) \quad (2-47)$$

As analyzed by Mei et al., the following Poisson equation is essentially satisfied if the steady state is reached

$$\nabla^2 p_0 = -\nabla \cdot (\mathbf{u}_0 \cdot \nabla \mathbf{u}_0) \quad (2-48)$$

With this iterative approach, the pressure acoustic oscillation and deficit of total energy can be eliminated well.

2.5 Boundary Condition

For the partial differential equation, appropriate boundary conditions should be imposed for the certain solutions. As aforementioned in the above context, the perfect shift scheme for the advection and nonlinear collision model for the diffusion term can determine the evolution process in a rule-based temporal and spatial approach. The spatial convergence accuracy is of strict second-order. Thus, the overall accuracy of the flow solver is affected by the boundary treatment remarkably. Moreover, the numerical stability and the calculation efficiency will also influenced by the boundary condition. In the lattice Boltzmann simulation, the boundary treatments are not as direct as those in the tradition CFD methods, as the basic variable for LBE is the particle distribution function. Therefore, the boundary condition of the particle distribution functions should be determined by the macroscopic values such as pressure, velocity and their derivatives based on some moderate hypothesis. In what follows, the usual boundary conditions used in our simulations are elaborated one by one.

2.5.1 Periodic boundary condition

Periodic boundary condition is usually applied in the pipe flow or channel flow for the periodic condition or infinite condition in some direction. It assumes that the particles that leave

one side at the present time step will stream into the other side the fluid domain. It can guarantee the absolute conservation of the mass and momentum. The ghost layers can be also set to make the streaming process convenient, as shown in Fig.2.3. Taking the x direction for example, the scheme can be implemented as

$$f_{\alpha}(0, y, z, t + \delta t) = f_{\alpha}^{\wedge}(N_x, y, z, t) \quad (2-49)$$

$$f_{\alpha}(N_x, y, z, t + \delta t) = f_{\alpha}^{\wedge}(0, y, z, t) \quad (2-50)$$

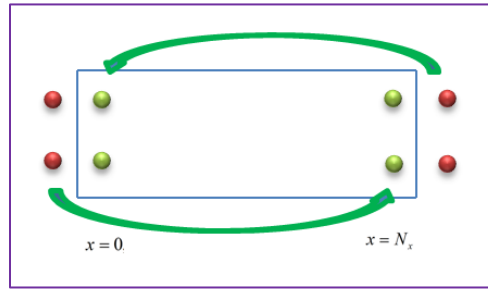


Fig.2.3 Periodic boundary condition

2.5.2 Symmetric boundary condition

In terms of the symmetric physic problems, this boundary treatment can be performed to save the computation resources. One ghost layer should be arranged near the outmost fluid layer. Taking the y-axis symmetric 2D scenario for example, this boundary scheme can be described as Fig.4. The particle distribution function on the ghost layer can be obtained by mirrored treatment with the y axis.

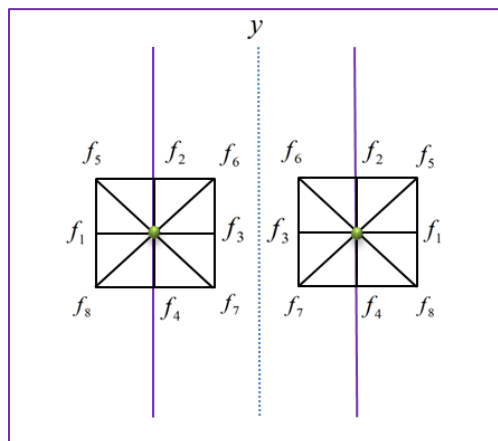


Fig.4 Symmetric boundary condition

2.5.3 Open boundary condition

Chapter 2 Basic theory of lattice Boltzmann equation

The zero normal gradient condition should be applied in the boundary that the macroscopic variables such as velocity and pressure are fully developed. In this case, it is extremely concise to deal with the distribution function. The particle distribution functions on the ghost layer are set to be identical to those on the neighboring outmost fluid layer. Taking the right-hand side boundary node for example, it can be written as

$$f_{\alpha}(N_x, y, z, t) = f_{\alpha}(N_x - 1, y, z, t) \quad (2-51)$$

2.5.4 Half-way bounce back scheme

To deal with the non-slip or free-slip boundary, the concise half-way bounce back scheme is addressed here.

Non-slip treatment

As the particle will be transported across one lattice within one time step in the inner fluid domain, it is assumed that the particles that collide with the boundary will come back to its original position after the distance of two half lattices in the half-way bounce back scheme. In this case, the stream step should be treated carefully. A ghost layer can be arranged near the outmost fluid layer to store the flag of the nodes. The stream step can be executed as

$$f_{\alpha}(\mathbf{x}, t + \delta_t) = f_{\bar{\alpha}}(\mathbf{x}, t) \quad (2-52)$$

where the direction $\bar{\alpha}$ points from fluid domain towards the boundary node. The half-way boundary treatment can be incorporated into the streaming process as executed by the following code:

```
If(flag[jy-ey[k]][jx-ex[k]]==fluid node)
    f[k][jy][jx] = f[k][jy-ey[k]][jx-ex[k]];
else if(flag[jy-ey[k]][jx-ex[k]]==boundary node)
    f[k][jy][jx] = f[inv_k][jy][jx];    (inv_k is inverse direction of k)
.....
```

In this kind of treatment, the corner nodes can be conveniently handled, in which the specific treatments are avoided.

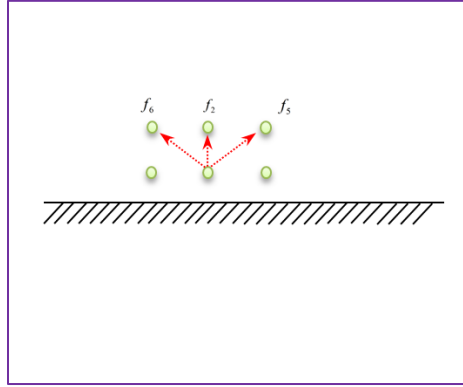


Fig.2.5 Half-way bounce back scheme for non-slip boundary

Free-slip treatment

In some physical boundary treatment, the velocity of the particle in one direction will keep the same while the other direction will be inversed. The analogy of mirror treatment is used, as shown in Fig.2.6. The particle f_4 hit the bottom wall and inverse its direction to be f_2 of the present node. The particle f_7 hit the bottom wall and reflected to its left-side neighboring node to be f_6 . The particle f_8 hit the bottom wall and reflected to its right-side neighboring node to be f_5 . Equivalently, the free-slip boundary treatment is applied to perform the symmetric boundary condition and vice versa.

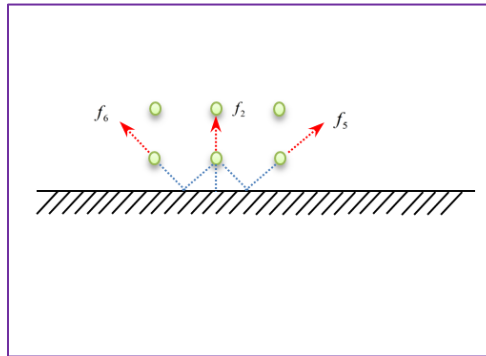


Fig.2.6 Half-way bounce back scheme for free-slip boundary

2.5.5 Non-equilibrium extrapolation scheme

Inspired by the work of Chen et al. (1996), Guo et al. (2002) proposed a new non-equilibrium extrapolation scheme. The basic idea underlying this approach is that the distribution functions on the boundary node can be divided into the equilibrium part and the non-equilibrium part, in which the former is calculated by the macroscopic values on the

Chapter 2 Basic theory of lattice Boltzmann equation

boundary nodes or the extrapolated macroscopic values while the latter is directly extrapolated from the neighboring fluid node. Taking the D2Q9 model for example, the scheme can be depicted in Fig.2.7, in which $O(x_b)$ is the boundary node and $NE, N, NW(x_f)$ are the fluid nodes. The boundary treatment can be regarded as a “collision” process for the boundary nodes. Thus after the stream and before the collision of the fluid domain, the unknown distribution functions on the boundary nodes should be specified. For the bottom of the fluid domain, the distribution functions of direction 2, 5, 6 are missing.

The equilibrium part can be evaluated by Eq.(2.21)

$$f_{\alpha}^{eq}(\mathbf{x}_b, t) = f_{\alpha}^{eq}(\rho(\mathbf{x}_f, t), \mathbf{u}_b) + O(\delta_x^3) \quad (2-53)$$

in which the boundary velocity and the extrapolated density are used.

The non-equilibrium part can be extrapolated from the neighboring fluid node

$$f_{\alpha}^{neq}(\mathbf{x}_b, t) = f_{\alpha}(\mathbf{x}_f, t) - f_{\alpha}^{eq}(\mathbf{x}_f, t) + O(\delta_x) \quad (2-54)$$

Then the distribution functions can be obtained by summing the zero-order equilibrium part and the first-order non-equilibrium part, which results in second-order accurate distribution functions on the boundary node, as demonstrated by Guo et al.

$$f_{\alpha}(\mathbf{x}_b, t) = f_{\alpha}^{eq}(\mathbf{x}_b, t) + f_{\alpha}^{neq}(\mathbf{x}_b, t) \quad (2-55)$$

The above explanation is based on the velocity boundary. It can also be applied to pressure boundary, in which the density is known and the velocity is extrapolated from neighboring fluid node. This non-equilibrium extrapolation scheme can be performed to accomplish the velocity or pressure boundary condition on the inflow and outflow.

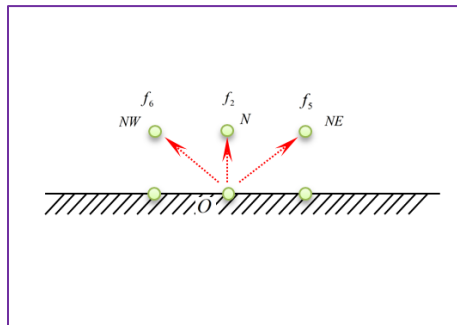


Fig.2.7 Non-equilibrium extrapolation scheme

2.5.6 Zou-He boundary condition

Based on the idea of bounce-back of the non-equilibrium distribution function, Zou &He (1997) proposed a general pressure and velocity boundary condition for the lattice Boltzmann model. The Dirichlet and Neumann boundary conditions are very common in the channel flow. As an example, take the inlet nodes in Fig.2.8 for explanation. After streaming, $f_0, f_2, f_3, f_4, f_6, f_7$ are known but f_1, f_5, f_8 are missed for the inlet boundary nodes. Suppose that u_x, u_y are specified on the left inlet boundary nodes and the momentums of the distribution functions can be used to determine the missing distribution functions f_1, f_5, f_8 .

$$f_1 + f_5 + f_8 = \rho_{in} - (f_0 + f_2 + f_3 + f_4 + f_6 + f_7) \quad (2-56)$$

$$f_1 + f_5 + f_8 = \rho_{in} u_x + (f_3 + f_6 + f_7) \quad (2-57)$$

$$f_5 - f_8 = -f_2 + f_4 - f_6 + f_7 \quad (2-58)$$

Consistency of Eq. (2-56) and Eq.(2-57) gives

$$\rho_{in} = \frac{1}{1-u_x} [f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)] \quad (2-59)$$

To close the equation system, the bounce-back rule for the non-equilibrium part of particle distribution function that is normal to the boundary is assumed (in this case $f_1 - f_1^{eq} = f_3 - f_3^{eq}$). Then the unknown distribution functions can be solved as

$$f_1 = f_3 + \frac{2}{3} \rho_{in} u_x \quad (2-60)$$

$$f_5 = f_7 - \frac{1}{2} (f_2 - f_4) + \frac{1}{6} \rho_{in} u_x \quad (2-61)$$

$$f_8 = f_6 + \frac{1}{2} (f_2 - f_4) + \frac{1}{6} \rho_{in} u_x \quad (2-62)$$

Analogously the outlet boundary can be imposed by the given macroscopic boundary condition, such as the fully developed pressure boundary condition. The density on the boundary nodes can be copied from the outmost fluid layer. In this way the unknown

distribution functions can be solved as

$$u_x = -1 + \frac{1}{\rho_{out}} [f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8)] \quad (2-63)$$

$$f_3 = f_1 - \frac{2}{3} \rho_{out} u_x \quad (2-64)$$

$$f_6 = f_8 - \frac{1}{2} (f_2 - f_4) - \frac{1}{6} \rho_{out} u_x \quad (2-65)$$

$$f_7 = f_5 + \frac{1}{2} (f_2 - f_4) - \frac{1}{6} \rho_{out} u_x \quad (2-66)$$

In terms of the corner node, taking the left-bottom corner node for example, the velocity is set to zero and the density is set to unit. Then the distribution functions can be solved as

$$f_1 = f_3, f_2 = f_4, f_5 = f_7 \quad (2-67)$$

$$f_6 = f_8 = \frac{1}{2} [\rho - (f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_7)] \quad (2-68)$$

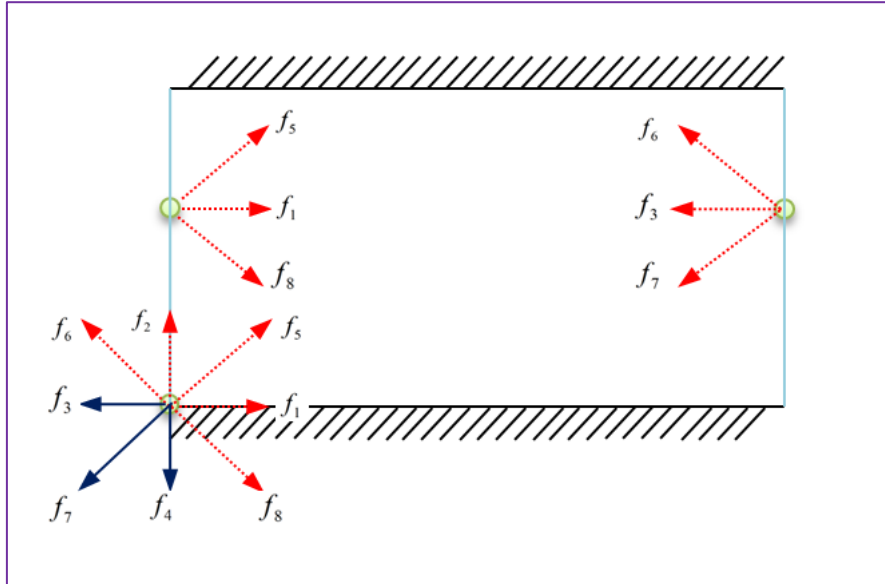


Fig.2.8 Zou-He boundary condition

2.5.7 Immersed boundary method

2.5.7.1 Partially saturated computational cell method (Nobel et al. 1998)

In order to consider the moving solid/fluid no-slip condition, Noble et al. (1998) proposed a partially saturated computational cell method (cut-cell-based method) to deal with the fluid-particulate interaction phenomenon. In this method, the lattice Boltzmann equation is modified by incorporating a term that concerns the proportion of the nodal cell that is partially covered by solid (see Fig.2.9). In this way, it could improve the boundary representation and smooth the hydrodynamic forces exerted on the obstacle in the moving boundary treatment. A control volume is introduced for each lattice node by setting a $\delta_x \times \delta_x$ square around the node. A solid covering ratio parameter ε_s with respect to the area of the whole cell is defined to consider the partial boundary representation. For those boundary lattice grids partially or fully covered by the obstacle, the modified LBE is given by

$$f_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t) - [1 - B_s(\varepsilon_s, \tau)] \frac{1}{\tau} [f_\alpha(\mathbf{x}, t) - f_\alpha^{(eq)}(\mathbf{x}, t)] + B_s(\varepsilon_s, \tau) \Omega_\alpha^s \quad (2-69)$$

where ε_s is the volume fraction of the solid part in each nodal cell which will be 0 or 1 corresponding to pure fluid or obstacle nodes far away from the boundary and $0 < \varepsilon_s < 1$ for nodal cells cut by the boundary and can be calculated by the polygonal approximations as the compromise of accuracy and efficiency. $B_s(\varepsilon_s, \tau)$ is a weighting function related to ε_s . Ω_α^s is an additional collision operator considering the interaction of the solid boundary and fluid. Here we use the following expressions for B_s and Ω_α^s

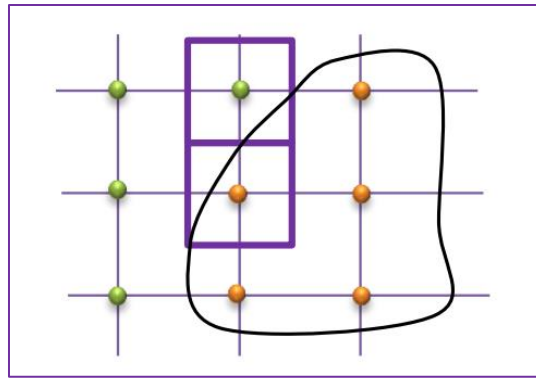


Fig.2.9 Schematic of the solid ratio

$$B_s(\varepsilon_s, \tau) = \frac{\varepsilon_s(\tau - 0.5)}{(1 - \varepsilon_s) + (\tau - 0.5)} \quad (2-70)$$

Chapter 2 Basic theory of lattice Boltzmann equation

$$\Omega_{\alpha}^s = [f_{\bar{\alpha}}(\mathbf{x}, t) - f_{\bar{\alpha}}^{(eq)}(\rho, \mathbf{u})] - [f_{\alpha}(\mathbf{x}, t) - f_{\alpha}^{(eq)}(\rho, \mathbf{u}_w)] \quad (2-71)$$

where $\bar{\alpha}$ is the opposite direction of α . The velocity of the solid boundary node $\mathbf{u}_w$ is computed as

$$\mathbf{u}_w = \mathbf{u}_c + \boldsymbol{\omega} \times \mathbf{r}_c \quad (2-72)$$

in which $\mathbf{u}_c$ and $\boldsymbol{\omega}$ are the translational and angular velocities at the obstacle's mass center, $\mathbf{r}_c$ is the distance vector between the mass center and the solid boundary node. The hydrodynamic force and torque exerted on the obstacle can be evaluated as

$$\mathbf{F} = C\delta_x \left[\sum_n (B_n \sum_{\alpha} \Omega_{\alpha}^s \mathbf{e}_{\alpha}) \right] \quad (2-73)$$

$$\mathbf{T} = C\delta_x \left[\sum_n (\mathbf{x}_n - \mathbf{x}_c) (B_n \sum_{\alpha} \Omega_{\alpha}^s \mathbf{e}_{\alpha}) \right] \quad (2-74)$$

where $\mathbf{x}_n$ and $\mathbf{x}_c$ are the position of boundary node and mass center respectively.

2.5.7.2 Direct forcing method with diffuse interface (DF explicit and implicit) (Kang et al. 2010)

As shown in Fig.2.10, a set of Lagrangian points are used to represent the boundary. The formula for evaluating the force density at the Lagrangian points is as follows:

$$\mathbf{F}(\mathbf{x}_b, t + \Delta t) = 2\rho(\mathbf{x}_b, t + \Delta t) \frac{\mathbf{U}^d - \mathbf{u}^{noF}(\mathbf{x}, t + \Delta t)}{\Delta t} \quad (2-75)$$

where $\mathbf{U}^d$ is the desired velocity at the Lagrangian point on the boundary and $\mathbf{u}^{noF}(\mathbf{x}, t + \Delta t)$ is interpolated from the unforced velocity $\mathbf{u}_{ij}^{noF}$ at the neighboring Cartesian grids by using the δ function $D(\mathbf{x}_{ij} - \mathbf{x}_b)$ as follows:

$$\mathbf{u}_b^{noF} = \sum_{i,j} \mathbf{u}_{ij}^{noF} D(\mathbf{x}_{ij} - \mathbf{x}_b) \quad (2-76)$$

The δ function $D(\mathbf{x}_{ij} - \mathbf{x}_b)$ is expressed as

$$D(\mathbf{x}_{ij} - \mathbf{x}_b) = \frac{1}{\delta_x^2} d_h\left(\frac{x_i - x_b}{\delta_x}\right) d_h\left(\frac{y_j - y_b}{\delta_x}\right)$$

$$d_h(r) = \begin{cases} 1 - |r|, & |r| \leq 1 \\ 0, & |r| > 1 \end{cases} \quad (2-77)$$

For the fluid-solid interaction, we should distribute the force density $\mathbf{F}(\mathbf{x}_b, t + \Delta t)$ onto the Cartesian fluid nodes:

$$\mathbf{F}_{ij} = \sum_{i,j} \mathbf{F}_b D(\mathbf{x}_{ij} - \mathbf{x}_b) \Delta s_b \quad (2-78)$$

where Δs_b is the length of segment for Lagrangian node.

Then the velocity on the Cartesian grid is updated as

$$\mathbf{u}_{ij} = \mathbf{u}_{ij}^{noF} + \frac{\Delta t}{2\rho} \mathbf{F}_{ij} \quad (2-79)$$

The surface force for the obstacle can be easily evaluated by using

$$\mathbf{F}_s = -\sum_b \mathbf{F}(\mathbf{x}_b) \Delta s_b = -\sum_{i,j} \mathbf{F}(\mathbf{x}_{ij}) \delta_x^2 \quad (2-80)$$

The solution procedure of this scheme can be summarized as follows:

- 1) Solve the equation (2-14) without force term and get the velocity at every Cartesian node using (2-39) without force term;
- 2) Obtain unforced velocity interpolation on Lagrangian node $\mathbf{x}_b$ using formula (2-76);
- 3) Evaluate the boundary force on Lagrangian node $\mathbf{x}_b$ using formula (2-75);
- 4) Distribute the force onto Cartesian node $\mathbf{x}_{ij}$ using formula (2-78);
- 5) Update the velocity on $\mathbf{x}_{ij}$ using formula (2-79);
- 6) For implicit-correction, repeat (3)-(5) for several times;
- 7) Calculate the surface force of the obstacle using formula (2-80)

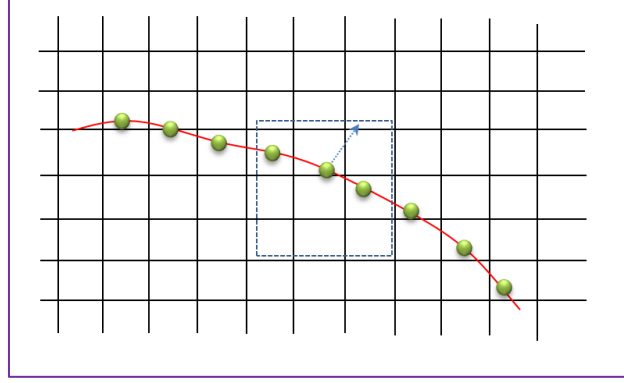


Fig.2.10 Lagrangian nodes on the boundary

2.6 Turbulence Model

In the ocean engineering applications, the violent free surface flows stay in the high Reynolds number flow regime, where turbulent flows contribute to the general flow pattern. To be capable with the turbulent flow with the lattice Boltzmann model, Manfred K. & Luo (2003) introduced the LES model into the MRT lattice Boltzmann model. In LES models, the large spatial-scale turbulent eddy structures which carry the majority of the energy is solved directly while the subgrid-scale eddies are modeled by a subgrid spatial filter. In this treatment, the effect of the subgrid-scale eddies are included in the simulation via an additional turbulent viscosity ν_t . The one-parameter Smagorinski model where ν_t depends on the shear rate is applied in the present work:

$$\nu_t = (C\Delta_x)^2 \|\mathbf{S}\| \quad (2-81)$$

where C is the Smagorinski constant, Δ_x is the filter length and $\mathbf{S}$ is the strain rate tensor as

$$S_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i) \quad (2-82)$$

The magnitude of the strain rate tensor is calculated as

$$\|\mathbf{S}\| = \sqrt{2 \sum_{i,j} S_{ij} S_{ij}} \quad (2-83)$$

The second momentum of the distribution function can be evaluated as

$$P_{ij} = \sum_{\alpha} e_{\alpha i} e_{\alpha j} f_{\alpha} = c_s^2 \rho \delta_{ij} + \rho u_i u_j - \frac{1}{s_{xx}} 2c_s^2 \rho S_{ij} \quad (2-84)$$

The strain rate tensor then can be expressed as

$$S_{ij} = \frac{s_{xx}}{2c_s^2 \rho} [c_s^2 \rho \delta_{ij} + \rho u_i u_j - P_{ij}] = \frac{s_{xx}}{2c_s^2 \rho} Q_{ij} \quad (2-85)$$

where $s_{xx} = \frac{1}{\tau_0 + \tau_t}$, Q_{ij} is the filtered mean momentum flux, of which the module can be calculated as

$$\bar{Q} = \sqrt{2 \sum_{i,j} Q_{ij} Q_{ij}} \quad (2-86)$$

Q_{ij} is the second-order momentum of the non-equilibrium distribution functions as

$$Q_{ij} = \sum_{\alpha} e_{\alpha i} e_{\alpha j} (f_{\alpha} - f_{\alpha}^{eq}) \quad (2-87)$$

The above equations induced a quadratic equation for eddy viscosity, the positive solution is retained

$$\tau_t = \frac{1}{2} (\sqrt{\tau_0^2 + 18(C\Delta_x)^2 \bar{Q}} - \tau_0) \quad (2-88)$$

The filter length is usually set to be identical to the lattice length.

From the aforementioned context, it can be found that the derivatives of velocities are avoided and the evaluation of the eddy viscosity is completely local, which is well-suite for the parallel computation on GPU.

2.7 Unit Transformation

In the lattice Boltzmann simulations, both the lattice length and the lattice time step are always identical to one. It is of significance to establish a perspicuous corresponding relationship with the real physical system. To guarantee the effectiveness of the lattice Boltzmann simulation, two principals should be noted. First, the simulation should be equivalent to the real physical system. Second, the flow field should be sufficiently resolved and the discrete time step should be sufficiently small such that the accuracy requirement could

Chapter 2 Basic theory of lattice Boltzmann equation

be satisfied. Here the work of Jonas Latt (2008) will be organized and explained combining with the practical engineering application in this thesis.

To make the unit transformation to be clear, two steps are conducted. In the first step, the physical system is converted into the non-dimensionless system if the characteristic parameters are selected such as the characteristic length and characteristic time. Then, the non-dimensionless system is transformed into the discrete LBM system. The similarity criterion should be kept the same such as the Reynolds number in the viscous flow. The transformation process can be found in Fig.2.11.

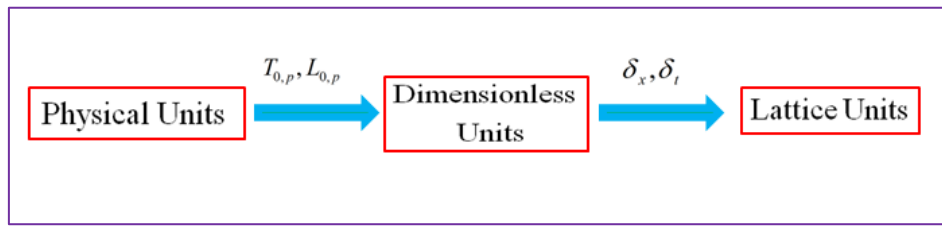


Fig.2.11 Unit transformation from physical units to lattice units

In an attempt to clarify the unit transformation relationship, the water entry case simulated in chapter 5 is described here to explain the two-step process, as shown in Fig.2.12.

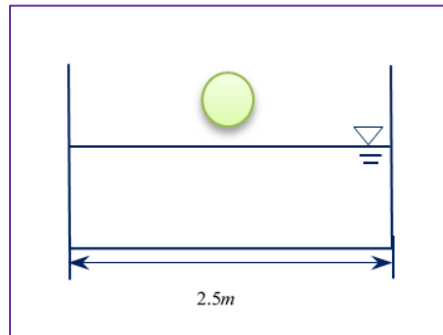


Fig.2.12 Configuration of water tank

The length and the height of the water tank are 2.5 meters and 1.4 meters, respectively. The depth of the water is 1.0 meter. The diameter of the cylinder is 0.125 m. The downward speed of the water entry is 0.639 m/s. The gravitational acceleration is 9.8 m/s^2 , and the kinematic viscosity is $\nu = 1.0 \times 10^{-6} \text{ m}^2 / \text{s}$. The significant similarity criteria are the Reynolds number and the Froude number, which are defined as $\text{Re} = UL / \nu$ and $\text{Fn} = U / \sqrt{gL}$, respectively.

Firstly the characteristic length and time should be determined. The diameter of the cylinder is chosen to be the physical characteristic length. The water entry velocity is selected to be the physical characteristic velocity. Then the physical characteristic time can be calculated by them.

$$L_{0,p} = 0.125m \quad (2-89a)$$

$$U_{0,p} = 0.639m / s \quad (2-89b)$$

$$T_{0,p} = L_{0,p} / U_{0,p} = 0.1956s \quad (2-89c)$$

In this case, the similarity criterions can be obtained as

$$Re = U_{0,p} L_{0,p} / \nu_{0,p} = 79875 \quad (2-90a)$$

$$Fn = U_{0,p} / \sqrt{g_{0,p} L_{0,p}} = 0.577 \quad (2-90b)$$

The physical variables can be normalized by the characteristic parameters. Thus the second step is to obtain the non-dimensionless values. In the non-dimensionless system, the length and time is normalized to be one. Subsequently the viscosity and the gravity could be calculated.

$$L_0 = 1 \quad T_0 = 1 \quad U_0 = 1 \quad (2-91a)$$

$$\nu_0 = \nu_{0,p} \cdot T_{0,p} / (L_{0,p} \cdot L_{0,p}) = 1.252 \cdot 10^{-5} \quad (2-91b)$$

$$g_0 = g_{0,p} \cdot (T_{0,p} \cdot T_{0,p}) / L_{0,p} = 3.0 \quad (2-91c)$$

Using the non-dimensionless variables, the Reynolds number and Froude number are identical to those calculated by the physical variables in Eq.(2-90a,b).

The third step is to determine the discretization scales of spatial interval and time step. Across the characteristic length, we should determine how many lattices to depict it. To simulate the characteristic time, how many time steps should be applied. Here, the reference length $L_0 = 1$ is represented by 50 lattices (cell based), which indicates that there exist 50 lattice across the diameter of the cylinder. The reference time $T_0 = 1$ is simulated by 2500 time steps, which indicates that 2500 time steps are identical to the real reference time length. The

Chapter 2 Basic theory of lattice Boltzmann equation

unit transformation between the non-dimensionless system and the lattice system can be achieved by the following relationships:

$$L_{LBM} = \delta_x = 1/50 = 0.02 \quad (2-92a)$$

$$T_{LBM} = \delta_t = 1/2500 = 0.0004 \quad (2-92b)$$

$$U_{LBM} = U_0 \cdot T_{LBM} / L_{LBM} = 0.02 \quad (2-92c)$$

$$\nu_{LBM} = \nu_0 \cdot T_{LBM} / (L_{LBM} \cdot L_{LBM}) = 1.252 \cdot 10^{-5} \quad (2-92d)$$

$$g_{LBM} = g_0 \cdot (T_{LBM} \cdot T_{LBM}) / L_{LBM} = 2.4 \cdot 10^{-5} \quad (2-92e)$$

This may be a little bit nonrepresentational to be understood. How large is the physical length of a lattice unit after two-step conversion? One have to multiply the first scale factor and the second scale factor together, that is $\delta_x \cdot L_{0,p} = 0.0025m$, which indicates that one lattice is identical to 0.0025m in the real world. To be followed, the conversion for the commonly used variables will be elaborated.

Velocity

The unit of velocity is [m/s]. The conversion of the physical system and the lattice system can be expressed as

$$U_{LBM} = U_{ph} \cdot T_{0,p} / L_{0,p} \cdot \delta_t / \delta_x = U_{ph} \cdot (T_{0,p} \delta_t) / (L_{0,p} \delta_x) \quad (2-93)$$

Viscosity

The unit of the viscosity is [m²/s]. The conversion of the physical system and the lattice system can be obtained by

$$\nu_{LBM} = \nu_{0,p} \cdot T_{0,p} / (L_{0,p} \cdot L_{0,p}) \cdot \delta_t / (\delta_x \cdot \delta_x) = \nu_{0,p} \cdot (T_{0,p} \delta_t) / (L_{0,p} \delta_x \cdot L_{0,p} \delta_x) \quad (2-94)$$

It is noteworthy to be addressed that the lattice viscosity will be very small in the high Reynolds number case and the eddy viscosity should be incorporated to stabilize the simulation.

Gravity

In the free surface flow simulations, the gravity is of significance. The conversion of the physical system and the lattice system can be realized by

$$g_{LBM} = g_{0,p} \cdot (T_{0,p} \cdot T_{0,p}) / L_{0,p} \cdot (\delta_t \cdot \delta_t) / \delta_x = g_{0,p} \cdot (T_{0,p} \delta_t \cdot T_{0,p} \delta_t) / (L_{0,p} \delta_x) \quad (2-95)$$

Force

The impact force is usually evaluated in the hydrodynamic numerical simulations. We use the formula $F = ma = \rho Va$ to scale it. The conversion of the physical system and the lattice system can be expressed as (2D case)

$$F_{physical} = F_{LBM} (\rho_{LBM} \delta_x \cdot \delta_x \cdot \delta_x / \delta_t) \cdot (\rho_p L_{0,p} \cdot L_{0,p} \cdot L_{0,p} / T_{0,p}) \quad (2-96)$$

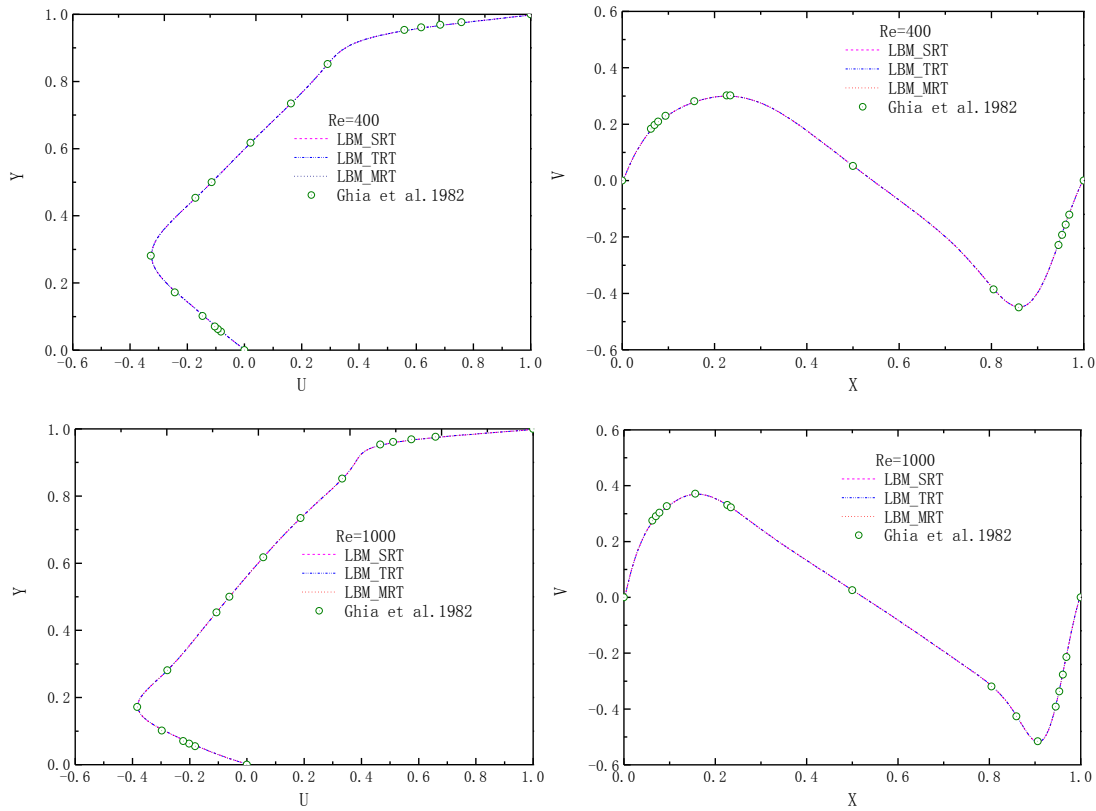
2.8 Benchmark Validation

2.8.1 2D Cavity Flow

The 2D lid-driven flow in a square domain is a common and standard test case for verification of the lattice Boltzmann flow solver based on different collision models. The fluid in the square domain is driven by the translation of the top plate with uniform velocity $U_c=0.1$. The non-slip boundary condition is implemented by the half-way bounce-back scheme on the left, right and bottom boundary of the square domain. Three collision models are tested with $Re=400, 1000, 5000, 7500$. For the MRT collision model, the fluid domain is resolved by the resolution of [256,256] for all Reynolds number. For the TRT collision model, the fluid domain is resolved by the resolution of [256,256] for $Re=400, 1000, 5000$ and [512,512] for $Re=7500$, which is not stable for the previous resolution. For the SRT collision model, the fluid domain is resolved by the resolution of [256,256] for $Re=400, 1000$ and [512,512] for $Re=5000, 7500$ for the consideration of the stability. The velocities of x-direction component along the vertical centerline and the y-direction component along the horizontal centerline calculated by different collision model are compared with the reference data by Ghia U. et al. (1982). As shown in Fig.2.13, it can be found that the numerical results simulated by different collision model agree well with the reference data. However, the discrepancy exists in the stability for different collision model. For the MRT collision model, the resolution of [256,256] is thoroughly stable for $Re=400, 1000, 5000, 7500$ and even 10000. However, the resolution of [256,256] is not stable at $Re=7500$ for TRT collision model and $Re=5000$ for SRT collision model. The

Chapter 2 Basic theory of lattice Boltzmann equation

increasing resolution will increase the numerical viscosity by the relationship $\nu = UN\delta_x / \text{Re}$, with the fixed Reynolds number and plate velocity, which can explain the increasing stability with increasing resolution for the SRT and TRT model. In Fig.2.14, the streamline and vorticity are depicted for different Reynolds numbers ($\text{Re}=400, 1000, 5000, 7500$). With the increasing Reynolds number, the flow field becomes more and more complicated. The primary and secondary vortex can be found with $\text{Re}=400, 1000$ at the center and bottom of the flow field. With $\text{Re}=5000$, the secondary vortex emerges at the left-top corner while the tertiary vortex emerges at the right-bottom corner when $\text{Re}=7500$. Similarly, the vorticity intensity is observed to be stronger and stronger with the increasing Reynolds number. All the simulations are conducted by the GPU version code and the code with MRT collision model (1024×1024) achieves the speed of 1146 MNUPS (million nodes updated per second).



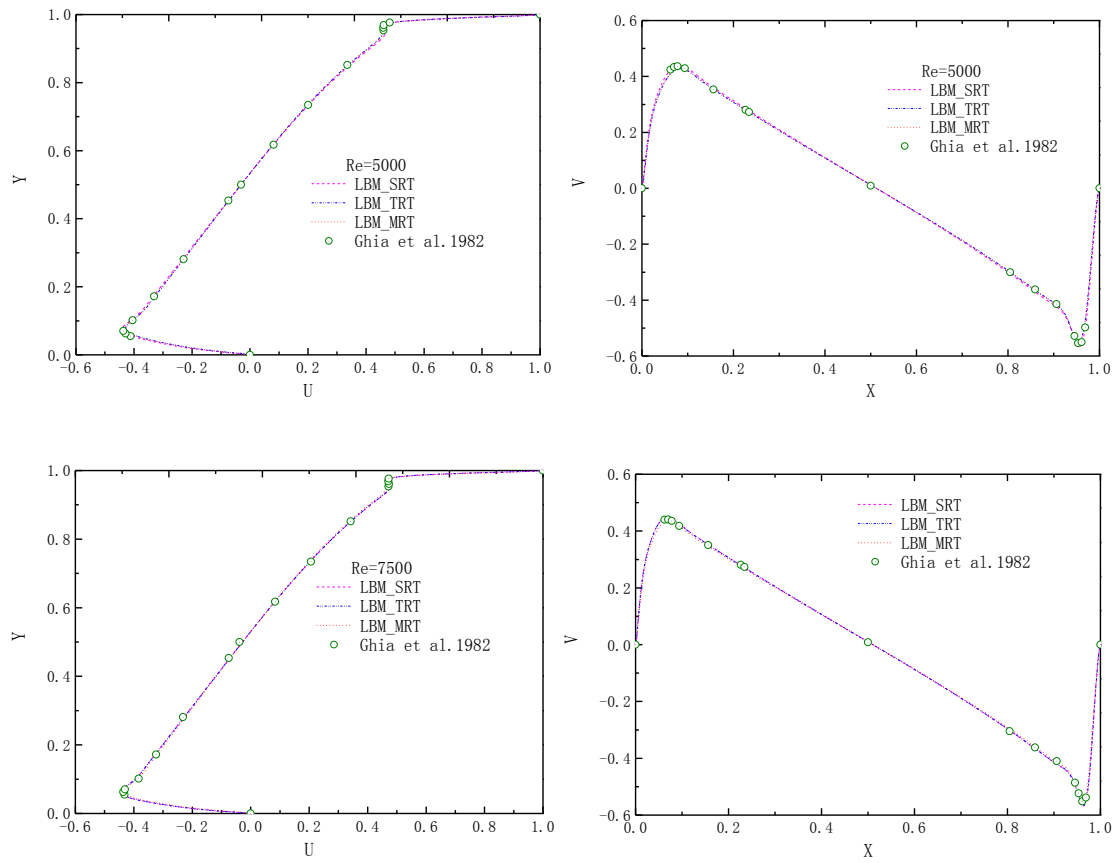
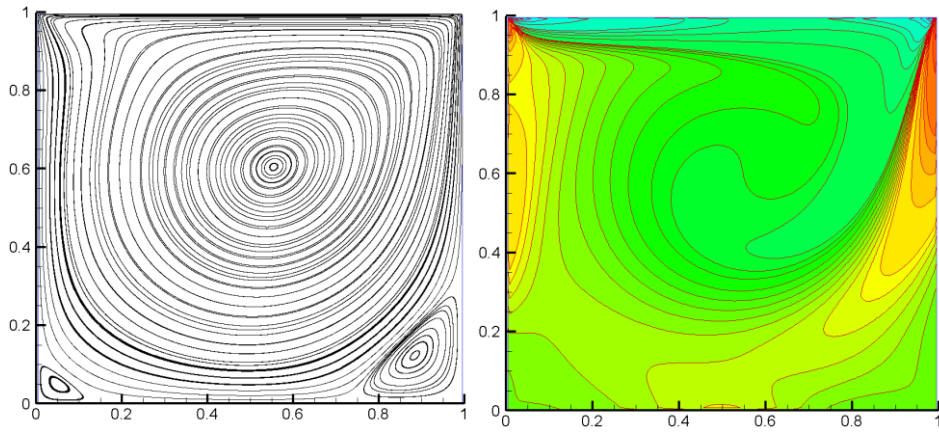


Fig.2.13 Velocity profile for $Re=400, 1000, 5000, 7500$



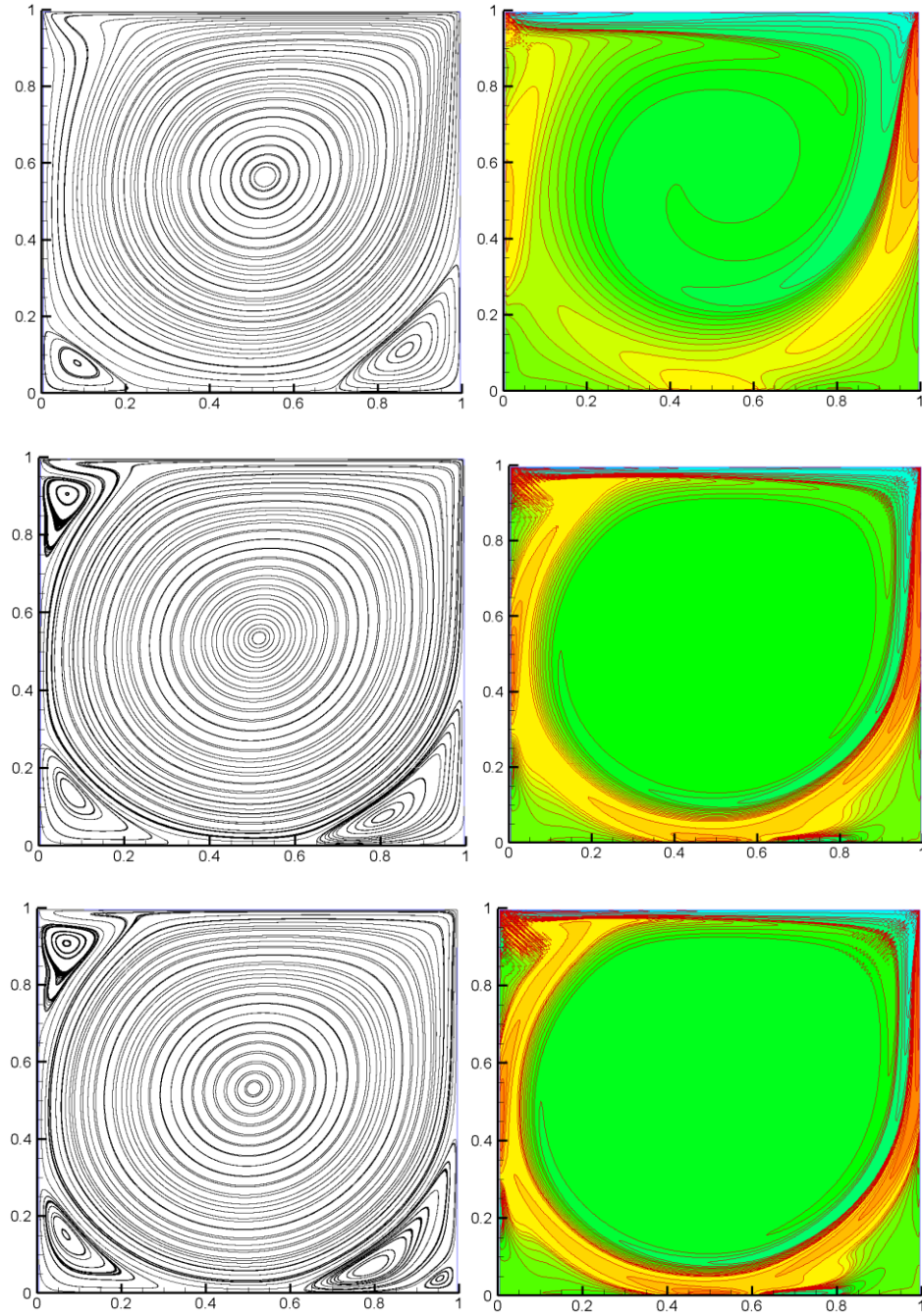


Fig.2.14 Streamline and vorticity calculated by MRT model at $Re=400, 1000, 5000, 7500$ (from top to bottom)

2.8.2 2D Taylor-Green Vortex

To validate the significance of initialization scheme, the simulation of unsteady Taylor-Green vortex case is carried out here. The 2D Taylor-Green vortex flow in a square domain has the analytical solution as followings:

$$u(x, y, t) = -U_0 \cos(kx) \sin(ky) e^{-2k^2 \nu t} \quad (2-97a)$$

$$v(x, y, t) = U_0 \sin(kx) \cos(ky) e^{-2k^2 \nu t} \quad (2-97b)$$

$$p(x, y, t) = P_0 - \frac{1}{4} U_0^2 [\cos(2kx) + \cos(2ky)] e^{-4k^2 \nu t} \quad (2-97c)$$

where the wave number is $k = \frac{2\pi}{L}$. The reference pressure is set as $P_0 = 0$.

To evaluate the effectiveness of the initial scheme, the normalized total kinetic energy and the normalized projection of the numerical pressure field are applied as followings:

$$K(t) = \frac{2}{N^2 U_0^2} \int_{\Omega} (u^2 + v^2) dx dy \quad (2-98)$$

$$P_2(t) = -\frac{16}{N^2 U_0^2} \int_{\Omega} p(x, y, t) \cos[k(x+y)] \cos[k(x-y)] dx dy \quad (2-99)$$

The periodic boundary conditions are applied in both x and y directions. The resolution is set to be [128,128] and shear viscosity is set to be 0.001 and 0.04. The relaxation rates for other modes are set as: $s_q = 1.7$, $s_e = 1.0$, $s_\varepsilon = 1.4$, $s_\chi = 1.0$ (for mass and momentum). Three initialization schemes are compared with the initial velocity field specified by Eq.(2-97a) and (2-97b): (a) the pressure field is initialized by the constant value and $f_\alpha = f_\alpha^{eq}$; (b) the pressure field is initialized by the analytical solution that is specified in Eq.(2-97c) and $f_\alpha = f_\alpha^{eq}$; (c) the pressure field and distribution functions are initialized by the iterative scheme proposed by Mei et al.(2006).

In Fig.2.15 and Fig.2.16, it can be seen that the initialization scheme (a) is suffered from the severe acoustic oscillation for both pressure field and total energy in the initial stage of the simulation. For the initialization scheme (b), the analytical pressure field is applied in which the correct equilibrium distribution functions f_α^{eq} are initialized. Thus the acoustic oscillation in the pressure field is eliminated while the deficit and oscillation for the total energy is retained, which is demonstrated to be the lack of the high-order non-equilibrium of the distribution function (Mei et al. 2006). With the iterative initialization scheme (c), the oscillations of the pressure field and total energy are completely eliminated, which indicates the significance of

Chapter 2 Basic theory of lattice Boltzmann equation

the consistence in the initialization process. Meanwhile, considering the discrepancy of Fig.2.15 and Fig.2.16, the inaccurate initialization approach will affect the high Reynolds number flow more remarkably. Fig.2.17 shows the pressure contour and velocity magnitude together with the vector plots at $T=1000$ with the iterative initialization scheme.

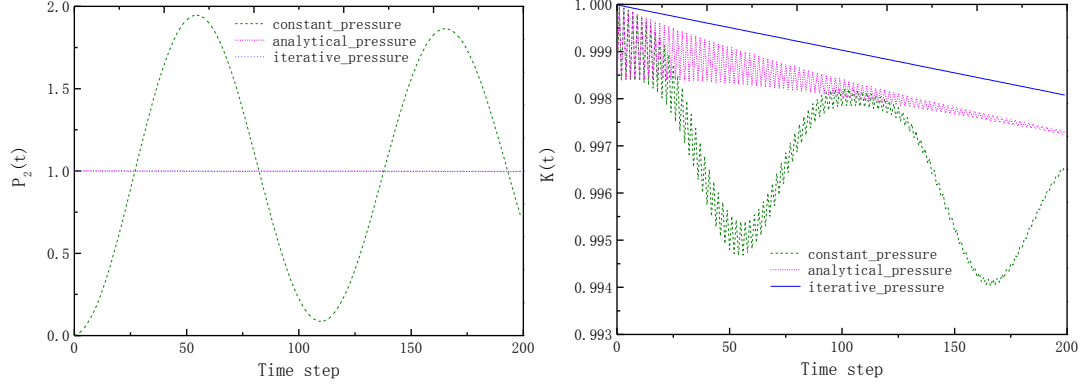


Fig.2.15 $P_2(t)$ (left) and $K(t)$ (right) with $\nu = 0.001$

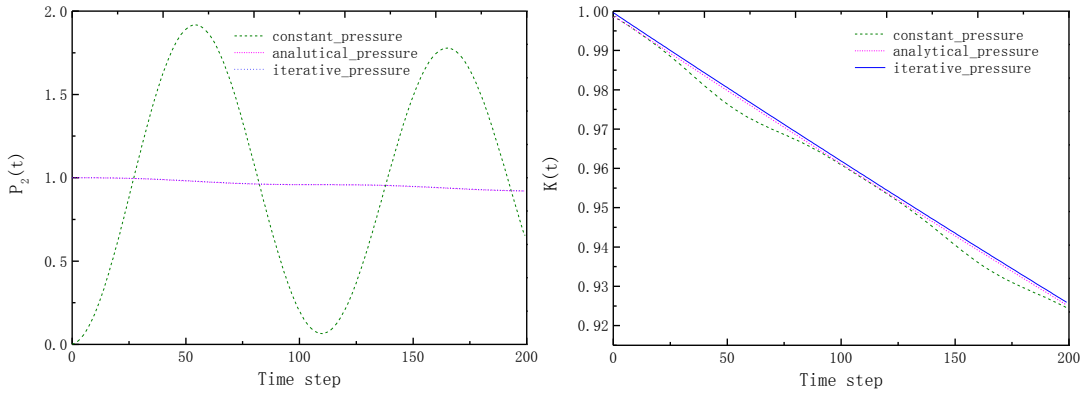


Fig.2.16 $P_2(t)$ (left) and $K(t)$ (right) with $\nu = 0.04$

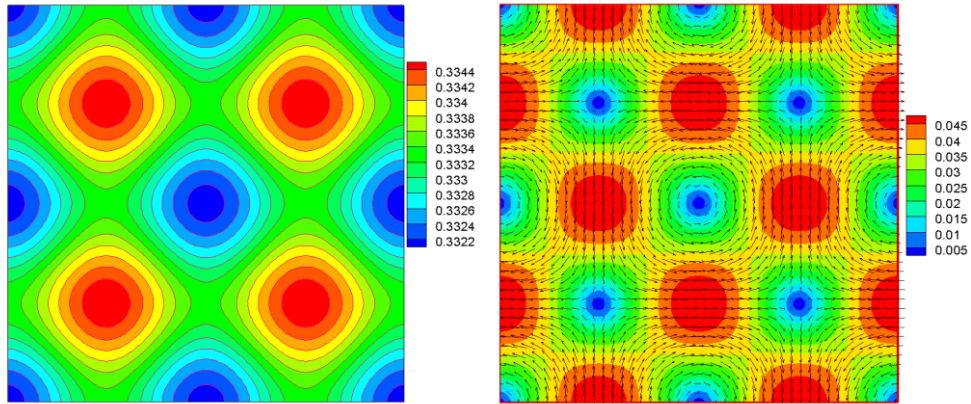


Fig.2.17 Pressure contour (left) and velocity magnitude, vector plots (right) at $T=1000$ with $\nu = 0.001$

2.8.3 Eccentric circular cylinder in a channel

The geometric arrangement of the flow field is shown in Fig.2.18. The boundary condition on the upper and bottom wall is no-slip condition that can be realized by Guo's non-equilibrium extrapolation scheme, the inlet is parabolic velocity condition and the outlet is constant pressure (initial pressure) condition that can be realized by Zou-He boundary scheme. The inlet parabolic velocity is

$$u(0, y) = 4u_{\max} \frac{y(H-y)}{H^2}, v = 0 \quad (2-100)$$

The average of the parabolic velocity $U = 2u_{\max}/3$ is taken as the characteristic velocity and the diameter of circular cylinder the characteristic length. The Reynold number is $Re = UD/\nu = 20$ and Mach number is $Ma = u_{\max}/c_s$. The fluid force exerted on the cylinder is written in dimensionless expressions:

$$C_d = \frac{F_x}{0.5\rho U^2 D}, C_L = \frac{F_y}{0.5\rho U^2 D} \quad (2-101)$$

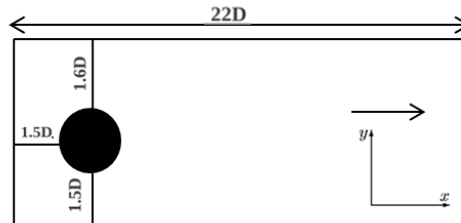


Fig.2.18 Eccentric circular cylinder flow

Table 2.1 Drag and lift coefficients (n=40 across the diameter)

Schemes	Cd	Cl
PSC	5.70	0.0103
DF_Explicit	5.81	0.0116
DF_Implicit	5.81	0.0125
NS solver (M. Schafer et al.)	5.57-5.59	0.0104-0.0110

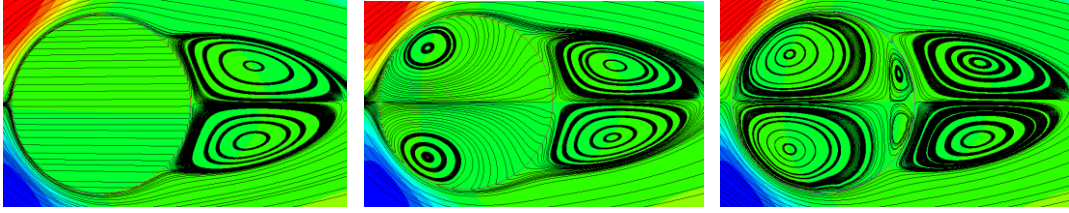


Fig.2.19 Streamline around the cylinder for different schemes

As shown in Table 2.1, it can be found that the drag and lift coefficients calculated by three diffuse immersed boundary schemes agree with those of the NS solver (M. Schafer et al., 1996) well. However, the streamlines penetrate the physical boundary for PSC and explicit direct-force diffuse immersed boundary scheme but the implicit direct-force diffuse immersed boundary scheme performs well, in which the non-slip boundary condition is strictly satisfied.

2.9 Concluding Remarks

In this chapter, the basic theory of the continuous Boltzmann equation, lattice Boltzmann equation and the body force model are described firstly. Subsequently, the initial condition and various boundary conditions are introduced for the preparation of the work in the following chapters. Then the application of LES turbulence model is introduced to be capable of the high Reynolds number flow. Finally, the unit transformation in the lattice Boltzmann simulation is elaborated. Three benchmarks have been conducted to verify the collision models, initial conditions and boundary conditions. The turbulence model and unit transformation will be addressed in chapter 5.

Chapter 3 GPU implementation of the lattice Boltzmann model

3.1 Introduction

Within the past decade, the applications of the modern General Purpose Graphics Processor Units (GPU) in high performance computation have fulfilled explosive achievements. With the features of low power consumption, high memory-bandwidth and intensive computational capability, GPU technique has been widely applied in both the industry field and the scientific calculation, such as oil and gas exploration, finance analysis, medical imaging, genetic analysis, computational fluid dynamics. Benefit from the application programming interface which is so-called Compute Unified Device Architecture (CUDA) released by NVIDIA, the parallel programming has become extremely friendly and easy for programmers to finish general purpose calculation. In the community of computational fluid dynamics, GPU acceleration technique has been quickly applied in the molecular dynamics, smooth particle hydrodynamics, lattice Boltzmann method and magneto-hydrodynamics. In the present work, the lattice Boltzmann models including basic benchmark validations for basic theory, fluid-solid interaction schemes, single phase free surface flow model and multiphase lattice Boltzmann model are implemented by CUDA C/C++ code.

In this chapter, the basic concepts in the NVIDIA CUDA GPU programming platform are introduced firstly. Next, the CUDA implementation of the lattice Boltzmann model will be elaborated. Subsequently, the multi-GPU technique will be described.

3.2 Overview of the CUDA programming platform

In order to make it easy to understand the after-mentioned programming techniques in the lattice Boltzmann simulation, it is necessary to briefly describe the hardware architecture and CUDA programming. More detailed information can be found in Programming guidance of NVIDIA Toolkit.

3.2.1 Overview of the hardware

Chapter 3 GPU implementation of the lattice Boltzmann model

As can be seen in Fig.3.1, most transistors are designed for data process and small proportion transistors are used for flow control and cache. In contrast, a huge number of transistors execute the function of cache and flow control while only small part of them are designed for the algorithm operation. The philosophy of the discrepancy of hardware architecture determines the features of them. GPU is especially well-suited to resolve the problems which can be expressed as data-parallel computations with high arithmetic intensity but low requirement for sophisticated flow control. Meanwhile, it utilizes the high arithmetic intensity to hide the memory latency for the lack of sophisticated cache mechanism. In this way, for the tens of thousands of light-weight cores, GPU results in more tremendous computational horsepower and much higher memory bandwidth than CPU. The comparison of them can be found in Fig.3.2 and Fig.3.3. As mentioned in the work of V. W Lee et al. (2010), the numerical simulation of lattice Boltzmann model is usually memory bandwidth bound. Therefore, it is extremely suitable to throw the calculation of lattice Boltzmann model onto the GPU card.

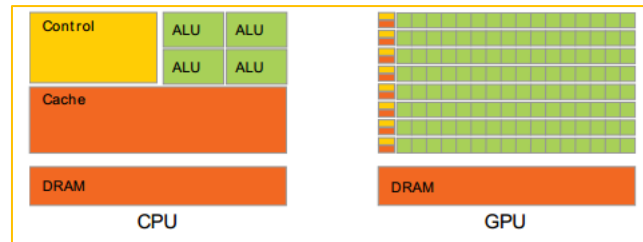


Fig.3.1 Transistors distribution for data process and flow control (CPU vs GPU) (NVIDIA Toolkit)

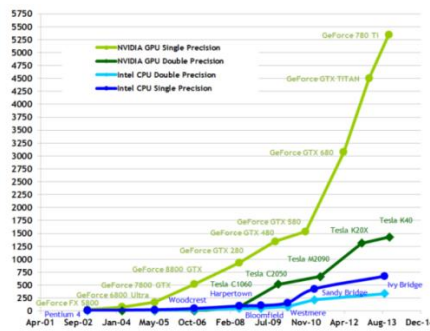


Fig.3.2 Floating-point operations per second for CPU and GPU (NVIDIA Toolkit)

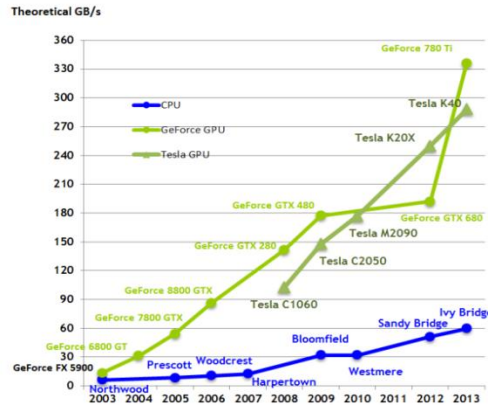
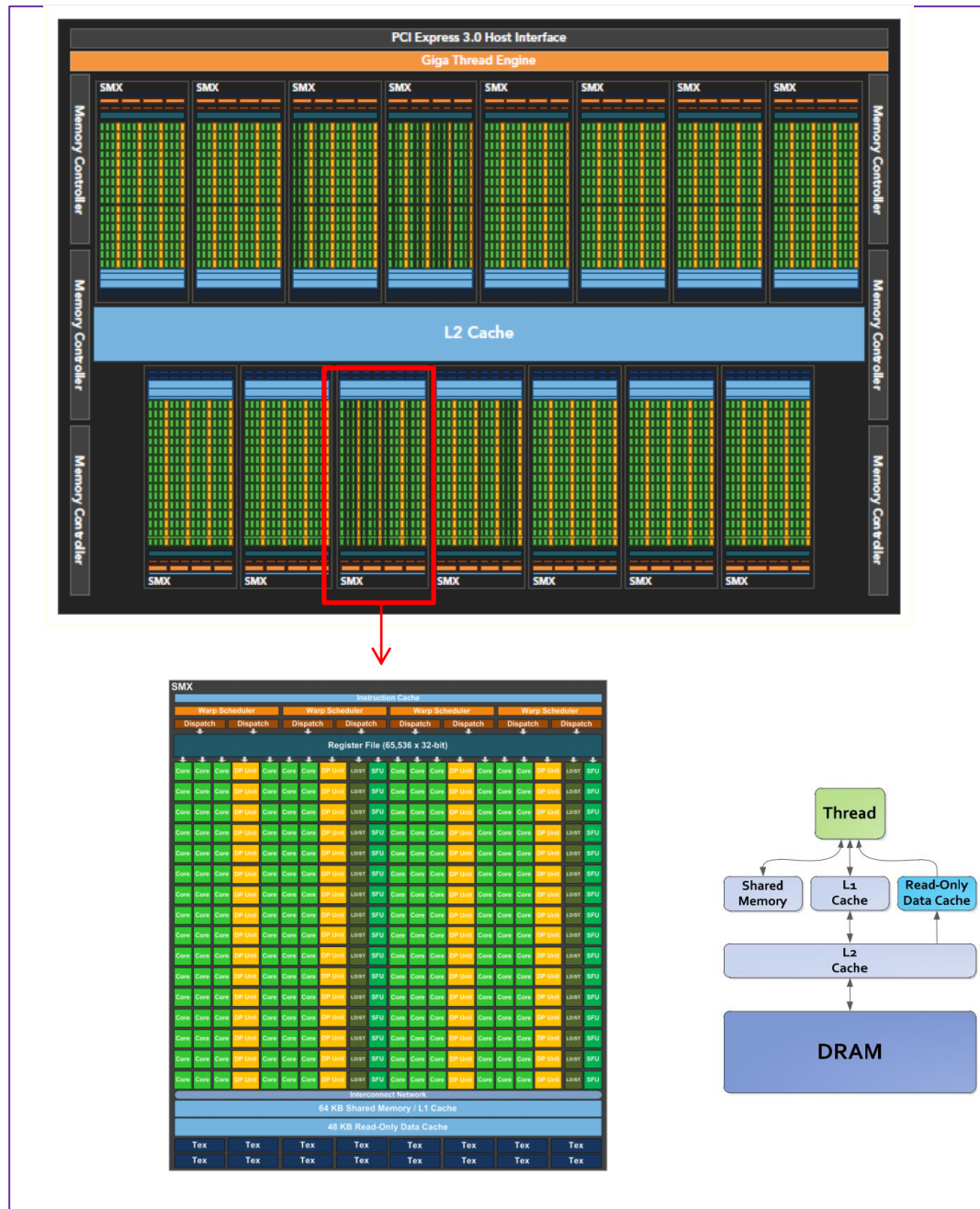


Fig.3.3 Memory bandwidth for the CPU and GPU (NVIDIA Toolkit CUDA v7.0)

Although there exist new generations of Maxwell and Pascal GTX GPU card, considering the double precision capability, robustness and ECC function and so on, a rough schematic of the current powerful Tesla series card – K40 is introduced here, as shown in Fig.3.4. The specification of K40 is elaborated referred to NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf. This upgraded device card consists of 15 stream multi-processes (SMX), within which 192 stream processes are equipped. To be detailed, there are 512 function units per SMX, including 192 units for algorithm logical operation or float point operation for single precision (green block), 64 float point units for double precision (orange block), 32 units for load/store (gray block), 32 units for spherical functions (dark green block). In total, K40 holds 2880 CUDA cores and 12 GB device memory. The device is connected to the host (CPU) through PCI-e 3.0 bus. The Giga-Thread global scheduler distributes thread blocks to SM thread schedulers. It is noteworthy that the memory hierarchy of GPU is more sophisticated than that of CPU. The memory types can be divided into two categories: on-chip memory and off-chip memory. The off-chip memories include global memory, constant memory, local memory and texture memory, while the on-chip memory consists of register file and shared memory/L1 cache. The global memory can be accessed by any stream process but with high latency. The constant memory is a kind of off-chip memory but equipped with cache, which can be used to store some data that is visited frequently but only read. The on-chip register file can be loaded extremely fast (1 clock cycle) and the maximum register file for each SMX is 64K. The shared memory can be also loaded efficiently but only can be visited by the stream processes in the same SMX. The size of shared memory and L1 cache can be switch. The local variables will be overflowed to off-chip local

Chapter 3 GPU implementation of the lattice Boltzmann model

memory once the on-chip register files are not enough, which will induce the performance crash. The texture memory is not used in the present simulations. Being familiar with the memory loading speed and using constraint condition will facilitate the CUDA programming.



language, which facilitates the programmer to make the parallel GPU code. CUDA C/C++ enriches the C/C++ language by incorporating some syntax and keywords together with an extensive application interface (API) enabling the code to be capable of device selection, memory allocation and copy, synchronization and so forth. The functions running on GPU are named kernels in CUDA C/C++ which are denoted by the `__global__` qualifier, as shown

```
__global__ void kernel_gpu <<< grid,block,smem,stream_id >>> (int a,double *b,double *c)
```

Different from the functions in C/C++, some parameters should be initialized when the kernel is launched onto the GPU. As mentioned in the aforementioned context, the GPU device is equipped with a huge number of cores, each of which executes a number of threads. The huge number threads are organized in a two-level parallelism mechanism. To be first, the threads are grouped in thread blocks. In a thread block, the threads are organized in the three dimensional way, which indicates that the thread in one block can be indexed by three coordinates, that is `threadIdx.x`, `threadIdx.y`, `threadIdx.z`. The number of threads in each direction of the block can be expressed by the build-in variables `blockDim.x`, `blockDim.y`, `blockDim.z`. Due to the rare register files and shared memory that can be used within the same block, the size of the block should be chosen in a suitable way, usually the multiple of 32 but less than the maximum 1024. Then the blocks are bundled in one grid, in which the block can be located by the three dimensional index, that is `blockIdx.x`, `blockIdx.y`, `blockIdx.z`. Similarly, the number of the blocks in each direction of the grid can be expressed by the build-in variables `gridDim.x`, `gridDim.y`, `gridDim.z`. The hierarchy of threads and its relationship with the hardware can be seen in Fig.5. The configuration of the first two parameters in the kernel parameters list can be declared as

```
dim3 grid(gridDim.x, gridDim.y, gridDim.z);  
dim3 block(blockDim.x, blockDim.y, blockDim.z);
```

The third parameter is the number of the shared memory and the fourth parameter is the stream ID. Due to the slave properties of the GPU, the memory allocation task should be conducted on the host (CPU) and the initialized variables have to be copied from the host

Chapter 3 GPU implementation of the lattice Boltzmann model

variables to the corresponding device variables. Meanwhile, since the device has no IO (input and output) mechanism, the intermediate results should be copied from the device to the host and then output from the host to the disk files. The memory allocation and memory copy operations are fulfilled by the following CUDA functions:

```
double *g, *g_gpu;
```

host end:

```
g = (double *) malloc( sizeof(double)*meshsize ); for non page-locked memory  
or cudaMallocHost((void**) &g, sizeof(double)*meshsize ); for page-locked memory
```

device end:

```
cudaMalloc ( (void**) &g_gpu, sizeof(double)*meshsize );
```

copy operation:

```
cudaMemcpy(g, g_gpu, sizeof(double)*meshsize, cudaMemcpyHostToDevice);  
cudaMemcpy(g_gpu, g, sizeof(double)*meshsize, cudaMemcpyDeviceToHost)
```

The copy operation between host and device is very time-consuming even the PCI-e bus bandwidth achieves 10 GB/s for K40. The copy operation will be more efficient if the page locked memory in the host is allocated for the variables.

In Fig.6, the kernel executions in Tesla/Fermi architecture and those in Kepler architecture are compared. In the early hardware architecture, the kernel can be only launched in the host function. However, for the Kepler architecture in the kernel the threads can also launch kernels with its own configuration, which improve the flexibility remarkably, especially for the adaptive mesh refinement technique in the lattice Boltzmann model.

To make it pellucid, the relationship of the software concepts thread, block and grid with the hardware SMX should be addressed here. One kernel or several kernels (at most 32 kernels for K40) can be launched onto the GPU card simultaneously. Each SMX on the GPU card will execute several blocks (maximum 16 blocks for K40). The arithmetic executions of each SMX will be conducted simultaneously. Within each block, each 32 threads actually are organized in one wrap. The threads in one wrap are executed by the same instruction. This is the so-called single instruction and multi-data process. If there exist branches in one wrap, such as the “if” command, the wrap will be conducted the number of the branches. Meanwhile, it had better to

access the continuous memory by one wrap. The concept of wrap can be described by a historical device in China, as shown in Fig.3.5. The arithmetic operation works are finished by ALUs, FPU, SFUs and so on. Based on the features of the hardware architecture of GPU, the threads within one block can access and exchange information on the shared memory and can be synchronized by the barrier operation, while the threads belong to different blocks can only exchange data through the high latency global memory. Therefore, the data transaction with the global memory should be reduced as less as possible, which can be realized by the temporary register file or shared memory to store the data loaded from the global memory and reuse them again and again or write the data into the global memory once after all the operations. Furthermore, since the warp will load or store the data on the global memory with the same instructor, the coalescence memory access is extremely significant to improve the hit-rate of the cache or write efficiency.

In general, the following principal should be kept in mind for the CUDA programming: a) reduce the global memory load/store as much as possible; b) the coalescence memory access pattern should be achieved; c) make full use of the shared memory and the register files; d) choose suitable block size and avoid the local variables overflowed to local memory; e) reduce the data communication between host and device.

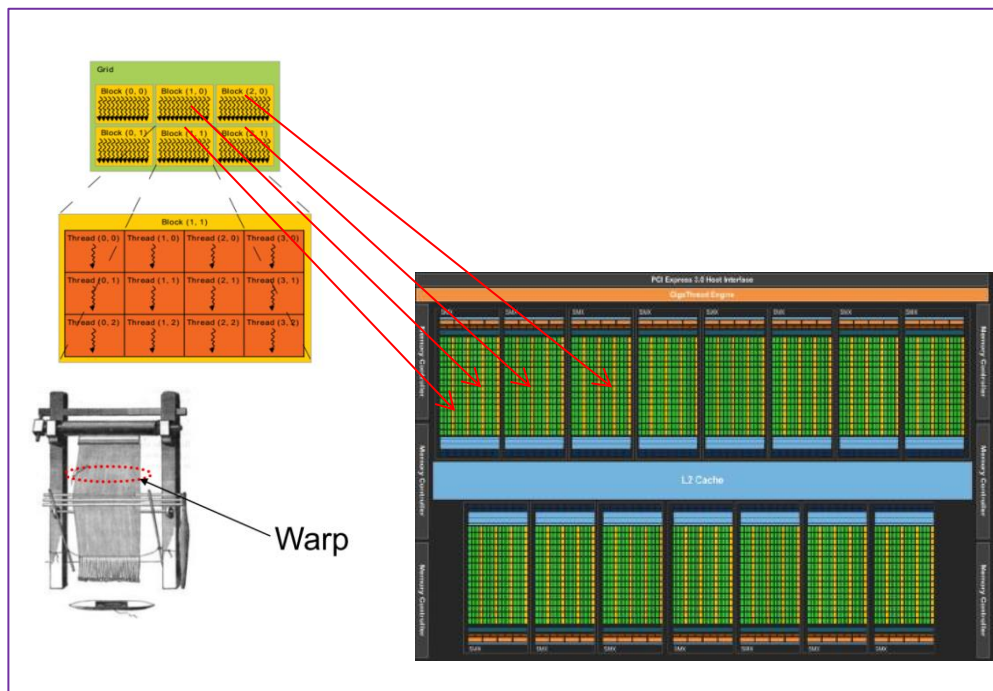


Fig.3.5 Hierarchy the threads in the CUDA programming and its relationship with hardware

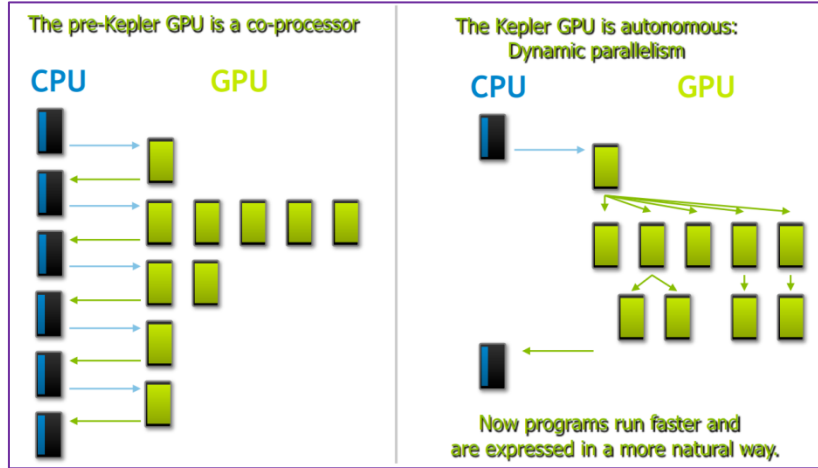


Fig.3.6 CUDA programming sketch

3.3 CUDA implementation in the lattice Boltzmann solver

It is interesting that in the lattice Boltzmann model the nonlinear collision term is completely local while the nonlocal advection term is linear. The nonlinear advection term in the traditional CFD is realized by the perfect shift with machine accuracy in the lattice Boltzmann model. The linear diffusion term in the traditional CFD is nonlinear in lattice Boltzmann model but is completely local. Meanwhile, the velocity field can be obtained by the momentum of the distribution functions and the pressure field is evaluated explicitly without solving the pressure Poisson equation. Furthermore, almost all the calculations in the lattice Boltzmann simulation are simple algebraic manipulations. As investigated by V. W Lee et al., the numerical implementation of lattice Boltzmann model suffers from the memory bandwidth bound, while the memory bandwidth of GPU is much higher than the CPU. Both the features of lattice Boltzmann model and the GPU hardware architecture make it natural to implement the lattice Boltzmann code on the GPU.

3.3.1 Corresponding relationship between threads and the fluid nodes

To launch the kernel onto the device, we should establish reasonable corresponding relationships between the thread indexes and the fluid nodes. The two dimensional case and the three dimension case will be discussed separately. For the two dimensional case, it is direct without the limitation of one dimensional configuration of the block and grid. Usually, two different configurations are applied in our CUDA code. One is to specify the configurations of blocks and grids in two-dimensional way. The grid is arranged as `grid(gridDim.x, gridDim.y, 1)`

and the block is configured as $\text{block}(\text{blockDim.x}, \text{blockDim.y}, 1)$ with $\text{gridDim.x} \times \text{blockDim.x}$ equal to the fluid nodes in x direction and $\text{gridDim.y} \times \text{blockDim.y}$ equal to the fluid nodes in y direction. The indexes of the fluid nodes in two directions can be expressed by the indexes of the threads as

$$j = \text{blockDim.y} \times \text{blockIdx.y} + \text{threadIdx.y}; \quad (3.1a)$$

$$i = \text{blockDim.x} \times \text{blockIdx.x} + \text{threadIdx.x}; \quad (3.1b)$$

Another way is purely one dimensional way, in which both the block and grid are arranged in one dimension. The configurations of grid and block are $\text{grid}(\text{gridDim.x}, 1, 1)$ and $\text{block}(\text{blockDim.x}, 1, 1)$. The corresponding relationship of the fluid nodes and the threads can be written as

$$j = \text{int}((\text{blockDim.x} \times \text{blockIdx.x} + \text{threadIdx.x}) / \text{xDim}); \quad (3.2a)$$

$$i = \text{blockDim.x} \times \text{blockIdx.x} + \text{threadIdx.x} - j \times \text{xDim}; \quad (3.2b)$$

In this way, it is more efficient and flexible to handle the corresponding relationship of the fluid nodes and the threads. However, in the three dimensional simulation, the one dimensional configuration might be not enough for the extremely huge resolution. The thread configuration for three-dimensional case is shown in Fig.3.7. The configuration of grid is $(\text{gridDim.x}, \text{gridDim.y}, 1)$, in which gridDim.x is equal to the number of the fluid nodes in y direction and gridDim.y is equal to the number of the fluid nodes in z direction. In this way, one fluid node in the y-z plane corresponds to a block. The configuration of the block can be one dimensional as $\text{block}(\text{blockDim.x}, 1, 1)$. The index of the thread threadIdx.x is equal to the node coordinate in x direction. If the number of the nodes in x direction is beyond the maximum of the threads number in a block or a moderate block size is selected, the block numbers in x direction can be extended. That means the configuration of the grid in z direction should be modified to be more than 1. Usually the minimum dimension of the fluid domain is chosen as the x direction and the maximum dimension of the fluid domain is chosen as the z direction. This choice is based on two considerations: one is to reduce the offset of the index increment in y and z direction when the non-local finite difference operators are conducted, the other point is to reduce the amount of the data transfer in the ghost layers when one-dimensional decomposition is conducted along the z direction in the multi-GPU implementation. By using this configuration, the memory access within a wrap can be coalesced and the cache hit rate will be increased. The following

data arrangement will demonstrate this viewpoint.

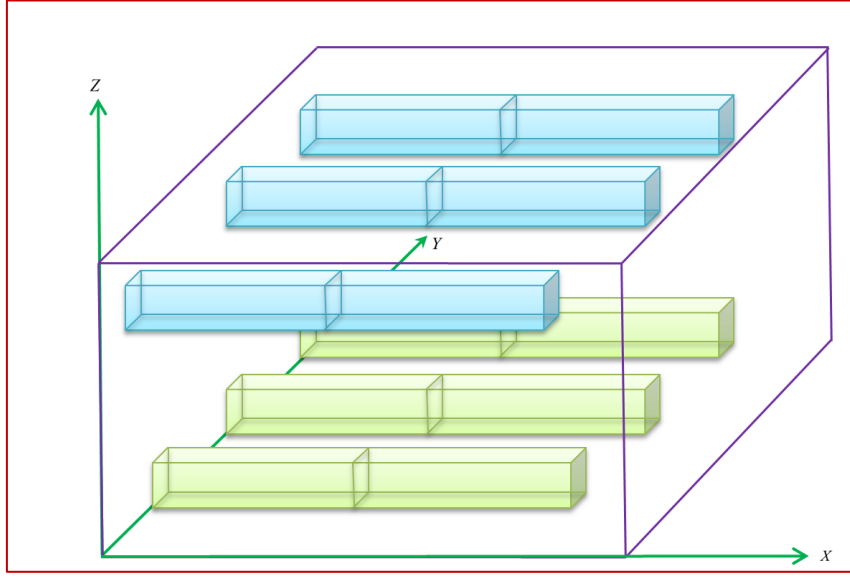


Fig.3.7 Threads map to the fluid nodes (each block maps to a fluid node in y and z direction, there are several one-dimensional blocks in x direction, each thread maps to a fluid node in x direction. The minimum dimension is selected as x dimension and the maximum dimension is selected as z dimension. Usually one block in x direction is enough.)

The data layout in the lattice Boltzmann simulation is of significance for the efficient GPU implementation. The following two arrangements are the usually applied data structure:

$$f[i*zDim*yDim*xDim+jz*yDim*xDim+jy*xDim+jx] \quad (3.3a)$$

$$f[jz*yDim*xDim+jy*xDim+jx+i] \quad (3.3b)$$

Let us analyze the discrepancy of the memory access of the above two layouts. In the lattice Boltzmann simulation, the collision process is executed as the follow expression (take the SRT model for example)

$$\begin{aligned} f[0*zDim*yDim*xDim+jz*yDim*xDim+jy*xDim+jx] = \\ f[0*zDim*yDim*xDim+jz*yDim*xDim+jy*xDim+jx]*(1-1/\tau)+feq[0]/\tau; \end{aligned}$$

$$\begin{aligned} f[1*zDim*yDim*xDim+jz*yDim*xDim+jy*xDim+jx] = \\ f[1*zDim*yDim*xDim+jz*yDim*xDim+jy*xDim+jx]*(1-1/\tau)+feq[1]/\tau; \end{aligned}$$

$$\begin{aligned} f[2*zDim*yDim*xDim+jz*yDim*xDim+jy*xDim+jx] = \\ f[2*zDim*yDim*xDim+jz*yDim*xDim+jy*xDim+jx]*(1-1/\tau)+feq[2]/\tau; \end{aligned} \quad (3.4)$$

.....

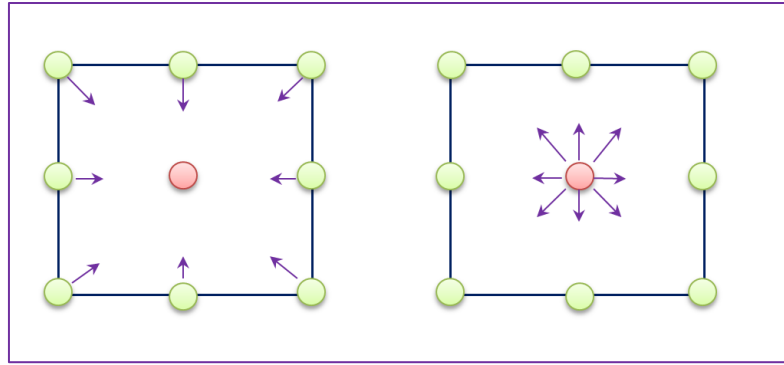


Fig.3.8 Perform the stream process in reading data (left) or writing data

From the viewpoint of the hardware, the wrap as a basic unit is scheduled and executed by the same instructor, which indicates that 32 threads will access the global memory together. If the memory addresses visited by the 32 threads are not continuous, the high latency of the global memory access may not be hidden and the cache hit rate will be extremely low. In this case, the bottleneck of memory access instead of the algorithm manipulations will hamper the efficient execution. If the data layout is selected as Eq.(3.3a), the collision operation in (3.4) is more efficient, because all the threads in a wrap can visit continuous memory when each command in Eq.(3.4) is executed. Nevertheless, if the data layout for the particle distribution functions is chosen as Eq.(3.3b), there always exists an offset when the threads in a wrap perform the same instructor. However, the data structure in Eq.(3.3b) performs better than that in Eq.(3.3a) for the CPU execution.

To choose the coalesced reading or coalesced writing in the stream process is another issue to be emphasized (Christian Obrecht et al., 2011). As shown in Fig.3.8, in the left pattern the stream process is performed in the data reading process of the global memory while the stream process is conducted in the data writing process in the right pattern. In the left pattern, the reading is not coalesced but the writing process is coalesced. In the right pattern, the reading operation is coalesced but the writing operation is not coalesced. There always exist non-coalesced memory fetch of the global memory in the stream process. Numerical experiments of the GPU code have demonstrated that the non-coalesced writing operation is more expensive than the non-coalesced reading operation. Thus, the left pattern is applied in the stream process for the present work.

In the finite difference operator such as central difference, biased upwind difference,

mixed form of them or even fifth-order WENO scheme, the nonlocal property is evident and the memory access will directly affect the efficiency of the GPU implementation. Take the isotropic central difference for example (see Fig.3.9), the surrounding eight nodes are applied to calculate the gradients. If the minimum dimension is chosen to be the x dimension, the offset to visit the neighboring nodes in y direction is xDim which is the size of x dimension. If the central difference is conducted in the three-dimensional domain, the neighboring nodes in z direction will be also accessed and the offset is $xDim*yDim$. This is why we choose the aforementioned strategy for the dimension size. Not only on the GPU programming but also on the CPU programming, the offset of the memory access should be noticed. In the fifth-order WENO scheme, the neighboring 3 nodes in each direction will be visited such that the offset is considerable in y and z directions. That is why the high-order scheme is not as efficient as the compact scheme on the GPU implementation, which also indicates the natural and intrinsic parallel property for the lattice Boltzmann model. However, the cache mechanism on GPU is not as advanced as that on CPU such that the coalesced memory access of the device memory is of significance.

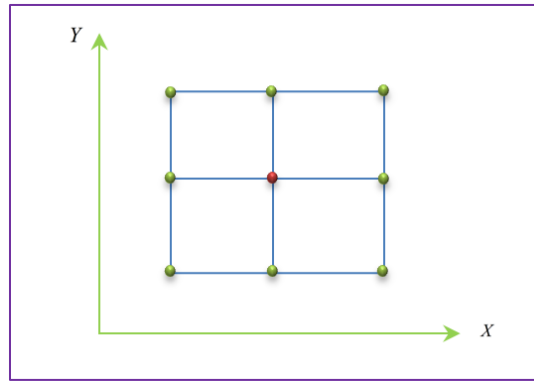


Fig.3.9 Finite difference operator in multiphase lattice Boltzmann model

In the present work, the structures of GPU codes vary with the lattice Boltzmann models. In Chapter 4, the general flow chart of the CUDA codes with different schemes for fluid-solid interaction is shown in Fig.3.10. In Chapter 5, the complicated interface capture algorithm has been applied to tackle the free surface flow. Thus the structure of the CUDA code is a little bit cumbersome. The details can be seen in Fig.3.11. In Chapter 6, the multiphase lattice Boltzmann model proposed by Abbas et al. (2016) will be applied to simulate the bubble dynamics. The memory access for this model is regular and most of the operations are local

except the stream and the finite difference operations in the collision term. The program structure can be found in Fig.3.12.

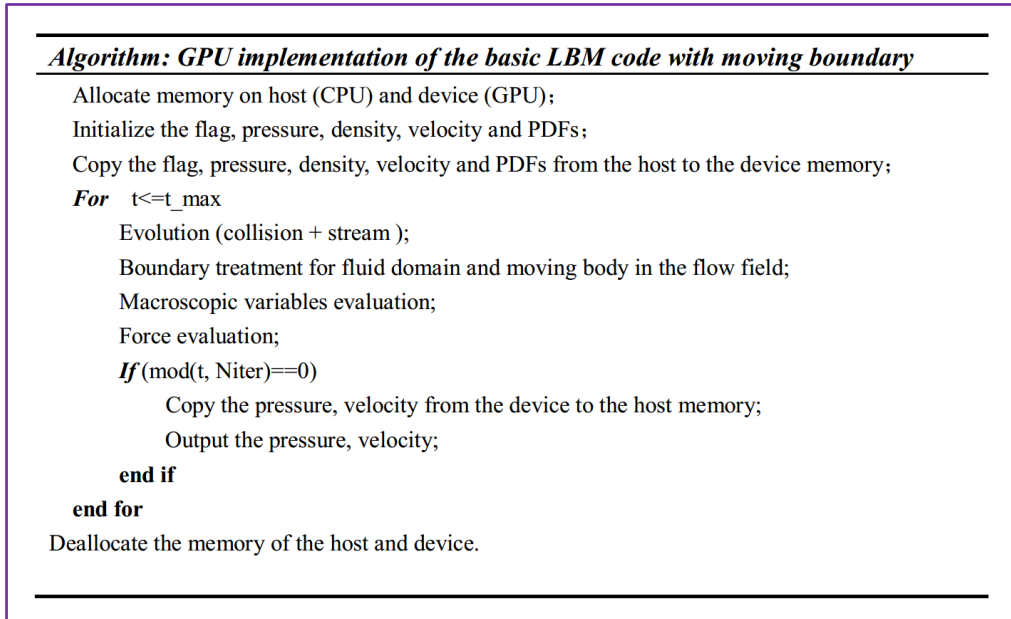


Fig.3.10 Algorithm for the GPU implementation of basic fluid-solid interaction

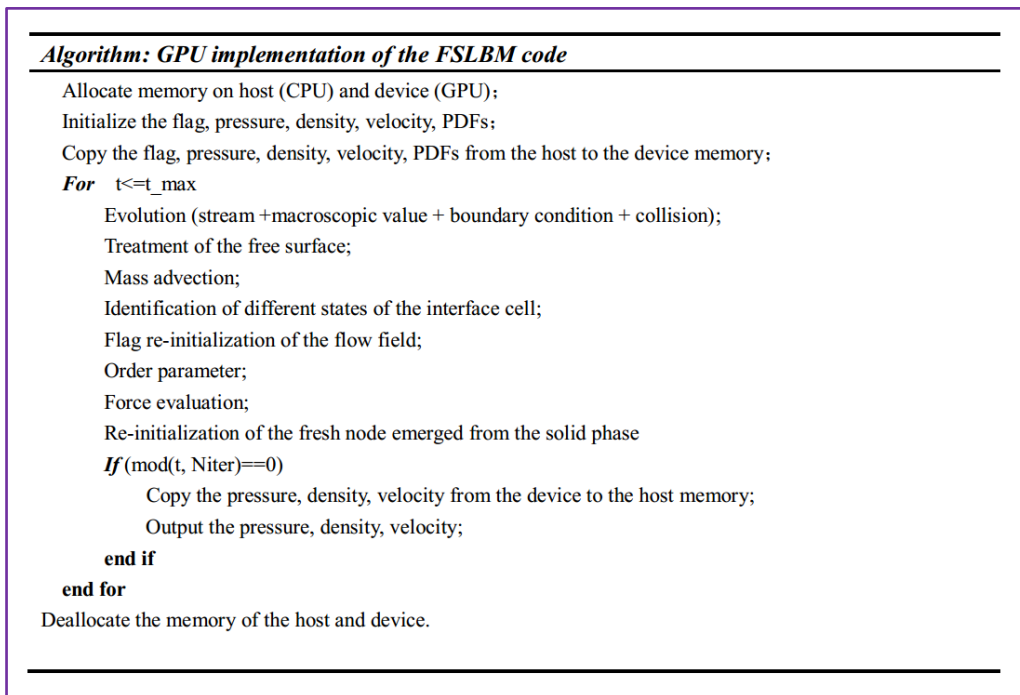


Fig.3.11 Algorithm for the GPU implementation of free surface flow

Algorithm: GPU implementation of the Multiphase lattice Boltzmann Model

```
Allocate memory on host (CPU) and device (GPU);
Initialize the flag, pressure, density, order parameter, velocity, chemical potential, gradients
of order parameter and normal vector of the interface, PDFs for phase field and flow field;
Copy them from the host to the device memory;
For t<=t_max
    Collision;
    Boundary condition for  $fbar$  and  $gbar$ ;
    Stream process;
    Order parameter and density evaluation;
    Boundary treatment for order parameter;
    Chemical potential evaluation;
    Gradients of order parameter and normal vector for interface;
    Velocity and pressure evaluation
    If (mod(t, Niter)==0)
        Copy the pressure, density, velocity from the device to the host memory;
        Output the pressure, density, velocity;
    end if
end for
Deallocate the memory of the host and device.
```

Fig.3.12 Algorithm for the GPU implementation of multiphase lattice Boltzmann model

3.3.2 Implementation examples

Among the structures of the above CUDA codes, there are some implementations that are of significance to be addressed. These treatments are frequently used in the programming of lattice Boltzmann models.

3.3.2.1 Force evaluation in LBM

The force evaluation approaches in the lattice Boltzmann models include the widely used momentum exchange method, stress integration method and the immersed boundary method. In these approaches, the reduce operation cannot be avoided. The force components across each lattice link in the momentum exchange method, the pressure and stress components on the discrete nodes of the boundary in the stress integration method, the force components on the Lagrange nodes of the boundary in the immerse boundary method should be summed as the total force. In the CPU serial code, this kind of operation is extremely simple and direct. However, the special treatment should be conducted to fulfill this operation on the GPU. In the

present work, the shared memory is applied to realize the summation operation. The synchronization function for the threads in one block and memory fence function are applied in the following two kernels. In the *Kernel 1*, the number of the force points should be less than the maximum of a block size (1024 for the present CUDA architecture). In this case, the force components can be completely loaded into the shared memory and reduced within one block. In the *Kernel 2*, the number of force points is not limited by the block size. The force components are distributed to many blocks. Each block sums part of the force components and then store the intermediate result into the global memory. The *threadfence* function is applied in this kernel. Its function is that the memory access to the global memory or the shared memory for the present thread is finished and the result is visible to other threads in the grid. The further reduce operation will be conducted by the last block which finishes the storage operation. If the number of the force points is too huge to be reduced with *Kernel 2*. Several similar kernels should be launched until the scale of the force components is less than the maximum size of a block.

This kind of reduce operation can be also applied in the search of the maximum value of some variables in the flow field, such as the maximum error in the fluid domain. In the Poisson equation solver, the present approach can be applied to check the maximum error in each iterative step but without the communication with the host which avoids the time-consuming copy operation.

Kernel 1:

```
extern __shared__ double sharef[];

//use only one block to calculate the summation
unsigned int tid = threadIdx.x;
// num_point is the number of force points
sharef[tid]=(tid<num_point)?f[tid]:0.0;
__syncthreads();
int offset=blockDim.x/2;
while(offset>0)
{
    if(tid<offset)
    {
        sharef[tid]+=sharef[tid+offset];
    }
}
```

Chapter 3 GPU implementation of the lattice Boltzmann model

```
offset>=1;
__syncthreads();
}
if(tid==0)
{
    // store the sum into the global memory
    *force=sharef[0];
}
```

Kernel 2:

```
extern __shared__ double sharef[];
```

```
unsigned int tid = threadIdx.x;
unsigned int i = blockIdx.x*blockDim.x + threadIdx.x;

// num_point is the number of force points
sharef[tid]=(i<num_point)?f[i]:0.0;
__syncthreads();
int offset=blockDim.x/2;
while(offset>0)
{
    if(tid<offset)
    {
        sharef[tid]+=sharef[tid+offset];
    }
    offset>=1;
    __syncthreads();
}
if(tid==0)
{
    // store the partial sum into the global memory array
    pf[blockIdx.x]=sharef[0];
    __threadfence();
    unsigned int value=atomicInc(&count,gridDim.x);
    isLastBlockDone=(value==gridDim.x-1);
}
__syncthreads();
if(isLastBlockDone)
{
    // the following command needs gridDim.x<=blockDim.x
    sharef[tid]=(tid<gridDim.x)?pf[tid]:0.0;
    __syncthreads();
    int offset=gridDim.x/2;
```

```

while(offset>0)
{
    if(tid<offset)
    {
        sharef[tid]+=sharef[tid+offset];
    }
    offset>>=1;
    __syncthreads();
}

// the 0th thread of the last block to store the values into forcex/forcey
if(threadIdx.x==0)
{
    *force=sharef[0];
    count=0;
}
}

```

3.3.2.2 Atomic operations in LBM

The atomic function performs a read-modify-write atomic operation on a 32-bit or 64-bit word on the global or shared memory. This operation is exclusive such that it is guaranteed to be executed without interference by other threads. This type function prevents the data on global or shared memory from disorder read or write operation when there exist competitive read or write by different threads.

In the CUDA code of the lattice Boltzmann simulation, the atomic operation is very common. The flux evaluation based on the interface between two cells in the FV-LBM, the count operation of the above reduce execution in *Kernel 2*, the excess mass distribution in the free surface capture algorithm in Chapter 4, cannot avoid the atomic operations. The atomic functions are shown in Eq.(3.5a,b). The memory read/write mechanism in the single CPU programming is not a problem. However, the competitive read/write mode in the multi-process or multi-thread programming should be carefully treated. In CUDA platform, the atomic functions are devised to guarantee the order of read/write from different threads. Here we take the operation of excess mass distribution in the free surface algorithm for example. The process can be found in Fig.3.13. Assume the cell $k-1$ and the cell $k+1$ are the interface cells just receiving enough mass transported from the other neighboring cells and the total volume in

these two cells exceeds the cell volume. In this case, the excess mass in this kind of interface cells will overflow to its neighboring interface cells that are not full filled, such as the interface cell k . However, the competitive read/write mode can be happened from two neighboring interface cells $k-1$ and $k+1$. In this case, the exclusive manipulation should be conducted to avoid the disorder write/read. The atomic functions in the CUDA architecture include operations for integer variable, single precision floating variable and other type variables except the double precision floating variable. This indicates that the accuracy of the double precision variable will be compulsively sacrificed to be the single precision.

atomicInc(&count,gridDim.x); (3.5a)

atomicAdd(&(mass[k]),accepted_mass); (3.5b)

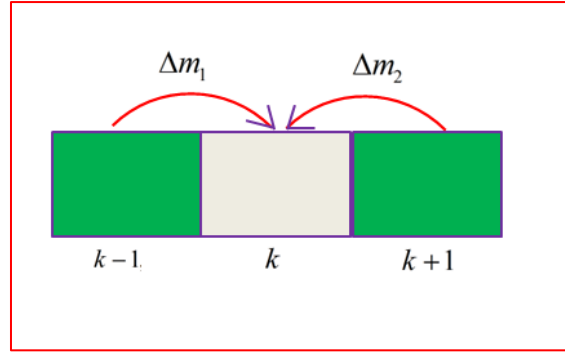


Fig.3.13 Excess mass distribution

3.3.2.3 Boundary treatments in LBM

For the simple physical problems, such as the cavity flow, Poiseuille flow and so on, the collision process, stream process, macroscopic evaluation and boundary treatment can be integrated into one kernel, which can considerably reduce the memory access of the global memory. With the coalescence of all the sub-processes, the CUDA version LBM code is extremely efficient. Nevertheless, the practical physical problems are usually not convenient to be handled. From the viewpoint of algorithm, the boundary treatments disturb the perfect and homogeneous algorithm in the lattice Boltzmann models. Meanwhile, for the purpose of general implementation of the boundary treatments in different lattice Boltzmann models, the boundary treatments are usually performed by exclusive kernels. For the two dimensional case, four sides should be addressed. Four kernels or a whole kernel that uses a flag to identify the

boundary nodes can be applied in the lattice Boltzmann boundary treatments. Owe to the devise of the stream mechanism in the CUDA architecture, the four kernels can be launched onto the device simultaneously but should be synchronized before other kernels launched to guarantee all the boundary treatments finished. The flow chart is shown as followings.

```
Kernel1_BC<<<grid1,block1,smem,stream1>>>();  
Kernel2_BC<<<grid2,block2,smem,stream2>>>();  
Kernel3_BC<<<grid3,block3,smem,stream3>>>();  
Kernel4_BC<<<grid4,block4,smem,stream4>>>();  
cudaStreamSynchronize();
```

3.4 Multi-GPU implementation of the lattice Boltzmann solver

Due to the limited device memory (GRAM) of the single GPU card (maximum 12GB for K40 or TITAN Z/X), the capability of the single device for the numerical simulation is constrained to the two dimensional case or three-dimensional case with low resolution. Thus, the multi-GPU implementation is pivotal for the large scale numerical simulation. The following context will introduce the multi-GPU technique in detail.

3.4.1 Peer-to-Peer technique

If the computing capability of the GPU card is 2.0 or higher and the application is run as 64-bit process, then each device may address each other's memory directly. To make use of this convenient memory access pattern, the logical value of the built-in function `cudaDeviceCanAccessPeer()` should be true for each device. Meanwhile another built-in function `cudaDeviceEnablePeerAccess()` should be called to activate the Peer-to-Peer function. The Peer-to-Peer coding mode for the devices sharing the same PCI-e bus is shown in Fig.3.14. However, some principals must be respected: 1) the copy operation cannot start before all the commands issued to each device finished. The function `cudaDeviceSynchronize` can be called before Peer-to-Peer copy; 2) only after the completion of the Peer-to-Peer copy operation, the subsequent command can be issued. The function `cudaDeviceSynchronize` should be called after Peer-to-Peer copy. The Peer-to-Peer memory access can facilitate a considerable memory utilization of 24 GB (say two K40 with 12GB each).

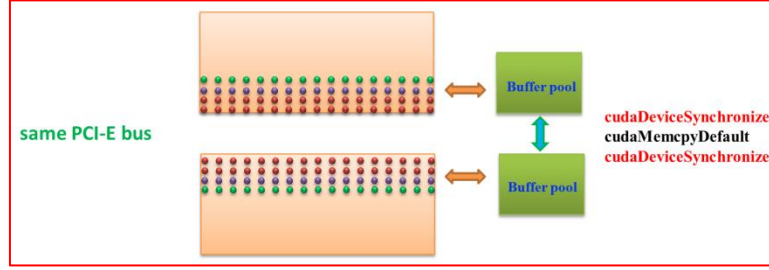


Fig.3.14 Peer to Peer memory access

3.4.2 CUDA with MPI

To achieve larger scale simulations, more devices that do not share the same PCI-e bus should be connected to each other. In this case, the data transfer among multi-GPU ought to be conducted by the GPU-CPU-GPU pattern if the NV-link technique cannot be used. The data is copied from GPU to CPU, after that communication is performed among CPUs, then data copied from CPU to GPU. If the CPUs are equipped on the same node, the shared memory programming mode can be applied, such as openMP, to avoid the communication among CPUs. If the GPUs and the CPUs to control them are distributed on different nodes, the data communications among different nodes have to be performed. In the present work, for the general purpose of programming, the MPI (Message Passing Interface) library is applied to execute the data transfer in the host ends. The general framework of multi-GPU code in the present work can be found in Fig.14. The hybrid data transfer pattern is applied in the present programming structure. For the GPUs that share the same PCI-e bus, the data exchange between the devices is conducted by the aforementioned Peer-to-Peer memory access. For the GPUs belong to different PCI-e buses, the data transfer should be completed with the assistance of MPI library among CPUs. In this approach, the partition of the fluid domain should be treated carefully.

The domain decomposition usually can be 1D, 2D or 3D strategy for the three dimensional numerical simulation. The strategy of 1D decomposition for the three-dimensional multiphase lattice Boltzmann model in Chapter 6 can be found in Fig.3.15. Three kinds of nodes are categorized: inner domain nodes, outer domain nodes and halos, which can be found in Fig.3.16. In order to overlap the communication time of GPU-CPU and CPU-CPU, the multi-stream technique is applied in the present multi-GPU code, as shown in Fig.3.17. The flow chart of

multi-GPU program for the multiphase LB model can be found in Fig.3.18. Part of the particle distribution functions and the order parameter should be exchanged each time step.

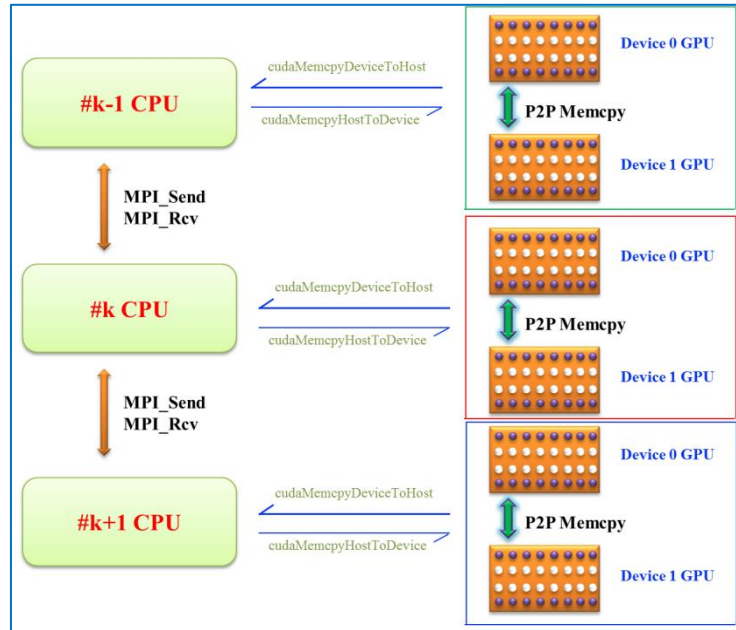


Fig.3.15 Multi-GPU coding framework

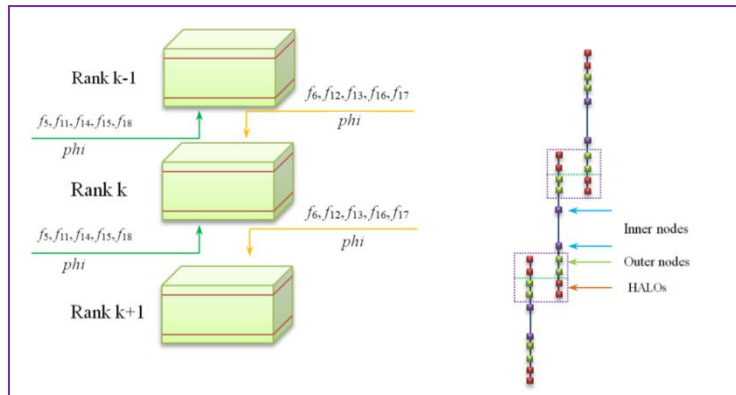


Fig.3.16 Decomposition strategy for the multi-GPU implementation

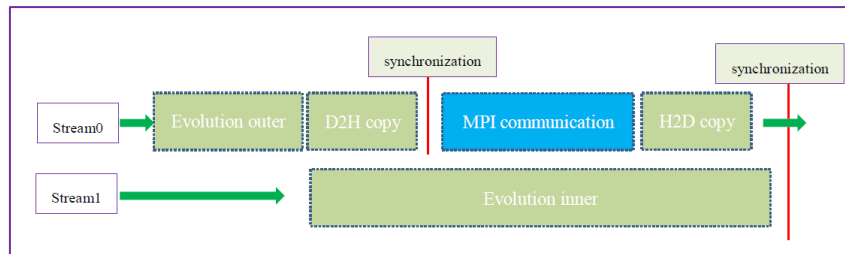


Fig.3.17 Overlap model for the communication

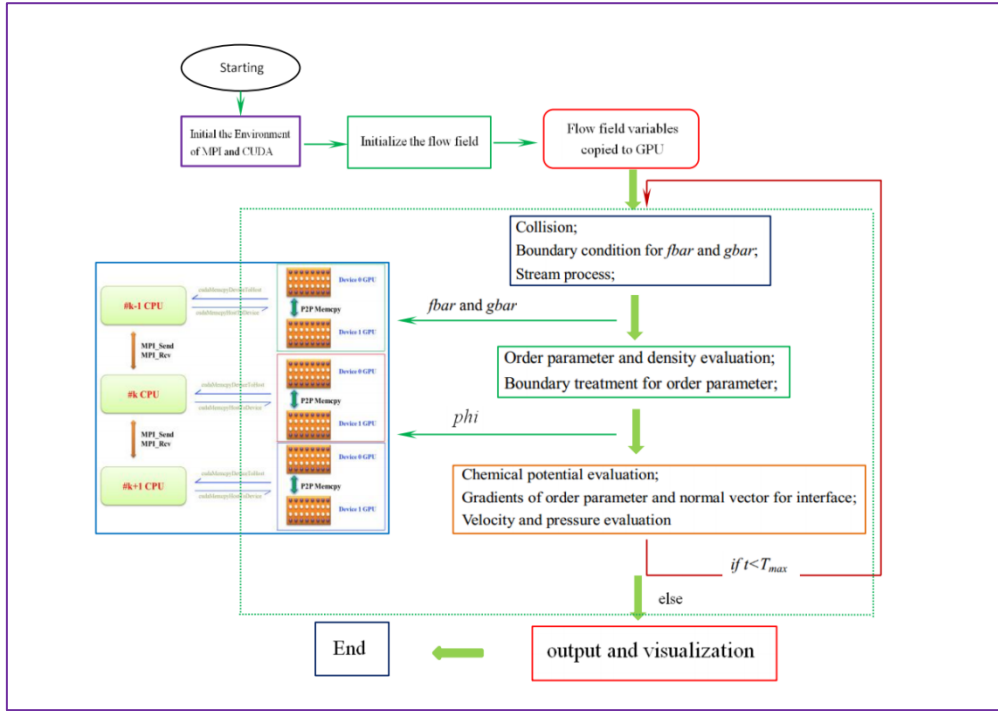


Fig.3.18 Flow chart of the multi-GPU implementation of the multiphase LB model

3.5 Concluding Remarks

In this chapter, the overview of the hardware of the GPGPU has been introduced firstly. Then the CUDA programming platform is described in detail. Subsequently, the applications of the CUDA programming in the lattice Boltzmann simulation are summarized. In order to resolve the large scale numerical simulation of lattice Boltzmann model, the multi-GPU programming framework and the data communication pattern are elaborated. In the present work, the GPU acceleration technique has been applied to the LB solvers in Chapter 2, 4, 5, 6.

Chapter 4 Moving boundary in the fluid-solid interactions

4.1 Numerical schemes for moving boundaries

4.1.1 Extrapolation scheme by Guo et al.(2002)

The standard evolution of the LBE process consists of two computation steps, i.e., the collision step and stream step. Before the stream step, each fluid node should be surrounded by fluid nodes or boundary nodes with reconstructed particle distribution functions (i.e. the “collision” updating process of the boundary nodes). As mentioned in Guo et al. (2002), the particle distribution function (PDF) can be divided into equilibrium PDF and non-equilibrium PDF, and the equilibrium part can be calculated by extrapolated density and velocity:

$$f_i^{+,eq}(\mathbf{r}_b, t) = Feq_i(\rho_b, \mathbf{u}_b, t) = \rho_b \omega_i \left[1 + \frac{3}{c^2} \mathbf{e}_i \cdot \mathbf{u}_b + \frac{9}{2c^4} (\mathbf{e}_i \cdot \mathbf{u}_b)^2 - \frac{3}{2c^4} \mathbf{u}_b \cdot \mathbf{u}_b \right] \quad (4-1)$$

where “+” denotes post-collision, $\bar{i}$ is the inverse direction of i and ω_i is the weighting coefficients. Subscripts b and w, f, ff used in the following formulas denote the physical variables on boundary node (i.e. ghost node), wall node and neighboring fluid nodes, of which the intuitive explanation can be found in Fig.4.1.

$$q \geq q_c, \rho_b = \rho(\mathbf{r}_f), \mathbf{u}_b = \frac{\mathbf{u}_w + (q-1)\mathbf{u}_f}{q} \quad (4-2)$$

$$q < q_c, \rho_b = \rho(\mathbf{r}_f), \mathbf{u}_b = q \frac{\mathbf{u}_w + (q-1)\mathbf{u}_f}{q} + (1-q) \frac{2\mathbf{u}_w + (q-1)\mathbf{u}_{ff}}{1+q} \quad (4-3)$$

The non-equilibrium part can be also extrapolated from fluid nodes:

$$q \geq q_c, f_i^{+,neq}(\mathbf{r}_b, t) = f_i(\mathbf{r}_f, t) - f_i^q(\mathbf{r}_f, t) \quad (4-4)$$

$$q < q_c, f_i^{+,neq}(\mathbf{r}_b, t) = q[f_i(\mathbf{r}_f, t) - f_i^{eq}(\mathbf{r}_f, t)] + (1-q)[f_i(\mathbf{r}_{ff}, t) - f_i^{eq}(\mathbf{r}_{ff}, t)] \quad (4-5)$$

where $q = |\mathbf{r}_f - \mathbf{r}_w| / |\mathbf{r}_f - \mathbf{r}_b|$ and $q_c = 0.75$. Thus the post-collision PDF on the boundary nodes can be obtained as follows.

$$f_i^+(\mathbf{r}_b, t) = f_i^{+,eq}(\mathbf{r}_b, t) + f_i^{+,neq}(\mathbf{r}_b, t) \quad (4-6)$$

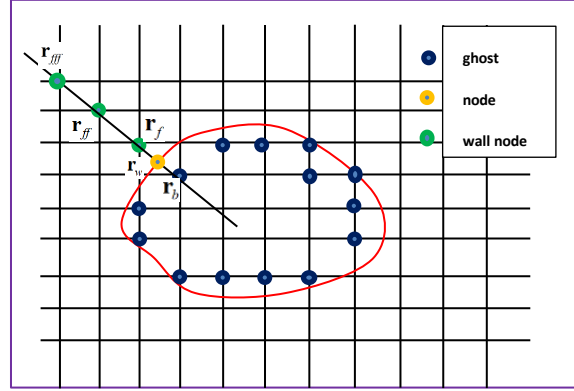


Fig.4.1 Boundary schematic

4.1.2 Interpolation bounce-back schemes

Bouzidi et al. (2001) and Lallemand et al. (2003) developed the first-order and second-order interpolation bounce back schemes to interpolate the PDFs streamed from the obstacle and use the momentum exchange method to evaluate the force exerted on the obstacle. The linear interpolation bounce-back (LIBB) scheme is expressed as follows:

$$q < 0.5$$

$$f_i(\mathbf{r}_b, t) = 2q f_i^+(\mathbf{r}_f, t) + (1-2q) f_i^+(\mathbf{r}_{ff}, t) + \frac{6}{c^2} \rho \omega_i \mathbf{u}_w \cdot \mathbf{e}_i \quad (4-7)$$

$$q > 0.5$$

$$f_i(\mathbf{r}_b, t) = \frac{1}{2q} f_i^+(\mathbf{r}_f, t) + \frac{2q-1}{2q} f_i^+(\mathbf{r}_{ff}, t) + \frac{3}{qc^2} \rho \omega_i \mathbf{u}_w \cdot \mathbf{e}_i \quad (4-8)$$

The quadratic interpolation bounce-back (QIBB) scheme is expressed as follows:

$$q < 0.5$$

$$f_i(\mathbf{r}_b, t) = q(1+2q) f_i^+(\mathbf{r}_f, t) + (1-4q^2) f_i^+(\mathbf{r}_{ff}, t) - q(1-2q) f_i^+(\mathbf{r}_{fff}, t) + \frac{6}{c^2} \rho \omega_i \mathbf{u}_w \cdot \mathbf{e}_i \quad (4-9)$$

$$q > 0.5$$

$$f_i(\mathbf{r}_b, t) = \frac{1}{q(2q+1)} f_i^+(\mathbf{r}_f, t) + \frac{2q-1}{2q} f_i^+(\mathbf{r}_f, t) + \frac{1-2q}{1+2q} f_i^+(\mathbf{r}_{ff}, t) + \frac{6}{q(1+2q)c^2} \rho \omega_i \mathbf{u}_w \cdot \mathbf{e}_i \quad (4-10)$$

4.1.3 Ghost fluid method

4.1.3.1 Scheme process

With the intrinsic consistency by using Cartesian mesh, it is naturally to incorporate the immersed boundary method into LBM. The ghost fluid method of sharp interface approach is one of the LBM-IBs. The Dirichlet as well as Neumann boundary conditions for pressure and velocity can be satisfied easily. In the ghost fluid based sharp interface scheme, boundary conditions are performed via ghost cells, which lie adjacent to the boundary but outside the fluid region, as shown in Fig.1. This kind of scheme can be organized as follows referred to Tiwari et al. (2012):

Step 1: Identification of the ghost nodes. As mentioned in above context, all of the interior fluid nodes should have neighboring nodes which are either fluid nodes after collision or ghost nodes after PDF reconstruction. The ghost node can be identified as the solid node which has at least one neighboring fluid node;

Step 2: Determine the image point of the ghost node mirrored along the normal direction with respect to the boundary;

Step 3: Interpolate the pressure (density), velocity and non-equilibrium PDFs on the image nodes through bilinear interpolation or extended quadratic interpolation using the flow information on fluid nodes or appropriate boundary condition.

Step 4: Extrapolate the information of ghost node from the image point and physical values on the boundary along the normal direction to preserve the non-penetration condition across the boundary.

In case of non-slip boundary condition, extrapolation takes the form

$$\mathbf{u}_g = 2\mathbf{u}_w - \mathbf{u}_{img}, \quad \rho_g = \rho_{img}, \quad f_{i_ghost}^{neq} = f_{i_image}^{neq} \quad (4-11)$$

Step 5: Calculate the equilibrium PDFs using the density and velocity obtained in *Step 4* with equilibrium distribution function;

Step 6: Add the non-equilibrium PDFs and equilibrium PDFs together to obtain the PDFs for streaming process.

4.1.3.2 Dirichlet boundary conditions

Without loss of generality, the flow values are uniformly defined as φ . The flow variable is evaluated at the image nodes with different interpolation schemes.

4.1.3.2.1 Bilinear reconstruction

In this paper, the formulas in Tiwari et al. (2012) are reconstructed similar to the way in Y.H. Tseng et al. (2003) for the convenience to extend it to the quadratic ghost fluid method below. In 2D case, if the flow variables are considered to be varied in a bilinear manner in x and y direction, the general physical value φ can be expressed as

$$\varphi = a_1 + a_2x + a_3y + a_4xy \quad (4-12)$$

The coefficients can be expressed in terms of the nodal values:

$$\mathbf{a} = B^{-1}\boldsymbol{\varphi} \quad (4-13)$$

Generally, there are three cases for the selection of the four interpolation points with different position of the image node relative to the boundary, as shown in Fig.4.2. Correspondingly, the coefficients matrix B can be written as

$$B = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 \end{bmatrix} \quad (4-14)$$

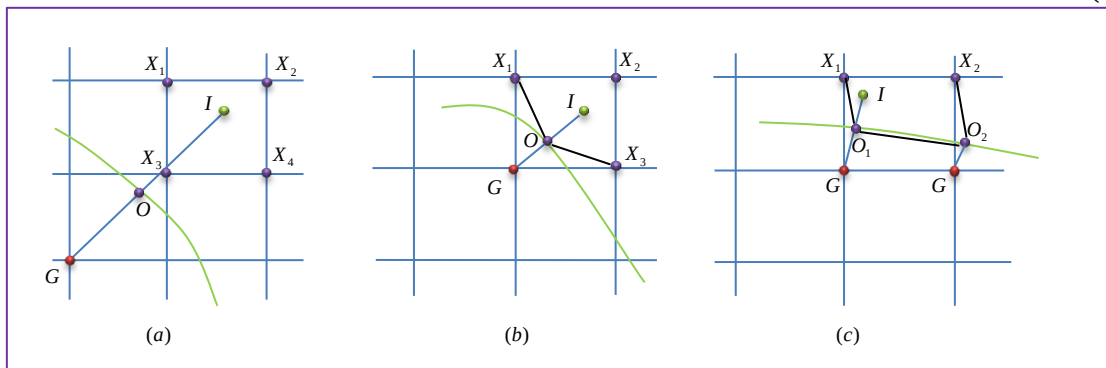


Fig.4.2 Bilinear interpolation schematic. X_1, X_2, X_3, X_4 are the fluid nodes, G is the ghost node, I is the image node, O or O_1, O_2 are the intersection node of the line GI which is vertical to local boundary (a) the image node is located at the quadrilateral of four fluid nodes; (b) the image node is located at the quadrilateral of three fluid nodes and a boundary node O ; (c) the image node is located at the quadrilateral of two fluid nodes and two boundary nodes, O_2 is intersection node of the boundary and the outward normal line extended from neighboring ghost node.

4.1.3.2.2 Quadratic reconstruction

In this paper, the quadratic ghost fluid method in Y.H. Tseng et al. (2003) has been firstly introduced into the lattice Boltzmann method. If the flow field are assumed to be varied in a quadratic manner in both x and y directions, the general physical value φ can be expressed as

$$\varphi = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 \quad (4-15)$$

There are three cases for the selection of the six interpolation points, as shown in Fig.4.3. If the image point is very close to the boundary in case (c), the case (c) in the Fig.4.2 will be more stable to replace it, which is similar to the treatment in the interpolation bounce-back scheme and Guo's scheme. Similarly, the coefficients matrix B can be written as

$$B = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 \\ 1 & x_3 & y_3 & x_3^2 & x_3y_3 & y_3^2 \\ 1 & x_4 & y_4 & x_4^2 & x_4y_4 & y_4^2 \\ 1 & x_5 & y_5 & x_5^2 & x_5y_5 & y_5^2 \\ 1 & x_6 & y_6 & x_6^2 & x_6y_6 & y_6^2 \end{bmatrix} \quad (4-16)$$

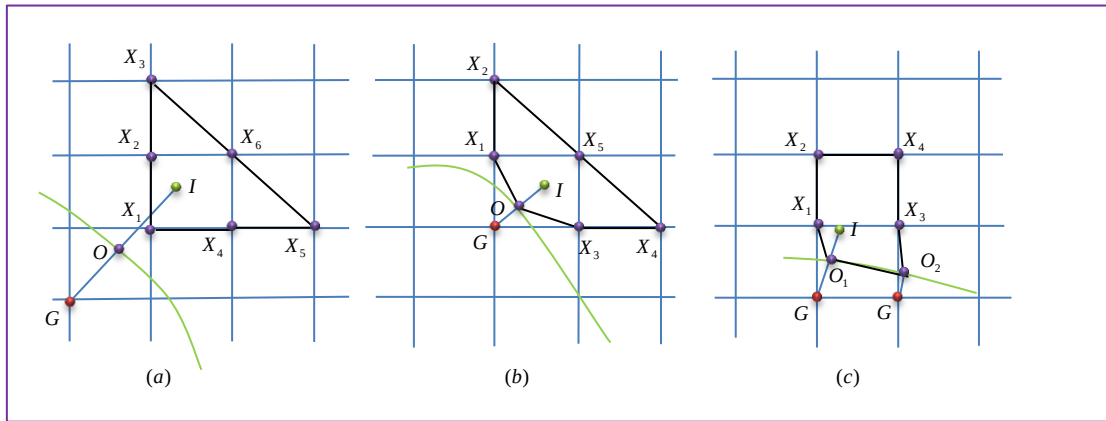


Fig.4.3 Quadratic interpolation schematic. $X_1, X_2, X_3, X_4, X_5, X_6$ are the fluid nodes, G is the ghost node, I is the image node, O or O_1, O_2 are the intersection node of the line GI which is vertical to local boundary (a) the image node is located at the polygon of six fluid nodes; (b) the image node is located at the polygon of five fluid nodes and a boundary node O ; (c) the image node is located at the polygon of four fluid nodes and two boundary nodes, O_2 is intersection node of the boundary and the outward normal line extended from neighboring ghost node.

4.1.3.3 Neumann boundary conditions

The Neumann boundary condition can be realized by the same way as Dirichlet boundary condition except a slight difference in the construction of the coefficient matrix (Y.H. Tseng et al. 2003). For instance, the pressure boundary condition needs zero normal derivative at the boundary node as

$$\left. \frac{\partial \rho}{\partial n} \right|_{\Omega} = 0 \quad (4-17)$$

The normal derivative on the boundary can be expanded as

$$\frac{\partial \rho}{\partial n} = \frac{\partial \rho}{\partial x} n_x + \frac{\partial \rho}{\partial y} n_y \quad (4-18)$$

where n_x and n_y are the components of the unit normal vector at the intersection point on the boundary. Considering this zero normal derivative condition, the corresponding row of the this point in the matrix B can be replaced as (taking quadratic interpolation case (b) for example)

$$B = \begin{bmatrix} 0 & n_x & n_y & 2x_1n_x & y_1n_x + x_1n_y & 2y_1n_y \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 \\ 1 & x_3 & y_3 & x_3^2 & x_3y_3 & y_3^2 \\ 1 & x_4 & y_4 & x_4^2 & x_4y_4 & y_4^2 \\ 1 & x_5 & y_5 & x_5^2 & x_5y_5 & y_5^2 \\ 1 & x_6 & y_6 & x_6^2 & x_6y_6 & y_6^2 \end{bmatrix} \quad (4-19)$$

Other cases in bilinear and quadratic interpolation schemes can be conducted in a similar way. The treatment of non-equilibrium PDFs interpolation is similar to the treatment of pressure.

4.2 Refilling of the fresh node

A typical and significant issue to be emphasized in moving boundary problems is the initialization of the fresh nodes which just emerge from the solid region to the fluid domain due to the moving boundary. The refilling technique of the flow information on the fresh nodes will dramatically affect the oscillation of the force evaluation. Putting deep insight into the Chapman-Enskog expansion (Shiyi Chen & Gary D. Doolen, 1998), one can find that zero-order PDF, i.e., equilibrium PDF, can only recover the Euler equation without viscous

term. The viscous term should be recovered from the non-equilibrium PDF. Therefore, if equilibrium PDF is used to initialize the fresh node without any indiscriminate treatment, the whole fluid field will be solved using NS equation except the fresh node which is partially solved by Euler equation at new time step. This kind of treatment is non-consistent from the physical point of view. A. Caiazzo et al. (2008) applied the asymptotic approach to expand PDF and pointed out that the non-equilibrium PDF associating with pressure and stress tensor contains velocity gradient term ∇u . However, it is difficult to be evaluated near the boundary. Revisiting the literature L.Chen et al. (2013), but different from their treatment, in this paper, we develop an alternative delicate and robust approach to interpolate the velocity, pressure and non-equilibrium PDF for initialization of the fresh node. As this process is performed after the stream process and the boundary updating, the PDFs which steam from previous old fluid nodes to the new fluid nodes will not be destroyed while only the PDFs steaming from solid nodes to the fresh nodes will be reinitialized. The velocity, density and non-equilibrium PDF will be obtained by bilinear interpolation with stencil construction of four nodes or three nodes (Fig.4.4). Similar to the treatment of section 3.3, the Dirichlet boundary condition is applied to deal with the velocity at boundary node O , the Neumann boundary condition (zero normal gradient condition) is utilized to treat the pressure and non-equilibrium PDSs. The information of the image node used for re-initialization in L.Chen et al. (2013) has been abandoned. It will be degraded to linear interpolation when either component of the local normal vector is close to zero.

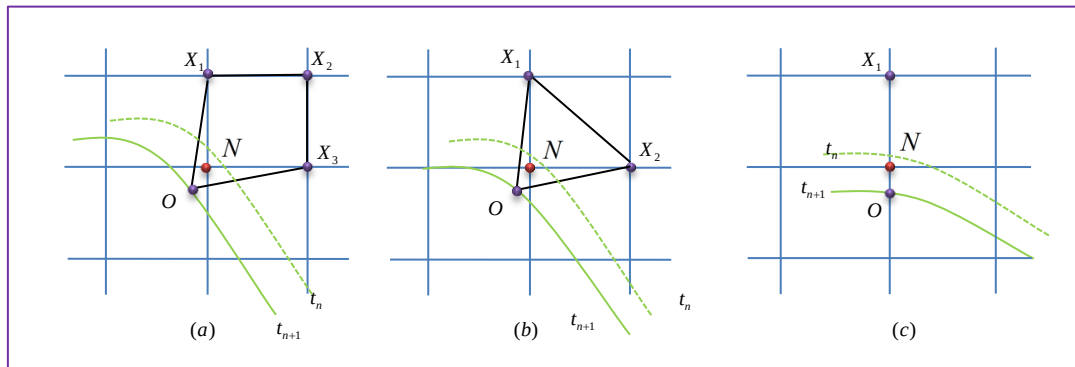


Fig.4.4 Re-initialization interpolation schematic. Point X_1 , X_2 , X_3 are the fluid nodes, the red point N is the fresh fluid node, O is the boundary node with ON vertical to the local boundary. (a) using three fluid nodes and a boundary node to interpolate the velocity, pressure and non-equilibrium distribution function at the new fluid node; (b) using two fluid nodes and a boundary node to interpolate the velocity, pressure and non-equilibrium distribution function at the new fluid node; (c) the local normal vector is almost parallel to the grid line, the linear interpolation is used.

The reconstruction of the missing PDF can be expressed as

$$f_i(\mathbf{x}, t + \delta t) = f_i^{eq}(\rho^I, \mathbf{u}^I) + f_i^{neq, I}(\mathbf{x}, t + \delta t) \quad (4-20)$$

where I denotes results of interpolation.

Meanwhile, other kinds of refilling techniques will be elaborated here for comparison:

(1) Equilibrium distribution function directly using the obstacle's velocity for re-initialization, as P. Lallemand & Luo LS (2003), $f_i(\mathbf{x}, t + \delta t) = f_i^{eq}(\rho^{average}, \mathbf{u}_w)$, $\rho^{average}$ is the density average value of the neighboring fluid nodes, i.e., refilling technique a;

(2) The sum of non-equilibrium distribution function directly copied from the neighboring fluid nodes as A. Caiazzo (2008) and the equilibrium part, $f_i(\mathbf{x}, t + \delta t) = f_i^{eq}(\rho^{average}, \mathbf{u}_w) + f_i(\mathbf{x} + \mathbf{e}_i \delta t, t + \delta t) - f_i^{eq}(\mathbf{x} + \mathbf{e}_i \delta t, t + \delta t)$ i.e., refilling technique b;

(3) Second-order extrapolation method as P. Lallemand & Luo (2003), $f_i(\mathbf{x}, t + \delta t) = 3f_i(\mathbf{x} + \mathbf{e}_k \delta t, t + \delta t) - 3f_i(\mathbf{x} + 2\mathbf{e}_k \delta t, t + \delta t) + f_i(\mathbf{x} + 3\mathbf{e}_k \delta t, t + \delta t)$, where the value of the lattice direction vector $\mathbf{e}_k$ dot the local boundary outward normal vector ($\mathbf{e}_k \cdot \mathbf{n}$) achieves the maximum among all of the possible lattice direction, i.e., refilling technique c;

4.3 Force evaluation methods

In this paper, two existing force evaluation methods are applied in the simulation to evaluate the accuracy and their influence on the pressure oscillation.

The conventional momentum exchange (CME) equation (Cyrus K. Aidun 1998, Mei RW 1999) to evaluate the force on a fluid-solid lattice-link can be written as

$$\mathbf{F}_i(\mathbf{r}_w) = \mathbf{e}_i f_i^+(\mathbf{r}_f, t) - \mathbf{e}_i f_i^+(\mathbf{r}_b, t) \quad (4-21)$$

where $\mathbf{e}_i f_i^+(\mathbf{r}_f, t)$ contributes a momentum increment and $-\mathbf{e}_i f_i^+(\mathbf{r}_b, t)$ contributes a momentum decrement.

However, in the literature Wen et al. (2014) pointed out that it is the relative velocity that contributes the momentum variation instead of the absolute lattice velocity. Thus, the particle

Chapter 4 Moving boundary in the fluid-solid interactions

$f_i^+(\mathbf{r}_f, t)$ has the velocity $(\mathbf{e}_i - \mathbf{u}_w)$ relative to the boundary and contributes a momentum increment $(\mathbf{e}_i - \mathbf{u}_w) f_i^+(\mathbf{r}_f, t)$. The modified Galilean invariant momentum exchange method (MME) can be expressed as

$$\mathbf{F}_i(\mathbf{r}_w) = (\mathbf{e}_i - \mathbf{u}_w) f_i^+(\mathbf{r}_f, t) - (\mathbf{e}_i - \mathbf{u}_w) f_i^+(\mathbf{r}_b, t) \quad (4-22)$$

The velocity of the boundary point can be calculated as

$$\mathbf{u}_w = \mathbf{U} + \mathbf{\Omega} \times (\mathbf{r}_w - \mathbf{r}_c) \quad (4-23)$$

where $\mathbf{r}_c$ is the mass center of the obstacle, $\mathbf{U}$ is its translational velocity and $\mathbf{\Omega}$ is the angular velocity.

The hydrodynamic force and torque exerted on the boundary of the obstacle can be evaluated as

$$\mathbf{F} = \sum \mathbf{F}_i(\mathbf{r}_w) \quad (4-24)$$

$$\mathbf{T} = \sum (\mathbf{r}_w - \mathbf{r}_c) \times \mathbf{F}_i(\mathbf{r}_w) \quad (4-25)$$

According to the stress integration method (SI) proposed by Inamuro et al.(2000), the stress tensor can be evaluated through the distribution function on the boundary directly instead of calculating the velocity gradient:

$$\sigma_{ij} = -\frac{1}{6\tau} \rho \delta_{ij} - (1 - \frac{1}{2\tau}) \sum_{\alpha} (\mathbf{e}_{\alpha i} - u_i)(\mathbf{e}_{\alpha j} - u_j) f_{\alpha} \quad (4-26)$$

The force and torque exerted on the boundary can be integrated as

$$\mathbf{F} = \int_{\partial\Omega} \{\boldsymbol{\sigma} \cdot \mathbf{n} - \rho \mathbf{u}[(\mathbf{u} - \mathbf{U}) \cdot \mathbf{n}]\} ds \quad (4-27)$$

$$\mathbf{T} = \int_{\partial\Omega} \mathbf{r} \times \{\boldsymbol{\sigma} \cdot \mathbf{n} - \rho \mathbf{u}[(\mathbf{u} - \mathbf{U}) \cdot \mathbf{n}]\} ds \quad (4-28)$$

where $\mathbf{u}$ is the velocity of the force point on the body surface, which can be evaluated by extrapolated f_{α} .

The extrapolation method in Huabing Li et al. (2004) is used in this paper to obtain the

PDFs on the boundary and a set of discrete Lagrangian points on the boundary are uniformly arranged to calculate the integration.

The force and torque acting on the boundary of the obstacle lead to the motion of the obstacle and updating of the boundary. The motion equation is

$$\frac{d\mathbf{U}}{dt} = \frac{\mathbf{F}}{M} \quad (4-29)$$

$$\frac{d\mathbf{\Omega}}{dt} = \frac{\mathbf{T}}{I} \quad (4-30)$$

where M and I are the mass and the moment of the inertia of the obstacle. The time process is performed using Euler explicit scheme.

4.4 Numerical simulation

In this section, the accuracy and pressure oscillation issues are investigated by simulating several benchmark test cases. First, the numerical case of an eccentric cylinder in a channel will be simulated by the interpolation or extrapolation schemes and ghost fluid methods to compare their efficiency, accuracy and invalidate the reliability of the codes. Then, the Cylindrical Couette flow will be adopted to compare the space convergence accuracy of different boundary treatment schemes. Subsequently, an impulsively started cylinder case with two different but equivalent reference coordinate is applied to compare different boundary treatment schemes and also investigate the effects of collision model, refilling technique, and force evaluation method on the pressure oscillation and accuracy. The fourth one is the oscillating cylinder in a channel with passive sinusoidal motion. This scenario aims to verify the mesh convergence effect. Finally, the practical particle suspension case is simulated to compare different boundary treatment schemes and investigate the effects of refilling technique and force evaluation method on the pressure oscillation and the motion of the particle.

4.4.1 Eccentric cylinder

This numerical case is simulated to validate the efficiency, accuracy of different boundary treatment schemes as well as the reliability of the codes. The TRT collision model and the modified momentum exchange method (B. Wen 2014) are applied in this case. The geometric

Chapter 4 Moving boundary in the fluid-solid interactions

arrangement of the flow field is shown in Fig.4.5. The boundary condition on the upper and lower wall is no-slip condition that can be realized by Guo's non-equilibrium extrapolation scheme (Guo 2002), the inlet is parabolic velocity condition and the outlet is constant pressure (initial pressure) condition that can be realized by Guo's non-equilibrium extrapolation (Guo 2002). The inlet parabolic velocity is

$$u(0, y) = 4u_{\max} \frac{y(H-y)}{H^2}, v = 0 \quad (4-31)$$

The average of the parabolic velocity $U = 2u_{\max} / 3$ is taken as the characteristic velocity and the diameter of circular cylinder is selected as the characteristic length. The Reynolds number is $Re = UD / \nu = 20$ and Mach number is $Ma = u_{\max} / c_s$. The fluid force exerted on the cylinder is written in dimensionless expressions:

$$C_D = \frac{F_x}{0.5\bar{\rho}U^2D}, C_L = \frac{F_y}{0.5\bar{\rho}U^2D} \quad (4-32)$$

where $\bar{\rho}$ is the average density of the whole fluid field.

The dimensionless form of recirculation length behind the cylinder can be calculated as

$$\bar{L} = L_r / D \quad (4-33)$$

The pressure difference between the front and back point of the cylinder can be also calculated as

$$\Delta \bar{p} = \frac{\Delta p}{0.5\bar{\rho}U^2} \quad (4-34)$$



Fig.4.5 Geometric arrangement of the flow field

In Table 1, the efficiency of different schemes are compared with the grid number across

the diameter of the cylinder $n=10$. The CPU properties are as followings: Inter(R) Core(TM) i7-4710MQ @ 2.50GHz, 16.0 RAM. Since the convergence of different schemes with time is almost the same (the accuracy of drag force, lift force, recirculation length and pressure difference keep 6 decimals unchanged), 200000 time steps are set for all the schemes. As shown in the Table 4.1, it can be found that there is little difference among the schemes Guo, LIBB, QIBB and BGFM, while the simulation time of QGFM increases around 10% compared with Guo's scheme, which is acceptable for the complicated stencil construction. Anyhow, the computational time associated with different boundary treatment schemes are not considerable.

Table 4.1 Comparison of Calculation time

Scheme($n=10$)	Time (s)
Guo	193.374
LIBB	193.643
QIBB	196.012
BGFM	198.755
QGFM	215.189

In Table 4.2, the drag force, lift force, recirculation length and pressure difference of different schemes with increasing resolution are systematically compared. In general, all schemes can obtain accurate results with the highest resolution compared with the reference results of NS solver (M. Schafer et al. 1996). However, the results of the quadratic interpolation bounce-back scheme and the quadratic ghost fluid method are better than those of their linear counterpart with rough resolution, especially for lift force. Generally, the present results with different schemes verify the capability of all schemes and the reliability of the codes.

Table 4.2 Comparison of drag, lift forces, recirculation length and pressure difference

Scheme(n)	C_D	C_L	$\bar{L}$	$\Delta \bar{p}$
LIBB($n=10$)	5.497492	0.015422	0.784415	5.491784
LIBB($n=20$)	5.539150	0.011462	0.831602	5.722959
LIBB($n=40$)	5.547793	0.010702	0.842857	5.803684
LIBB($n=80$)	5.550285	0.010575	0.845022	5.831888
QIBB($n=10$)	5.463932	0.012403	0.793939	5.358970
QIBB($n=20$)	5.534948	0.010604	0.838528	5.657733
QIBB($n=40$)	5.546803	0.010542	0.844588	5.780458
QIBB($n=80$)	5.550099	0.010544	0.845887	5.822530
BGFM($n=10$)	5.524729	0.019856	0.798701	5.811097

Chapter 4 Moving boundary in the fluid-solid interactions

BGFM(n=20)	5.536529	0.012726	0.829437	5.877013
BGFM(n=40)	5.542570	0.010991	0.842424	5.864859
BGFM(n=80)	5.546306	0.010670	0.845021	5.852657
QGFM(n=10)	5.527363	0.014444	0.817748	5.455456
QGFM(n=20)	5.527428	0.011665	0.830303	5.855303
QGFM(n=40)	5.540521	0.011019	0.841125	5.863627
QGFM(n=80)	5.545197	0.010754	0.844156	5.851021
Guo(n=10)	5.622609	0.013974	0.832467	5.694705
Guo(n=20)	5.562751	0.011283	0.838528	5.797433
Guo(n=40)	5.548544	0.010711	0.843290	5.821849
Guo(n=80)	5.548521	0.010580	0.845022	5.834165
NS solver (M. Schafer et al.)	5.57-5.59	0.0104-0.0110	0.8420-0.8520	5.86-5.88

4.4.2 Cylindrical Couette Flow

The accuracy comparison work associating with interpolation bounce-back (IBB) (M. Bouzidi et al. 2001, P. Lallemand et al. 2003) and conventional diffused interface LBM-IB (Feng Z-G et al. 2004, 2005) has been conducted by Yan P. et al. (2008). In this section, the spatial convergence is tested with different extrapolation/interpolation bounce-back schemes (Guo, LIBB, QIBB) and the sharp-interface ghost fluid immersed boundary methods (BGFM and QGFM). The MRT collision model is applied in this numerical case with the relaxation parameters $s_0 = 0$, $s_1 = 1.64$, $s_2 = 1.54$, $s_3 = 0$, $s_4 = 1.9$, $s_5 = 0$, $s_6 = 1.9$, $s_7 = 1/\tau$, $s_8 = 1/\tau$. Since the interpolations or extrapolations for the flow information on the ghost nodes can be classified into two branches: one is along the lattice-link direction which is classical in LBM field (Guo scheme, LIBB and QIBB schemes) and the other is along the local boundary normal vector direction which is introduced from traditional CFD branches such as finite difference method (BGFM and QGFM). Thus it is worth clarifying the accuracy and space convergence rate for these different schemes. For this purpose, the numerical simulation of the Cylindrical Couette flow is performed here. Cylindrical Couette flow is a typical problem that has been widely used to investigate the consistency and order of accuracy of numerical approaches with curved boundaries. In present case, two 2-D cylinders are placed concentrically. The space between these two cylinders is fluid domain. The outer cylinder keeps still and the inner cylinder rotates with constant angular speed ω . With the non-slip boundary condition, the analytical solution can be given as

$$\frac{u(r)}{\omega r_1} = \frac{r_1 r_2}{(r_2^2 - r_1^2)} \left(\frac{r_2}{r} - \frac{r}{r_2} \right) \quad (4-35)$$

Using the speed ωr_1 as the reference velocity and the diameter of inner cylinder $2r_1$ as the reference length, the Mach number is $Ma = \omega r_1 / c_s = 0.1$ and the Reynold number is $Re = (\omega r_1)(2r_1) / \nu = 50$. The aspect ratio $A = (r_2 - r_1) / r_1 = 3$ is chosen in the analysis. Numerical simulation are conducted with different resolution 41×41 , 81×81 , 161×161 , 321×321 . The geometric arrangement is depicted in Fig.4.6.

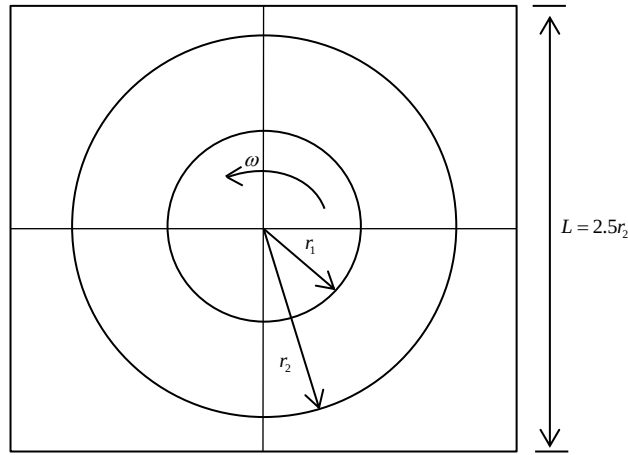


Fig.4.6 Cylindrical Couette flow geometry arrangement

The overall and local convergence accuracy can be measured by L_2, L_∞ as shown below.

$$L_2 = \frac{1}{\omega r_1} \sqrt{\frac{\sum_{i=1}^n [(u_{xi}^N - u_{xi}^a)^2 + (u_{yi}^N - u_{yi}^a)^2]}{n}} \quad (4-36)$$

$$L_\infty = \max_{i=1,n} \frac{\sqrt{[(u_{xi}^N - u_{xi}^a)^2 + (u_{yi}^N - u_{yi}^a)^2]}}{\omega r_1} \quad (4-37)$$

where N stands for the solution of the numerical simulation, superscript a stands for the analytical solution and n stands for the number of fluid nodes for comparison.

Fig.4.7(a) shows comparison of the velocity profile obtained by different boundary treatment schemes (41×41 resolution) and the analytical solution. In general, the velocity profiles of different schemes agree well with the analytical solution. Fig.4.7(b) shows the relative error of velocity profiles of different boundary treatment schemes (41×41 resolution)

compared with the analytical solution. It is found that the accuracy of QGFM is better than that of BGFM and similarly the accuracy of QIBB is better than that of LIBB, which indicates that accuracies of the quadratic schemes are superior to the linear schemes. The accuracies of QGFM and Guo's scheme are comparable while the accuracies of interpolation bounce back schemes are better than other schemes.

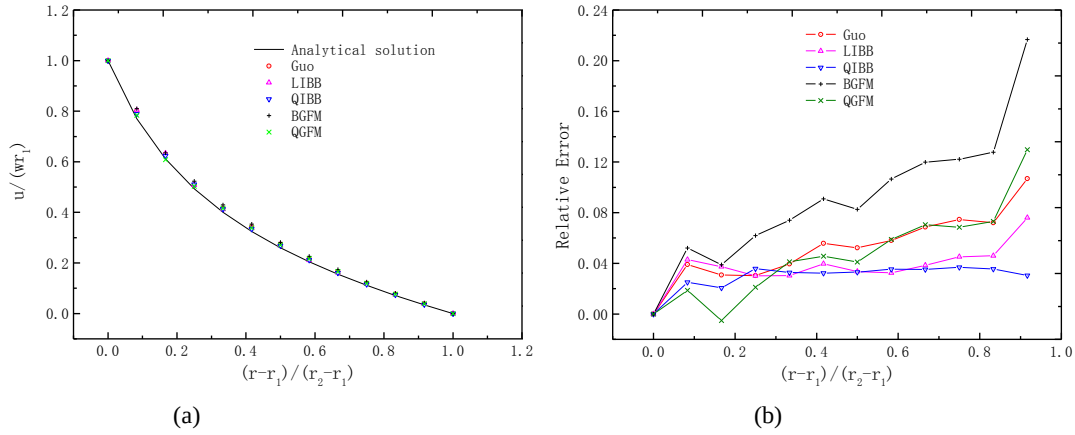


Fig.4.7 Comparison of the accuracy of velocity profile: (a) Comparison of velocity profile with analytical solution; (b) Relative error of different schemes with analytical solution

The overall error L_2 and maximum error L_∞ of the whole fluid domain are plotted on a log-log scale in Fig.4.8. It can be found that all of the schemes are near second-order accuracy. Similarly, the convergence accuracies of QGFM and QIBB are better than their linear counterpart BGFM and LIBB. The convergence accuracy of LIBB and QIBB are better than the other schemes. The convergence accuracy of QGFM and Guo's scheme are comparable. Overall, the accuracies of the ghost fluid methods have no ascendancy over the traditional boundary treatment in lattice Boltzmann method such as Guo's scheme, LIBB and QIBB. In the lattice Boltzmann method, the particles on the lattice are always streaming along the characteristic line in the lattice system such that the interpolation/extrapolation performed along the streaming direction is more reasonable. Nevertheless, in the ghost fluid method, which introduced from traditional CFD branches, the macroscopic variables are handled directly similar to the treatment in traditional CFD. On the one hand, too much transformation between distribution functions and the macroscopic variables incorporates considerable numerical errors, on the other hand, the interpolation/extrapolation along the streaming direction might be advantageous than the normal vector direction of the local boundary in the

category of lattice Boltzmann method.

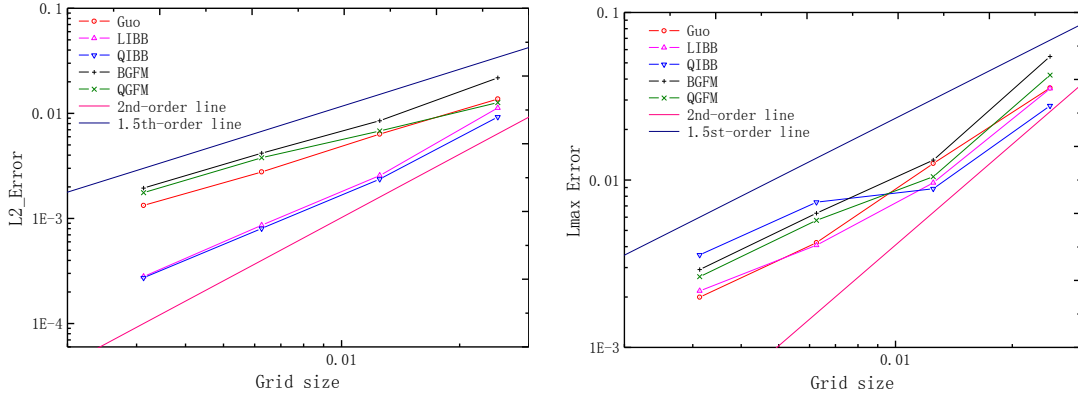


Fig.4.8 Spatial convergence comparison of different schemes (a) L_2 error of different schemes; (b) $L_{\max}$ error of different schemes

4.4.3 Impulsively started cylinder in a channel

To elaborately investigate the effecting extent of different boundary treatment schemes, collision models, refilling techniques, and force evaluation methods on the pressure oscillation in moving boundary issue, the scenario of impulsively started cylinder in a channel (P. Lallemand, Luo LS 2003) is considered. The flow field configuration is depicted in Fig.4.9. The cylinder is placed asymmetrically in the channel. We consider two cases. The first case is that the cylinder starts impulsively towards x direction with uniform velocity $U_c = 0.04$ while the fluid, top and bottom walls keep stationary initially. In this case, the moving boundary of the cylinder is treated. The second case is that the cylinder keeps stationary while the fluid as well as the top and bottom walls moving to the left with uniform velocity $U_0 = -U_c$. Theoretically, these two cases are equal to each other. The periodic boundary condition is applied at the inlet and outlet of the channel and non-equilibrium extrapolation scheme for non-slip boundary condition by Guo et al. (2002) is used at the bottom and top walls. In both cases, the mesh size is 1001×101 , i.e., the width of the channel is 100 and the length is 1000 in lattice units. The initial position of the cylinder is (60.3,54). The cylinder radius is 12 lattice units. The Reynolds number is $Re = 2rU_c / \nu = 200$. The relaxation parameters for the MRT model are $s_0 = 0$, $s_1 = 1.5$, $s_2 = 1.4$, $s_3 = 0$, $s_4 = 1.5$, $s_5 = 0$, $s_6 = 1.5$, $s_7 = 1/\tau$, $s_8 = 1/\tau$. The relaxation parameters s_1, s_2 of the TRT model are set to be $1/\tau$ while other relaxation parameters are the same as those used in the MRT model. With non-dimensional solution, the drag and lift coefficients can be calculated by $C_d = F_d / (0.5\rho U_c^2 D)$ and $C_l = F_l / (0.5\rho U_c^2 D)$, in which D is the

diameter of the cylinder.

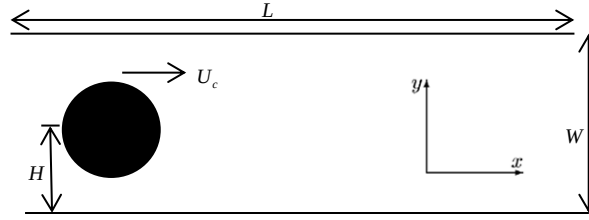


Fig.4.9. Geometric arrangement for the flow field

4.4.3.1 Source of pressure oscillation

Fig.4.10 describes the locally magnified drag force calculated by Guo's scheme with refilling technique a. It is easily found that the drag force with moving boundary case fluctuates around the stationary case with an approximate period of $1/U_c=25$ which is the number of time steps a ghost node transformed into a fluid node or a neighboring fluid node transformed into a ghost node. Within 25 time steps, the moving cylinder with uniform velocity will sweep a lattice distance. These results numerically verify that the original source of pressure oscillation with moving boundary comes from the role conversion of the Cartesian nodes that induces the non-continuity of the flow field (P. Lallemand, Luo LS 2003).

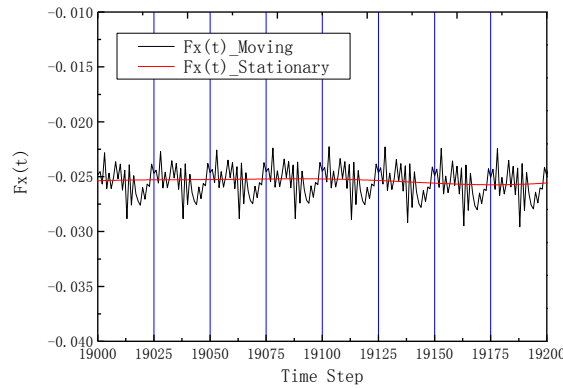


Fig.4.10 Locally magnified comparison for drag force

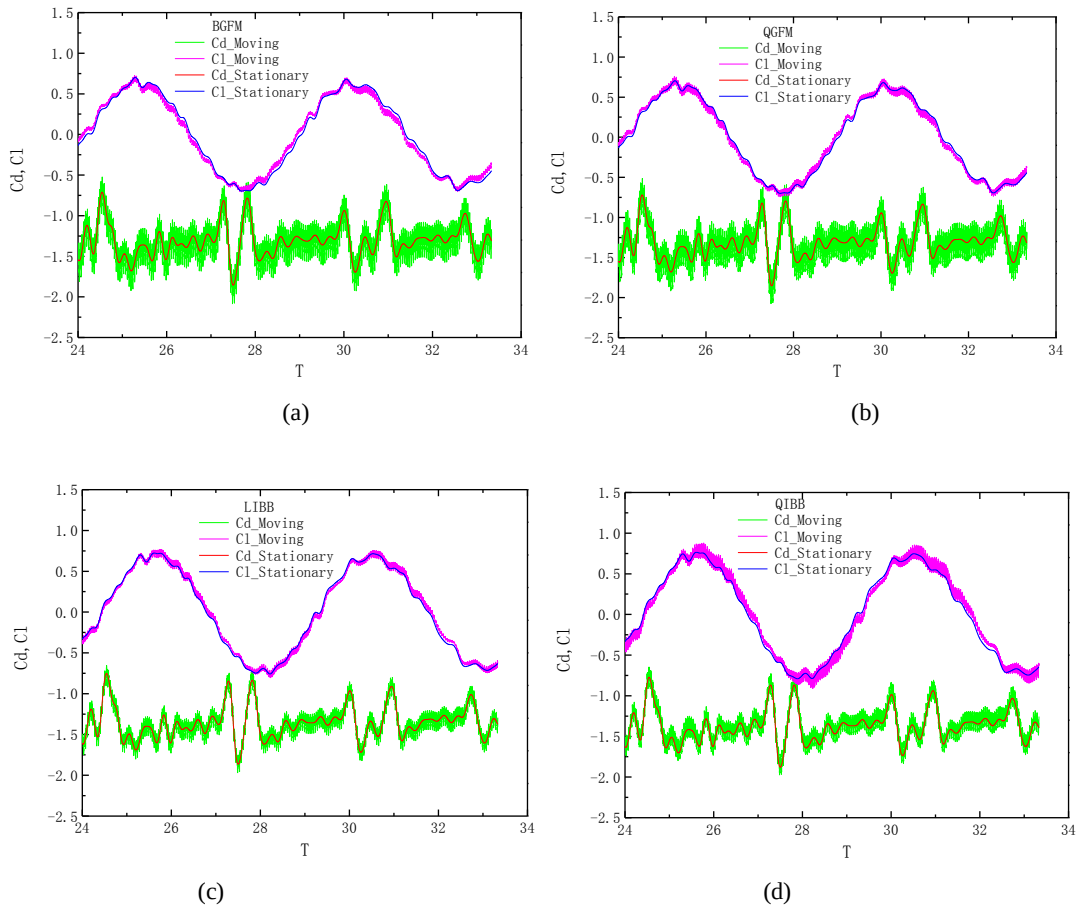
4.4.3.2 Comparison of different schemes

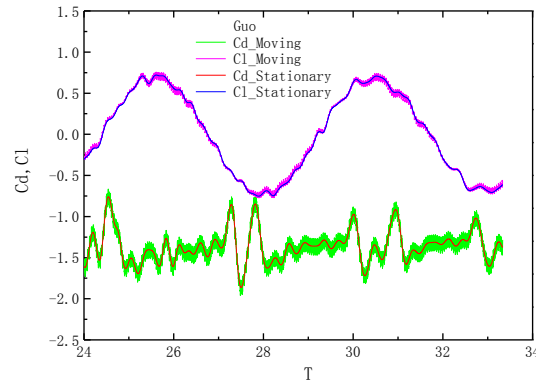
In this subsection, the MRT collision model, the proposed refilling technique and the modified momentum exchange method (B.Wen 2014) are chosen for all the schemes. In order to quantify the force fluctuation of the moving boundary case around the stationary case, the mean squared error is applied:

$$\sigma = \sqrt{\frac{\varepsilon_1^2 + \varepsilon_2^2 + \dots + \varepsilon_n^2}{n}} = \sqrt{\frac{\sum \varepsilon_i^2}{n}} \quad (4-38)$$

The data from T=24 to the final are chosen to calculate the deviation extent of the results of moving boundary case from those of stationary case.

Fig.4.11 (a), (b), (c), (d), (e) compare the drag and lift coefficients of the moving boundary case with the stationary boundary case using the schemes BGFM, QGFM, LIBB, QIBB and Guo's scheme. Table 4.3 states the mean squared error of drag force and lift force of the moving boundary case deviating from the stationary case. It is found that the performance of Guo's scheme is the best with the least oscillation and no phase shift while there is phase deviation of lift force of the moving boundary case from the stationary boundary case for the other four schemes. It is also found that the oscillation magnitudes of the ghost fluid methods (BGFM, QGFM) are more violent than the classic LIBB, QIBB and Guo's scheme. The performance of the LIBB is superior to the QIBB scheme for this numerical case and approximates to the performance of Guo's scheme.





(e)

Fig.4.11 Comparison of drag and lift coefficients of the moving boundary case with the stationary boundary case using different schemes (a) BGFM (b) QGFM (c) LIBB (d) QIBB (e) Guo

Table 4.3 Mean squared error of different schemes

Scheme	σ_x	σ_y
BGFM	0.113067281	0.054462822
QGFM	0.115637299	0.044052355
LIBB	0.056587499	0.034896275
QIBB	0.069370808	0.058266397
GUO	0.055115660	0.021153156

4.4.3.3 Effect of refilling technique

In this subsection, the numerical simulation is conducted by the Guo's scheme with the MRT collision model and the forces are evaluated by the modified momentum exchange method (B. Wen 2014). Different refilling techniques are compared with this numerical example.

From Fig.4.12 and Table 4.4, it can be observed that the pressure oscillation calculated by Guo's scheme with the present refilling technique is suppressed to the most extent and there are no average value deviation and phase shift for the moving boundary case from the stationary boundary case. However, the force oscillations with the refilling technique a and c are remarkable for this numerical scenario. These results indicate that the physical consistence and the continuity of fluid domain are significant in suppressing pressure oscillation.

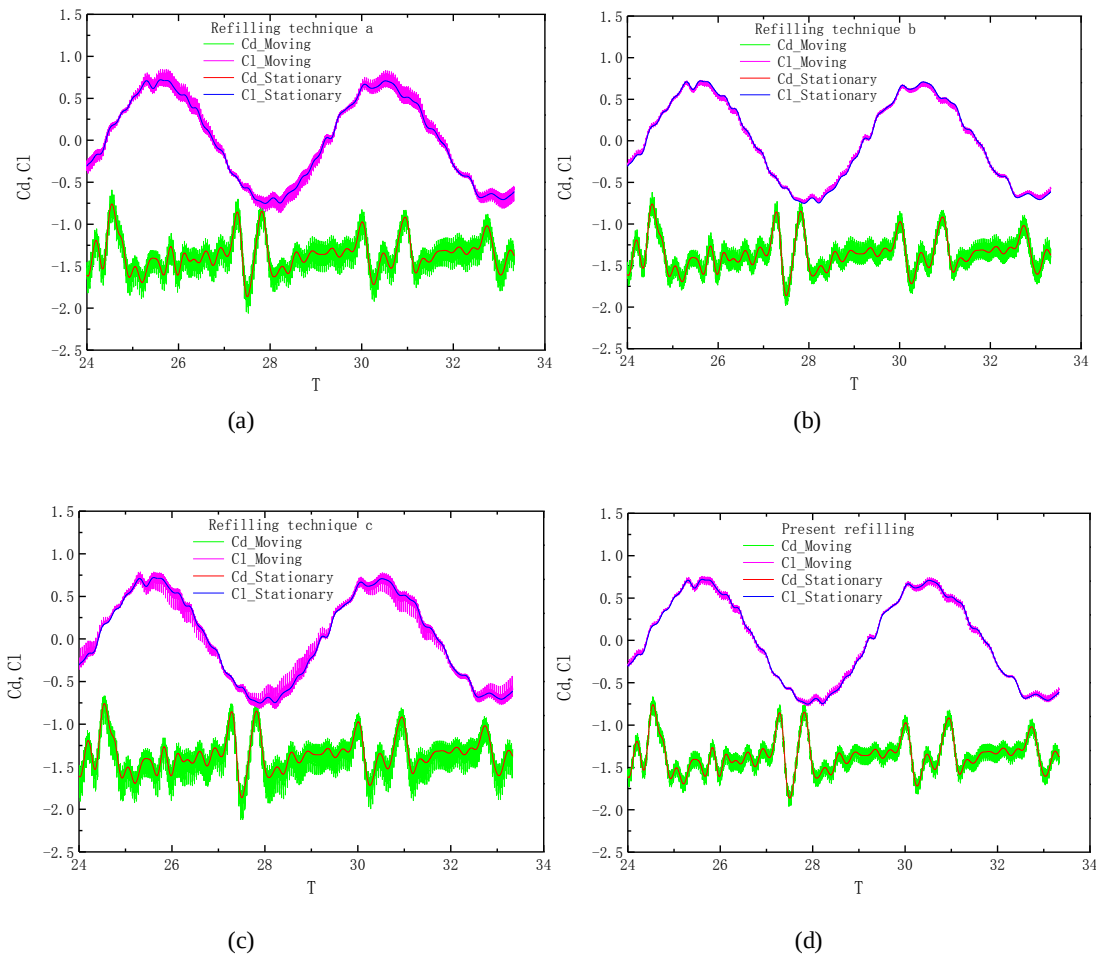


Fig.4.12 Comparison of drag and lift coefficients of the moving boundary case with the stationary boundary case using refilling technique a, b, c and the present refilling technique

Table 4.4 Mean squared error of different refilling techniques

Refilling method	σ_x	σ_y
refilling a	0.081833190	0.040830258
refilling b	0.069095846	0.025474104
refilling c	0.093231449	0.037505413
present refilling	0.055115660	0.021153156

4.4.3.4 Effects of collision models

In this section, different collision models, including SRT, TRT and MRT, are applied in the numerical simulation relatively. For all cases, simulations are conducted by Guo's extrapolation scheme with the present refilling technique and the forces are evaluated by modified momentum exchange method (B. Wen 2014).

Fig.4.13 (a) and (b) describe the comparison of the drag and lift force coefficients of

Chapter 4 Moving boundary in the fluid-solid interactions

stationary boundary case and moving boundary case between the MRT model, TRT model and SRT model. It is noteworthy that the forces evaluated with the SRT model and the TRT model nearly overlap each other. For the stationary boundary case, the spurious pressure oscillation can be found in both the SRT model and the TRT model, which verifies the significance of the relaxation rates of non-conserved terms—energy term and energy square term on stabilize the pressure field (Luo et al. 2011). For the moving boundary case, the spurious pressure oscillation is more distinct in the results of the SRT model, which further demonstrates the superiority of the MRT model in eliminating the spurious pressure oscillation in moving boundary issue. The discrepancy of relaxation rate parameters between the TRT model and the MRT model is only that s_e and s_ϵ in the TRT model are set to be equal to s_v , however, the results of the TRT model almost degrade to the same as those of the SRT model for this scenario, which validate the pivotal effect of dissipation of energy modal on pressure field. If the s_q in the TRT model is set as Eq.(2-32), the forces seem to oscillate more violently for this scenario, which are not shown here. These results validate that the dissipation of sound waves or pressure fluctuation associated with the bulk viscosity (s_e) is critical in the treatment of moving boundaries of the lattice Boltzmann method. If the results of the stationary case simulated by the MRT model are thought of as the standard solution, the mean squared error of the SRT model (stationary and moving boundary case) and TRT model (stationary and moving boundary case) and MRT model (moving boundary case) can be calculated, which are stated in Table 4.5. Take the results in Table 4.4 into consideration, it can be found that the force oscillation due to the collision model are several times violent than the force oscillation induced by the rough refilling techniques, which insinuates that the influence extent of the collision model on the pressure fluctuation with the moving boundary might be more remarkable than the refilling techniques.

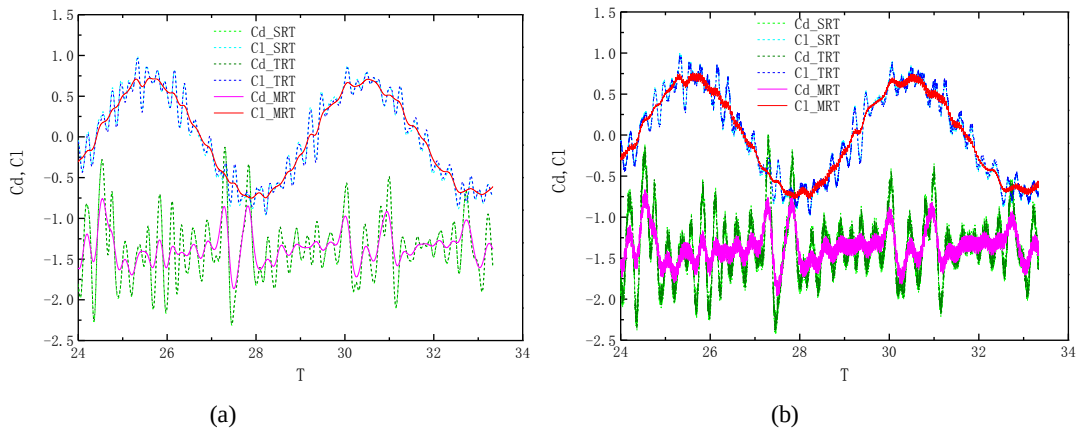


Fig.4.13 Results comparison with different collision models (a) stationary boundary case; (b) moving boundary case

Table 4.5 Mean squared error of different collision models

Cases	σ_x	σ_y
SRT (stationary)	0.23775438	0.116448903
SRT (moving)	0.25154915	0.114812734
TRT (stationary)	0.2369527	0.112014959
TRT (moving)	0.24783215	0.114670593
MRT (moving)	0.05514218	0.020729062

4.4.3.5 Effects of force evaluation methods

In this section, the modified momentum exchange method (B. Wen 2014) and the stress integration method proposed by Inamuro et al. (2000) are applied to calculate the forces exerted on the obstacle. For all the cases, simulations are conducted by Guo's extrapolation scheme using the MRT model with refilling technique a. Essentially, for this numerical scenario, the velocity field and the pressure field are the same in the simulation process while only the force exerted on the obstacle are evaluated by different approaches.

From Fig.4.14, obviously it can be found that the pressure oscillation can be remarkably eliminated by the utilization of the stress integration method compared with the modified momentum method, especially for the lift force. With the rough refilling technique a, the stress integration method also can achieve the slighter force oscillation, which mainly contributes to the large number of force points arranged on the boundary (800 points in the simulation) while the lattice link numbers in the modified momentum exchange method are much smaller than the force points in the stress integration method. Since the stress integration method concerns complicated extrapolation treatment, it is necessary to check the efficiency of these two approaches. The code with these two force evaluation methods are tested on the CPU specified in Section 6.1, the computation times are 328.793s and 317.809s for the stress integration method and the modified momentum exchange method, respectively, which indicates that the simulation time increases around 3.45% of the total simulation time when the stress integration method is utilized to evaluate the forces.

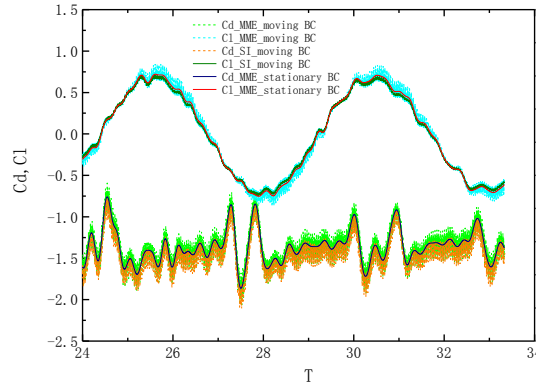


Fig.4.14 Results comparison of SI and MME with refilling technique a

4.4.4 Oscillation cylinder in a channel

In this numerical case, different schemes will be compared and the force evaluation methods on the accuracy will be investigated again. Furthermore, the resolution will also be verified to improve the accuracy and suppress the pressure oscillation. The influence of the collision models and refilling techniques on the force oscillation are not remarkable such that the comparisons concerning these two issues are neglected in this numerical scenario. The geometric schematic diagram of the flow field is depicted in Fig.4.15. The cylinder with diameter of D is initially placed at the center of the fluid region (a $4D \times 4D$ square area).

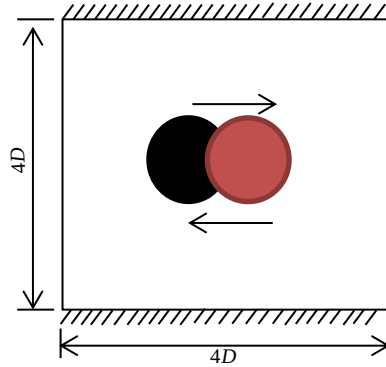


Fig.4.15 Geometric schematic of flow field

The cylinder oscillates sinusoidally with the following motion equation:

$$x_c(t) = x_c(0) + X[1 - \cos(2\pi ft)] \quad (4-39)$$

The moving velocity can be obtained as

$$u_c(t) = U \sin(2\pi ft) \quad (4-40)$$

where the characteristic velocity is $U = 2\pi fX = 0.1$, the oscillation amplitude is X , and the oscillation period is $T = 1/f$. Reynolds and Strouhal numbers are defined as $Re = UD/\nu$ and $St = fD/U$. One dimensionless parameter $C_{pd} = F_x / (0.5D^3 f^2)$ is defined to consider the drag force exerting on the cylinder. The amplitude, Reynold number and Strouhal number are set as $X = 0.125D$, $Re = 78.5$, $St = 1.27$. For the initial conditions, the flow field is stationary and the equilibrium PDF is set as the initial values for PDF. On the left and right of the flow domain, the boundary conditions associated with the velocity and pressure are all set as Neumann boundary condition (zero normal gradient condition), while the up and down are set as non-slip boundary condition. The relaxation parameters for MRT model are the same as the second simulation case. The number of the grid node across the diameter of the cylinder takes $n = 8, 16, 32, 64$.

4.4.4.1 Comparison of different schemes

In this subsection, the MRT collision model, the present refilling technique and the modified momentum exchange method (B. Wen 2014) are chosen for all the boundary treatment schemes. Different resolutions are tested to verify its validity on suppressing the force oscillation and improving the accuracy.

Fig.4.16 (a, b, c, d, e) depict the results calculated by BGFM, QGFM, LIBB, QIBB and Guo's scheme with various resolutions. It can be found that the drag force of all the schemes can converge with the increasing resolution and the non-smooth of the force gradually disappears with the increasing resolution, which indicates that the high resolution can effectively cure both the issues of force oscillation and accuracy of the force. In Fig.4.16 (f), the drag forces of all the schemes with the highest resolution are compared. It is obvious that the results of all the schemes coincide with each other precisely, from which it can be made the conclusion that all the schemes can well handle the force evaluation issue of moving boundary with the fine resolution.

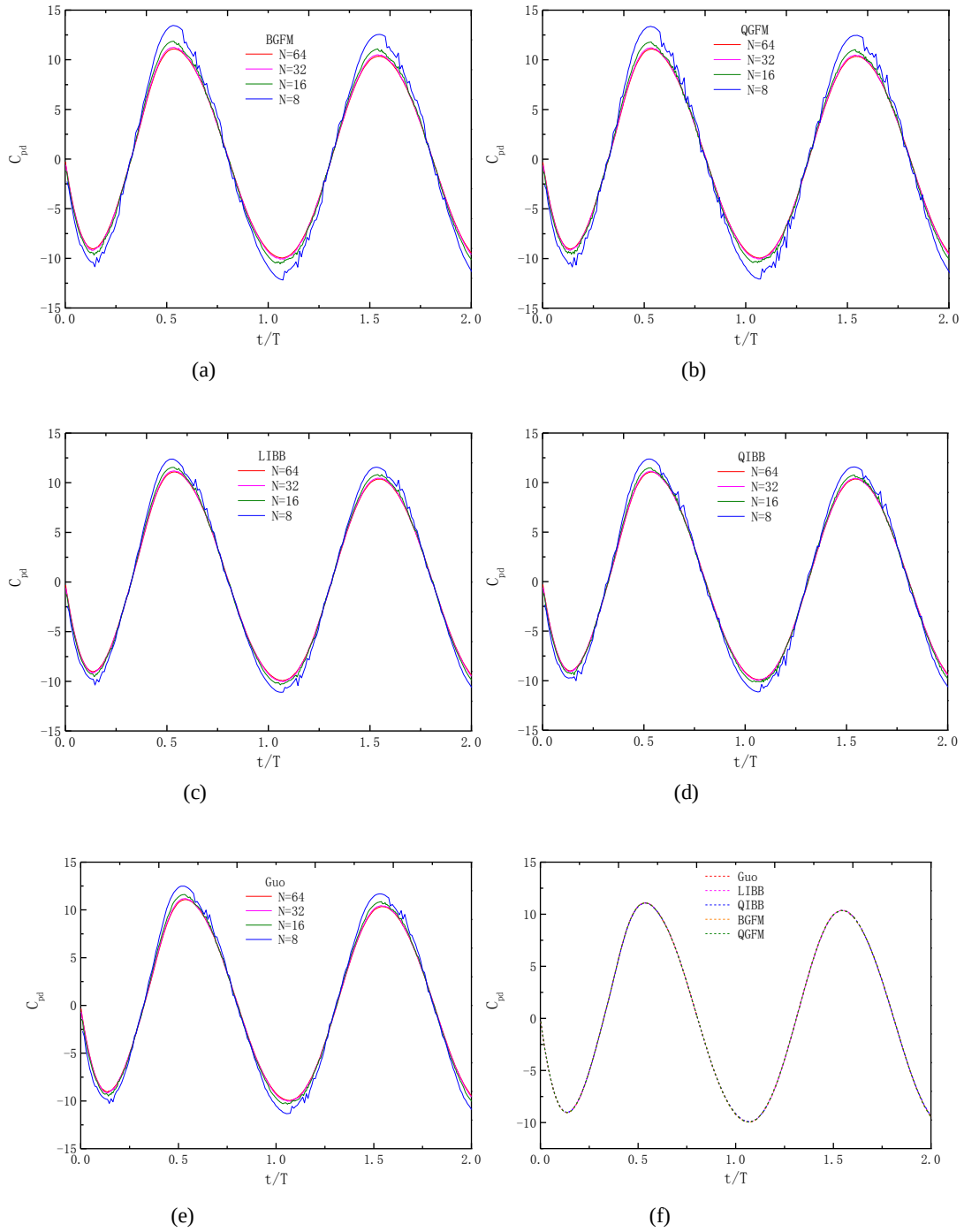


Fig.4.16 Comparison of different schemes for the drag coefficient (a) BGFM (b) QGFM (c) LIBB (d) QIBB (e) Guo (f) all schemes with $N=64$

4.4.4.2 Effects of force evaluation methods

In this section, the stress integration method (Inamuro et al. 2000) and the modified momentum exchange method (B. Wen et al. 2014) are performed to calculate the forces exerted on the obstacle. The simulations are conducted by Guo's extrapolation scheme with the present

refilling technique and the MRT model.

Fig.4.17 presents the results comparison of the two force evaluation methods with the highest resolution. The two curves coincide with each other and thus are regarded as the reference results which will be used in the later comparison.

Fig.4.18 (a), (b) describe the comparison of the two force evaluation schemes with the reference result at $D = 8\Delta x$ and $D = 16\Delta x$. The results of MME deviate from the reference result considerably while the results of SI are close to the reference result even at low resolution. In Fig.4.18 (c), the two curves coincide with the reference result well, in which the resolution plays a significant role.

The results in Section 6.3.5 and these results indicate that the forces evaluated by different approaches present significant discrepancy even with the same pressure field (the velocity and motion of the obstacle are given). With insight from these results, several points can be addressed. Firstly, the momentum exchange method is concise and easy to be conducted and implemented. However, it only counts the momentum exchange on the finite lattice links at low resolution, which can be regarded as the original inducement of the non-accuracy of the force evaluation from the viewpoint of statistics. As a result, the pressure oscillation can be observed in the force evaluation and the evaluated forces are not accurate with low resolution. Secondly, the stress integration method can provide accurate results even with low resolution while it requires complex extrapolation operation, especially for the three dimensional case, but the efficiency is not a problem.

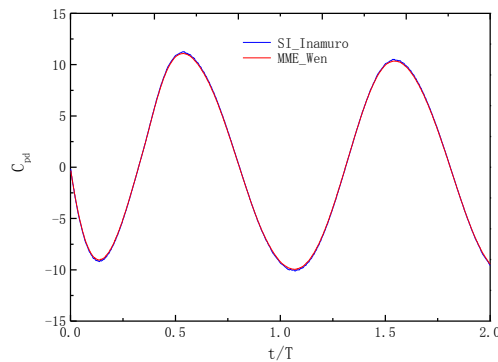


Fig.4.17 Drag coefficient evaluated by different force evaluation schemes with $D = 64\Delta x$

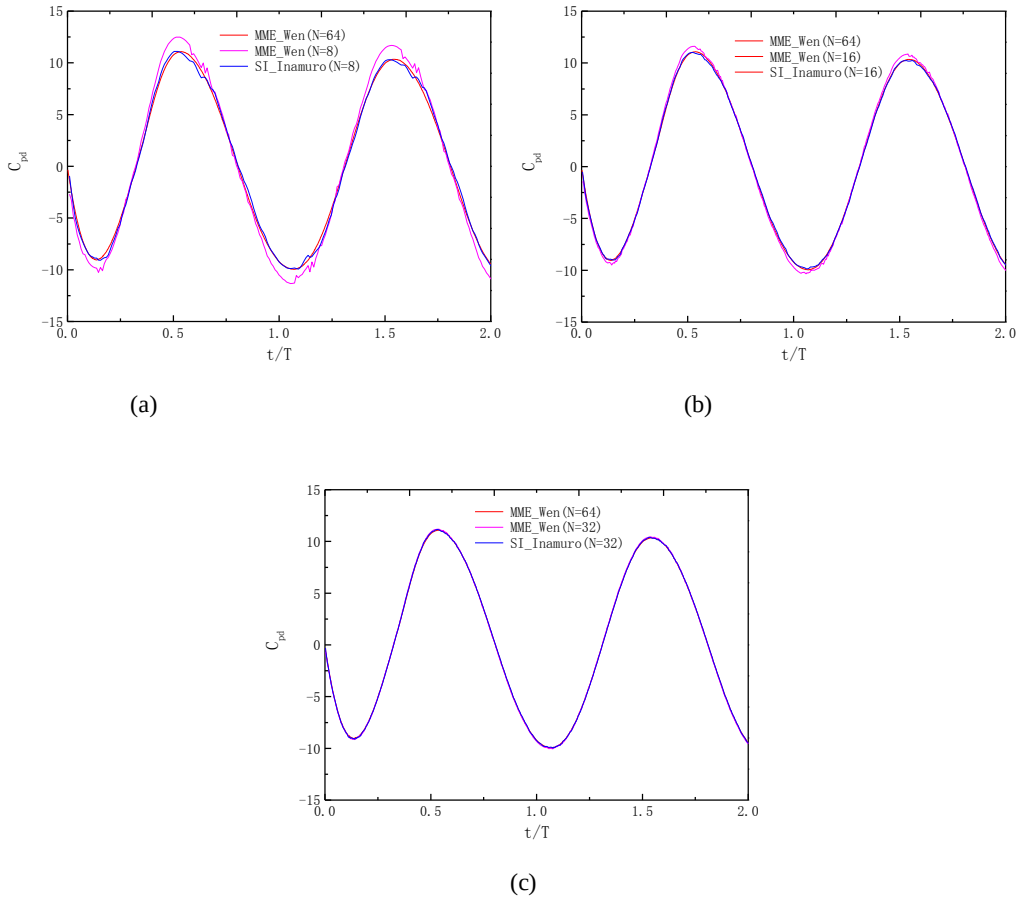


Fig.4.18 Drag coefficient evaluated by different force evaluation schemes with (a) $N=8$ (b) $N=16$ (c) $N=32$ grid number across the diameter of the cylinder

4.4.5 Particle suspension

The last case is the popular particle suspension problem which has been checked by many other LBM schemes (Eric Lorenz et al. 2009, J.R.Clausen,C.K.Aidun 2009, G.F.Zhou 2011, B. Wen et al. 2014, Y. Chen et al. 2013) and FEM (H.H. Hu, D.D. Joseph 1992, J. Feng et al. 1994), all of which aim to validate their ability to cope with moving complex boundaries. Different boundary treatment schemes, refilling techniques and force evaluation methods will be compared in detail while the effect of collision model can be neglected for this particle flow of low Reynolds number such that it is excluded in this numerical scenario.

The geometric arrangement of the flow field is shown as Fig.4.19. The channel width is 0.4 cm and the diameter of the particle d is 0.1 cm. The particle will be released at 0.076 cm away from the left wall, and then sink due to the gravity $g=980 \text{ cm}^2/\text{s}$. The fluid kinematic viscosity and its density are $0.01 \text{ cm}^2/\text{s}$ and 1 g/cm^3 . In lattice unit system, the relaxation time

τ in the SRT model is 0.6. The relaxation parameters for MRT model are $s_0 = 0, s_1 = 1.64, s_2 = 1.54, s_3 = 0, s_4 = 1.9, s_5 = 0, s_6 = 1.9, s_7 = 1/\tau, s_8 = 1/\tau$. The diameter of the particle is $d=26$ lattice units. The width and length of the channel are $L=104$ lattice units and 2200 lattice units. The particle is placed initially at 19.76 lattice units away from the left wall in the horizontal direction and at the center of the channel in the vertical direction. In terms of the boundary condition, the inlet is zero velocity condition while the zero normal derivative condition is applied at the outlet. The translation of the mass center and the rotation of the solid particle are updated at each time step by the Euler explicit scheme. We investigate the case with solid-fluid density ratio 1.03.

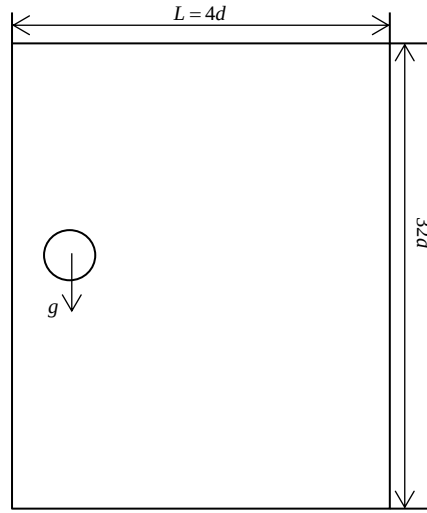


Fig.4.19 Schematic for the flow field arrangement

4.4.5.1 Comparison of different schemes

In this subsection, the MRT collision model, the present refilling technique and the modified momentum exchange method (B. Wen et al. 2014) are chosen for all of the schemes.

Fig.4.20 describes the time-dependent forces and torque exerted on the particle. It is obvious that the oscillations of Guo's scheme, QIBB scheme and LIBB scheme are slighter than those of the BGFM scheme and QGFM scheme. Fig.4.21 depicts the time-dependent trajectory, translation velocity and rotational velocity with different boundary treatment schemes. The results of all the schemes agree with the reference results (FEM) (H.H. Hu, D.D. Joseph 1992, J. Feng et al. 1994) very well, which indicates that all of the schemes are capable of the particle suspension flow of low Reynolds number. These results demonstrate that the simple LIBB

Chapter 4 Moving boundary in the fluid-solid interactions

scheme and Guo's scheme can be chosen to handle the low Reynolds particle suspension problem, especially for the flow with a large amount of particles for their simplicity.

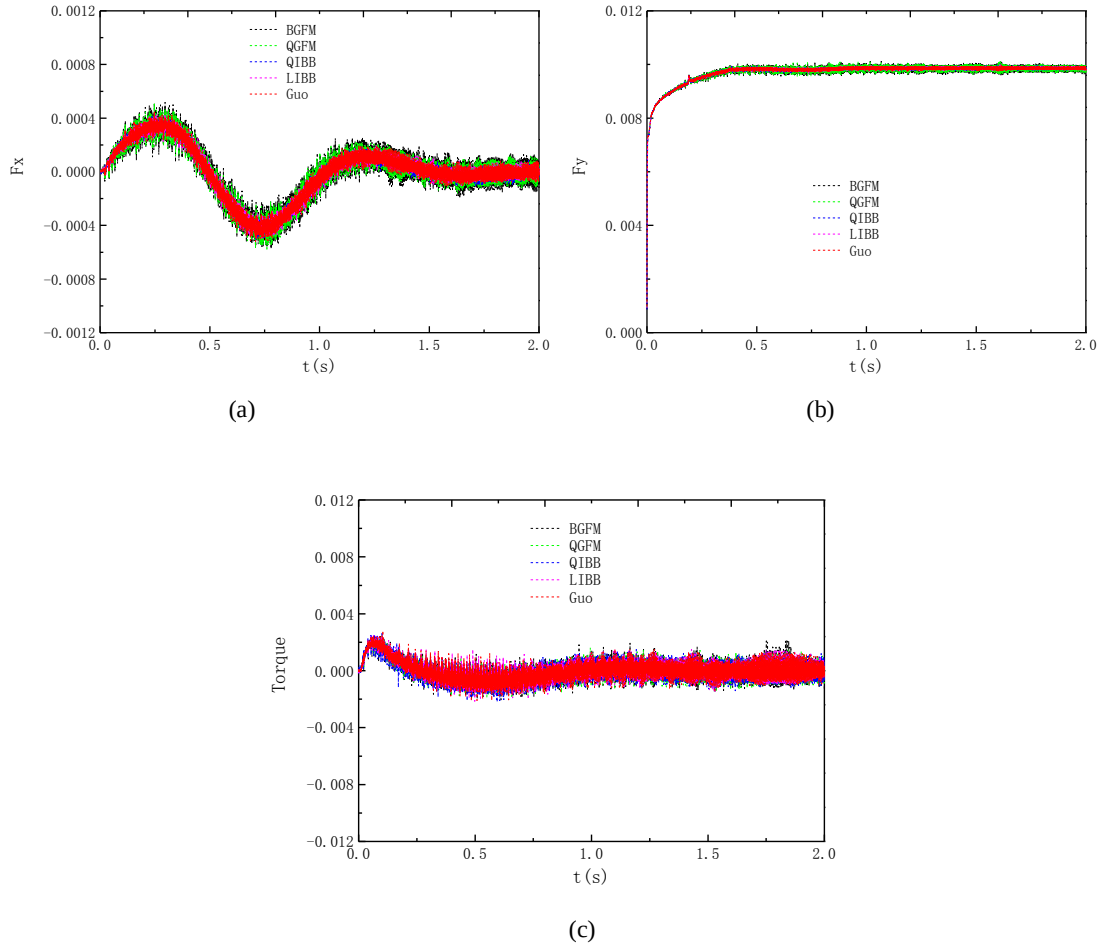
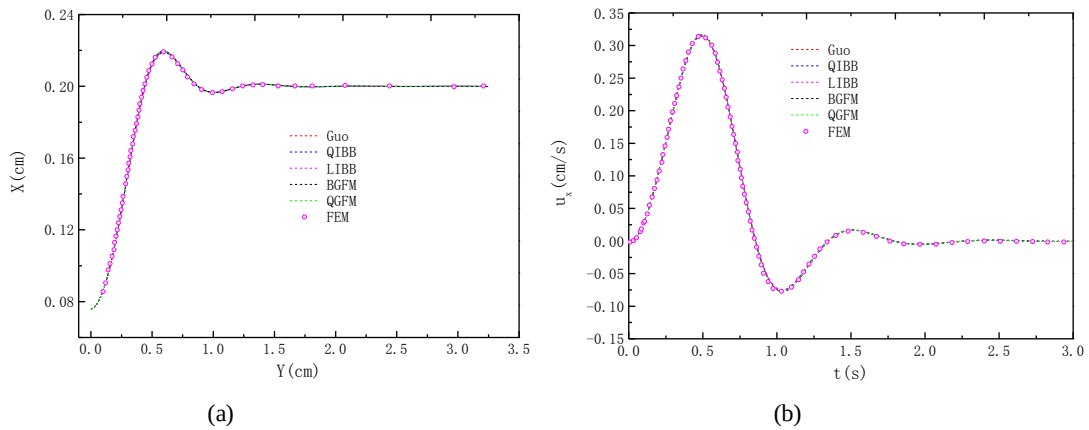


Fig. 4.20 The time-dependent forces and torque with different schemes



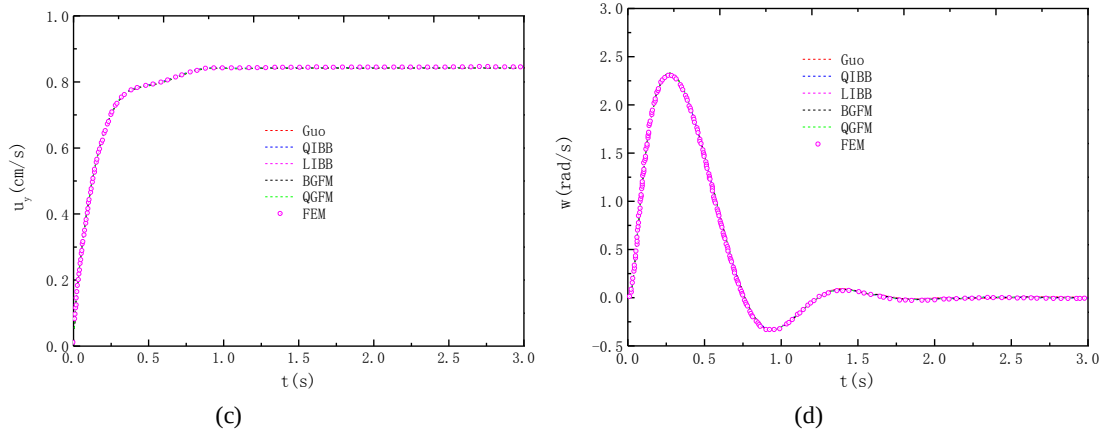
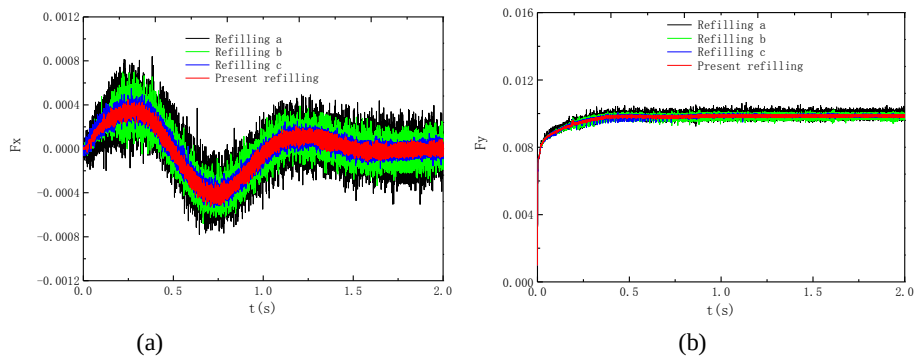


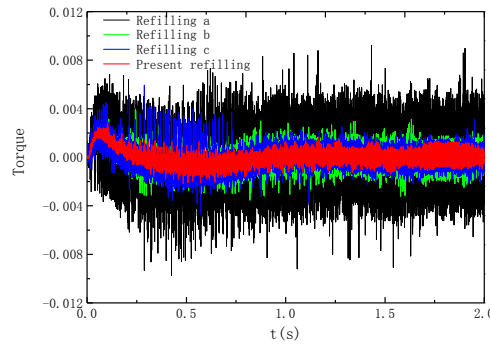
Fig.4.21 The time –dependent trajectory and velocities for different schemes

4.4.5.2 Effects of refilling technique

In this subsection, all simulations are conducted by Guo’s extrapolation scheme with the MRT model and the force is evaluated by the modified momentum exchange method (B. Wen et al. 2014). Different refilling techniques, including the refilling technique a, b, c and the present refilling technique, are applied in the simulation.

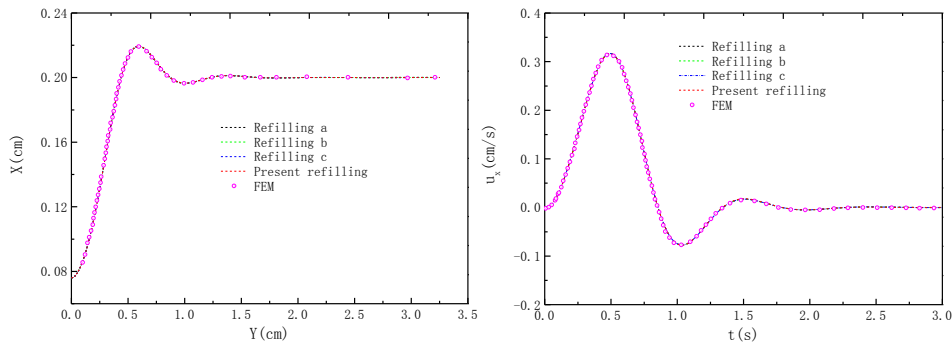
From Fig.4.22, it is found that the present refilling technique can suppress the force oscillation remarkably compared with the refilling technique a, b and c, which indicate the superiority of the present refilling technique. In Fig.4.23, it can be found that there exist almost no discrepancy among different refilling techniques for the velocities and trajectories of the particle. Actually, it is not strange for this kind of physical phenomenon. The velocities are the integral variables of forces, which insinuate that the oscillations of forces could be cancelled with each other. Furthermore, the trajectory is the integral variable of velocity, which indicates that the error of motion will be the higher-order error such that difference can be hardly found among the results of different refilling techniques.





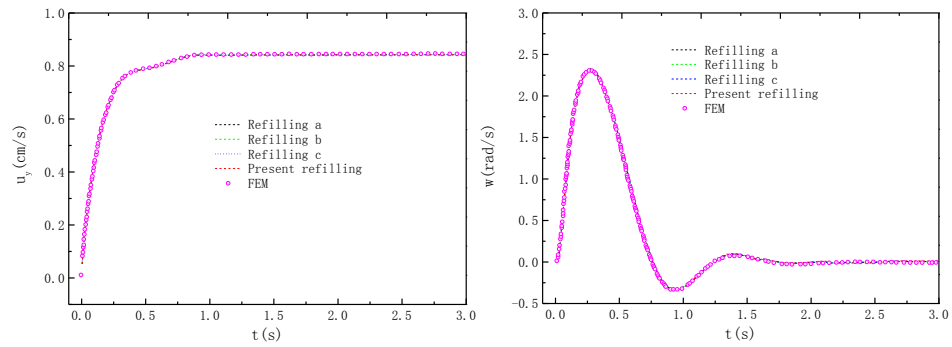
(c)

Fig.4.22 The time-dependent forces and torque with different refilling techniques



(a)

(b)



(c)

(d)

Fig.4.23 The time –dependent trajectory and velocities for different refilling techniques

4.4.5.3 Effects of force evaluation methods

In this subsection, all simulations are conducted by Guo's extrapolation scheme with the MRT model and the present refilling technique. The modified momentum exchange method (B. Wen et al. 2014) and the stress integration method proposed by Inamuro et al.(2000) are applied in the simulation.

From Fig.4.24, it can be found that the force oscillations of stress integration are a little bit slighter than those of the modified momentum exchange method, especially for the torque. Nevertheless, in Fig.4.25, the slight discrepancies among the velocities could be found compared with the reference data (H.H. Hu, D.D. Joseph 1992, J. Feng et al. 1994) for the stress integration method, while the trajectory of both methods can agree with the reference data well, which indicates that the modified momentum exchange method is not only a concise but also accurate method for the force evaluation method in lattice Boltzmann equation. Considering the physical phenomena in all subsections of this numerical case, it could be concluded that the most significant factor that affects the motions of the particle suspension at low Reynolds number is the force evaluation method while the oscillations of forces can be cancelled when solve the velocities and furtherly smoothened when obtain the motion.

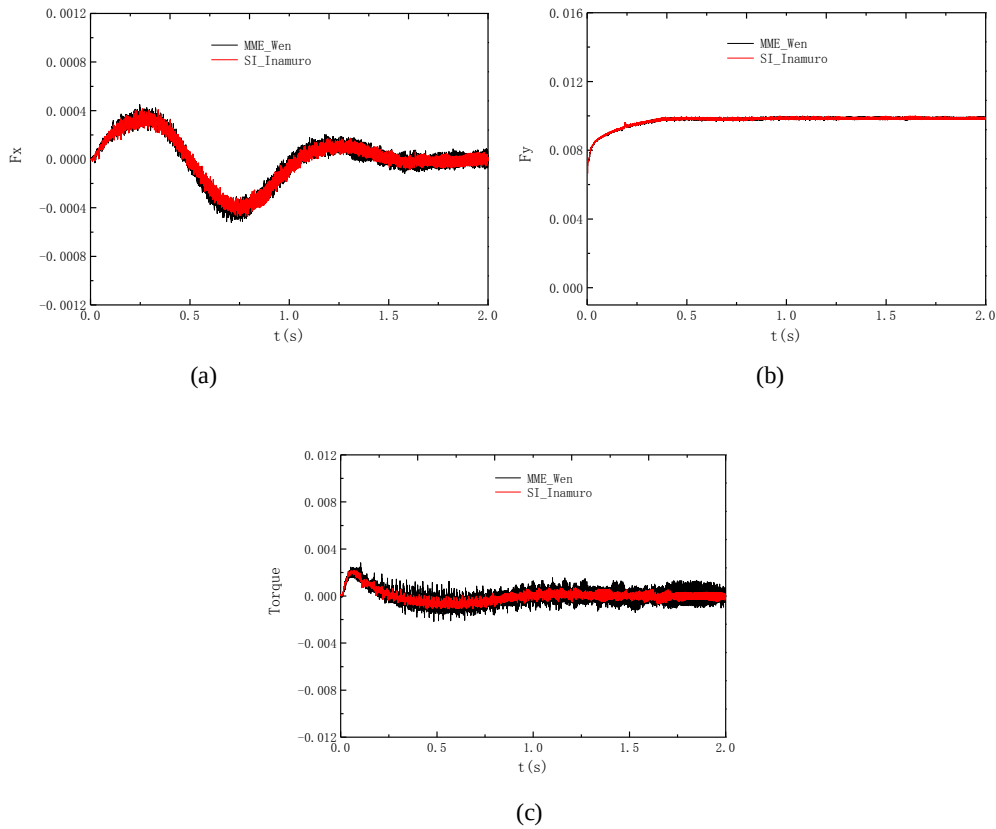


Fig.4.24 The time-dependent forces and torque with different force evaluation methods

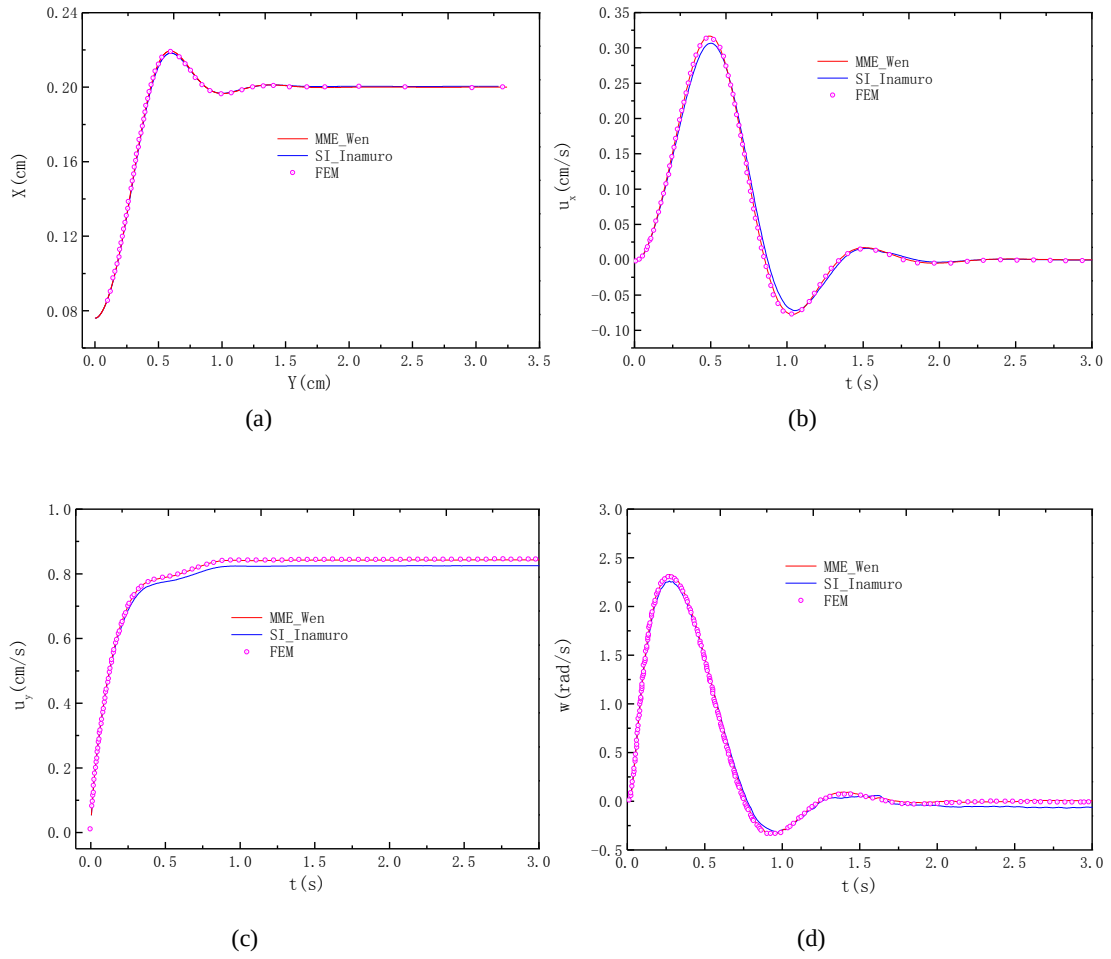


Fig.4.25 The time –dependent trajectory and velocities for different force evaluation methods

4.5 Concluding Remarks

In this paper, the ghost fluid immersed boundary lattice Boltzmann methods (BGFM and QGFM) have been compared with the conventional interpolation or extrapolation bounce-back schemes. The accuracy and the pressure oscillation feature in moving boundary problems for different collision models, refilling techniques and force evaluation methods have also been investigated and discussed. The following conclusions can be addressed.

- The interpolation bounce-back schemes (LIBB, QIBB) , the extrapolation bounce-back scheme (Guo) and the ghost fluid methods (BGFM, QGFM) have been compared in terms of efficiency, accuracy and pressure oscillation phenomenon through five numerical examples. It can be found that the efficiency of LIBB, QIBB, Guo and BGFM are comparable while the computation time of QGFM is about 10% more than that of other schemes, which is mainly due to the complex stencil construction. The accuracies

of QGFM and QIBB are better than those of their linear counterparts BGFM and LIBB. Nevertheless, the accuracies of BGFM and QGFM are worse than those of the conventional bounce-back schemes (LIBB, QIBB and Guo). The accuracy of QGFM is only comparable to the Guo's scheme. As for the pressure oscillation suppression, the Guo's scheme shows the best performance while the BGFM scheme shows the worst. These numerical results demonstrate that the BGFM and QGFM, of which the origins goes back to the Navier-Stokes solver, have no evident advantages over the LBM treatments, such as LIBB, QIBB and Guo's extrapolation scheme in which the interpolation or extrapolation direction is only along the lattice-link direction.

- From the numerical results of the impulsively started cylinder case, it can be found that for the force oscillation issue, the effect of the collision model is much more significant than that of the refilling technique and the force evaluation method, which indicates that the collision model will affect the fluctuation of the pressure field from the fundamental approach;
- The proposed refilling technique can suppress the pressure oscillation to a lower level than those of other refilling techniques. The weighting strategy for density, velocity and the non-equilibrium distribution function plays significant role in the re-initialization of distribution function on fresh fluid nodes to smooth the flow field near the boundary.
- In terms of the force evaluation methods, the recently developed modified momentum exchange method proposed by Wen et al.(2014) is compared with the stress integration method proposed by Inamuro et al.(2000). The numerical results indicate that the stress integration method generally performs better than the modified momentum exchange method for low spatial resolution calculation. If fine resolution is used, the performance of modified momentum exchange method can be better. However, it should be noted that such pressure oscillation may not affect the computation accuracy of the particle velocity and trajectory if suitable force evaluation method is chosen, such as the modified momentum exchange method proposed by Wen et al.(2014).

In general, the collision model matters the fluctuation of the pressure field; the refilling technique influences the consistence of flow filed with moving boundary; the force evaluation approach affects the accuracy of the force calculation.

Chapter 5 Free Surface Model and Its Applications

In this chapter, the single-phase free surface flow is investigated and enhanced to model the violent nonlinear impact waves on the structures in ocean engineering field. Based on the analysis of the accuracy and pressure oscillation for the fluid-solid interaction of LB solver in chapter 4, all the possible factors are considered in the present free surface lattice Boltzmann solver (FSLB solver) to eliminate the spurious pressure oscillation.

5.1 Free surface flow with floating bodies

In the present solver, a flag array is applied to identify different phases: fluid, gas and solid. Meanwhile, there exit three interfaces among them: fluid-gas interface, fluid-solid interface and solid-gas interface, as shown in Fig. 5.1. In the present model, the gas and the solid regions are not necessary to be resolved. The lattice Boltzmann equations are only resolved in the fluid domain. This treatment actually saves a huge computation resource. However, the interface treatment seems to be complicated. Attributting to the negligence of the gas phase, the atmospheric pressure should be incorporated into the solver by a dynamic pressure boundary condition on the moving free surface. The fluid-solid interface should be also handled by the moving boundary condition for fluid-solid interaction which can be found in our previous work (Xuhui Li et al., 2016) and Chapter 4. Fortunately, the interface between solid and gas is not necessary to be treated. In the following context, we elaborated them separately.

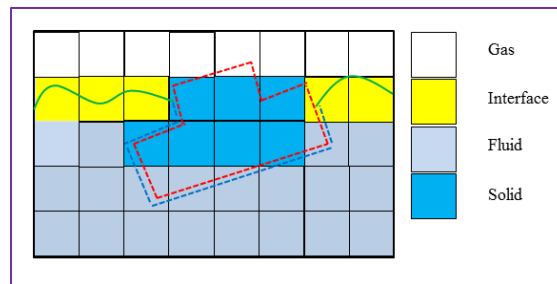


Fig. 5.1 Schematic of the wave-body interaction

5.1.1 Treatment of the free surface

5.1.1.1 Free surface boundary condition

On the free surface, both the velocity boundary condition and dynamic boundary condition should be satisfied. In the present model, the gas phase is ignored and the velocity boundary condition is consistent naturally. For the dynamic boundary condition, the force owing to the atmospheric pressure has to be balanced by the force from the fluid phase. This indicates that the force balance state should be imposed on the interface cell. In general, the momentum exchange method is widely performed to evaluate the hydrodynamics force. The momentum from the fluid side can be obtained directly from the distribution functions streaming from the fluid cell while the momentum contributing to the gas side is unknown for the missing distribution functions streaming from the gas phase. Therefore, the reconstruction procedure for the missing distribution function from the gas cell is indispensable. The underlying philosophy of the reconstruction procedure is to apply the relationship that the known atmospheric pressure should be identical to the hydrodynamic force on the interface cell to resolve the missing distribution functions. The total hydrodynamic force originates from the particle currents from the liquid to the gas and vice versa. As shown in Fig. 5.2, the particles with $\mathbf{n} \cdot \mathbf{e}_i < 0$ will transport from liquid side to the gas side and the particles with $\mathbf{n} \cdot \mathbf{e}_i > 0$ will transport from the gas side to liquid side.

The total hydrodynamic force exerted on the interface cell (interface area $S(\mathbf{x})$) consists of the momentum contribution from the liquid side and the gas side:

$$\mathbf{F} = -\mathbf{n} \cdot S(\mathbf{x}) \left[\sum_{i, \mathbf{n} \cdot \mathbf{e}_i < 0} \hat{f}_i(\mathbf{x}, t) (e_{i,\alpha} - u_\alpha) (e_{i,\beta} - u_\beta) + \sum_{i, \mathbf{n} \cdot \mathbf{e}_i \geq 0} \hat{f}_i(\mathbf{x} - \mathbf{e}_i, t) (e_{i,\alpha} - u_\alpha) (e_{i,\beta} - u_\beta) \right] \quad (5-1)$$

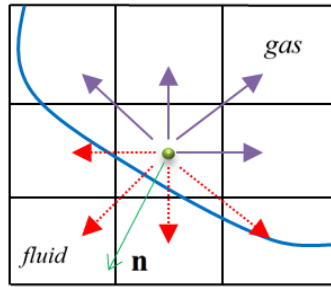


Fig. 5.2 Missing distribution functions at interface cell after advection

In the above equation, to be rigorous the pure advection term is eliminated by subtracting the interface velocity from the particle velocities. In fact, the final result is the same if the advection velocity is not subtracted from the particle velocity.

Chapter 5 Free Surface Model and Its Applications

The left thing is to reconstruct the unknown missing distribution functions to guarantee that the hydrodynamic force $\mathbf{F}$ is balanced by the gas pressure. Making use of the given gas pressure and the velocity in the interface cell, the missing distribution functions are reconstructed as followings:

$$f_i^{\wedge}(\mathbf{x}-\mathbf{e}_i, t) = f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u}) + f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u}) - f_i^{\wedge}(\mathbf{x}, t), \forall i: \mathbf{n} \cdot \mathbf{e}_i \geq 0 \quad (5-2)$$

where $\rho^G = 3p^G$ is the gas pressure and $\mathbf{u}$ is the velocity of the interface cell.

In order to verify the correctness of the above reconstruction of missing distribution functions, substitute Eq.(5-2) into Eq.(5-1), we have

$$\begin{aligned} \mathbf{F} / S(\mathbf{x}) &= -\mathbf{n} \cdot \left[\sum_{i, \mathbf{n} \cdot \mathbf{e}_i < 0} f_i^{\wedge}(\mathbf{x}, t)(e_{i,\alpha} - u_{\alpha})(e_{i,\beta} - u_{\beta}) \right. \\ &\quad \left. + \sum_{i, \mathbf{n} \cdot \mathbf{e}_i \geq 0} f_i^{\wedge}(\mathbf{x}-\mathbf{e}_i, t)(e_{i,\alpha} - u_{\alpha})(e_{i,\beta} - u_{\beta}) \right] \\ &= -\mathbf{n} \cdot \sum_{i, \mathbf{n} \cdot \mathbf{e}_i < 0} f_i^{\wedge}(\mathbf{x}, t)(e_{i,\alpha} - u_{\alpha})(e_{i,\beta} - u_{\beta}) \\ &\quad - \mathbf{n} \cdot \sum_{i, \mathbf{n} \cdot \mathbf{e}_i \geq 0} [f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u}) + f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u}) - f_i^{\wedge}(\mathbf{x}, t)](e_{i,\alpha} - u_{\alpha})(e_{i,\beta} - u_{\beta}) \\ &= -\mathbf{n} \cdot \sum_{i, \mathbf{n} \cdot \mathbf{e}_i \geq 0} [f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u}) + f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u})](e_{i,\alpha} - u_{\alpha})(e_{i,\beta} - u_{\beta}) \\ &= -\mathbf{n} \cdot \sum_i f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u})(e_{i,\alpha} - u_{\alpha})(e_{i,\beta} - u_{\beta}) \\ &= -\mathbf{n} \cdot \left[\sum_i f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u}) e_{i,\alpha} e_{i,\beta} - u_{\alpha} \sum_i f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u}) e_{i,\beta} \right. \\ &\quad \left. - u_{\beta} \sum_i f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u}) e_{i,\alpha} + u_{\alpha} u_{\beta} \sum_i f_i^{eq}(\mathbf{x}, \rho^G, \mathbf{u}) \right] \\ &= -\mathbf{n} \cdot [\rho^G u_{\alpha} u_{\beta} + p^G \delta_{\alpha\beta} - \rho^G u_{\alpha} u_{\beta} - \rho^G u_{\alpha} u_{\beta} + \rho^G u_{\alpha} u_{\beta}] \\ &= -\mathbf{n} \cdot p^G \mathbf{I} \end{aligned} \quad (5-3)$$

From the above derivation, it can be found that the hydrodynamic force per unit area exerted on the interface is identical to the gas pressure if the missing particle distribution functions are reconstructed by Eq. (5-2). It is noteworthy to be addressed that all the distribution functions with $\mathbf{n} \cdot \mathbf{e}_i \geq 0$ are reconstructed by Eq. (5-2) aiming at guaranteeing the symmetry of the interface and the force balance. Generally, half of the particle distribution functions are reconstructed.

5.1.1.2 Movement of Free surface

The algorithm developed in N. Thurey (2007) is applied to capture the free surface flow. The algorithm uses the volume of fluid (VOF) to track the interface motion. Two values, the mass m and the fluid fraction ε are introduced and their relationship is as follows:

$$\varepsilon = m / \rho \quad (5-4)$$

The mass evolution is more direct than the interface treatments in the traditional finite differential method or finite volume method. The change of mass is directly computed from the particle distribution functions which are streamed between the neighborhood cells in every direction. One can get the relation:

$$\Delta m_i(\mathbf{x}, t + \Delta t) = \frac{1}{2} [\varepsilon(\mathbf{x} + \mathbf{e}_i \Delta t, t) + \varepsilon(\mathbf{x}, t)] s_e$$

$$\text{with } s_e = [f_{\tilde{i}}(\mathbf{x} + \mathbf{e}_i \Delta t, t) - f_i(\mathbf{x}, t)] \quad (5-5)$$

$\tilde{i}$ is the inverse direction of i . Thus the mass at the new time step at the cell center can be obtained by summing the mass change of all directions.

$$m(\mathbf{x}, t + \Delta t) = m(\mathbf{x}, t) + \sum_{i=1}^q \Delta m_i(\mathbf{x}, t + \Delta t) \quad (5-6)$$

The algorithm described in N. Thurey (2007) actually incorporated some artifacts. For the single interface cells that are left alone, which means that there is no fluid neighboring node around it, it will be forced to be empty when its mass drops to be below 0.1ρ . For the interface cells that are trapped in the fluid, which means that there is no gas neighboring node around it, it will be forced to be filled when its mass is over 0.9ρ . Considering these artificial operation, the term s_e in Eq.(5-5) should be replaced by corresponding term as listed in Table 5.1.

Chapter 5 Free Surface Model and Its Applications

Table 5.1 Replace s_e with corresponding operation to realize the artificial operation

	standard cell at $\mathbf{x}_{nb}$	no fluid neighbors at $\mathbf{x}_{nb}$	no empty neighbors at $\mathbf{x}_{nb}$
standard cell at $(\mathbf{x})$	$f_i(\mathbf{x}_{nb}, t) - f_i(\mathbf{x}, t)$	$f_i(\mathbf{x}_{nb}, t)$	$-f_i(\mathbf{x}, t)$
no fluid nbs at $(\mathbf{x})$	$-f_i(\mathbf{x}, t)$	$f_i(\mathbf{x}_{nb}, t) - f_i(\mathbf{x}, t)$	$-f_i(\mathbf{x}, t)$
no empty nbs at $(\mathbf{x})$	$f_i(\mathbf{x}_{nb}, t)$	$f_i(\mathbf{x}_{nb}, t)$	$f_i(\mathbf{x}_{nb}, t) - f_i(\mathbf{x}, t)$

After the mass advection step finished, two kinds of interface cells should be identified: one is the interface cell filled and the other is the interface cell emptied.

$$m(\mathbf{x}, t + \Delta t) > (1 + \kappa)\rho(\mathbf{x}, t + \Delta t) \quad (5-7a)$$

$$m(\mathbf{x}, t + \Delta t) < (0 - \kappa)\rho(\mathbf{x}, t + \Delta t) \quad (5-7b)$$

where $\kappa = 0.001$ is used to avoid the reconverting phenomenon of the interface cell conversion.

5.1.1.3 Flag Re-initialization

After the mass advection step and identification of two kinds of interface cells, the flag of different cell should be updated. For the interface cells already filled or emptied, the role of them should be changed and the excess positive or negative mass should be distributed into the neighboring cells. Meanwhile, the neighboring cells of these two types of interface cells should be changed correspondingly. The neighboring gas cells of the filled interface cells should be converted into interface cells while the neighboring fluid cells of the emptied interface cells should be converted into interface cells. From the viewpoint of programming, we can use two passes to finish the conversion and re-initialization. In the first loop, the filled interface cells are treated. The neighboring gas cells are converted to the interface cells and the distribution functions are re-initialized by the Maxwell equilibrium distribution, while the macroscopic variables such as density and velocity are the averaged extrapolated values from

the surrounding fluid or interface cells. In the second loop, the emptied interface cells are treated. The neighboring fluid cells are converted to interface cells, while the distribution functions are just inherited. For both the filled interface cells and the emptied interface cells, the flags of them should be changed to fluid cells and gas cells but restore the flag into a temporary flag array. After the two loops, the temporary flag can be assigned to the flag array. After these operations, the alone interface cell without fluid neighboring cells and the trapped interface cells without gas neighboring cells should be identified and store them into an array. Subsequently, the excessive mass distribution should be conducted. For the filled interface cell, the excessive mass is $m(\mathbf{x}, t + \Delta t) - \rho(\mathbf{x}, t + \Delta t)$ while for the emptied interface cell, the excessive mass is $m(\mathbf{x}, t + \Delta t)$. The weighted distribution is performed on the neighboring interface cells as followings:

For the filled cells,

$$\omega_i = \begin{cases} -\mathbf{n} \cdot \mathbf{e}_i, (\mathbf{n} \cdot \mathbf{e}_i < 0) \\ 0, (otherwise) \end{cases} \quad (5-8a)$$

For the emptied cells,

$$\omega_i = \begin{cases} \mathbf{n} \cdot \mathbf{e}_i, (\mathbf{n} \cdot \mathbf{e}_i > 0) \\ 0, (otherwise) \end{cases} \quad (5-8b)$$

Therefore,

$$m(\mathbf{x}, t + \Delta t) = m(\mathbf{x}, t + \Delta t) + m^{ex}(\omega_i / \omega_{total}) \quad (5-9)$$

ω_{total} is the summation of the weights ω_i .

After this procedure, the fraction of all cells needs to be calculated. From the above statement of the treatment of free surface, we can found that no direct conversions from fluid cells to gas cells and vice versa, which indicates that the fluid cells and gas cells are always separated by the interface cells.

5.1.2 Conversions of the liquid-gas-solid system

The fluid-solid moving boundary and the refilling algorithm of the fresh fluid node

emerged from the obstacle is of significance in dealing with the fluid-solid interface. For the moving boundary on the free surface, the similar treatments are necessary. The details can be found in Chapter 4. In order to maintain the closed interface layer, a valid strategy should be performed in the interaction of free surface and structure. Here, the conversion rule in Simon Bogner (2013) is used in our solver.

The solid-liquid-gas conversion rule is shown in Fig.5.3. The bottom liquid cell-liquid cell-gas cell conversion rule has been elaborated in the aforementioned context. Here only the conversion rule between the solid cell and the liquid, interface, gas cells is described.

Case a: solid cell without any gas neighboring cell at previous time step;

Case b: solid cell with both liquid and gas neighboring cell at previous time step;

Case c: solid cell without liquid neighboring cell at previous time step.

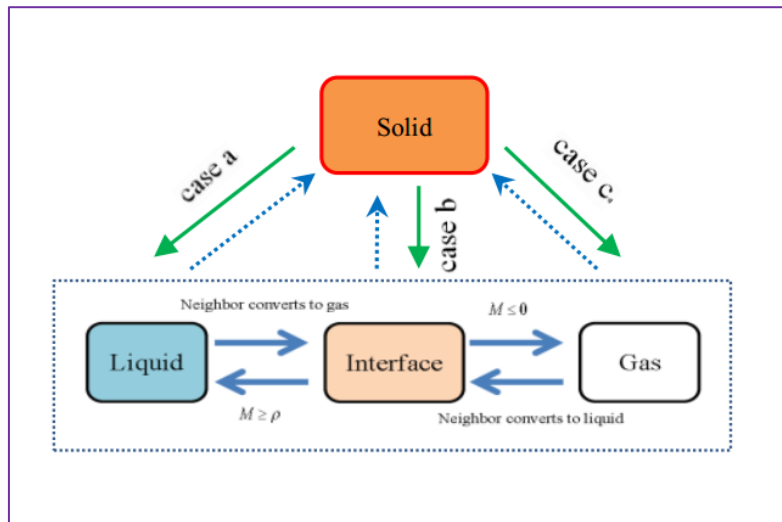


Fig. 5.3 Cell conversion rule for solid-gas-liquid

5.2 Numerical benchmarks

5.2.1 Dam breaking

To verify the accuracy and efficiency of the present GPU accelerated free surface solver, the experiment conducted by Hu & Sueyoshi (2010) has been used. The initial experiment configuration can be found in Fig.5.4. Length, height and width of the water tank are 0.8m, 0.6m and 0.2m, respectively. A water column with 0.2m length, 0.4m height and 0.2m width is initially placed at the left side with a removable vertical plate. During the simulation, the

position of the surge front is recorded. The comparison between the numerical simulation and the experiment data can be seen in Fig.5.5, in which L is the length of the water column. Good agreement and a convergent behavior can be observed in Fig.5.5. Meanwhile, the free surface evolutions of some specific times have been chosen to make further comparison. As can be observed in Fig.6.6, the free surface profiles captured by the present solver agree with the experiment snapshots very well. These results effectively validate the accuracy of the present solver.

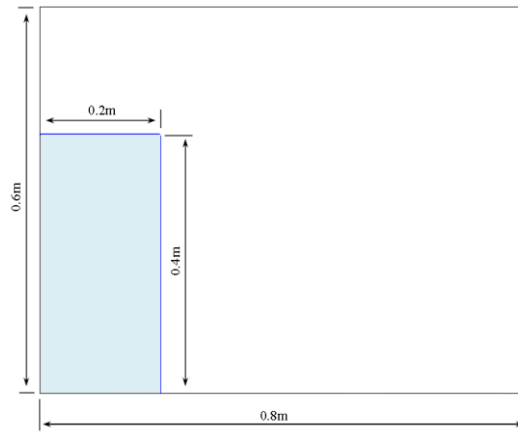


Fig. 5.4 Geometry configurations

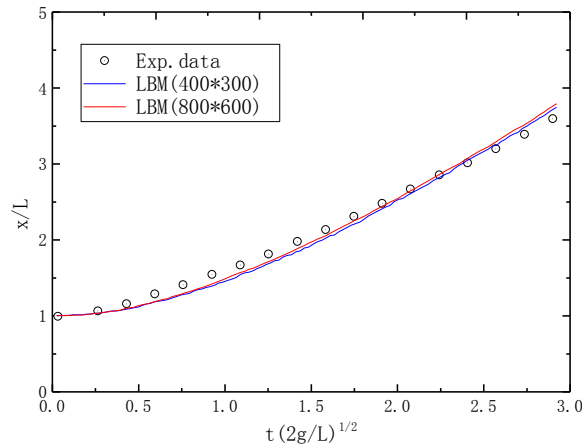
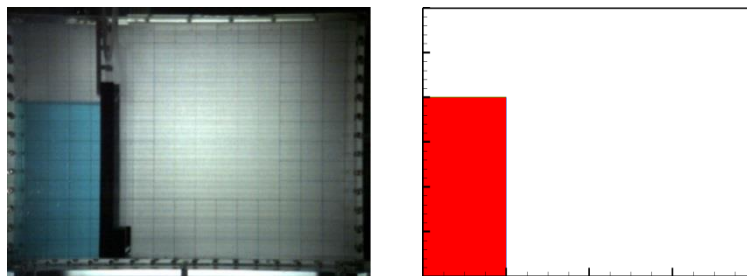


Fig. 5.5 Free surface front position (Exp. v.s. LBM)



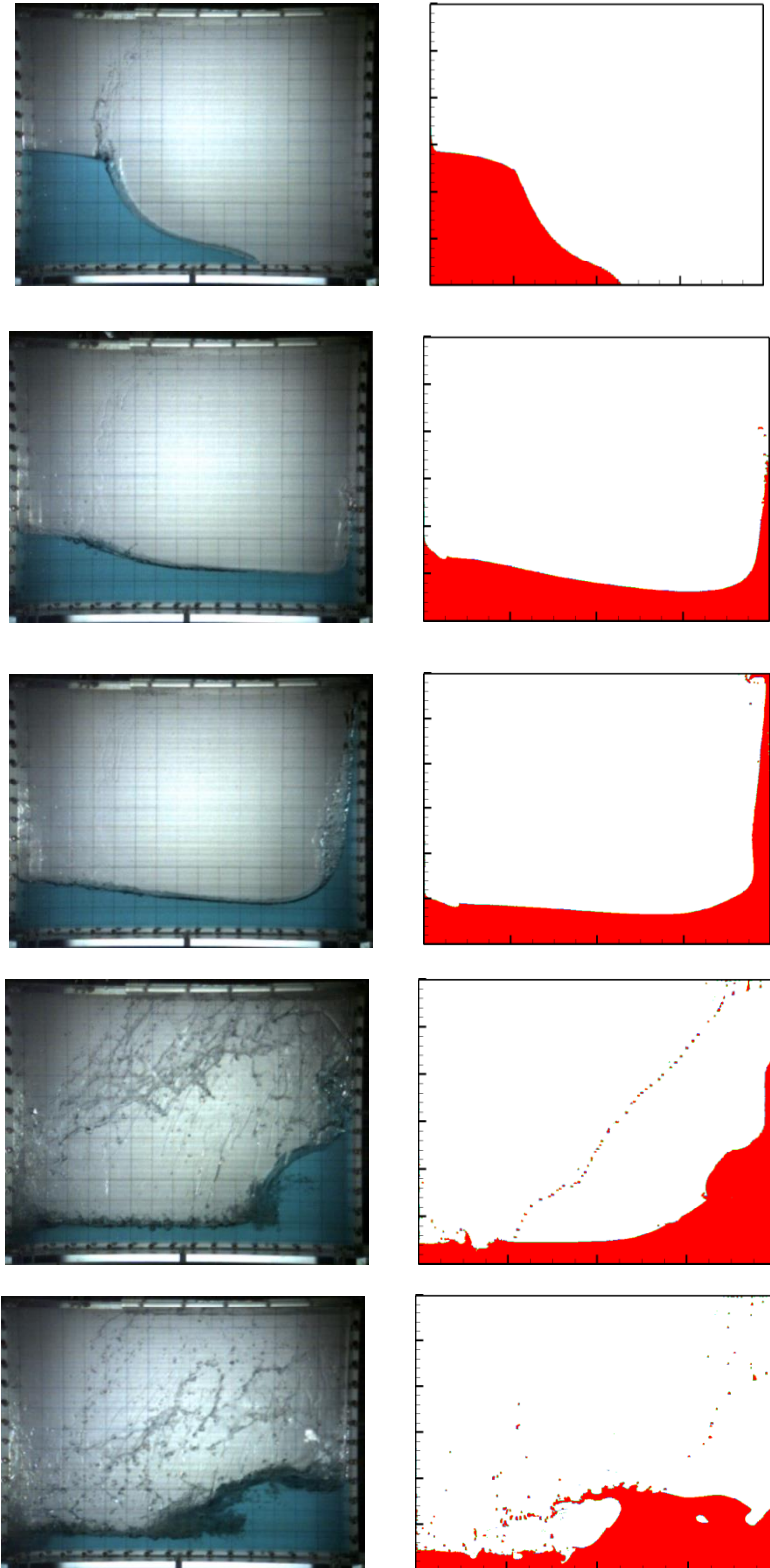


Fig. 5.6 Free surface evolution at $t=0, 0.18\text{s}, 0.39\text{s}, 0.52\text{s}, 0.85\text{s}, 0.99\text{s}$ (Exp. vs LBM)

5.2.2 Dam breaking with obstacle

The dam breaking experiment with an obstacle conducted by Andreas Kolke (2005) is

revisited here to make comparison with the numerical simulations. The experiment configuration is described in Fig.5.7, in which a water column positioned at the left side and an obstacle is placed at a distance 0.292 m from the left side. The parameters are listed in Table 5.2. Some snapshots of the experiment have been compared with the numerical results of the present solver. Firstly, a latency of free surface profile can be observed. Subsequently, the jet extends towards the top side of the water tank, and then it hits the top and falls down due to the gravity. In general, the numerical results agree well with the free surface profiles of the experiment qualitatively. This case indicates that the present solver is stable enough to handle the violent free surface flow issue in ocean engineering.

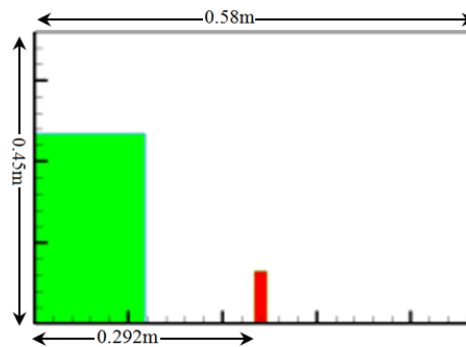


Fig. 5.7 Experiment configuration

Table 5.2 Parameters for the dam breaking with obstacle

Param.	Value
Domain	0.58m*0.45m
Water	0.146m*0.292m
Obstacle	0.015m*0.08m
Lattice	464*360
Δx	0.00125m

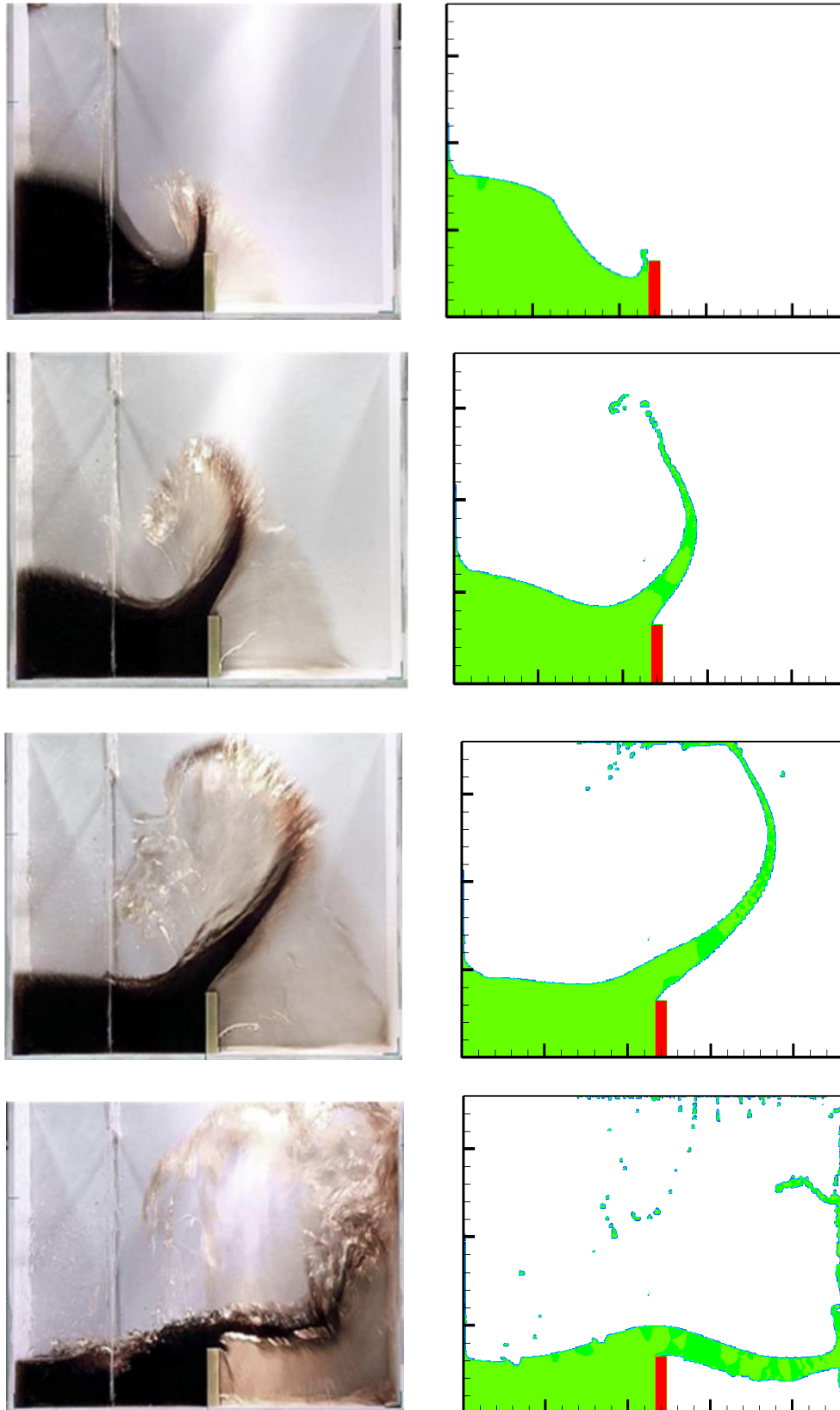


Fig. 5.8 Evolution of the free surface at specific time step

5.3 Water entry with constant speed

Water entry phenomena exist widely in the interactions of marine structure and surface waves, such as the motions of high-speed vessels, torpedo launch and so on. The impact force on the structure at the initial stage is considerable. The traditional mathematic model usually can only solve the linear problem while the conventional CFD tools such as finite difference method, finite volume method are very time-consuming. Therefore, an alternative CFD tool is deserved to be developed to tackle the water entry problem.

In this section, the experiments conducted by Miao (1989) in Marine Technology Center in Trondheim are used to verify our GPU accelerated FSLB solver. The tank is 28m long, 2.5m wide and 1.0m depth. The test cylinder has a diameter $D=0.125\text{m}$ and length $L=0.25\text{m}$. Theoretically, it can be regarded as the two-dimensional case. The experiment parameters can be found in the following table.

Table 5.3 Parameters for the water entry tests (Miao 1989)

Test No.	No.1	No.2	No.3	No.4
$V(\text{m/s})$	0.5124	0.639	0.76	1.024
Fn	0.4627	0.577	0.6863	0.9247

where $F_n = U / \sqrt{gD}$ is the Froude number based on the vertical downward constant speed of the cylinder and its diameter.

5.3.1 Convergence Test

Before comparing with the experiment results, it is of significance to test the convergence of the time step and mesh size for the present numerical solver.

In terms of the effect of time step, the No.1 case with resolution of 1000×560 (50 grid across the cylinder diameter) is selected. As is observed in Fig.5.9, it is completely incorrect for the largest time step. The force drop occurs when the water is collapsed onto the upper surface of the cylinder at the time to be totally submerged, especially with rough time step. However, the high frequency oscillation of the slamming force becomes more remarkable with smaller time step. Since the mesh size and the time step should comply with the diffusive scale in the present physical problem, that is $\Delta x \sim \Delta t^2$. With the present resolution, $\Delta x = 0.02$, the time step should be set as $\Delta t \sim O(0.0001)$. However, the diffusive scale is not respected in the case of the most rough time step, $\Delta t \sim O(0.001)$, which can explain why the numerical result deviates

Chapter 5 Free Surface Model and Its Applications

the experiment data considerably. In the following cases, the time step $\Delta t = T / 4000$ is chosen.

Three different mesh sizes ($D = 25\Delta x, D = 50\Delta x, D = 100\Delta x$) are set to test the effect of the resolution. As can be found in Fig.5.10, the mesh size seems no remarkable effect on the slamming coefficient. In the hereafter context, the resolution with $D = 50\Delta x$ is performed.

The pressure wave generated by the impact of the cylinder on the free surface should to be addressed. From Fig.5.11, at initial impact stage, the ring pressure wave can be observed and then propagates in the flow field and hits the wall of the fluid domain. In the present FSLB solver, the bounce-back boundary condition is imposed at all the sides of the fluid domain. The pressure wave will also be bounced back and reflected into the fluid domain. As can be observed obviously, the pressure noise is always remarkable near the cylinder and the free surface. In the present solver, the gas phase is neglected and the atmosphere is imposed on the free surface as the pressure boundary condition. The pressure evolution of the present solver is obviously influenced by this single-phase treatment actually. The MRT collision model, the advanced re-initialization scheme of the fresh node emerged from the solid phase, the stress integration method for force evaluation are applied in the present solver. With these treatments, the unphysical pressure noise can be eliminated to some extent.

a. Effect of time step

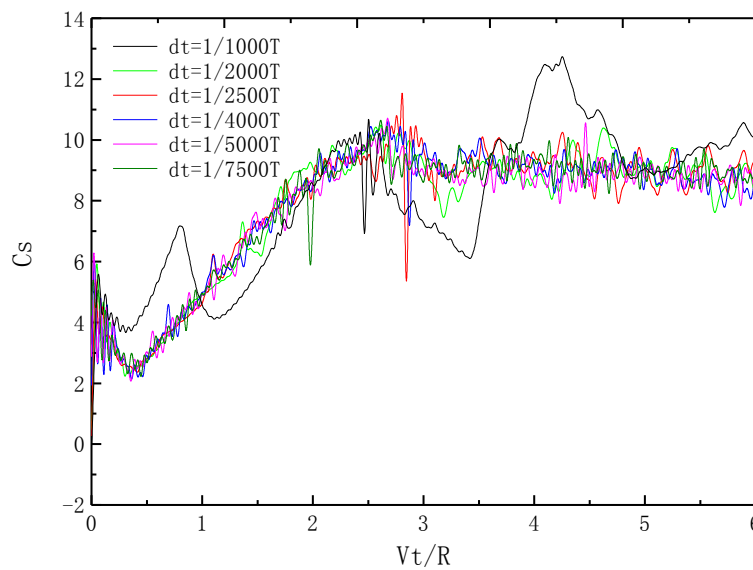


Fig. 5.9 Convergence of time step (after low pass filter)

b. Effect of mesh size

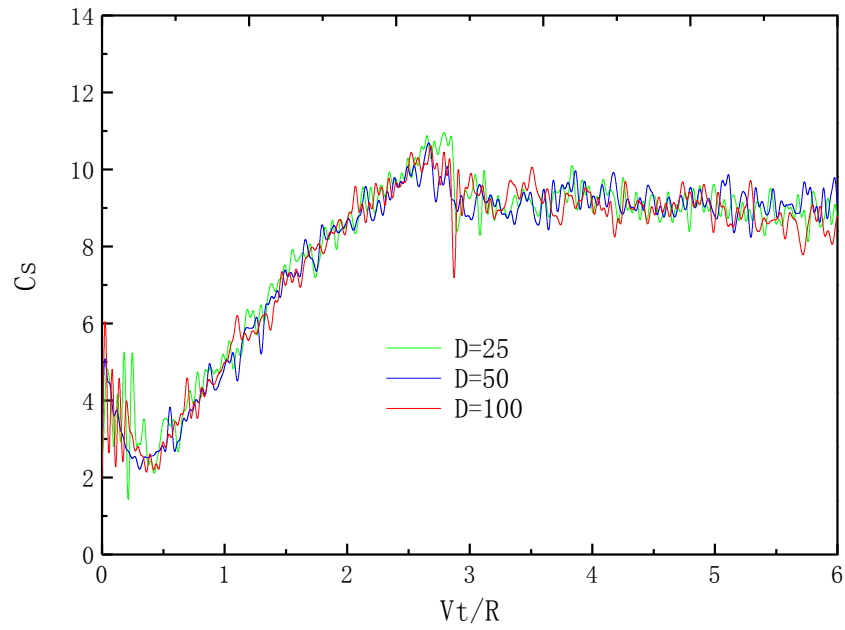
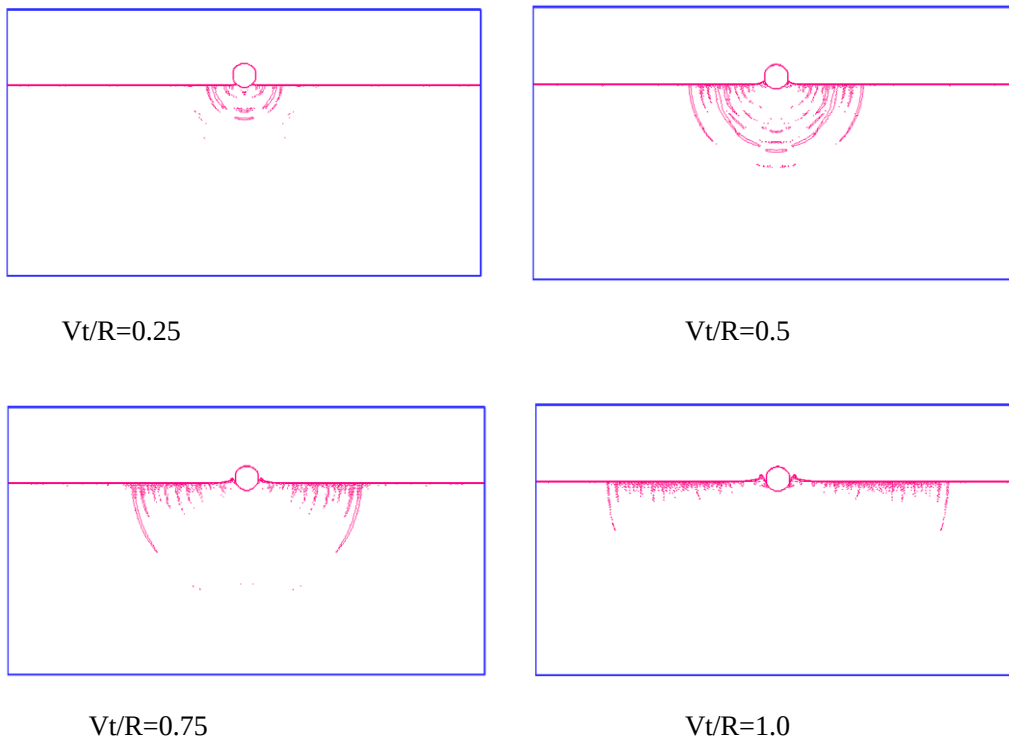


Fig. 5.10 Convergence of mesh size (after low pass filter)

c. Pressure wave



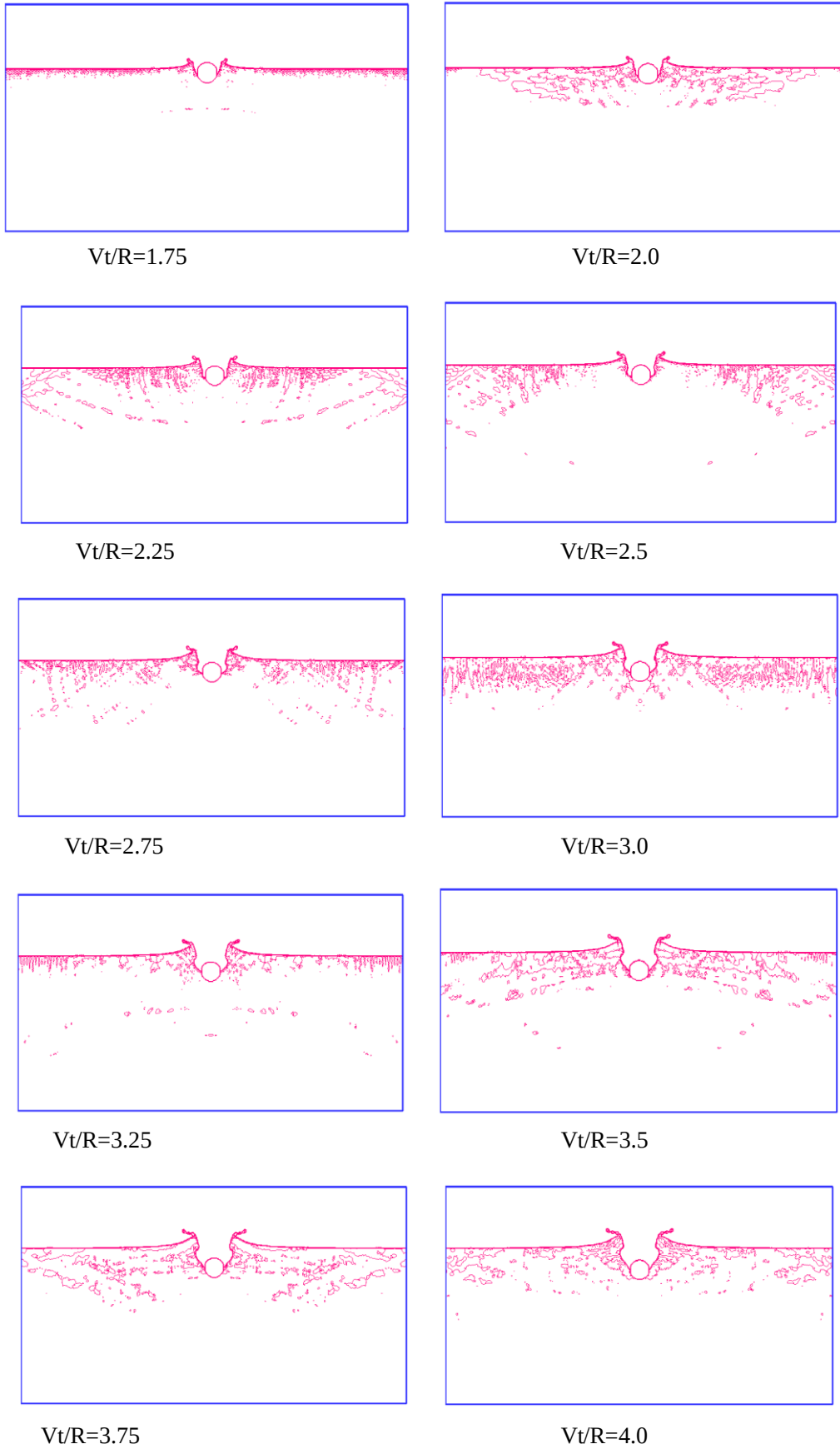


Fig. 5.11 Pressure contour of the flow field at specific time steps

5.3.2 Numerical simulation vs the experiment result

In this section, the numerical results of slamming coefficient for four cases are compared with the experiment data obtained by Miao (1989). From Fig.5.12-15, it can be seen that the slamming coefficients calculated by the present FSLB solver agree with the experiment data very well. Since the experiment data is detached from Zhu (2006), the initial stage experiment data is hard to get for the high frequency oscillation. It also can be found that the trends of the slamming coefficients are completely correct while the evaluation of the slamming force at the very initial stage when the cylinder hits on the free surface seems not ideal. In fact, the results seem to be reasonable for the present solver since the present solver is intrinsic a single-phase solver. When the cylinder hit the free surface, the velocity of the fluid node contacting with the cylinder is zero while the cylinder has a certain velocity, which is not true for the real physical phenomenon. According to the Wagner (1932) theory, the time varying rate of the wet surface in the outflow domain is proportional to $1/\sqrt{t}$, which indicates that it is singular at the time when cylinder hits the free surface. This initial stage is extremely short and much shorter than the real time step. It is impossible for the present single-phase model to evaluate the slamming force at initial stage.

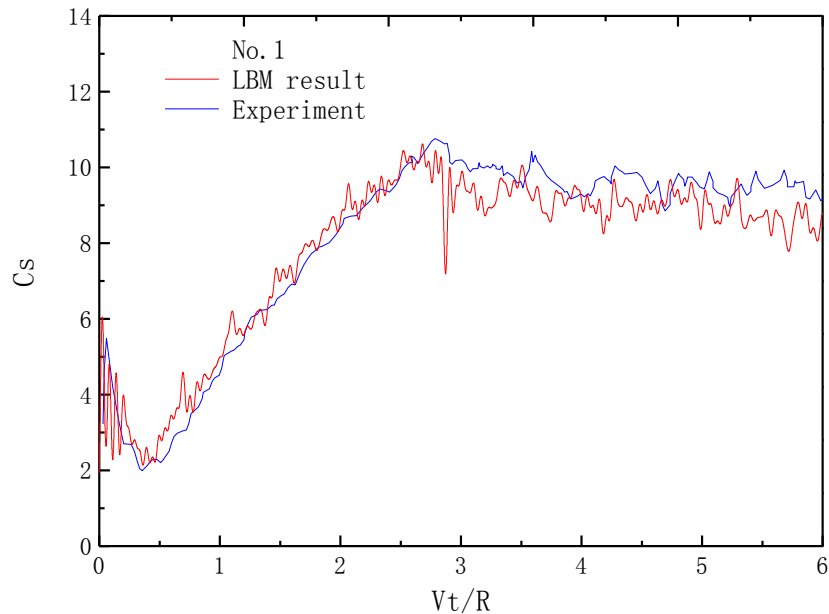


Fig.5.12 Slamming coefficient C_s for the water entry test No.1

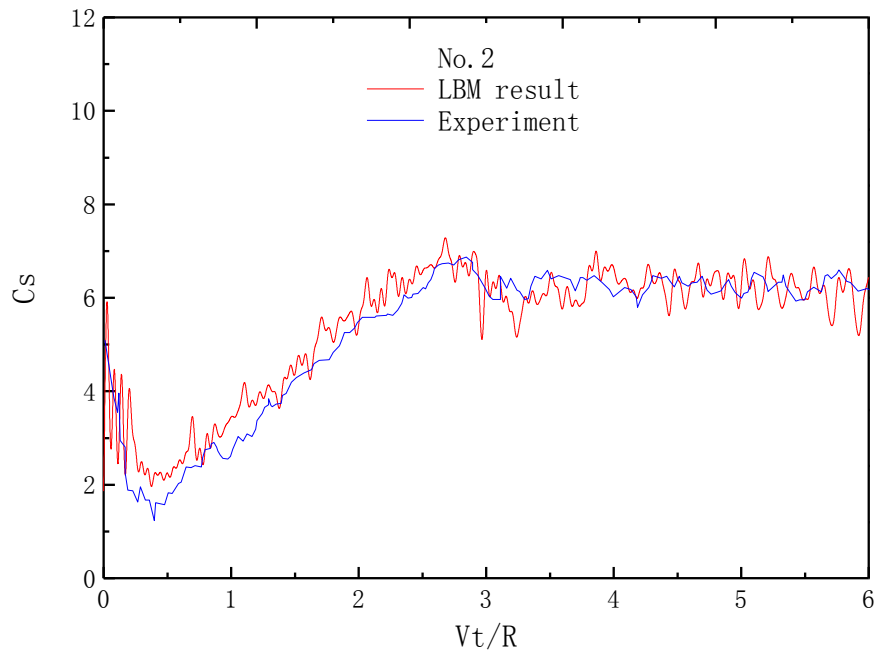


Fig. 5.13 Slamming coefficient C_s for the water entry test No.2

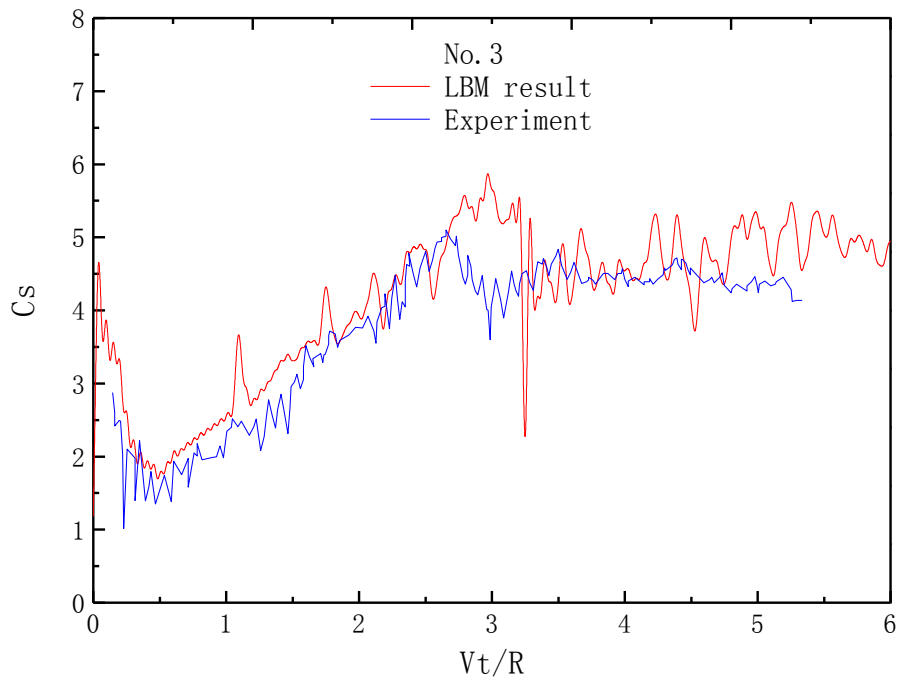


Fig. 5.14 Slamming coefficient C_s for the water entry test No.3

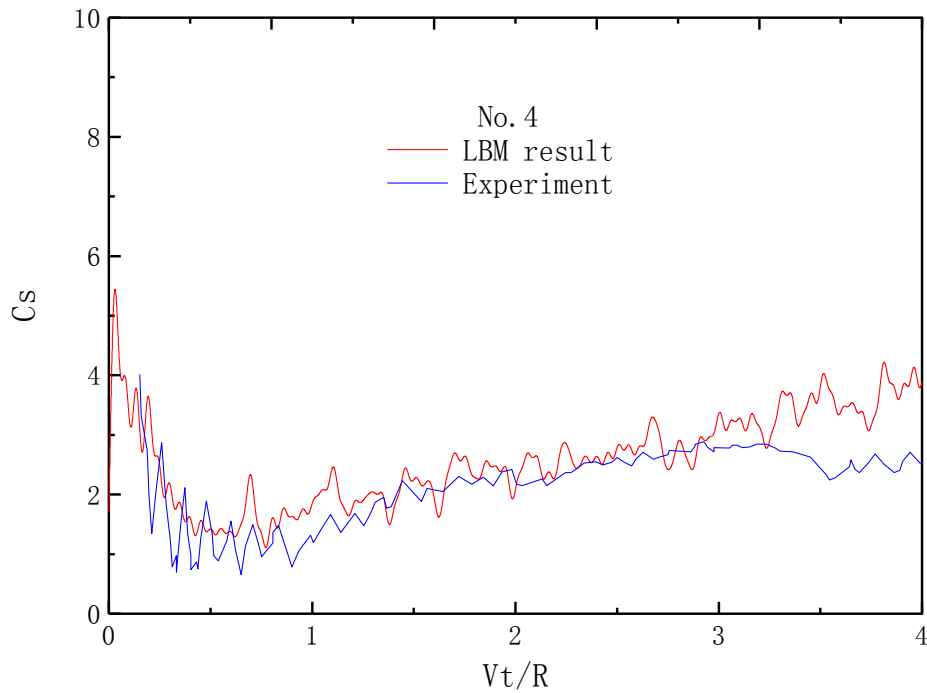


Fig. 5.15 Slamming coefficient C_s for the water entry test No.4

5.4 Water exit with constant speed

In this section, the water exit phenomenon is simulated by the present FSLB solver. The experiment results conducted by Miao (1989) are used to verify the present solver. The parameters of the experiment are listed in Table 5.4.

Table 5.4 Parameters for water exit, Miao 1989

Test no.	No. 1	No. 2
V (m/s)	0.5124	0.7644
Fn	0.4627	0.6903

Initially, the cylinder is positioned below the free surface with 5.5 radius of the cylinder. The cylinder is accelerated to a constant velocity at 5 radius distance below the free surface with a harmonic function $\sin(\omega t)$. Then the cylinder rises with constant velocity. From Fig.5.16 and Fig.5.17, it can be found that the numerical results agree well with the experiment data in general trend. In terms of the force oscillation of the experiment at the initial stage of the exit process, Zhu (2006) thought that the initial oscillation of the exit drag coefficient of the experiment data is owed to the unstable behavior of the hydraulic system. In general, the present numerical results can agree well with the experiment data.

Chapter 5 Free Surface Model and Its Applications

The free surface profiles at specific time steps can be found in Fig.5.18 with $Fn=1.0$. The numerical results of CIP-FDM by Zhu (2006) are included to compare with the present results. It can be observed that the free surface evolution of two different numerical tools agree with each other well, which validates the accuracy of the present solver.

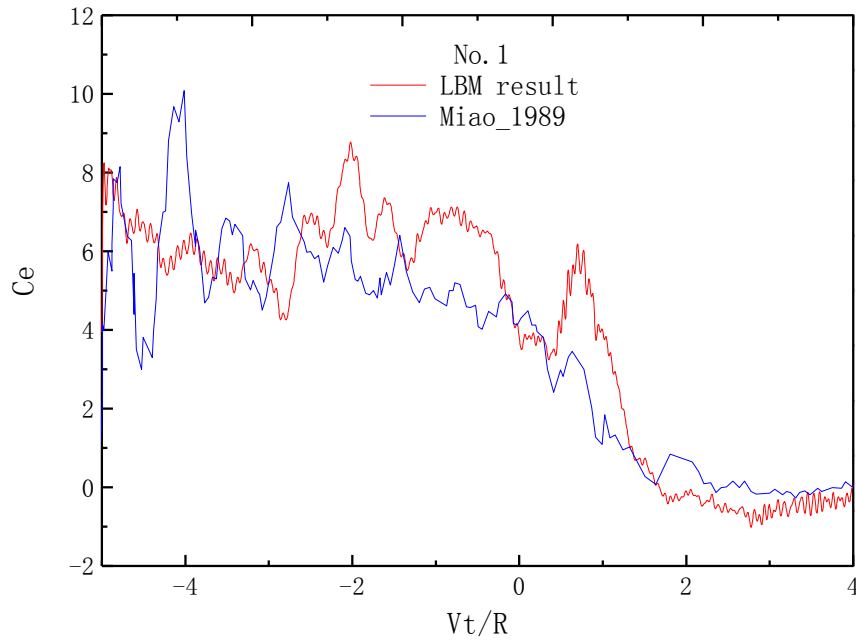


Fig. 5.16 Exit coefficient C_e with the non-dimensional time (No.1)

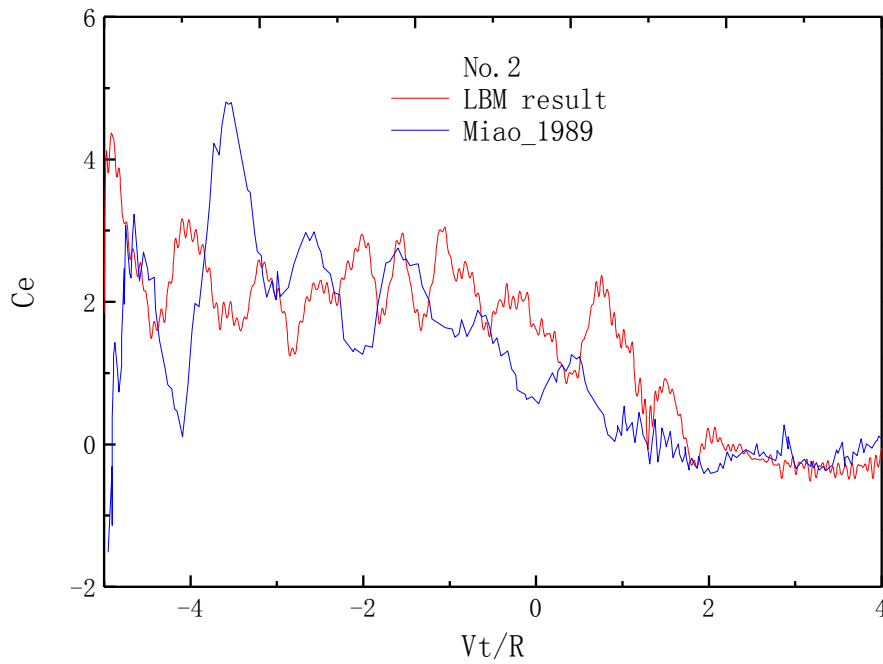


Fig. 5.17 Exit coefficient C_e with the non-dimensional time (No.2)

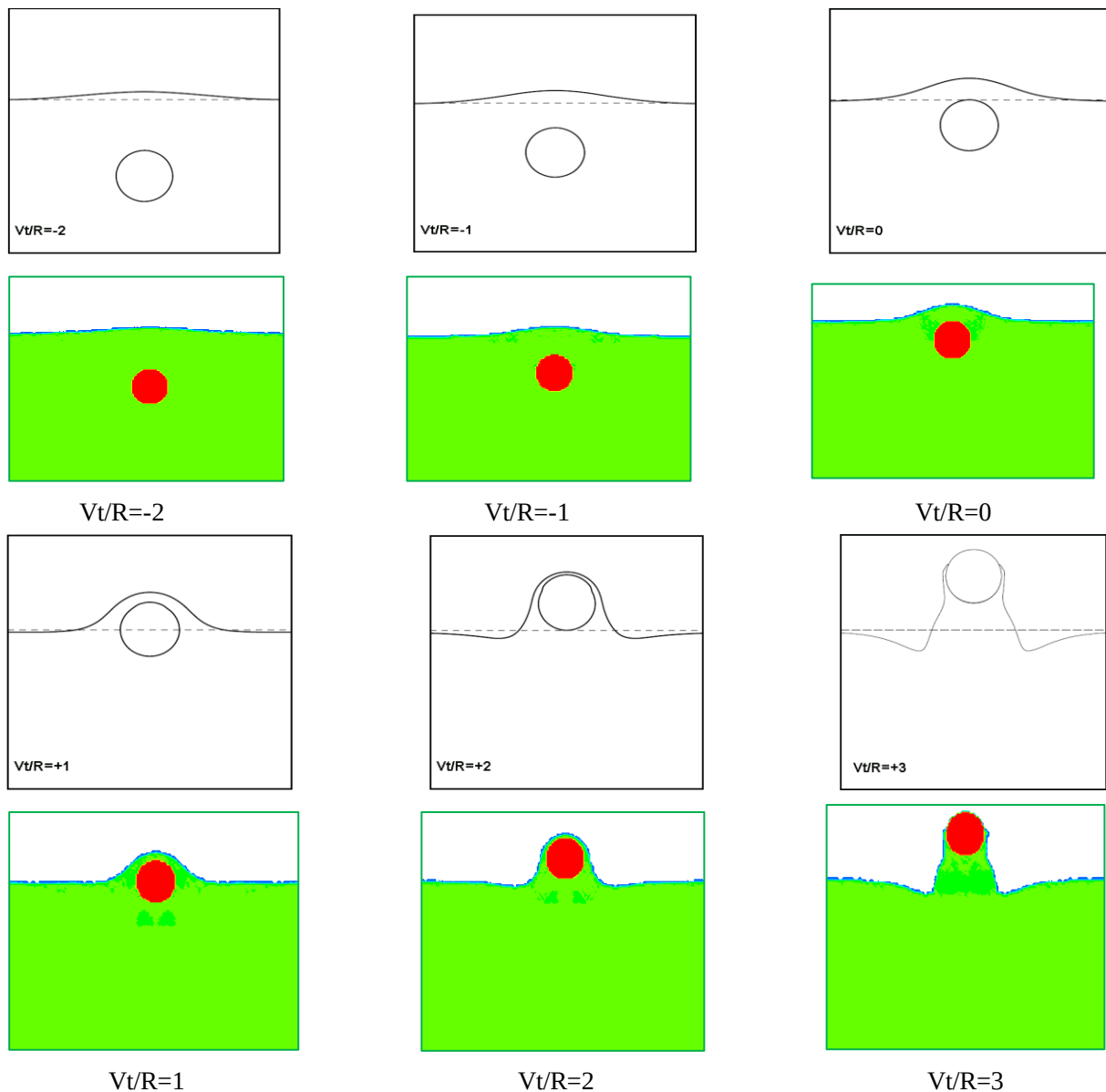


Fig. 5.18 Free surface evolution for water exit with $Fn=1.0$, CIP-FDM vs LBM

5.5 Water entry with free fall

In this section, the free fall cylinder water entry will be simulated by the present solver. The experiments conducted by Greenhow & WM Lin are simulated here. A cylinder with diameter 0.11m falls free from an initial height 0.5 m into a water tank with 0.3 m water depth. Two cases with different cylinder-water density ratio are conducted. One case is neutrally buoyant and the other is half-buoyant.

5.5.1 Neutrally buoyant free fall cylinder

Chapter 5 Free Surface Model and Its Applications

In Fig.5.20, the slamming force can be found for the neutrally buoyant free fall cylinder. It has a peak for the impact force when the cylinder slaps on the free surface. Subsequently, the slamming force decreases with time but fluctuates with time. However, the fluctuation for the slamming force in the later period is not found in the result of Zhu (2006). It can be explained that the pressure wave is difficult to be decayed with little dissipation. Another issue should be addressed is that the non-slip boundary condition is realized by the bounce-back scheme which reflects the particle into the fluid domain. The sponge boundary condition will be applied in our future work. In Fig.5.21, it can be found that the penetration depth into the free surface of the present solver agrees well with both the experiment result and FDM. It can be interpreted that the motion of the cylinder actually is the integration of the external force such that the fluctuation might be canceled.

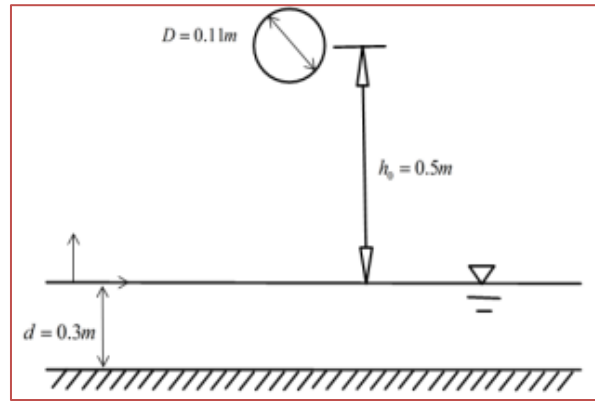


Fig. 5.19 Schematic of the experiment configuration

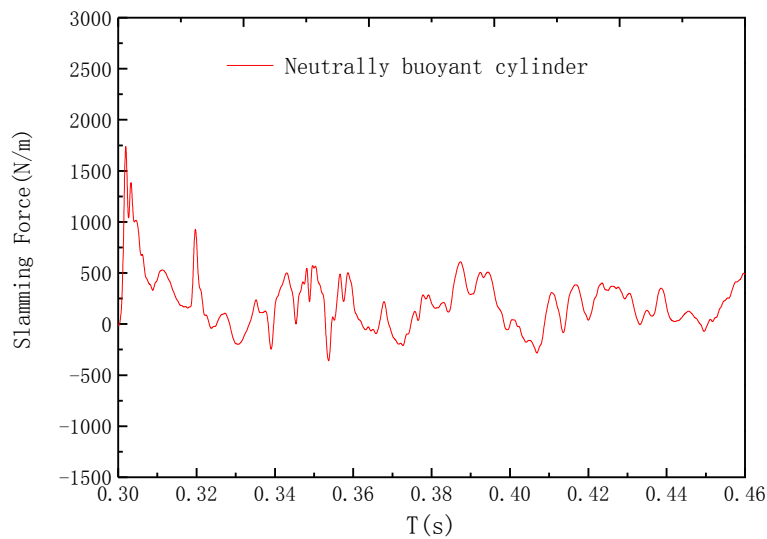


Fig. 5.20 Slamming force for neutrally buoyant cylinder

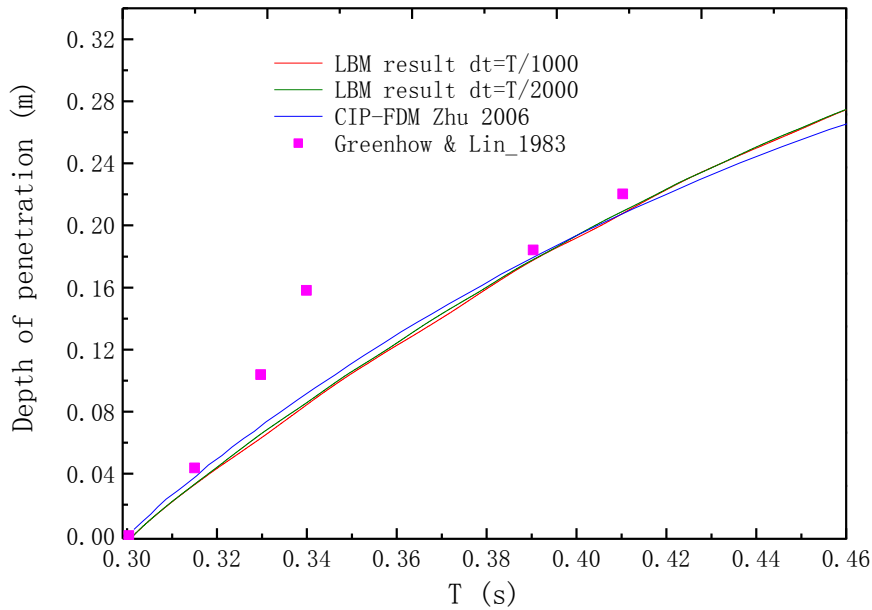


Fig. 5.21 Penetration depth of the neutrally buoyant cylinder

5.5.2 Half-buoyant free fall cylinder

For the half-buoyant free fall cylinder, similar comparisons are performed. Fig.5.22 describes the slamming coefficient of the cylinder impacting on the free surface. The similar fluctuation can be found in the later stage of the water entry process. In Fig.5.23, the penetration depth of the cylinder is compared with the experiment data and CIP-FDM by Zhu (2006). In the initial stage, the numerical result of the present solver coincides with the experiment data and reference result well, while the cylinder seems to move a little faster than the real case. Fig.5.24 describes the free surface profiles of the water entry for the present FSLB solver, CIP-based FDM solver and the experiment. When the cylinder reaches the free surface, the free surface sprays to both sides of the cylinder. It can be found that both the numerical results simulated by the FSLB solver and the CIP-based FDM solver agree well with those of the experiment.

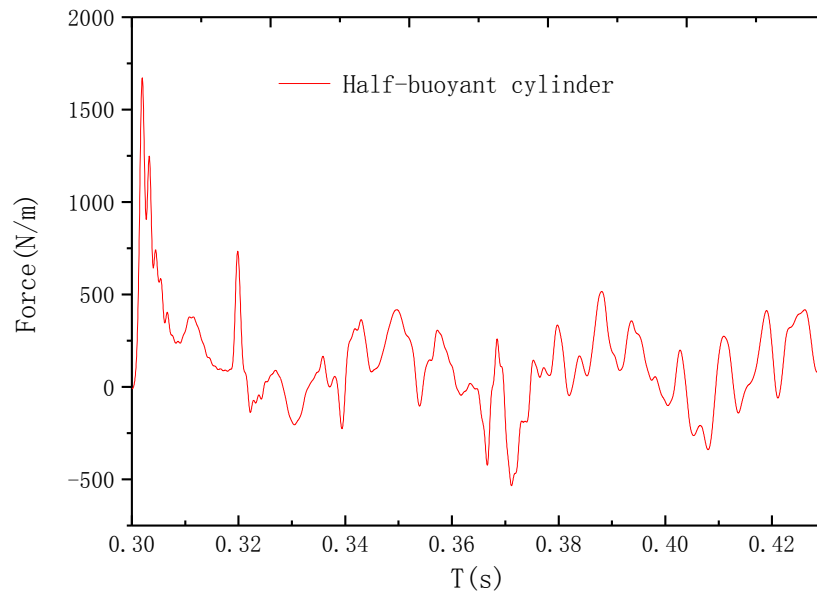


Fig. 5.22 Slamming force for half-buoyant cylinder

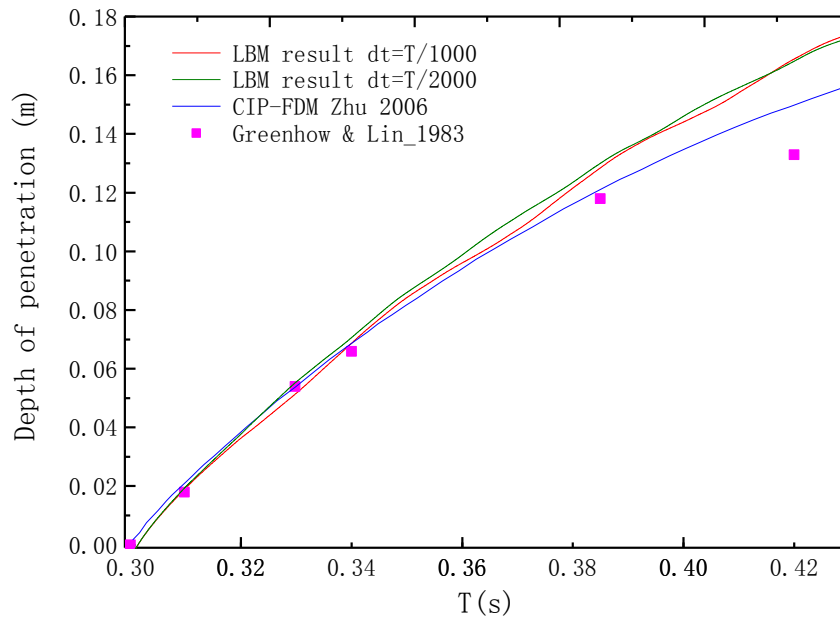
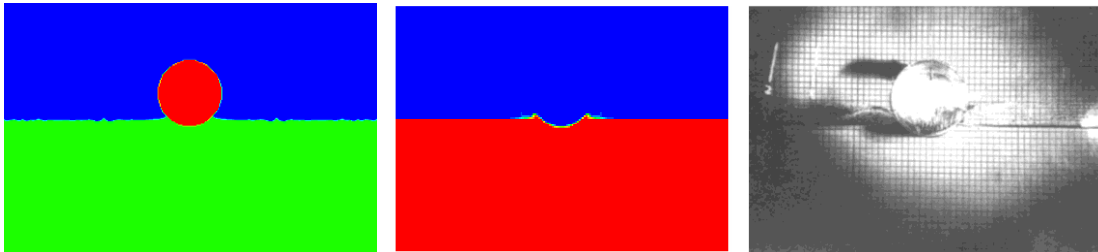


Fig. 5.23 Penetration depth of the half-buoyant cylinder



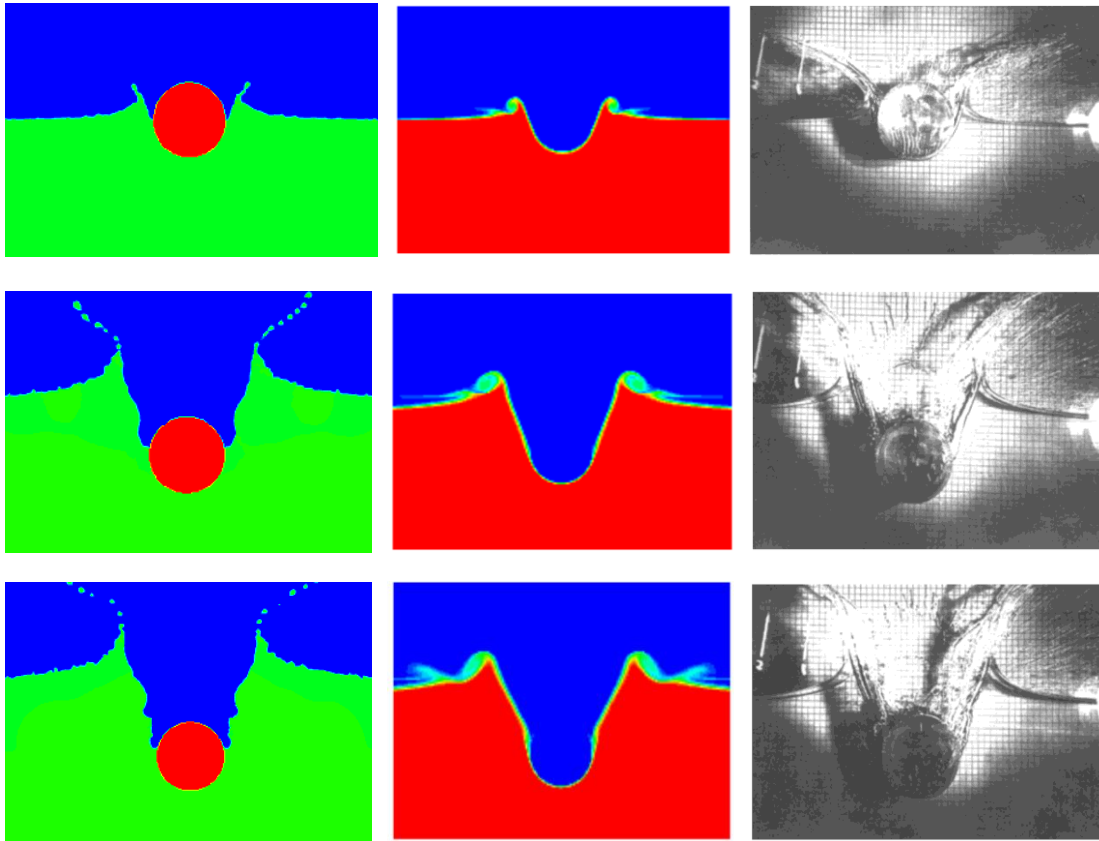


Fig. 5.24 Free surface deformation a) LBM, b) CIP-FDM c)Exp. at $t=0.305s, 0.330s, 0.385s, 0.420s$

5.6 Concluding Remarks

In the present work, the GPU accelerated FSLB solver has been developed and enhanced to deal with the violent nonlinear free surface flow. The numerical simulation of the water entry and exit scenario can be completed within one minute. Compared with the execution time of several hours for the FDM solver, it is demonstrated to be an extremely efficient solver. The nonlinear free surface flows of dam breaking, dam breaking with obstacle, water entry with constant speed, water exit with constant speed, free fall water entry are successfully simulated by the present FSLB solver. Meanwhile, the present numerical results are compared with the experiment data and the CIP-FDM solver. Based on the comparison, it can be concluded that the present solver can be capable of the free surface flow and the wave impact phenomenon in ocean engineering field. However, the impact force at the initial stage of the water entry problem cannot be captured accurately for the single-phase approximation and the time step limitation of numerical simulation. Suitable potential flow model should be incorporated into the present solver in the future work. Furthermore, the pressure noise is

Chapter 5 Free Surface Model and Its Applications

remarkable in the present solver. It can be explained that the pressure wave are always reflected into the fluid domain with the bounce-back boundary condition and the pressure boundary condition at the free surface. In the future work, the sponge boundary condition and improved pressure boundary condition should be applied and developed.

Chapter 6 Multiphase Lattice Boltzmann Model and Its Applications

6.1 Theory of multiphase models

In the present work, the multiphase models based on the discrete Boltzmann equation have been studied systematically and then applied to solve the engineering problems. In this chapter, the underlying relationships of four multiphase models based on the discrete Boltzmann equation are elaborated. The state-of-art model developed by F. Abbas et al. (2016) is adopted in the present applications.

6.1.1 He-Cheng-Zhang two-phase model (1999)

The Boltzmann equation for non-ideal fluids with the intermolecular interaction is as followings:

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \xi \cdot \nabla f = -\frac{1}{\lambda} (f - f^{eq}) + \frac{(\xi - \mathbf{u}) \cdot (\mathbf{F} + \mathbf{G})}{\rho c_s^2} f^{eq} \quad (6-1)$$

f is the particle distribution function in the phase space, ξ is the microscopic velocity, $\mathbf{F}$ is the effective molecular interaction force, $\mathbf{G}$ is the gravity, λ is the relaxation time, and the equilibrium distribution function satisfies the Maxwell distribution:

$$f^{eq} = \frac{\rho}{(2\pi RT)^{D/2}} \exp\left[-\frac{(\xi - \mathbf{u})^2}{2RT}\right] \quad (6-2)$$

D is the dimension of the space.

The intermolecular force can be expressed as

$$\mathbf{F} = -\nabla \psi + \mathbf{F}_s \quad (6-3)$$

$\mathbf{F}_s = \kappa \rho \nabla \nabla^2 \rho$ is the surface tension and ψ is a function of density

$$\psi(\rho) = P - \rho RT \quad (6-4)$$

P is the thermodynamic pressure.

Chapter 6 Multiphase Lattice Boltzmann Model and Its Applications

For the van der Waals equation of state

$$P = \frac{\rho RT}{1 - b\rho} - a\rho^2 \quad (6-5)$$

For the Carnahan-Starling equation

$$P = \rho RT \frac{1 + b\rho/4 + (b\rho/4)^2 - (b\rho/4)^3}{(1 - b\rho/4)^3} - a\rho^2 \quad (6-6)$$

where a and b are parameters related to the intermolecular attraction and exclusion-volume effect. The term $\nabla\psi$ in the intermolecular force term is steep cross the interface of binary fluids. Any numerical errors in evaluating this term will trigger the numerical instability. A numerical trick has been applied to reduce the numerical errors. A new distribution function has been introduced

$$g = fc_s^2 + \psi(\rho)\Gamma(0) \quad (6-7)$$

where $\Gamma(\mathbf{u})$ is the equilibrium function divided by the density as

$$\Gamma(\mathbf{u}) = \frac{1}{(2\pi RT)^{D/2}} \exp\left[-\frac{(\boldsymbol{\xi} - \mathbf{u})^2}{2RT}\right] \quad (6-8)$$

Then the reconstructed equilibrium function can be expressed as

$$g^{eq} = c_s^2 f + \Gamma(0)\psi(\rho) \quad (6-9)$$

Some derivations of the new distribution function are as followings

$$\frac{Dg}{Dt} = c_s^2 \frac{Df}{Dt} + \Gamma(0) \frac{D\psi(\rho)}{Dt} \quad (6-10)$$

The second term in the above equation can be derived as

$$\begin{aligned} \frac{D\psi(\rho)}{Dt} &= \frac{d\psi(\rho)}{d\rho} \frac{D\rho}{Dt} \\ &= \frac{d\psi(\rho)}{d\rho} \left[\frac{\partial\rho}{\partial t} + (\boldsymbol{\xi} \cdot \nabla)\rho \right] \end{aligned}$$

$$\begin{aligned}
 &= \frac{d\psi(\rho)}{d\rho} (\xi - \mathbf{u}) \nabla \rho \\
 &= (\xi - \mathbf{u}) \nabla \psi(\rho)
 \end{aligned} \tag{6-11}$$

where $\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + (\mathbf{u} \cdot \nabla) \rho = 0$ is used in above equation.

From Eq.(6-11), it can be obtained

$$\frac{D\psi(\rho)}{Dt} = (\xi - \mathbf{u}) \nabla \psi(\rho) \tag{6-12}$$

From Eq.(6-1), Eq.(6-3), Eq.(6-12), it has

$$\frac{Dg}{Dt} = -\frac{g - g^{eq}}{\lambda} + (\xi - \mathbf{u}) [\Gamma(\mathbf{u})(\mathbf{F}_s + \mathbf{G}) - (\Gamma(\mathbf{u}) - \Gamma(0)) \nabla \psi(\rho)] \tag{6-13}$$

Comments should be remarked here. Before introducing the new distribution function, the gradient term of $\psi(\rho)$ only multiplied by $(\xi - \mathbf{u})\Gamma(\mathbf{u})$. However, with the numerical trick, this instable term is multiplied by $(\xi - \mathbf{u})(\Gamma(\mathbf{u}) - \Gamma(0))$, which is at least $O(\mathbf{u}^3)$. For the second-order spatial accurate LBE solver, it is reasonable and acceptable. Finally, this model becomes stable for the two-phase simulation.

For the phase-field interface tracking, the simplified form of Eq. (6-1) without the surface tension and gravity term is applied as

$$\frac{Df}{Dt} = -\frac{1}{\lambda} (f - f^{eq}) - \frac{(\xi - \mathbf{u}) \cdot \nabla \psi(\rho)}{c_s^2} \Gamma(\mathbf{u}) \tag{6-14}$$

where the equilibrium distribution function can be expressed as

$$f^{eq} = \phi \Gamma(\mathbf{u}) \tag{6-15}$$

The discrete forms of Eq.(6-13) and (6-14) can be established through discretizing them in velocity space and temporal space.

For two-dimensional case, the popular discrete velocity space can be expressed as

$$\mathbf{e}_\alpha = \begin{cases} (0, 0) & \alpha = 0 \\ (\cos[(\alpha-1)\pi/2], \sin[(\alpha-1)\pi/2])c & \alpha = 1, 2, 3, 4 \\ \sqrt{2}(\cos[(\alpha-5)\pi/2 + \pi/4], \sin[(\alpha-5)\pi/2 + \pi/4])c & \alpha = 5, 6, 7, 8 \end{cases} \quad (6-16)$$

The equilibrium function for f_α, g_α are

$$f_\alpha^{eq} = w_\alpha \phi [1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2}] \quad (6-17)$$

$$g_\alpha^{eq} = w_\alpha [p + \rho (\frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2})] \quad (6-18)$$

The temporal discretization is conducted by integrating Eq.(6-13) and Eq.(6-14) along characteristics over time step δt . To maintain the explicit time-marching scheme of the evolution, the reconstructed distribution functions for the phase-field and flow-field are as followings

$$\bar{f}_\alpha = f_\alpha + \frac{(\mathbf{e}_\alpha - \mathbf{u}) \cdot \nabla \psi(\rho)}{2c_s^2} \Gamma_\alpha(\mathbf{u}) \delta t \quad (6-19)$$

$$\bar{g}_\alpha = g_\alpha - \frac{1}{2} (\mathbf{e}_\alpha - \mathbf{u}) \cdot [\Gamma_\alpha(\mathbf{u})(\mathbf{F}_s + \mathbf{G}) - (\Gamma_\alpha(\mathbf{u}) - \Gamma_\alpha(0)) \nabla \psi(\rho)] \delta t \quad (6-20)$$

Finally, the discrete evolution equations are as followings

$$\begin{aligned} \bar{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta t, t + \delta t) &= \bar{f}_\alpha(\mathbf{x}, t) - \Lambda(\bar{f}_\alpha(\mathbf{x}, t) - f_\alpha^{eq}) \\ &\quad - (1 - \frac{1}{2\tau}) \frac{(\mathbf{e}_\alpha - \mathbf{u}) \cdot \nabla \psi(\rho)}{c_s^2} \Gamma_\alpha(\mathbf{u}) \delta t \end{aligned} \quad (6-21)$$

$$\begin{aligned} \bar{g}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \delta t, t + \delta t) &= \bar{g}_\alpha(\mathbf{x}, t) - \Lambda(\bar{g}_\alpha(\mathbf{x}, t) - g_\alpha^{eq}) \\ &\quad - (1 - \frac{1}{2\tau}) (\mathbf{e}_\alpha - \mathbf{u}) \cdot [\Gamma_\alpha(\mathbf{u})(\mathbf{F}_s + \mathbf{G}) - (\Gamma_\alpha(\mathbf{u}) - \Gamma_\alpha(0)) \nabla \psi(\rho)] \delta t \end{aligned} \quad (6-22)$$

where $\tau = \lambda / \delta t$.

The macroscopic variables are evaluated by the moments of distribution functions as

$$\varphi = \sum \bar{f}_\alpha \quad (6-23)$$

$$p = \sum \bar{g}_\alpha - \frac{1}{2} \mathbf{u} \cdot \nabla \psi(\rho) \delta t \quad (6-24)$$

$$\rho \mathbf{u} = \frac{1}{c_s^2} \sum e_\alpha \bar{g}_\alpha + \frac{1}{2} (\mathbf{F}_s + \mathbf{G}) \delta t \quad (6-25)$$

The numerical stability of the He-Cheng-Zhang model is beneficial from the reduced discretization numerical error of the intermolecular force (reduced from second-order to third-order error). However, the density ratio of this model is constrained to be finite value that is much lower than that of water-air density ratio.

6.1.2 Lee-Lin two phase model (2005)

In order to overcome the numerical instability in HCZ model for high density ratio issue, Lee & Lin proposed a stable discretization scheme to deal with the derivatives in the intermolecular force term and applied the so-called surface tension of stress form and potential form in the flow-field discrete Boltzmann equation and phase-field discrete Boltzmann equation, respectively. The pressure updating scheme is claimed to be developed based on the non-ideal gas.

The original intermolecular force in this model is

$$F_i = \frac{\partial}{\partial x_j} (\rho c_s^2 - P) \delta_{ij} + \kappa \frac{\partial}{\partial x_j} \left[\left(\frac{1}{2} \frac{\partial \rho}{\partial x_k} \frac{\partial \rho}{\partial x_k} + \rho \frac{\partial^2 \rho}{\partial x_k \partial x_k} \right) \delta_{ij} - \frac{\partial \rho}{\partial x_i} \frac{\partial \rho}{\partial x_j} \right] \quad (6-26)$$

Then the above equation can be recast into stress form using a vector identity

$$F_i^S = \frac{\partial}{\partial x_j} (\rho c_s^2 - p) \delta_{ij} + \kappa \frac{\partial}{\partial x_j} \left(\frac{\partial \rho}{\partial x_k} \frac{\partial \rho}{\partial x_k} \delta_{ij} - \frac{\partial \rho}{\partial x_i} \frac{\partial \rho}{\partial x_j} \right) \quad (6-27)$$

in which the so-called hydrodynamic pressure p that contains the thermodynamic pressure and the pressure due to the inclusion of curvature

$$p = P - \kappa \rho \frac{\partial^2 \rho}{\partial x_k \partial x_k} + \frac{\kappa}{2} \frac{\partial \rho}{\partial x_k} \frac{\partial \rho}{\partial x_k} \quad (6-28)$$

As for the potential form, if conduct some algebraic manipulation of the second term of

right hand of Eq.(6-26) as

$$\begin{aligned}
 & \kappa \frac{\partial}{\partial x_j} \left[\left(\frac{1}{2} \frac{\partial \rho}{\partial x_k} \frac{\partial \rho}{\partial x_k} + \rho \frac{\partial^2 \rho}{\partial x_k \partial x_k} \right) \delta_{ij} - \frac{\partial \rho}{\partial x_i} \frac{\partial \rho}{\partial x_j} \right] \\
 &= \kappa \frac{\partial}{\partial x_i} \left(\frac{1}{2} \frac{\partial \rho}{\partial x_k} \frac{\partial \rho}{\partial x_k} + \rho \frac{\partial^2 \rho}{\partial x_k \partial x_k} \right) - \kappa \frac{\partial^2 \rho}{\partial x_i \partial x_j} \frac{\partial \rho}{\partial x_j} - \kappa \frac{\partial \rho}{\partial x_i} \frac{\partial^2 \rho}{\partial x_j \partial x_j} \\
 &= \kappa \frac{\partial \rho}{\partial x_k} \frac{\partial^2 \rho}{\partial x_i \partial x_k} + \kappa \frac{\partial \rho}{\partial x_i} \frac{\partial^2 \rho}{\partial x_k \partial x_k} + \kappa \rho \frac{\partial}{\partial x_i} \left(\frac{\partial^2 \rho}{\partial x_k \partial x_k} \right) - \kappa \frac{\partial^2 \rho}{\partial x_i \partial x_j} \frac{\partial \rho}{\partial x_j} - \kappa \frac{\partial \rho}{\partial x_i} \frac{\partial^2 \rho}{\partial x_j \partial x_j} \\
 &= \kappa \rho \frac{\partial}{\partial x_i} \left(\frac{\partial^2 \rho}{\partial x_k \partial x_k} \right)
 \end{aligned} \tag{6-29}$$

Thus the intermolecular force can be written as

$$F_i = \frac{\partial}{\partial x_j} (\rho c_s^2 - P) \delta_{ij} + \kappa \rho \frac{\partial}{\partial x_i} \left(\frac{\partial^2 \rho}{\partial x_k \partial x_k} \right) \tag{6-30}$$

The above form is equal to Eq.(6-3) in HCZ model. The following reshaped potential form is applied in the Lee-Lin model by the thermodynamic identity:

$$F_i^p = \frac{\partial \rho c_s^2}{\partial x_j} \delta_{ij} - \rho \frac{\partial}{\partial x_i} \left(\mu_0 - \kappa \frac{\partial^2 \rho}{\partial x_j \partial x_j} \right) \tag{6-31}$$

in which μ_0 is the bulk chemical potential that origins from the bulk energy of the system

and the term $\kappa \frac{\partial^2 \rho}{\partial x_j \partial x_j}$ owing to the surface tension. The detailed derivation can be found in

the subsequent section.

In Lee-Lin model, the stress form of intermolecular force Eq.(6-27) is applied in the flow-field discrete Boltzmann equation to guarantee the momentum conservation while the potential form Eq.(6-31) is applied in the phase-field discrete Boltzmann equation for the convenient phase separation.

The second remarkable contribution of Lee-Lin model is the realization of the large density ratio and large viscosity ratio of binary fluids, which owes to the consistent discretization of the intermolecular force. As mentioned in Lee & Lin (2005), the second-order

central discrete scheme performs dissatisfactory oscillation at the leading edge of the wave while obtains good results at the end edge of the wave. Nevertheless, the second-order biased difference scheme suffers from small undershoot at the end edge of the wave but performs great at the leading edge of the wave. To realize the different discretization schemes of the molecular force at different stages, Lee-Lin model divided the temporal evolution of the discrete Boltzmann equations into three steps: pre-stream collision step, stream step and the post-stream collision step. In the first step, the mixed or biased discretization for the intermolecular is applied. The second-order central discretization is performed in the post-stream step. Meanwhile, the isotropic discretization or weighted directional derivatives are utilized to calculate the gradient operator and Laplacian operator.

Based on the above mentioned two main contributions, Lee-Lin model has successfully simulated the high-density ratio and high-viscosity ratio binary fluids. However, the numerical dissipation is considerable due to the finite difference discretization of the first-order, second-order or mixed partial derivatives of the density, chemical potential. Meanwhile, the high-Reynolds number problem is beyond the capability of the original SRT Lee-Lin model (2005).

6.1.3 Lee-Liu two-phase model (2010)

In the subsequent work of Lee et al. (2010), the unified intermolecular force potential form is applied in the two-phase model. Similar to the HCZ model, the reconstructed lattice Boltzmann equations are applied to solve the phase-field and flow-field equations, while different discretization schemes for the force term are applied in the equilibrium distribution function and the collision step. Another remarkable difference from the previous Lee-Lin model is the discrete Boltzmann equation for the order parameter, which introduces complicated non-local Laplace derivatives across two time steps in dealing with the material derivatives of the term ϕ/ρ . This change involves not only fourth-order derivative of phase-field but also non-local operation in the Laplace operator of the neighboring node, which introduce extra memory access and boundary treatment.

The intermolecular force in this modified two-phase model can be regarded as potential form as mentioned in Lee-Lin model (2005). The total pressure consists of thermodynamic

Chapter 6 Multiphase Lattice Boltzmann Model and Its Applications

pressure, hydrodynamic pressure and the pressure due to the inclusion of curvature $-\kappa\phi\nabla^2\phi + \kappa/2|\nabla\phi|^2$. The total pressure can be written as

$$P_{total} = p + p_0 - \kappa\phi\nabla^2\phi + \frac{\kappa}{2}|\nabla\phi|^2 \quad (6-32)$$

Since the bulk chemical potential is the derivative of free energy with order parameter $\mu_0 = \partial E_f / \partial \phi$, and the thermodynamic pressure is related to the free energy E_f as

$$p_0 = \phi \frac{\partial E_f}{\partial \phi} - E_f = \phi\mu_0 - E_f \quad (6-33)$$

Then it can be obtained

$$\frac{\partial p_0}{\partial \phi} = \phi \frac{\partial \mu_0}{\partial \phi} \quad (6-34)$$

The spatial derivatives of thermodynamic pressure can be derived as

$$\frac{\partial p_0}{\partial x_i} = \phi \frac{\partial \mu_0}{\partial x_i} + \mu_0 \frac{\partial \phi}{\partial x_i} - \frac{\partial E_f}{\partial \phi} \frac{\partial \phi}{\partial x_i} = \phi \frac{\partial \mu_0}{\partial x_i} \quad (6-35)$$

That is

$$\nabla p_0 = \phi \nabla \mu_0 \quad (6-36)$$

Therefore the intermolecular force can be recast as in the form of Lee & Liu (2010)

$$\begin{aligned} \mathbf{F} &= \nabla(\rho c_s^2 - P_{total}) \\ &= \nabla \rho c_s^2 - \nabla p - \nabla p_0 - \kappa \nabla \phi \nabla^2 \phi - \kappa \phi \nabla \nabla^2 \phi + \kappa \nabla \phi \nabla^2 \phi \\ &= \nabla \rho c_s^2 - \nabla p - \phi \nabla \mu_0 - \kappa \phi \nabla \nabla^2 \phi \\ &= \nabla \rho c_s^2 - \nabla p - \phi \nabla (\mu_0 - \kappa \nabla^2 \phi) \\ &= \nabla \rho c_s^2 - \nabla p - \phi \nabla \mu_\phi \end{aligned} \quad (6-37)$$

where $\mu_\phi = \mu_0 - \kappa \nabla^2 \phi$ is the chemical potential and ∇p enforces the incompressibility.

As mentioned in some work (T.Lee 2006), this potential form can suppress the spurious current across the interface for the integration of the thermodynamic pressure and surface tension into one gradient operator to reduce the discretization error. This is the so-called trick to eliminate the force imbalance between thermodynamic pressure and surface tension as deeply investigated by Guo et al. (2011).

In this two-phase model, the particle distribution function for order parameter is set as $h_\alpha = f_\alpha C / \rho$, if taken the material derivatives and substitute the intermolecular force it will give

$$\begin{aligned} \frac{\partial h_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla h_\alpha &= \frac{C}{\rho} \left[-\frac{1}{\lambda} (f_\alpha - f_\alpha^{eq}) + \frac{(\mathbf{e}_\alpha - \mathbf{u}) \cdot \mathbf{F}}{c_s^2} \Gamma_\alpha \right] \\ &= -\frac{1}{\lambda} (h_\alpha - h_\alpha^{eq}) + \frac{C}{\rho c_s^2} (\mathbf{e}_\alpha - \mathbf{u}) \cdot [\nabla \rho c_s^2 - (\nabla p + C \nabla \mu_C)] \Gamma_\alpha + f_\alpha \frac{D}{Dt} \left(\frac{C}{\rho} \right) \end{aligned} \quad (6-38)$$

And for the last term it will give the following form if the phase-field governing equation Eq.(6-22) in Lee & Liu (2010) is used

$$\begin{aligned} f_\alpha \frac{D}{Dt} \left(\frac{C}{\rho} \right) &\approx \left(\frac{\partial C}{\partial t} + \mathbf{e}_\alpha \cdot \nabla C \right) \Gamma_\alpha - \frac{C}{\rho} \left(\frac{\partial \rho}{\partial t} + \mathbf{e}_\alpha \cdot \nabla \rho \right) \Gamma_\alpha \\ &= [(\mathbf{e}_\alpha - \mathbf{u}) \cdot \nabla C + M \nabla^2 \mu_C] \Gamma_\alpha - \frac{C}{\rho} (\mathbf{e}_\alpha - \mathbf{u}) \cdot \nabla \rho \Gamma_\alpha \end{aligned} \quad (6-39)$$

Then the discrete Boltzmann equation for order parameter is

$$\begin{aligned} \frac{\partial h_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla h_\alpha &= -\frac{1}{\lambda} (h_\alpha - h_\alpha^{eq}) \\ &\quad + M \nabla^2 \mu_C \Gamma_\alpha + (\mathbf{e}_\alpha - \mathbf{u}) \cdot \left[\nabla C - \frac{C}{\rho c_s^2} (\nabla p + C \nabla \mu_C) \right] \Gamma_\alpha \end{aligned} \quad (6-40)$$

From the above derivation, it can be found the material derivative of C / ρ introduces the Laplace operation of chemical potential, which is actually the fourth-order derivative of order parameter. The non-local properties will be more severe after integrated Eq.(6-36) to obtain the lattice Boltzmann equation for order parameter. Furthermore, the artificial free energy should

be incorporated to avoid the numerical collapse when the negative order parameter occurs. Meanwhile, there exist finite difference discretization of pressure, order parameter and chemical potential. The boundary treatment not only concerns the particle distribution functions but also for the macroscopic order parameter, pressure and chemical potential, which makes this model a little bit complicated and cumbersome.

6.1.4 Abbas-Martin-Lee two-phase model (2016)

In the state-of-art Abbas-Martin-Lee two phase model (2016), the interface is captured by solving a conservative phase-field LBE (Martin Geier et al. 2015) and the adaptive mesh refinement technique (F. Abbas et al. 2016) together with the MRT collision model have also been applied to solve the pressure equation and momentum equation. Recently, the Allen-Cahn equation (1976), which is not conservative but only concerns the second-order derivatives, has been improved into a phase-field model (Y. Sun and C. Beckermann 2007) and then reshaped into a conservative form (P.-H. Chiu and Y.-T. Lin 2011). Based on these work, Martin & Abbas & Lee (2016) applied the LBGK model to solve this partial differential equation. To solve the normal vector of the interface, the second-order isotropic central difference should be conducted to obtain the gradients of the order parameter, which indicates that only first-order derivatives are concerned in the collision term. The fourth-order derivatives in the phase model of Lee-Liu model (2010) have been avoided. The derivatives of the pressure are eliminated in the phase-field equation. The derivatives of the density can be also transformed into those of the phase field since the density can be written in the linear function of the order parameter. Furthermore, the surface tension $-\phi \nabla \mu_\phi$ can be replaced by the equivalent continuum surface tension form $\mu_\phi \nabla \phi$. Finally, this two-phase model only concerns the first-order derivatives of phase field, which not only simplifies the manipulation of finite difference operator but also saves considerable memory assumption. Meanwhile, the cumbersome treatment of the pressure, density and especially the chemical potential can be released when the distribute-memory programming such as MPI-CPU or MPI-GPU is applied to simulate the high-resolution numerical simulation. To simulate the physical phenomenon efficiently, the AMR technique is applied in this two-phase model. The high Reynolds number cases can be also resolved by incorporating the MRT collision model. However, currently this two-phase model is not capable of the high Reynolds number and high density ratio problems simultaneously, which

are the standard physical problem in the field of ocean engineering such as the interactions between vessels or platforms and waves.

In this chapter, we mainly applied the Abbas-Martin-Lee model to do engineering applications. The theory formulae will be elaborated here.

6.1.4.1 Conservative Phase-field lattice Boltzmann equation

The discrete Boltzmann equation for conservative phase-field can be written as

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot \varphi \mathbf{u} = \nabla \cdot \left[M \left(\nabla \varphi - \frac{1 - 4(\varphi - \varphi_{avg})^2}{W} \mathbf{n} \right) \right] \quad (6-41)$$

φ is the order parameter and range in $[0,1]$, $\varphi_{avg} = (\varphi_l + \varphi_h)/2$ is the average value of the two-phase order parameter. $\mathbf{n}$ is the normal vector of the interface and can be determined by

$$\mathbf{n} = \frac{\nabla \varphi}{|\nabla \varphi|} \quad (6-42)$$

At equilibrium state, the order parameter is determined by the hyperbolic tangent function according to the distance along the normal vector direction of the interface

$$\varphi(\xi) = \varphi_{avg} + \frac{(\varphi_h - \varphi_l)}{2} \tanh\left(\frac{2\xi}{W}\right) \quad (6-43)$$

The following lattice Boltzmann equation can be used to solve the partial differential equation (6-37)

$$\frac{\partial h_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla h_\alpha = -\frac{h_\alpha - h_\alpha^{eq}}{\lambda_\varphi} \quad (6-44)$$

where h_α is the particle distribution function and λ_φ is the relaxation time, and the equilibrium distribution function is as followings

$$h_\alpha^{eq} = \varphi \Gamma_\alpha + \theta \omega_\alpha \mathbf{e}_\alpha \cdot \mathbf{n} \quad (6-45)$$

where Γ_α is

$$\Gamma_{\alpha}(\mathbf{u}) = w_{\alpha} \left[1 + \frac{\mathbf{e}_{\alpha} \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_{\alpha} \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right] \quad (6-46)$$

and the additional term $\theta \omega_{\alpha} \mathbf{e}_{\alpha} \cdot \mathbf{n}$ is the anti-diffusion term, in which

$$\theta = \frac{M}{c_s^2} \left[\frac{1 - 4(\varphi - \varphi_{avg})^2}{W} \right] \quad (6-47)$$

The lattice Boltzmann equation (6-40) can be conducted by two steps: collision step and the stream step:

$$h_{\alpha}^{+}(\mathbf{x}, t) = h_{\alpha}(\mathbf{x}, t) - \frac{1}{\tau + 0.5} [h_{\alpha}(\mathbf{x}, t) - h_{\alpha}^{eq}(\mathbf{x}, t)] \quad (6-48)$$

$$h_{\alpha}(\mathbf{x} + \mathbf{e}_{\alpha} \delta t, t + \delta t) = h_{\alpha}^{+}(\mathbf{x}, t) \quad (6-49)$$

where $\tau = \lambda_{\varphi} / \delta t$ is the dimensionless relaxation time, $M = \tau c_s^2 \delta t$ is the mobility rate which is related to the relaxation time. The order parameter can be obtained by the zero-order moment of h_{α}

$$\varphi = \sum_{\alpha} h_{\alpha} \quad (6-50)$$

6.1.4.2 Pressure and momentum evolution lattice Boltzmann equations

The discrete Boltzmann equation for non-ideal fluid system can be written as

$$\frac{\partial f_{\alpha}}{\partial t} + \mathbf{e}_{\alpha} \cdot \nabla f_{\alpha} = -\Lambda(f_{\alpha} - f_{\alpha}^{eq}) + \frac{(\mathbf{e}_{\alpha} - \mathbf{u}) \cdot \mathbf{F}}{c_s^2} \Gamma_{\alpha} \quad (6-51)$$

where $f_{\alpha}^{eq} = \rho \Gamma_{\alpha}$ is the equilibrium distribution function and $\mathbf{F}$ is the external force including the intermolecular force and body force

$$\mathbf{F} = \nabla(\rho c_s^2 - p) + \mathbf{F}_s + \mathbf{G} \quad (6-52)$$

$\mathbf{G}$ is the gravity. The continuum surface tension form is used in this two-phase model

$$\mathbf{F}_s = \mu_{\varphi} \nabla \varphi \quad (6-53)$$

where the chemical potential can be written as

$$\mu_{\varphi} = 4\beta(\varphi - \varphi_l)(\varphi - \varphi_h)(\varphi - \varphi_{avg}) - \kappa \nabla^2 \varphi \quad (6-54)$$

in which the parameter $\beta = 12\sigma / W$ and $\kappa = 3\sigma W / 2$ are related to the surface tension and interface thickness.

Similar to the HCZ model, a new distribution function is constructed as

$$g_\alpha = f_\alpha c_s^2 + w_\alpha (p - \rho c_s^2) \quad (6-55)$$

It is noteworthy that the pressure p here is the hydrodynamic pressure instead of the thermodynamic pressure in the HCZ model. If neglect the high-order term $\mathbf{u} \cdot \nabla p \sim O(Ma^3)$, then substitute Eq.(6-55) into Eq.(6-51)

$$\frac{\partial g_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla g_\alpha = -\Lambda(g_\alpha - g_\alpha^{eq}) + (\mathbf{e}_\alpha - \mathbf{u}) \cdot [(\Gamma_\alpha - w_\alpha) \nabla \rho c_s^2 + \Gamma_\alpha (\mu_\phi \nabla \phi + \mathbf{G})] \quad (6-56)$$

in which the equilibrium distribution function is $g_\alpha^{eq} = f_\alpha^{eq} c_s^2 + w_\alpha (p - \rho c_s^2)$

In addition, the gradient of density can be transformed to the gradient of order parameter

$$\nabla \rho = \frac{\partial \rho}{\partial \phi} \nabla \phi = (\rho_h - \rho_l) \nabla \phi \quad (6-57)$$

Therefore we can get the simplified discrete Boltzmann equation of flow-field

$$\begin{aligned} \frac{\partial g_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \nabla g_\alpha = & -\Lambda(g_\alpha - g_\alpha^{eq}) \\ & + (\mathbf{e}_\alpha - \mathbf{u}) \cdot [(\Gamma_\alpha - w_\alpha)(\rho_h - \rho_l)c_s^2 + \Gamma_\alpha \mu_\phi] \nabla \phi + \Gamma_\alpha (\mathbf{e}_\alpha - \mathbf{u}) \cdot \mathbf{G} \end{aligned} \quad (6-58)$$

In order to construct an explicit time scheme for the discrete Boltzmann equation, the following transformation form is used

$$\begin{aligned} \bar{g}_\alpha(\mathbf{x}, t) = & g_\alpha + \frac{1}{2} \mathbf{S}(g_\alpha - g_\alpha^{eq}) \\ & - \frac{\delta t}{2} (\mathbf{e}_\alpha - \mathbf{u}) \cdot [(\Gamma_\alpha - w_\alpha)(\rho_h - \rho_l)c_s^2 + \Gamma_\alpha \mu_\phi] \nabla^C \phi \\ & - \frac{\delta t}{2} \Gamma_\alpha (\mathbf{e}_\alpha - \mathbf{u}) \cdot \mathbf{G} \end{aligned} \quad (6-59)$$

Chapter 6 Multiphase Lattice Boltzmann Model and Its Applications

Then the collision step is

$$\begin{aligned}
 g_{\alpha}^{+}(\mathbf{x}, t) = & \bar{g}_{\alpha}(\mathbf{x}, t) - \mathbf{M}^{-1} \mathbf{S} \mathbf{M} [\bar{g}_{\alpha}(\mathbf{x}, t) - \bar{g}_{\alpha}^{eq}(\mathbf{x}, t)]|_{(\mathbf{x}, t)} \\
 & + \delta t (\mathbf{e}_{\alpha} - \mathbf{u}) \cdot [(\Gamma_{\alpha} - w_{\alpha})(\rho_h - \rho_l) c_s^2 + \Gamma_{\alpha} \mu_{\varphi}] \nabla^M \varphi|_{(\mathbf{x}, t)} \\
 & + \delta t \Gamma_{\alpha} (\mathbf{e}_{\alpha} - \mathbf{u}) \cdot \mathbf{G}|_{(\mathbf{x}, t)}
 \end{aligned} \tag{6-60}$$

in which the mixed discretization scheme is applied. $\mathbf{M}$ is the transformation matrix and $\mathbf{S}$ is the relaxation matrix,

The stream step is

$$\bar{g}_{\alpha}(\mathbf{x} + \mathbf{e}_{\alpha} \delta t, t + \delta t) = g_{\alpha}^{+}(\mathbf{x}, t) \tag{6-61}$$

The modified equilibrium distribution function is

$$\begin{aligned}
 \bar{g}_{\alpha}^{eq} = & g_{\alpha}^{eq} - \frac{\delta t}{2} (\mathbf{e}_{\alpha} - \mathbf{u}) \cdot [(\Gamma_{\alpha} - w_{\alpha})(\rho_h - \rho_l) c_s^2 + \Gamma_{\alpha} \mu_{\varphi}] \nabla^C \varphi \\
 & - \frac{\delta t}{2} \Gamma_{\alpha} (\mathbf{e}_{\alpha} - \mathbf{u}) \cdot \mathbf{G}
 \end{aligned} \tag{6-62}$$

in which the central difference discretization is used.

The above discrete Boltzmann equation finally recovers the following macroscopic equation for the flow field

$$\frac{1}{\rho c_s^2} \frac{\partial p}{\partial t} + \nabla \cdot \mathbf{u} = 0 \tag{6-63a}$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot [\mu (\nabla \mathbf{u} + \mathbf{u} \nabla)] + \mathbf{F}_s + \mathbf{G} \tag{6-63b}$$

The hydrodynamic variables can be calculated by the moment of the modified distribution function and the additional modification of the moment of the external force

$$p = \sum_{\alpha} \bar{g}_{\alpha} + \frac{\delta t}{2} (\rho_h - \rho_l) c_s^2 \mathbf{u} \cdot \nabla^C \varphi \tag{6-64}$$

$$\rho c_s^2 \mathbf{u} = \sum_{\alpha} \bar{g}_{\alpha} \mathbf{e}_{\alpha} + \frac{\delta t}{2} c_s^2 (\mu_{\varphi} \nabla^C \varphi + \mathbf{G}) \quad (6-65)$$

6.2 Numerical verification

In this section, some benchmark tests have been conducted to verify the conservativeness, stability and accuracy of the multiphase lattice Boltzmann model developed by F. Abbas et al. (2016). Firstly, the numerical scenario of circular interface in a shear flow and deformation of a circular interface are applied to validate the conservative property of the phase-field model. Then, the Rayleigh-Taylor instability, drop splashing on a thin film and 3D Laplace law have been tested to check the stability and accuracy of the present two-phase lattice Boltzmann model.

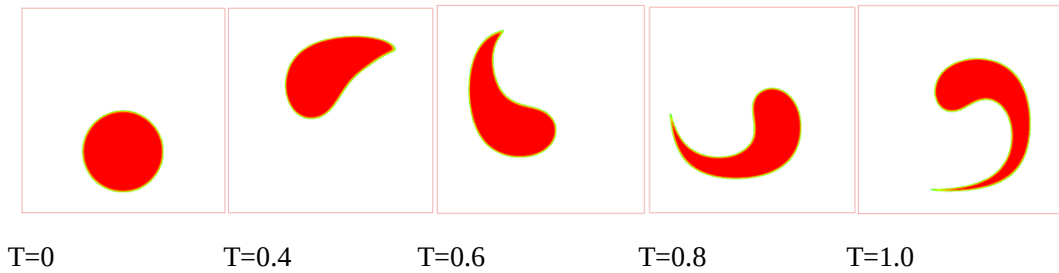
6.2.1 Circular interface in a shear flow

In this case, a circular interface is positioned in a shear flow, of which the velocity field can be expressed as

$$u(x, y) = -U_0 \pi \cos[\pi(x/L_0 - 0.5)] \sin[\pi(y/L_0 - 0.5)] \quad (6-66a)$$

$$v(x, y) = U_0 \pi \sin[\pi(x/L_0 - 0.5)] \cos[\pi(y/L_0 - 0.5)] \quad (6-66b)$$

A circular interface with the radius $R = L_0/5$ is initially placed at $(x, y) = (100, 60)$, where $L_0 = 200$ is the length of the side. The velocity field is reversed at $T = T_f$. Therefore, it will go back to its original place theoretically. The simulation will be stopped at $T = 2T_f$. The evolution of the circular interface is described in Fig.6.1. As can be seen, the interface of the circular drop keeps smooth and finally goes back to its original position.



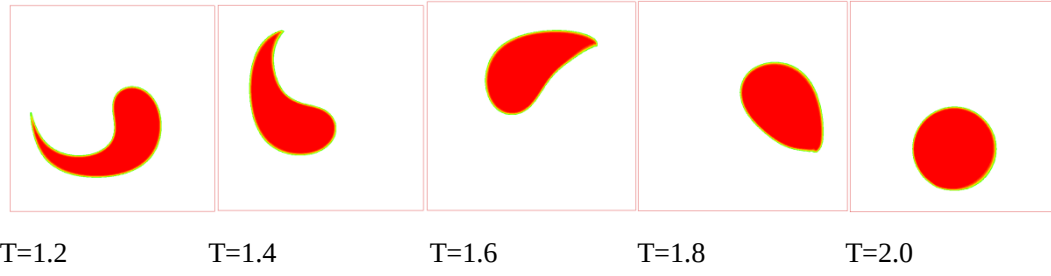


Fig.6.1 Evolution of the circular interface

6.2.2 Deformation of a circular interface

In this case, a more complicated deformation will be found due to the following velocity field:

$$u(x, y) = -U_0 \sin[4\pi(x/L_0 + 0.5)] \sin[4\pi(y/L_0 + 0.5)] \cos(\pi T/T_f) \quad (6-67a)$$

$$v(x, y) = -U_0 \cos[4\pi(x/L_0 + 0.5)] \cos[4\pi(y/L_0 + 0.5)] \cos(\pi T/T_f) \quad (6-67b)$$

In the above velocity field setting, the velocity field will gradually develop and then reverse smoothly with the function $\cos(\pi T/T_f)$. A circular interface with radius $R = L_0/5$ is placed at the center of the square domain $[L_0, L_0]$ with $L_0 = 500$. The evolution of the circular interface is shown in Fig.6.2. It can be found that the complicated interface evolution is captured and the interface is recovered to its original position.

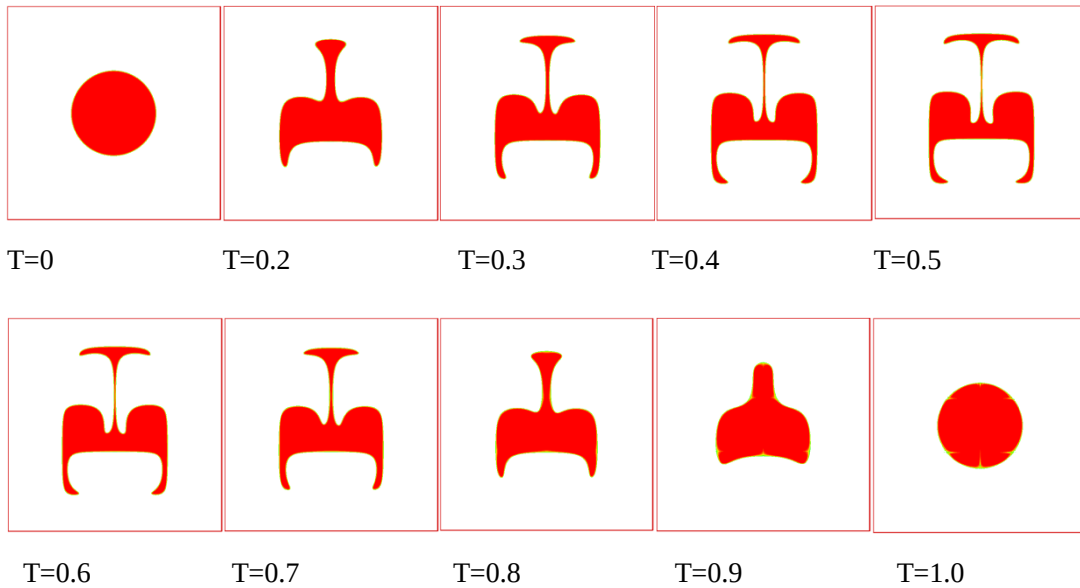


Fig.6.2 Evolution of the circular interface

6.2.3 2D Rayleigh-Taylor instability

The classical Rayleigh-Taylor instability of the interface with two immiscible fluids is tested to verify the stability and accuracy of the present multiphase lattice Boltzmann model. The heavier fluid (ρ_H) is positioned upon the light fluid (ρ_L) with the initial interface profile $y(x) = 0.1d \cos(2\pi x/d)$ in a box region $[-d/2, d/2] \times [-2d, 2d]$ exposed under gravity. Non-slip boundary conditions are applied at the bottom and top and periodic boundary conditions are applied to the left and right sides. The width of the flow domain is $d = 128$, gravity is defined as $\sqrt{dg} = 0.01$, Reynolds number is defined as $Re = \sqrt{dg} \cdot d / \nu = 256$ with the identical viscosity for the two fluids and $At = (\rho_H - \rho_L) / (\rho_H + \rho_L) = 0.5$ with $\rho_L = 1$. The numerical resolution is 128×512 in the width and height direction.

Fig.6.3 depicts the interface position of the interface front with time evolution. The present results agree well with the results of the multiphase lattice Boltzmann flux solver of Wang et al. (2014) and the results of Navier-Stokes solver of Nourgaliev et al. (2004). Fig.6.4 shows the interface evolution process at the specific times, in which the basic physical phenomenon can be vividly depicted.

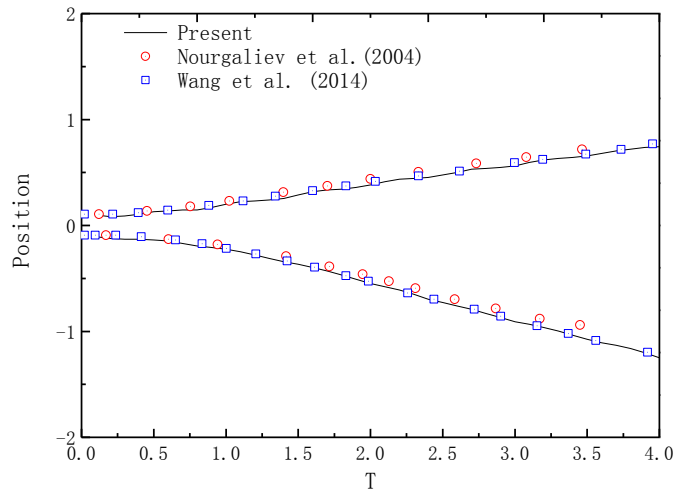


Fig.6.3 Positions of the bubble and spike fronts for the Rayleigh-Taylor instability at $Re=256$

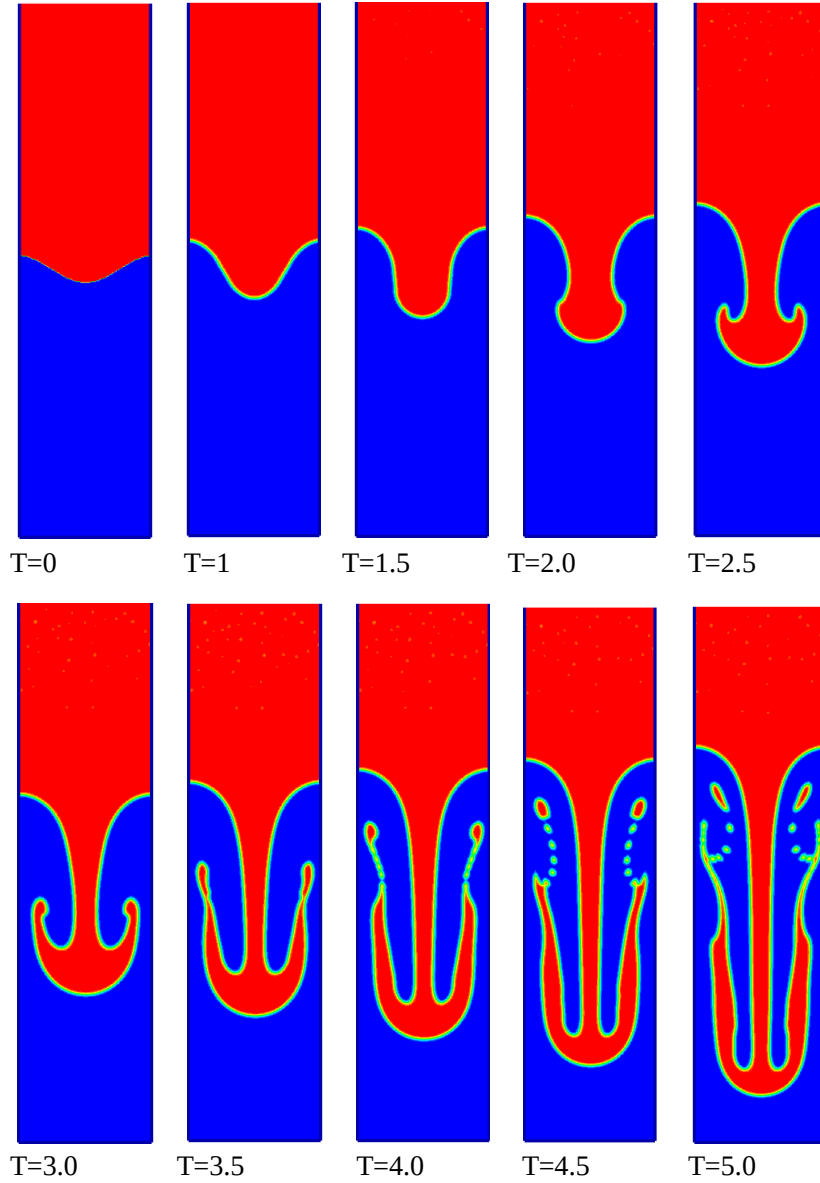


Fig.6.4 Evolution of the fluid interface for the Rayleigh-Taylor instability at $Re=256$

6.2.4 Drop splashing on a thin film

Droplet splashing issue is significant for a wide variety of engineering applications, such as ink-jet printing, internal combustion engines and turbine blades. Previous experimental work demonstrated that there exist two patterns when the droplet impacts on the thin film with different velocities: deposition and splashing. In the present work, in order to investigate the different physical phenomenon with different Reynolds number ($Re = 2\rho_H UR / \mu_H$), the cases of $Re=20, 100, 500, 1000$ have been calculated. The fluid parameters are set referred to Lee et al.(2005): the density ratio is set to $\rho_H / \rho_L = 1000$, and the viscosity ratio is given as $\mu_H / \mu_L = 40$, the Weber number is set as $We = 2\rho_H U^2 R / \sigma = 8000$. The radius of the droplet

is set to 100 and the height of the thin film is set to 30. The initial geometry arrangement can be found in Fig.6.5. The numerical resolution is set as 1532*512 in the lateral and height direction. Non-slip boundary condition is applied at the top and bottom while the periodic boundary condition is applied on the left and right sides. As shown in Fig.6.6-9, the decomposition phenomenon can be found at low Reynolds number ($Re=20$). With $Re=100$, the splashing can be found at the later stage of the evolution. At high Reynolds number regime ($Re=500, 1000$), both the splashing and breaking phenomena can be observed.

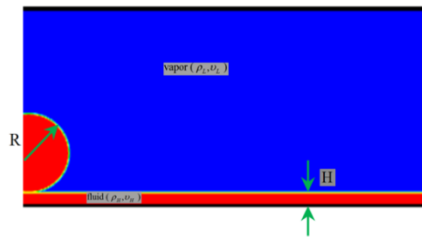


Fig.6.5 Schematic of the droplet splashing on a thin film

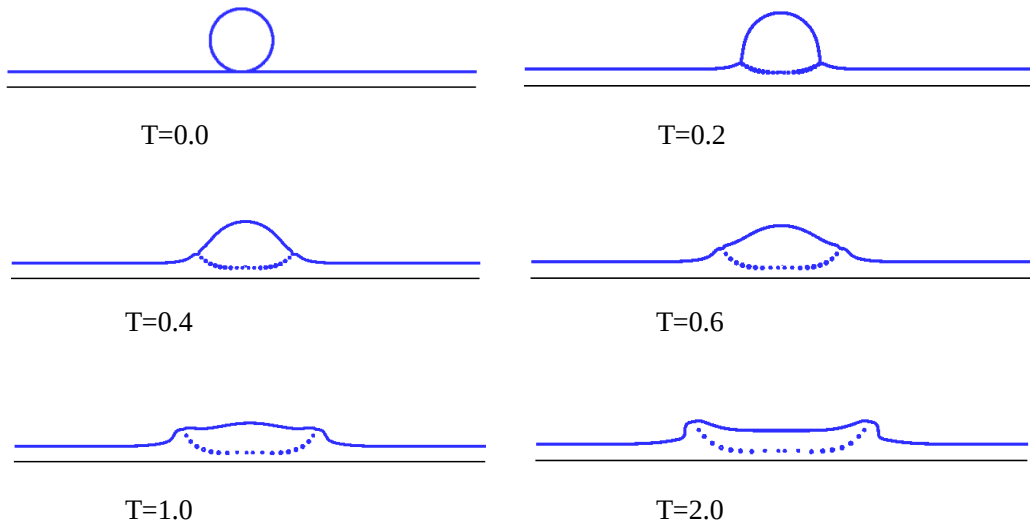
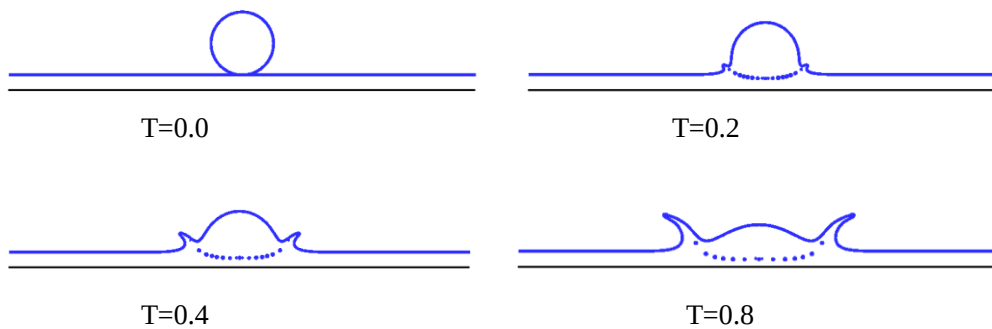


Fig. 6.6 Evolution of the interface at specific time steps at $Re=20$



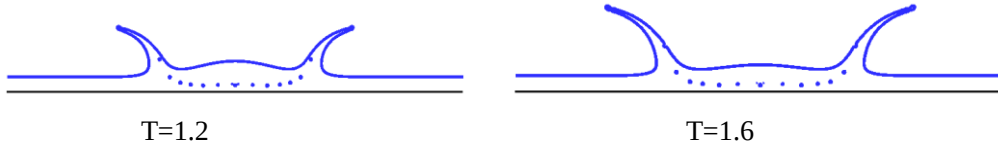


Fig. 6.7 Evolution of the interface at specific time steps at $Re=100$

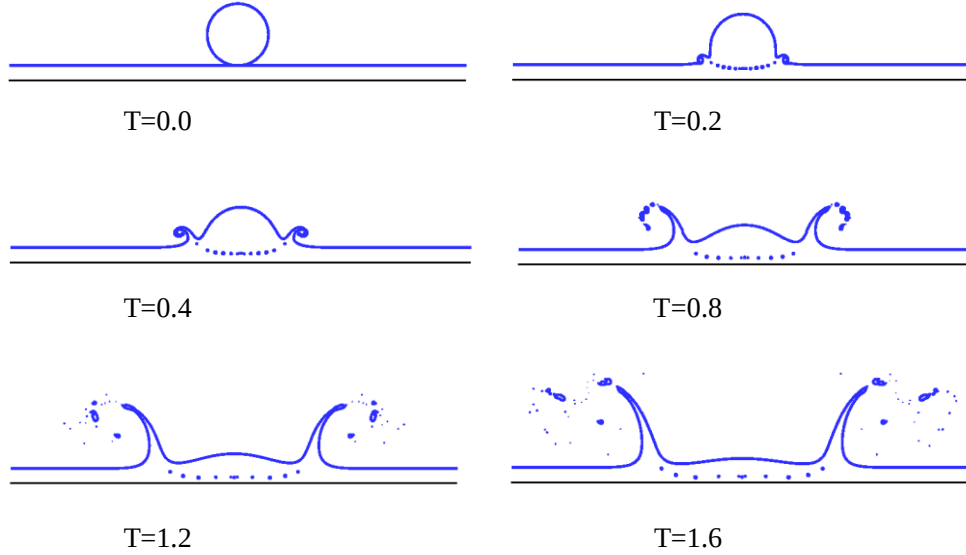


Fig. 6.8 Evolution of the interface at specific time steps at $Re=500$

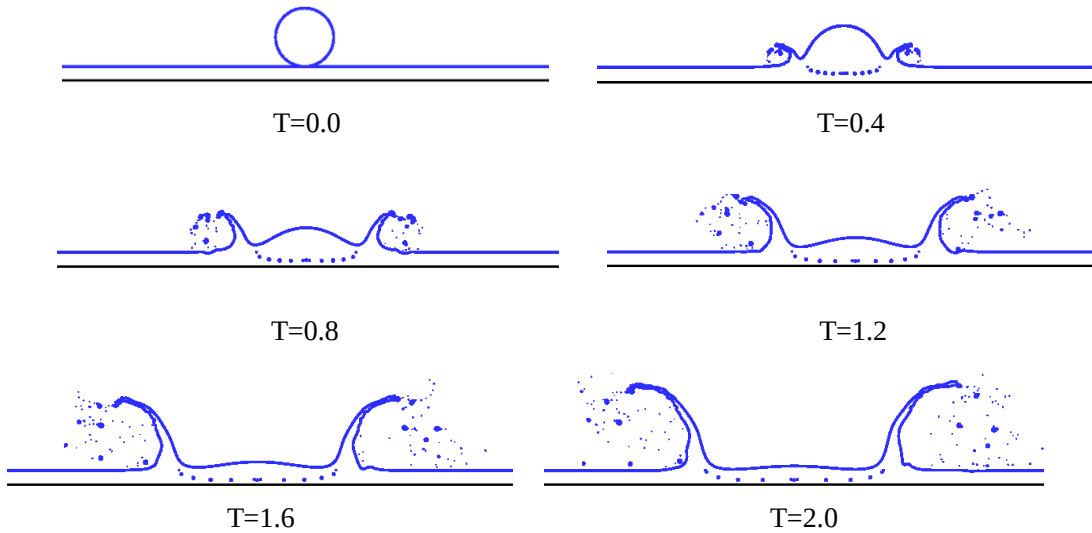


Fig. 6.9 Evolution of the interface at specific time steps at $Re=1000$

6.2.5 3D Laplace Law

The Laplace law for a three-dimensional static drop with density ρ_H that immersed in another fluid with density ρ_L could be examined by the present two-phase lattice Boltzmann

model. Theoretically, the pressure difference across the interface of the droplet can be analytically expressed as

$$p_{in} - p_{out} = \frac{2\sigma}{R} \quad (6-68)$$

where p_{in} and p_{out} are the average pressures inside and outside the droplet, respectively. σ is the surface tension and R is the radius of the droplet. Fig.6.10 is the calculation domain [128*128*128]. The droplet is positioned at the center of the domain and the periodic boundary condition is applied at all sides. The surface tension is set as 0.001 and the density ratio of two fluids is set as 1000. Six cases are test with the radius of the droplet as 20, 25, 30, 35, 40, 45. The pressure difference simulated by the two-phase model is plotted versus the analytical solution as a function of $1/R$ in Fig.6.11. As can be observed, the numerical solutions agree well with analytical solutions.

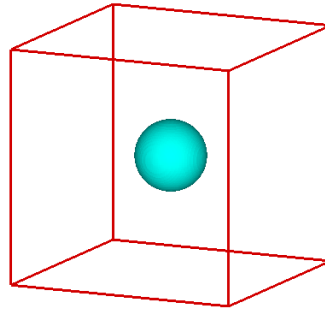


Fig. 6.10 Geometry arrangement of the droplet

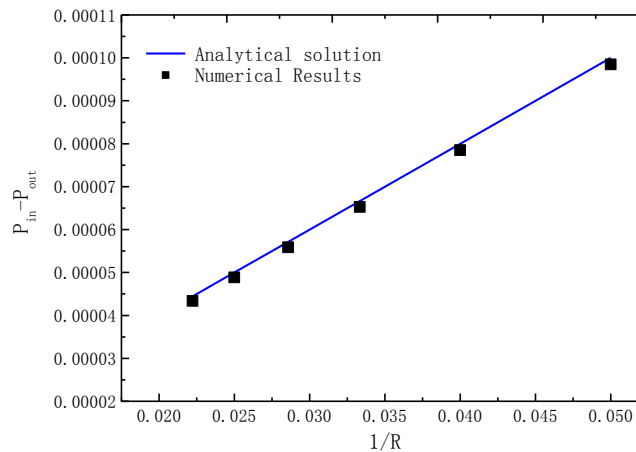


Fig. 6.11 Laplace law verification

6.3. Bubble dynamics

The rising of bubbles in the viscous liquid due to buoyancy is one of typical multiphase problems. A fundamental understanding of the mechanism in the bubble dynamics is essential in various practical applications, such as the underwater explosion, the rise of steam bubble in boiler tubes, gas bubbles in oil well. However, a comprehensive knowledge of the flow phenomena and mechanism of such multiphase systems in full flow regimes is still unclear, even a number of experimental, theoretical analysis and numerical studies have addressed this problem. Lattice Boltzmann method is a concise and convenient tool to deal with surface tension and viscosity. Therefore, the bubble dynamics are investigated carefully in this section.

6.3.1 Single bubble

Single rising bubble is studied based on the experiment cases by D.Bhaga et al. (1981). The parameters such as Bond number (Eu), Morton number (Mo) are set according to the those in the experiments (D.Bhaga et al., 1981). These characteristic parameters are defined as

$$Eu = \frac{\rho g D^2}{\sigma} \quad (6-69)$$

$$Mo = \frac{g \mu^4}{\rho \sigma^3} \quad (6-70)$$

The diameter of the bubble is set to $D = 32$. The fluid domain is resolved by the resolution $[6D, 6D, 12D]$. The bubble is initially positioned at $[3D, 3D, 2D]$ and released at the $T=0$. Due to the synthetic effect of gravity, surface tension and viscosity, the bubble will rise with deformation. The rising velocity of the bubble will increase dramatically and then gradually slowly. Finally, the bubble will rise at a stable velocity. The terminal shape and velocity of the bubble for cases A1-A4 and B1-B4 are compared with the experiment data and the numerical results by J.Hua et al. (2007) in Table I and Table II. As can be found, the present results can agree with both of them very well.

Table 6.1 Comparison of terminal bubble shapes observed in experiments (D.Bhaga et al. 1981), other literature (J. Hua et al. 2007) and the present simulations (A1-A4)

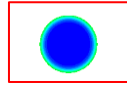
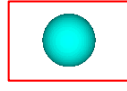
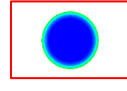
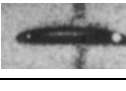
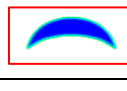
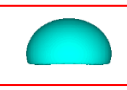
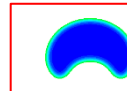
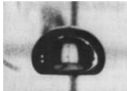
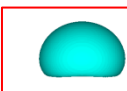
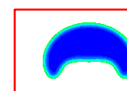
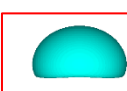
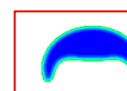
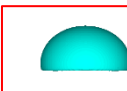
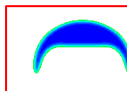
Test No.	Experiments		Present simulations			J.Hua et al.(2007) simulations	
	Test parameters	Terminal bubble shapes	3D terminal shape	Slices	Terminal Re	Terminal shapes	Terminal Re
A1	Eo=8.67 Mo=711 Re=0.078				0.0427		0.0705
A2	Eo=17.7 Mo=711 Re=0.232				0.140		0.211
A3	Eo=32.2 Mo=8.2E-4 Re=55.3				52.81		52.96
A4	Eo=243 Mo=266 Re=7.77				8.09		8.397

Table 6.2 Comparison of terminal bubble shapes observed in experiments (D.Bhaga et al. 1981), other literature (J. Hua et al. 2007) and the present simulations (B1-B4)

Test No.	Experiments		Present simulations			J.Hua et al.2007 simulations	
	Test parameters	Terminal bubble shapes	3D terminal shape	Slices	Terminal Re	Terminal shapes	Terminal Re
B1	Eo=116 Mo=848 Re=2.47				2.477		2.317
B2	Eo=116 Mo=266 Re=3.57				3.893		3.621
B3	Eo=116 Mo=41.1 Re=7.16				7.296		7.002
B4	Eo=116 Mo=5.51 Re=13.3				13.464		13.167

6.3.2 Bubble interaction

In this section, the bubble interactions with different initial arrangement and bubble size are investigated in detail.

6.3.2.1 Co-axial coalescence of two bubbles

In Case I, a small bubble with radius $0.5R$ positioned at $2.75D$ and a large bubble with radius R at $1.5D$. They are released at $T=0$ and rise with time. From Fig.6.12, it is found that both the small bubble and the larger bubble are deformed into oblate shape gradually. However, it is obvious that the large bubble rises much faster than the smaller one and kisses the small bubble at $T=3$. The large bubble dominates the coalescence process. Finally, the merged bubble rise like a single bubble.

In Case II and Case III, a large bubble with radius R is placed at the initial height $3D$ while a small bubble with radius $0.65R$ or $0.7R$ is positioned at the initial height $1.5D$. From Fig.6.13, it can be seen that the small bubble is affected by the wake of the large bubble when they rise together. The small bubble is elongated into a vertical ellipsoid by the wake of the large bubble. Unfortunately, the rising velocity of the small bubble is not high enough even it is accelerated by the wake field of the large bubble. Finally the two bubbles keep a distance and can't contact each other. If the small bubble size is increased to $0.7R$, the coalescence phenomenon is observed. From Fig.6.14, the interesting phenomenon can be observed that influenced by the wake field of the large bubble before they contact each other the small bubble is deformed into an upside-down pear shape. Quickly, the large bubble wraps the small pear bubble and then they merge together.

It is noteworthy to be addressed that there should exit a critical radius for the small bubble. If the distance of the large bubble over the small bubble is fixed, there always exits a critical radius for the small bubble. Larger than the critical size, the small bubble will catch up with the large one, while smaller than the critical size the small bubble can't catch up with the large one. On the contrary, there also exits a critical distance for them to be coalesced or not if the bubble sizes of the large bubble and the small bubble are fixed. If the bubble size ratio (radius of small bubble versus the radius of the large bubble) is regarded as the x axis and the distance between them regarded as the y axis, it seems to be that there is an area that two bubbles can merge

together while there is also an area that the two bubbles cannot merge together. Therefore, it can be concluded that the bubble interaction is affected not only by the bubble size but also their relative position.

Case I: Small bubble with radius $0.5R$ over the large bubble with radius R (density ratio 1000, viscous ratio 100)

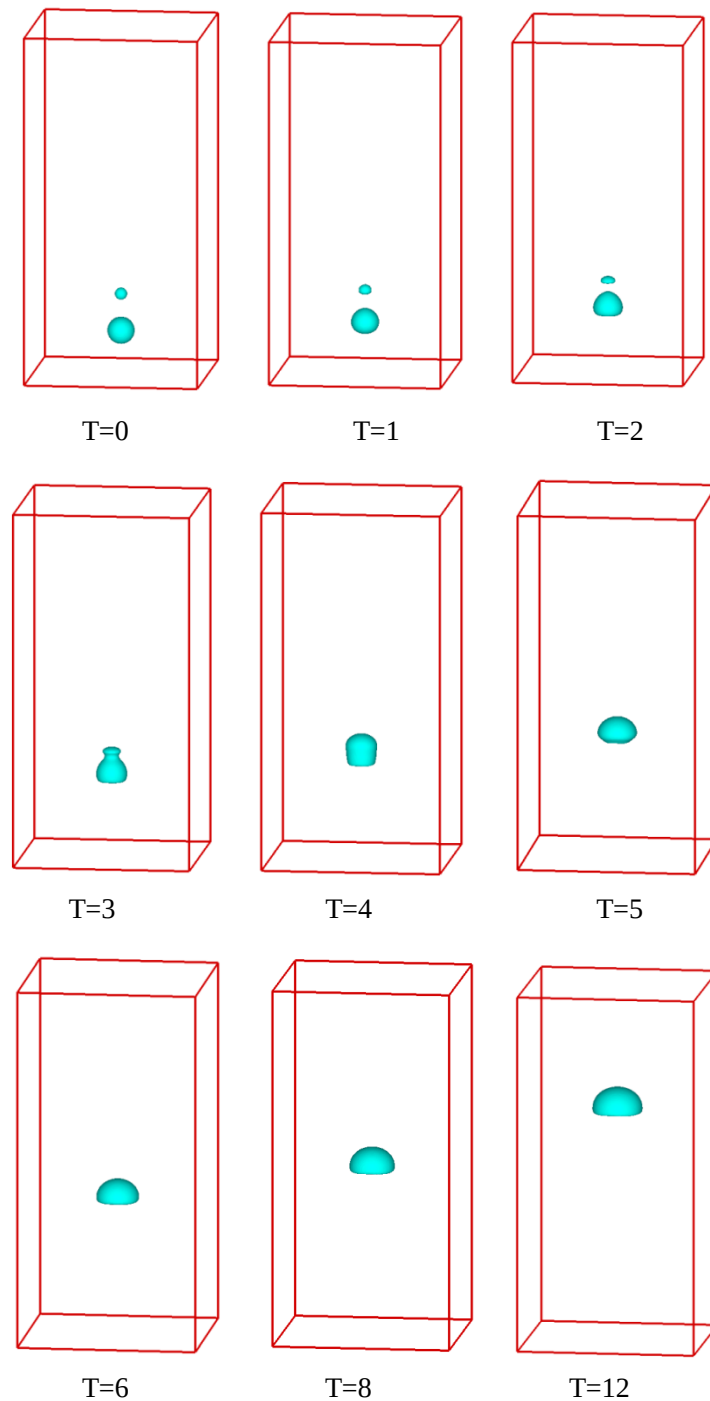


Fig.6.12 Evolution of the coalescence of two different size bubbles (small bubble over large bubble)

Case II: small bubble with $0.65R$ under a large bubble with R (density ratio 1000, viscous ratio 100)

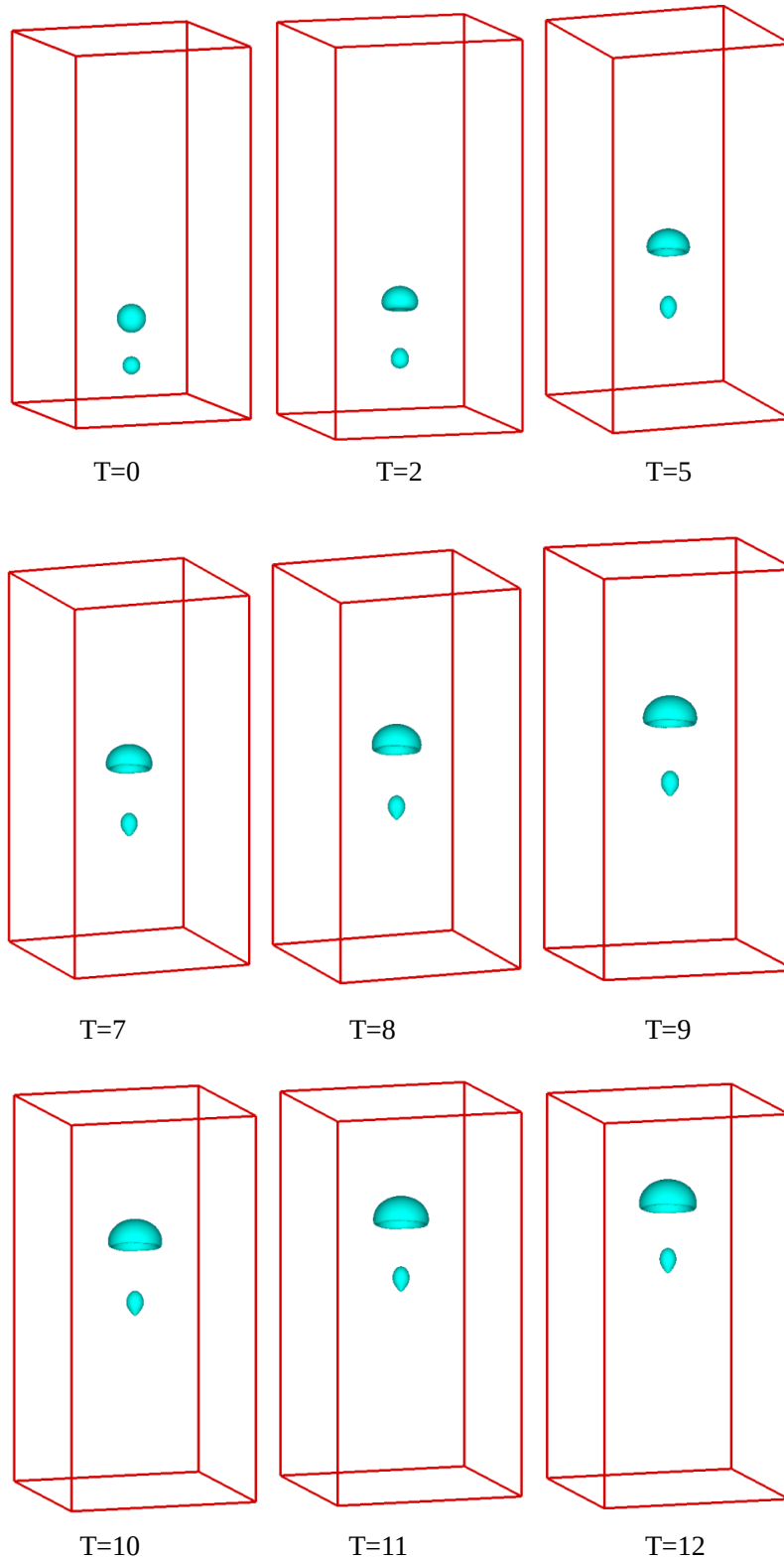


Fig.6.13 Evolution of the coalescence of two different size bubbles (large bubble over small bubble with $0.65R$)

Case III: small bubble with $0.70R$ under a large bubble with R (density ratio 1000, viscous ratio 100)

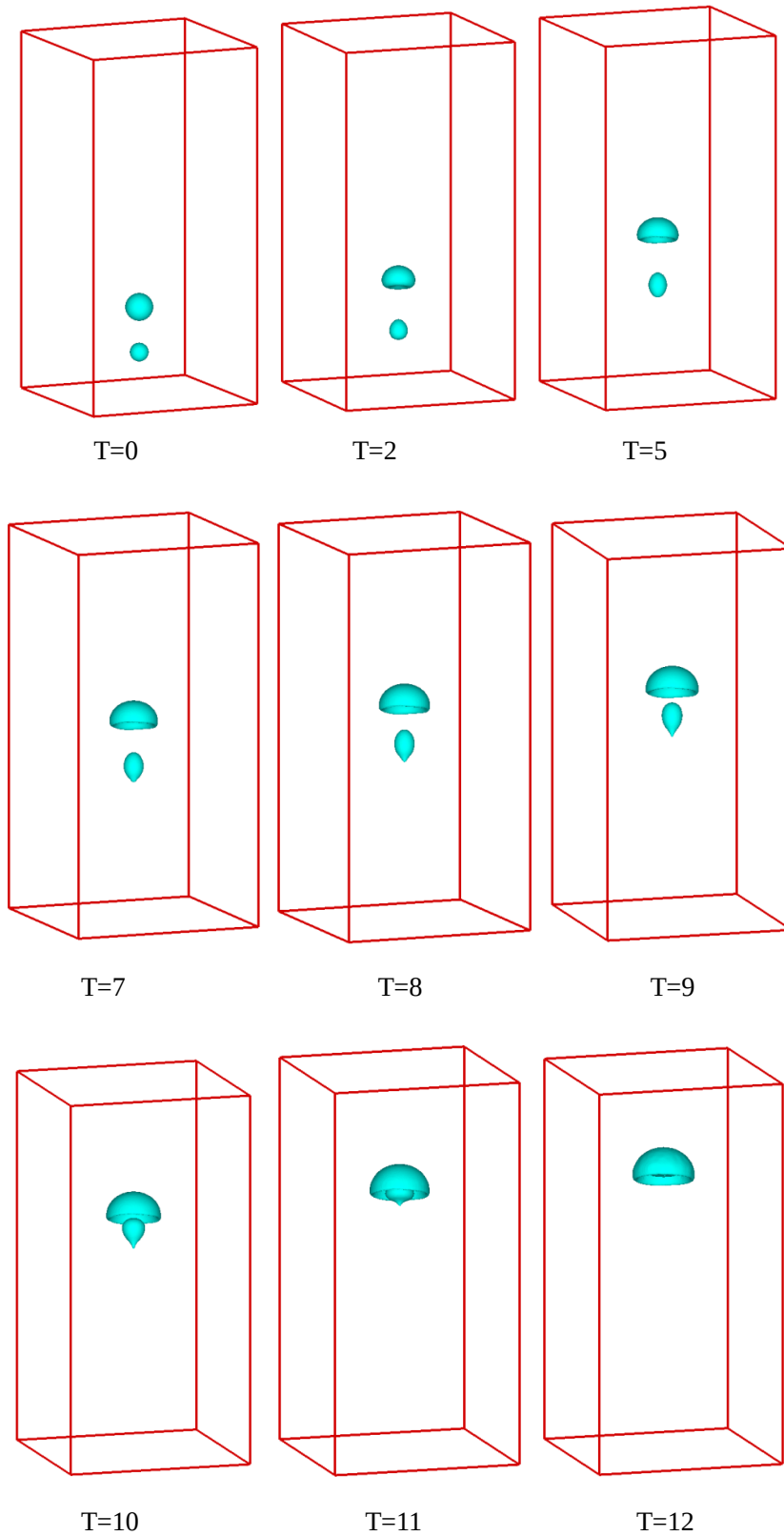
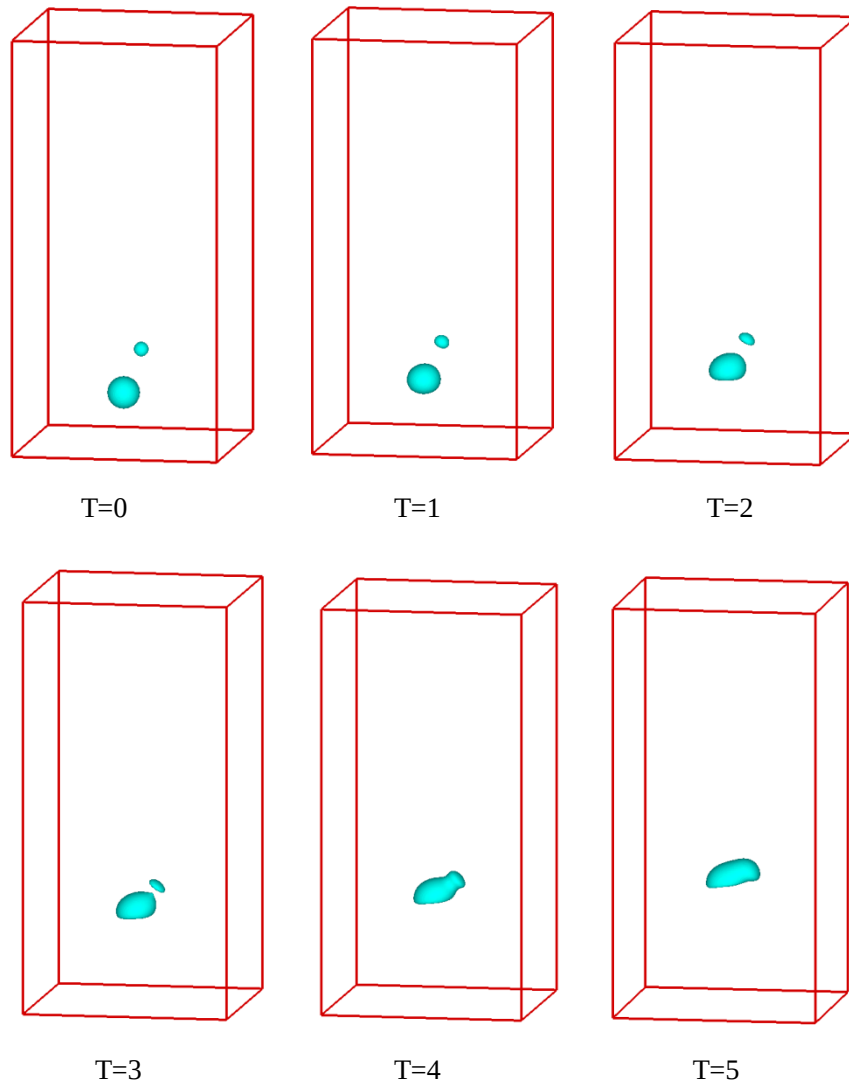


Fig.6.14 Evolution of the coalescence of two different size bubbles (large bubble over small bubble with $0.7R$)

6.3.2.2 Oblique coalescence of two bubbles

In this section, two oblique bubbles with $0.5D$ offset in x direction are positioned at height of $1.5D$ and $2.75D$ from the bottom. The fluid domain is the same as the first section. As can be seen in Fig.6.15, similar coalescence can be found in the oblique bubbles. The large bubble is affected by the wake of the upper small bubble and deforms towards the small bubble. Gradually they merge together and behave as a single bubble.

Case I: large bubble with R under small bubble with $0.5R$ (density ratio 1000, viscous ratio 100)



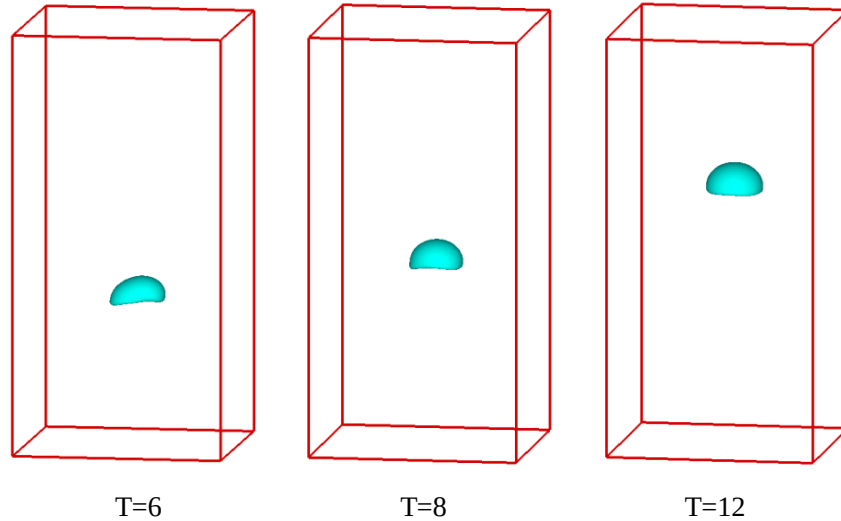


Fig.6.15 Evolution of the oblique coalescence of two bubbles with different size

6.3.2.3 Bubble flow

In the work of M.Cheng et al.(2010) and Inamuro et al. (2004), the bubble flows with decades of bubbles of same size have been simulated. While in order to capture the more natural and authentic phenomenon of bubble flows, we conduct the simulation of 45 random diameter bubbles rising in a square duct. The diameter of the bubbles range from $0.375D_0$ to $0.75D_0$ with $D_0=32$, and the fluid domain is set as $[6D_0, 6D_0, 12D_0]$. In the lateral direction the non-slip boundary condition is applied, the periodical boundary condition is imposed at the bottom and top sides. The Bond number is set to 116 and the Morton number is set as 42.1. The density ratio is 1000 and the viscosity ratio is 100.

Fig.6.16 shows the bubble flow evolution process. The wake of each bubble influences other bubbles. The strength of the wake depends on the bubble size, the distance and angle between each other. With the surrounding bubbles of different size, different distance and relative position, every bubble is rising with distortion and deformation. It is interesting that the small bubbles seem to be attracted into the large bubbles and engulfed quickly. The large bubbles are also affected by the wake of the above large bubbles. However, the small bubbles far away from other bubbles are abandoned by the bubble flow. With the coalescence of each other, more large bubbles are formed and more strong wakes are generated after them. Gradually, a lot of bubbles merge into several large bubbles in the center region of the duct from $T=8$, while only several small bubbles are left far away from the wake. These results indicate

Chapter 6 Multiphase Lattice Boltzmann Model and Its Applications

that the relative size, relative angle and relative distance will remarkably affect the features of the bubble evolution in the complicated wake field of the bubble flow.

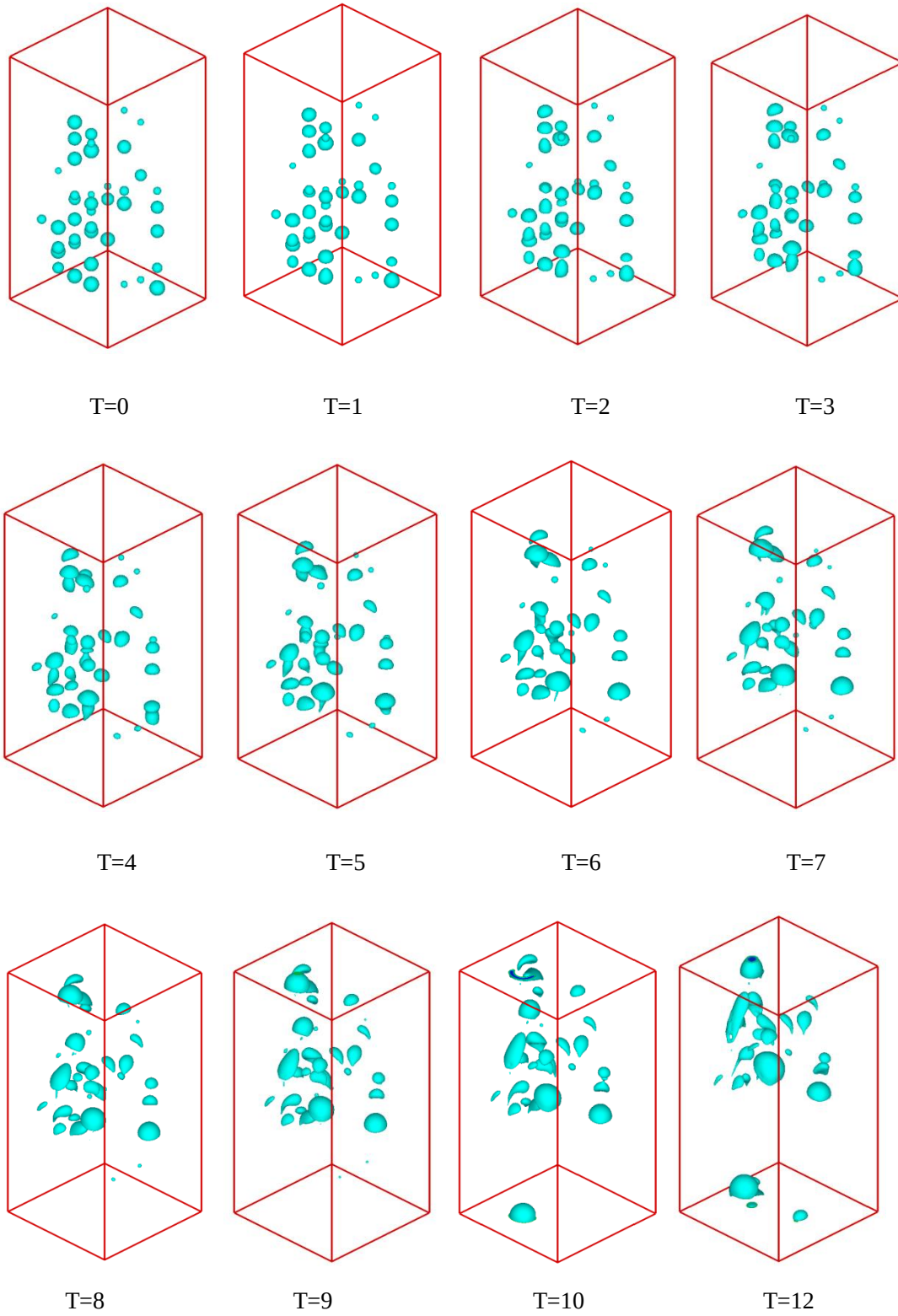


Fig. 6.16 Evolution of 45 different size bubbles with random settings and positions

6.3.3 Motion of Bubble Ring

In this section, a bubble ring with a circular cross-section of radius R_0 , circulation γ_0 and density ρ_g immersed in the viscous fluid with density ρ_l is simulated. The density ratio is $\rho_l / \rho_g = 100$ and the viscosity ratio is $\mu_l / \mu_g = 100$. The resolution of the flow domain is $[10R_0, 10R_0, 12R_0]$ with $R_0=28.8$, which is constrained by the GPU memory (24GB for 4 C2070 cards). The bubble ring center is initially positioned at $[5R_0, 5R_0, 3R_0]$. The core radius of the bubble ring is set as $r_0 = 0.25R_0$. The initial arrangement can be found in Fig.6.17.

The initial vorticity distribution of the bubble is assigned by a Gaussian function (M. Cheng et al., 2010), which indicates that the velocity distribution around the ring is related to the circulation γ_0 as the following formula

$$\mathbf{u}_0 = \frac{\gamma_0}{2\pi r} [1 - e^{-(r/r_0)^2}] \mathbf{n} \quad (6-71)$$

where r is the radial distance from the center of the core and $\mathbf{n}$ is the unit vector tangential to circulation circles about the ring core.

In initial state, the ring vorticity is perturbed in the radial direction of the plane of the ring bubble by modifying the axis of the vortex core using a sine-wave perturbation. The initial perturbation can be expressed as the following formula

$$R = R_0 + \varepsilon \sin(N\theta) \quad (6-72)$$

where $\varepsilon = 0.02$ is the perturbation amplitude and N is the wavenumber.

The characteristic variables governing the flow and defined here are as follows:

$$T = t\sqrt{g/R_0} \quad \Gamma = \frac{2\gamma}{\pi R_0 \sqrt{gR_0}} \quad \lambda = \rho_l / \rho_g \quad R_e = \frac{R_0 \sqrt{gR_0}}{\nu} \quad E_o = \frac{(\rho_l - \rho_g)gr_0^2}{\sigma} \quad (6-73)$$

where g is the gravitational acceleration and ν is the kinematic viscosity.

In the present simulation, Reynold number is fixed as $Re=40$, the initial circulation strength is set as $\Gamma=3.0$, the Bond number that characterize the surface tension is set as $E_o=50$. Fig.6.18 and Fig.6.19 are the experiment phenomenon and our numerical simulation. It can be found that 1) the vortex ring bubble rises with expanding radius and decreasing core

Chapter 6 Multiphase Lattice Boltzmann Model and Its Applications

radius to maintain the conservative volume; 2) the deformation in radial direction and vertical direction can be observed with the bubble rising; 3) the instability of the vortex ring bubble becomes more and more obvious with the evolution of the vortex ring bubble. Finally, the vortex ring bubble breaks into small N bubbles, which is identical to the initial setting wavenumbers. The vortex effects together with the surface tension, viscosity, initial circulation and gravity are more complex than those in single bubble rising phenomena. No stable rising velocity of the vortex ring bubble has been observed.

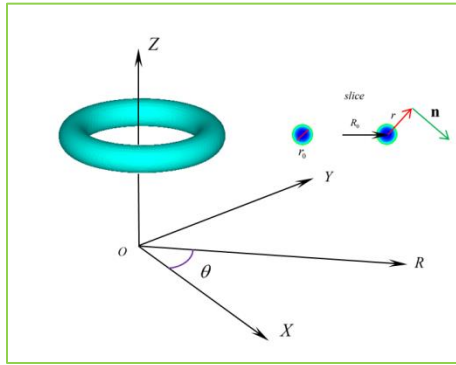


Fig. 6.17 Schematic diagram of the vortex ring bubble

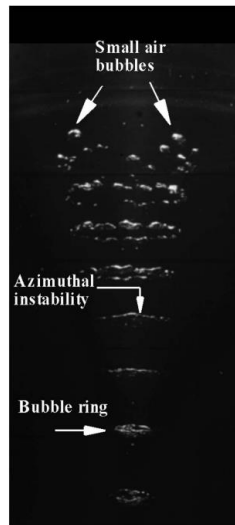


Fig.6.18 The snapshot of evolution of a bubble ring (M. Cheng et al., 2013)

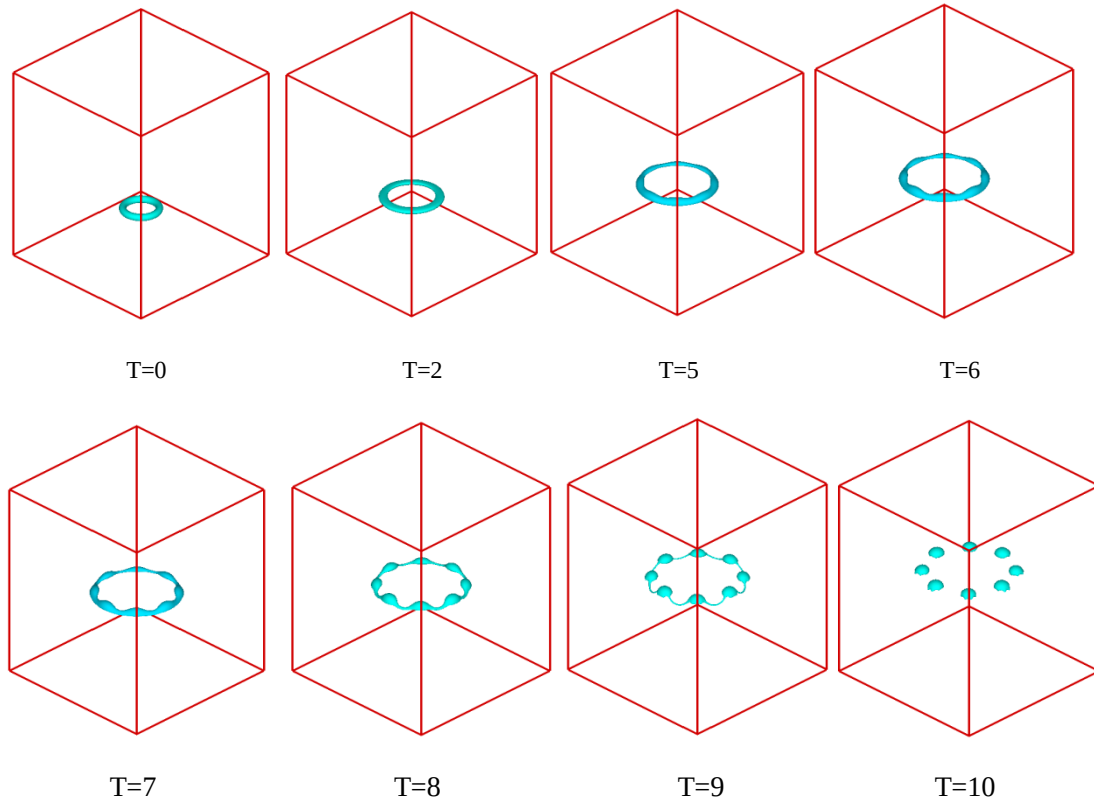


Fig.6.19 Interface evolution of the ring bubble with time ($Re=40$, $Eo=50$, $\Gamma=3.0$)

6.4 Concluding Remarks

In this chapter, the multiphase lattice Boltzmann models have been investigated in detail. Firstly, the basic theory, underlying connections and the difference among the four multiphase models are stated and analyzed. The Abbas-Martin-Lee model is finally chosen to conduct the high density ratio and high Reynolds number flow in the present work. The multiphase lattice Boltzmann model is verified by five numerical benchmarks: the circular interface in a shear flow, complicated interface deformation, Rayleigh-Taylor instability, drop splashing on a thin film, three-dimensional Laplace law. Subsequently, the bubble dynamics has been investigated systematically. Single bubble rising, bubble-bubble interaction, bubble flows and vortex ring bubble have been investigated and analyzed in detail. In the future work, the present liquid-gas two-phase model will be extended to the liquid-gas-solid model that incorporates the immersed boundary method in chapter 2.

Chapter 7 Conclusions

The lattice Boltzmann method has attracted increasing popularity in simulating various complex fluid flows because of its implementation simplicity, excellent execution efficiency and intrinsic parallelism. As a young CFD branch, LBM has not yet completely applied into complicated free surface problems in the ocean engineering field. The work described in the present thesis is devoted to investigate the potential of the lattice Boltzmann method in ocean engineering applications. It is well known that the wave-body interaction such as ship and platform motion in waves, as a typical ocean engineering problem usually involves two major numerical difficulties for LBM: moving solid boundary and complicated free surface. The present work is devoted to handle these two topics. The main results are as follows.

(1) For the moving solid boundary, the difficulty is the spurious pressure oscillation in the treatment of moving fluid-solid boundary, which is systematically investigated in this study. It is found that firstly the fundamental collision model in the lattice Boltzmann method causes intrinsic pressure fluctuation. Secondly the re-initialization algorithm for the fresh fluid node that just emerged from the solid region remarkably affects the oscillation of the force exerted on the moving body immersed in the fluid domain. Moreover, the force evaluation approach (such as various momentum exchange methods, stress integration method, immersed boundary method) is also relevant to the accuracy of the pressure evaluation and the oscillation of the force on the moving body. To improve the accuracy of force evaluation of the moving boundary, a bi-quadratic ghost fluid immersed boundary method has been proposed. To suppress the pressure oscillation issue, a new refilling scheme for the fresh fluid node has been developed.

(2) For the free surface, two approaches have been explored to model the violent free surface flow: the single-phase free surface model and the high-density ratio multiphase model. To develop a stable, accurate and efficient free surface capture algorithm is a key challenge for both of them. In the present study, a VOF (volume-of-fluid) type algorithm is applied to capture the free surfaces in the single-phase free surface model which ignores the gas phase and model the atmosphere pressure as the kinetic pressure boundary condition on the free

surface. The classical engineering applications such as dam breaking, dam breaking with obstacle, water entry and exit of circular cylinder with constant speed, water entry of circular cylinder with free falling have been successfully simulated. Meanwhile, the multiphase lattice Boltzmann models which are capable of high-density ratio and high Reynold number simulations have also been investigated. The mass conservative phase-field interface-tracking equation has been applied to treat the interfacial dynamics. The state-of-art mass-conservative multiphase lattice Boltzmann model developed by Abbas & Martin & Lee has been applied in this thesis. The bubble dynamics including single bubble behavior, bubble-bubble interactions and vortex ring bubble have been investigated deeply. The numerical simulation results agree well with the existing experimental data and other numerical results.

(3) To efficiently implement the numerical code of different LB solver in the thesis, the modern GPU accelerated technique has been realized and equipped to the present programs. In order to achieve the large-scale three-dimensional numerical simulation, the multi-GPU technique has been also developed due to the limited device memory on single GPU card. The accelerated program has fulfilled the speed-up ratio up to several hundreds, which dramatically decreases the simulation time and improve the research efficiency.

In general, the multi-GPU accelerated in-house LB solvers have been developed successfully to handle the problems in ocean engineering field in the present work.

However, there exist many unfinished works to be done in the future research.

- 1) The surface capturing algorithm will be developed to express complicated geometry of practical ocean structures;
- 2) More accurate free surface capturing algorithm will be applied in the present FSLB solver and the pressure boundary condition will be improved;
- 3) The adaptive mesh refinement technique and its application on multi-GPU will be developed based on the dynamic parallel technique and load balance. Meanwhile, the two-dimensional or three-dimensional decomposition strategy will be applied to improve the communication efficiency;

Chapter 7 Conclusions

- 4) The solver of water-air-structure based on the multiphase model of high density ratio will be developed;
- 5) More advanced turbulence model will be incorporated into the present LBE solver to handle the high Reynolds number issue.

Appendix A

From lattice Boltzmann equation to Navier-Stokes Equation

In this appendix, the derivation from discrete lattice Boltzmann equation at mesoscopic scale to the macroscopic fluid governing equations is elaborated (Chen & Doolen 1998, Yaling He et al.)

The multi-scale expansion technique, Chapman-Enskog expansion, which is proposed by Chapman & Enskog during 1910-1920, is applied to derive the Navier-Stokes equation from the lattice Boltzmann equation. In this multi-scale expansion technique, three time scales are used:

- 1) time scale for collision, ε^0 . ε^0 is an arbitrarily small quantity, which indicates the collision process is extremely rapid;
- 2) time scale for convection, ε^{-1} , which is slower than the collision process;
- 3) time scale for diffusion, ε^{-2} , which is slower than the convection process.

The lattice Boltzmann equation is expressed as

$$f_{\alpha}(\vec{x} + \vec{e}_{\alpha}\delta_t, t + \delta_t) - f_{\alpha}(\vec{x}, t) = -\frac{1}{\tau} [f_{\alpha}(\vec{x}, t) - f_{\alpha}^{(eq)}(\vec{x}, t)] \quad (\text{A-1})$$

In the multi-scaling expansion, three time scales of t^* (discrete), $t_1 = \varepsilon t^*$ and $t_2 = \varepsilon^2 t^*$ (continuous), two spatial scales of r^* (discrete) and $r_1 = \varepsilon r^*$ (continuous) are performed. Thus it has the following forms for time derivative and spatial derivative:

$$\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \quad (\text{A-2})$$

$$\frac{\partial}{\partial \vec{x}} = \varepsilon \frac{\partial}{\partial \vec{x}_1} \quad (\text{A-3})$$

Similarly, the distribution function can be also expanded with the multi-scaling technique

Appendix

$$f_\alpha = f_\alpha^{eq} + f_\alpha^{neq} = f_\alpha^{(0)} + \varepsilon f_\alpha^{(1)} + \varepsilon^2 f_\alpha^{(2)} + \dots = f_\alpha^{eq} + \varepsilon f_\alpha^{(1)} + \varepsilon^2 f_\alpha^{(2)} + \dots \quad (\text{A-4})$$

The definitions of macroscopic density and velocity are as followings

$$\rho = \sum_\alpha f_\alpha \quad (\text{A-5})$$

$$\vec{u} = \frac{1}{\rho} \sum_\alpha \vec{e}_\alpha f_\alpha \quad (\text{A-6})$$

The Maxwell equilibrium distribution function is applied to calculate the equilibrium distribution function

$$\begin{aligned} f_\alpha^{eq} &= \rho (2\pi RT)^{-D/2} \exp\left[-\frac{(\mathbf{e}_\alpha - \mathbf{u})^2}{2RT}\right] \\ &= \rho \omega_\alpha \left[1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{RT} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2R^2 T^2} - \frac{u^2}{2RT}\right] + O(u^3) \end{aligned} \quad (\text{A-7})$$

where $\omega_\alpha = (2\pi RT)^{-D/2} \exp(-\frac{e_\alpha^2}{2RT})$ is the weighting coefficients that vary with different discrete model.

Take the moment of the equilibrium distribution function (A-7), the following constraints can be obtained

$$\sum_\alpha f_\alpha^{(eq)} = \rho \quad (\text{A-8})$$

$$\sum_\alpha \vec{e}_\alpha f_\alpha^{(eq)} = \rho \vec{u} \quad (\text{A-9})$$

$$\Pi_{ij}^{(0)} = \sum_\alpha e_{\alpha i} e_{\alpha j} f_\alpha^{(eq)} = \rho u_i u_j + p \delta_{ij} \quad (\text{A-10})$$

$$\sum_\alpha e_{\alpha i} e_{\alpha j} e_{\alpha k} f_\alpha^{(eq)} = \rho c_s^2 (\delta_{ij} u_k + \delta_{ik} u_j + \delta_{jk} u_i) \quad (\text{A-11})$$

Considering the Equations (A-5), (A-6) and (A-8), (A-9), it has the following constraints for the non-equilibrium distribution function

$$\sum_\alpha f_\alpha^{(n)} = \sum_\alpha \vec{e}_\alpha f_\alpha^{(n)} = 0, \quad n = 1, 2, \dots \quad (\text{A-12})$$

If the Taylor expansion is conducted to the left side of Eq. (A-1), the following equation can be obtained

$$\delta_t \left(\frac{\partial}{\partial t} + \vec{e}_\alpha \cdot \nabla \right) f_\alpha + \frac{\delta_t^2}{2} \left(\frac{\partial}{\partial t} + \vec{e}_\alpha \cdot \nabla \right)^2 + \frac{1}{\tau} [f_\alpha - f_\alpha^{(eq)}] + O(\delta_t^3) = 0 \quad (\text{A-13})$$

Substitute Equations (A-2), (A-3), (A-4) into Eq. (A-13)

$$\varepsilon^0 : f_\alpha^{(0)} = f_\alpha^{(eq)} \quad (\text{A-14})$$

$$\varepsilon^1 : \left(\frac{\partial}{\partial t_1} + \vec{e}_\alpha \cdot \nabla_1 \right) f_\alpha^{(0)} + \frac{1}{\tau \delta_t} f_\alpha^{(1)} = 0 \quad (\text{A-15})$$

$$\varepsilon^2 : \frac{\partial f_\alpha^{(0)}}{\partial t_2} + \frac{\delta_t}{2} \left(\frac{\partial}{\partial t_1} + \vec{e}_\alpha \cdot \nabla_1 \right)^2 f_\alpha^{(0)} + \left(\frac{\partial}{\partial t_1} + \vec{e}_\alpha \cdot \nabla_1 \right) f_\alpha^{(1)} + \frac{1}{\tau \delta_t} f_\alpha^{(2)} = 0 \quad (\text{A-16})$$

From Eq. (A-15) it can be found that the non-equilibrium distribution function can be derived from the temporal and spatial derivatives of the equilibrium distribution function, which actually provides an approach to resolve the high-order part of the distribution function based on the low-order part.

Substitute (A-15) into (A-16), it has the simplified form

$$\frac{\partial f_\alpha^{(0)}}{\partial t_2} + \left(1 - \frac{1}{2\tau} \right) \left(\frac{\partial}{\partial t_1} + \vec{e}_\alpha \cdot \nabla_1 \right) f_\alpha^{(1)} + \frac{1}{\tau \delta_t} f_\alpha^{(2)} = 0 \quad (\text{A-17})$$

Take the zero-order momentum of Eq. (A-15) and Eq. (A-17), we can get the following two macroscopic equations at the t_1 and t_2 time scales

$$\frac{\partial \rho}{\partial t_1} + \nabla_1 \cdot (\rho \vec{u}) = 0 \quad (\text{A-18a})$$

$$\frac{\partial \rho}{\partial t_2} = 0 \quad (\text{A-18b})$$

Eq. (A-18a) indicates that the mass is conserved at convection level while the density is also implied to be unchanged by the diffusion effect.

Appendix

Take the first-order momentum of Eq. (A-15) and Eq. (A-17), we can get the following two macroscopic equations at the t_1 and t_2 time scales

$$\frac{\partial}{\partial t_1}(\rho u_j) + \frac{\partial}{\partial x_{1i}}(\rho u_i u_j) = -\frac{\partial p}{\partial x_{1j}} \quad (\text{A-19a})$$

$$\frac{\partial}{\partial t_2}(\rho u_j) + \left(1 - \frac{1}{2\tau}\right) \frac{\partial}{\partial x_{1i}} \left(\sum_{\alpha} e_{\alpha i} e_{\alpha j} f_{\alpha}^{(1)} \right) = 0 \quad (\text{A-19b})$$

From Eq. (A-19a) it can be found that the Euler equation can be derived at the convection time scale without the viscous dissipation term while the shear stress term can be obtained through the high-order momentum flux tensor as shown in Eq. (A-19b).

For the diffusion term in Eq. (A-19b), to make more specific derivation as followings

$$\begin{aligned} \Pi_{ij}^{(1)} &= \sum_{\alpha} e_{\alpha i} e_{\alpha j} f_{\alpha}^{(1)} \\ &= -\tau \delta_t \sum_{\alpha} e_{\alpha i} e_{\alpha j} \left(\frac{\partial}{\partial t_1} + \bar{e}_{\alpha} \cdot \nabla_1 \right) f_{\alpha}^{(eq)} \\ &= -\tau \delta_t \left[\frac{\partial}{\partial t_1} \Pi_{ij}^{(0)} + \frac{\partial}{\partial x_{1k}} \left(\sum_{\alpha} e_{\alpha i} e_{\alpha j} e_{\alpha k} f_{\alpha}^{(eq)} \right) \right] \\ &= -\tau \delta_t \left[\frac{\partial}{\partial t_1} (\rho u_i u_j + \rho c_s^2 \delta_{ij}) + \frac{\partial}{\partial x_{1k}} \left(\sum_{\alpha} e_{\alpha i} e_{\alpha j} e_{\alpha k} f_{\alpha}^{(eq)} \right) \right] \end{aligned} \quad (\text{A-20})$$

in which Eq. (A-15) is used to transform the high-order distribution function to the derivatives of the equilibrium distribution function. Also it can be found that the high-order momentum flux tensor could be transformed to be the summation of the time derivative of the low-order momentum flux tensor and the spatial derivative of the third-order moment of the equilibrium distribution function.

The first term of Eq. (A-20) can be divided as

$$\frac{\partial}{\partial t_1} (\rho u_i u_j + \rho c_s^2 \delta_{ij}) = \frac{\partial (\rho u_i u_j)}{\partial t_1} - c_s^2 \nabla_1 \cdot (\rho \bar{u}) \delta_{ij} \quad (\text{A-21})$$

Using the relationship of time derivative and spatial derivative (time changing rate of the momentum is identical to the spatial changing rate of the low-order momentum flux tensor) in

Eq. (A-19a), we can expand the first term of the above identity as

$$\begin{aligned}
 & \frac{\partial(\rho u_i u_j)}{\partial t_1} \\
 &= u_j \frac{\partial(\rho u_i)}{\partial t_1} + u_i \frac{\partial(\rho u_j)}{\partial t_1} - u_i u_j \frac{\partial \rho}{\partial t_1} \\
 &= u_j \left[-\frac{\partial}{\partial x_{1k}} (\rho u_i u_k + \rho c_s^2 \delta_{ik}) \right] + u_i \left[-\frac{\partial}{\partial x_{1k}} (\rho u_j u_k + \rho c_s^2 \delta_{jk}) \right] - u_i u_j \frac{\partial \rho}{\partial t_1} \\
 &= -u_j \frac{\partial(\rho u_i u_k)}{\partial x_{1k}} - c_s^2 u_j \frac{\partial \rho}{\partial x_{1i}} - u_i \frac{\partial(\rho u_j u_k)}{\partial x_{1k}} - c_s^2 u_i \frac{\partial \rho}{\partial x_{1j}} + u_i u_j \frac{\partial(\rho u_k)}{\partial x_{1j}} \\
 &= -c_s^2 u_j \frac{\partial \rho}{\partial x_{1i}} - c_s^2 u_i \frac{\partial \rho}{\partial x_{1j}} - \rho u_j u_k \frac{\partial u_i}{\partial x_{1k}} - u_i \frac{\partial(\rho u_j u_k)}{\partial x_{1j}} \\
 &= -c_s^2 u_j \frac{\partial \rho}{\partial x_{1i}} - c_s^2 u_i \frac{\partial \rho}{\partial x_{1j}} - \frac{\partial(\rho u_i u_j u_k)}{\partial x_{1k}}
 \end{aligned} \tag{A-22}$$

Using the constraint Eq. (A-11), the second term of Eq. (A-20) can be derived as

$$\begin{aligned}
 & \frac{\partial}{\partial x_{1k}} \left(\sum_{\alpha} e_{\alpha i} e_{\alpha j} e_{\alpha k} f_{\alpha}^{(eq)} \right) = \frac{\partial}{\partial x_{1k}} \left[\rho c_s^2 (\delta_{ij} u_k + \delta_{ik} u_j + \delta_{jk} u_i) \right] \\
 &= c_s^2 \nabla_1 \cdot (\rho \vec{u}) \delta_{ij} + \rho c_s^2 \frac{\partial u_j}{\partial x_{1i}} + c_s^2 u_j \frac{\partial \rho}{\partial x_{1i}} + \rho c_s^2 \frac{\partial u_i}{\partial x_{1j}} + c_s^2 u_i \frac{\partial \rho}{\partial x_{1j}}
 \end{aligned} \tag{A-23}$$

Combine Eq. (A-20)-(A-23), we can get the high-order momentum flux tensor as

$$\Pi_{ij}^{(1)} = \sum_{\alpha} e_{\alpha i} e_{\alpha j} f_{\alpha}^{(1)} = -\tau \delta_i \left[\rho c_s^2 \left(\frac{\partial u_j}{\partial x_{1i}} + \frac{\partial u_i}{\partial x_{1j}} \right) - \frac{\partial(\rho u_i u_j u_k)}{\partial x_{1k}} \right] \tag{A-24}$$

If the high-order velocity term is ignored in Eq. (A-24) and the discrete lattice effect is incorporated, then the high-order momentum flux tensor (diffusion term) can be expressed as

$$\begin{aligned}
 \Pi_{ij}^{(1)} &= (1 - \frac{1}{2\tau}) \sum_{\alpha} e_{\alpha i} e_{\alpha j} f_{\alpha}^{(1)} \\
 &= -(\tau - \frac{1}{2}) \delta_i c_s^2 \rho \left(\frac{\partial u_j}{\partial x_{1i}} + \frac{\partial u_i}{\partial x_{1j}} \right) = -\rho \nu \left(\frac{\partial u_j}{\partial x_{1i}} + \frac{\partial u_i}{\partial x_{1j}} \right)
 \end{aligned} \tag{A-25}$$

where the kinetical viscosity can be expressed as

Appendix

$$\nu = c_s^2(\tau - \frac{1}{2})\delta_t \quad (\text{A-26})$$

Combine Eq. (A-18a) and Eq. (A-18b) at t_1 and t_2 time scale, we can obtain the mass equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{u}) = 0 \quad (\text{A-27a})$$

Combine Eq. (A-19a), Eq. (A-19b) at t_1 and t_2 time scale and consider Eq. (A-25), we can get the momentum equation

$$\frac{\partial(\rho \bar{u})}{\partial t} + \nabla \cdot (\rho \bar{u} \bar{u}) = -\nabla p + \nabla \cdot [\rho \nu (\nabla \bar{u} + (\nabla \bar{u})^T)] \quad (\text{A-27b})$$

It is noteworthy to be addressed that we can get the macroscopic density, momentum, convection term, pressure term and dissipation term from moments of different order of the distribution function. To be detailed, density can be obtained from the zero-order moment of the equilibrium distribution function (A-8), momentum can be obtained from the first-order moment of the equilibrium distribution function (A-9), convection term and pressure term can be calculated by the second-order moment of the equilibrium distribution function (A-10), diffusion term can be obtained by the second-order moment of the non-equilibrium distribution function (A-25).

Appendix B:

Matrix M for D2Q9 MRT model

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix}$$

Matrix M for D3Q19 MRT model

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

Appendix C

Derivation of the Body Force

In the ocean engineering field, the treatment of gravity force is very common and of significance in many physical problems, such as water waves, bubble dynamics and so forth. In order to handle these problems, the force term in the lattice Boltzmann models must be treated appropriately to resolve the accurate hydrodynamics. In this appendix, the force term considering the discrete lattice effects designed by Guo et al. (2002) is elaborated.

The lattice Boltzmann equation without the force term can be written as

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau} [f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t)] \quad (\text{C-1})$$

The density and momentum can be evaluated by the distribution function as followings

$$\rho = \sum_i f_i, \rho \mathbf{u} = \sum_i \mathbf{e}_i f_i \quad (\text{C-2})$$

Taking the body force $\mathbf{F} = \rho \mathbf{g}$ into account, the lattice Boltzmann equation can be modified as the following form

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = -\frac{1}{\tau} [f_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t)] + \Delta t F_i \quad (\text{C-3})$$

where the equilibrium distribution function $f_i^{(eq)}(\mathbf{x}, t)$ is defined as

$$f_i^{(eq)} = E_i(\rho, \mathbf{u}^*) = \omega_i \rho \left[1 + \frac{\mathbf{e}_i \cdot \mathbf{u}^*}{c_s^2} + \frac{\mathbf{u}^* \mathbf{u}^* : (\mathbf{e}_i \mathbf{e}_i - c_s^2 \mathbf{I})}{2c_s^4} \right]$$

$$\text{with } \rho \mathbf{u}^* = \sum_i \mathbf{e}_i f_i + m \mathbf{F} \Delta t \quad (m \text{ is a constant to be determined}) \quad (\text{C-4})$$

The force term F_i can be constructed as the power series in the particle velocity

$$F_i = \omega_i \left[A + \frac{\mathbf{B} \cdot \mathbf{e}_i}{c_s^2} + \frac{\mathbf{C} : (\mathbf{e}_i \mathbf{e}_i - c_s^2 \mathbf{I})}{2c_s^4} \right] \quad (\text{C-5})$$

where $A, \mathbf{B}, \mathbf{C}$ are functions of $\mathbf{F}$, which are determined by the consistence of the moments of F_i and the hydrodynamic equations. The zeroth to second moments of F_i can be

obtained based on Eq. (C-5)

$$\sum_i F_i = A, \quad \sum_i \mathbf{e}_i F_i = \mathbf{B}, \quad \sum_i \mathbf{e}_i \mathbf{e}_i F_i = c_s^2 A \mathbf{I} + \frac{1}{2} [\mathbf{C} + \mathbf{C}^T] \quad (\text{C-6})$$

Similar to the multi-scaling expansion in Appendix B, we make some preparation for the recovery from lattice Boltzmann equation to Navier-Stokes equations

$$f_i = f_i^{(0)} + \varepsilon f_i^{(1)} + \varepsilon^2 f_i^{(2)} + \dots \quad (\text{C-7a})$$

$$\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2}, \quad \nabla = \varepsilon \nabla_1 \quad (\text{C-7b})$$

$$F = \varepsilon F_1, A = \varepsilon A_1, B = \varepsilon B_1, C = \varepsilon C_1 \quad (\text{C-7c})$$

Applying Taylor expansion to Eq. (C-3), we can get the following expression

$$D_i f_i + \frac{\Delta t}{2} D_i^2 f_i = -\frac{1}{\tau \Delta t} (f_i - f_i^{(eq)}) + F_{1i} \quad (\text{C-8})$$

where $D_i = \partial_t + c_{i\alpha} \partial_\alpha$

Substitute Eq. (C-7 a,b,c) into the above equation, we can get the following equations of different scales

$$\varepsilon^{(0)} : f_i^{(0)} = f_i^{(eq)} \quad (\text{C-9a})$$

$$\varepsilon^{(1)} : D_{1i} f_i^{(0)} = -\frac{1}{\tau \Delta t} f_i^{(1)} + F_{1i} \quad (\text{C-9b})$$

$$\varepsilon^{(2)} : \frac{\partial f_i^{(0)}}{\partial t_2} + D_{1i} f_i^{(1)} + \frac{\Delta t}{2} D_{1i}^2 f_i^{(0)} = -\frac{1}{\tau \Delta t} f_i^{(2)} \quad (\text{C-9c})$$

Substitute Eq. (C-9b) into Eq. (C-9c), a simplified form can be obtained

$$\varepsilon^{(2)} : \frac{\partial f_i^{(0)}}{\partial t_2} + (1 - \frac{1}{2\tau}) D_{1i} f_i^{(1)} = -\frac{1}{\tau \Delta t} f_i^{(2)} - \frac{\Delta t}{2} D_{1i} F_{1i} \quad (\text{C-9c})$$

where $D_{1i} = \partial / \partial t_1 + \mathbf{e}_i \cdot \nabla_1$.

Appendix

From (C-9a) and (C-4) we have the following relationship

$$\rho = \sum_i f_i = \sum_i f_i^{(0)} \quad (\text{C-10a})$$

$$\rho \mathbf{u}^* = \sum_i \mathbf{e}_i f_i + m \Delta t \mathbf{F} = \sum_i \mathbf{e}_i f_i^{(0)} \quad (\text{C-10b})$$

$$\sum_i f_i^{(k)} = 0, k > 0 \quad (\text{C-10c})$$

$$\sum_i \mathbf{e}_i f_i^{(1)} = -m \Delta t \mathbf{F}_1 \quad (\text{C-10d})$$

$$\sum_i \mathbf{e}_i f_i^{(k)} = 0, k > 1 \quad (\text{C-10e})$$

$$\Pi_{\alpha\beta}^{(0)} = \sum_i e_{i\alpha} e_{i\beta} f_i^{(0)} = \rho \mathbf{u}_\alpha^* \mathbf{u}_\beta^* + \rho c_s^2 \delta_{\alpha\beta} \quad (\text{C-10f})$$

Assume $\mathbf{B}_1 = n \mathbf{F}_1$. Take the zero-order and first-order moment of (C-9b), we can have

$$\frac{\partial \rho}{\partial t_1} + \nabla_1 \cdot (\rho \mathbf{u}^*) = A_1 \quad (\text{C-11a})$$

$$\frac{\partial \rho \mathbf{u}^*}{\partial t_1} + \nabla_1 \cdot \Pi^{(0)} = (n + \frac{m}{\tau}) \mathbf{F}_1 \quad (\text{C-11b})$$

To recover the Euler equation correctly, we must choose the following coefficients

$$A_1 = 0, \quad n + \frac{m}{\tau} = 1 \quad (\text{C-12})$$

To calculate the zero-order and first-order moments of Eq. (C-9c), we make some preparations. It is of significance to derive the first-order moment of the momentum flux.

From Eq. (C-9b), we can transform the non-equilibrium distribution function to the temporal and spatial derivatives of the equilibrium distribution function and a force term

$$f_i^{(1)} = -\tau \Delta t D_{1i} f_i^{(0)} + \tau \Delta t F_{1i} \quad (\text{C-13})$$

Thus the second-order moment of the non-equilibrium distribution function (high-order momentum flux tensor) can be expressed as

$$\Pi^{(1)} = \sum_i \mathbf{e}_i \mathbf{e}_i f_i^{(1)}$$

$$= -\tau\Delta t \left[\frac{\partial}{\partial t_1} \left(\sum_i \mathbf{e}_i \mathbf{e}_i f_i^{(0)} \right) + \nabla_1 \cdot \left(\sum_i \mathbf{e}_i \mathbf{e}_i \mathbf{e}_i f_i^{(0)} \right) \right] + \tau\Delta t \sum_i \mathbf{e}_i \mathbf{e}_i F_{1i} \quad (\text{C-14})$$

Give the constraints again

$$\sum_i \mathbf{e}_i \mathbf{e}_i f_i^{(0)} = \rho \mathbf{u}_i^* \mathbf{u}_j^* + \rho c_s^2 \delta_{ij} \quad (\text{C-15a})$$

$$\sum_i \mathbf{e}_i \mathbf{e}_i \mathbf{e}_i f_i^{(0)} = \rho c_s^2 (u_i^* \delta_{jk} + u_j^* \delta_{ik} + u_k^* \delta_{ij}) \quad (\text{C-15b})$$

$$\sum_i \mathbf{e}_i \mathbf{e}_i F_{1i} = \frac{1}{2} [C_{1ij} + C_{1ji}] \quad (\text{C-15c})$$

To be simplicity, we derive each term of Eq. (C-14) with Eq. (C-11 a, b) separately

$$\frac{\partial}{\partial t_1} \left(\sum_i \mathbf{e}_i \mathbf{e}_i f_i^{(0)} \right) = \frac{\partial}{\partial t_1} (\rho u_i^* u_j^* + \rho c_s^2 \delta_{ij}) = \frac{\partial \rho u_i^* u_j^*}{\partial t_1} - c_s^2 \nabla_1 \cdot (\rho \mathbf{u}^*) \delta_{ij} \quad (\text{C-16a})$$

And

$$\begin{aligned} \frac{\partial \rho u_i^* u_j^*}{\partial t_1} &= u_j^* \frac{\partial \rho u_i^*}{\partial t_1} + u_i^* \frac{\partial \rho u_j^*}{\partial t_1} - u_i^* u_j^* \frac{\partial \rho}{\partial t_1} \\ &= u_j^* \left[-\frac{\partial \rho u_i^* u_k^*}{\partial r_{1k}} - \frac{\partial \rho c_s^2}{\partial r_{1k}} \delta_{ki} + F_{1i} \right] + u_i^* \left[-\frac{\partial \rho u_j^* u_k^*}{\partial r_{1k}} - \frac{\partial \rho c_s^2}{\partial r_{1k}} \delta_{kj} + F_{1j} \right] + u_i^* u_j^* \frac{\partial \rho u_k^*}{\partial r_{1k}} \\ &= u_j^* F_{1i} + u_i^* F_{1j} - u_j^* c_s^2 \frac{\partial \rho}{\partial r_{1i}} - u_i^* c_s^2 \frac{\partial \rho}{\partial r_{1j}} - u_j^* \frac{\partial \rho u_i^* u_k^*}{\partial r_{1k}} - u_i^* \frac{\partial \rho u_j^* u_k^*}{\partial r_{1k}} + u_i^* u_j^* \frac{\partial \rho u_k^*}{\partial r_{1k}} \\ &= u_j^* F_{1i} + u_i^* F_{1j} - u_j^* c_s^2 \frac{\partial \rho}{\partial r_{1i}} - u_i^* c_s^2 \frac{\partial \rho}{\partial r_{1j}} - \frac{\partial \rho u_i^* u_j^* u_k^*}{\partial r_{1k}} \end{aligned} \quad (\text{C-16b})$$

Thus we can have the temporal derivative of the zero-order momentum flux tensor

$$\begin{aligned} \frac{\partial}{\partial t_1} \left(\sum_i \mathbf{e}_i \mathbf{e}_i f_i^{(0)} \right) &= \\ &= u_j^* F_{1i} + u_i^* F_{1j} - u_j^* c_s^2 \frac{\partial \rho}{\partial r_{1i}} - u_i^* c_s^2 \frac{\partial \rho}{\partial r_{1j}} - c_s^2 \nabla_1 \cdot (\rho \mathbf{u}^*) \delta_{ij} - \frac{\partial \rho u_i^* u_j^* u_k^*}{\partial r_{1k}} \end{aligned} \quad (\text{C-16c})$$

Appendix

The second term of Eq. (C-14) can be derived as

$$\begin{aligned}
 \nabla_1 \cdot (\sum_i \mathbf{e}_i \mathbf{e}_i f_i^{(0)}) &= \\
 &= \frac{\partial}{\partial r_{1k}} [\rho c_s^2 (u_i^* \delta_{jk} + u_j^* \delta_{ik} + u_k^* \delta_{ij})] \\
 &= \frac{\partial \rho u_k^*}{\partial r_{1k}} c_s^2 \delta_{ij} + c_s^2 \frac{\partial \rho u_i^*}{\partial r_{1j}} + c_s^2 \frac{\partial \rho u_j^*}{\partial r_{1i}} \\
 &= c_s^2 \delta_{ij} \nabla_1 \cdot (\rho \mathbf{u}^*) + \rho c_s^2 \frac{\partial u_i^*}{\partial r_{1j}} + u_i^* c_s^2 \frac{\partial \rho}{\partial r_{1j}} + \rho c_s^2 \frac{\partial u_j^*}{\partial r_{1i}} + u_j^* c_s^2 \frac{\partial \rho}{\partial r_{1i}} \quad (\text{C-16d})
 \end{aligned}$$

Combine Eq. (C-14), Eq. (C-15c) and Eq. (C-16 a, b, c, d), we can obtain the first-order momentum flux tensor

$$\begin{aligned}
 \Pi^{(1)} &= -\tau \Delta t [(u_j^* F_{1i} + u_i^* F_{1j}) + \rho c_s^2 (\frac{\partial u_i^*}{\partial r_{1j}} + \frac{\partial u_j^*}{\partial r_{1i}}) - \frac{\partial \rho u_i^* u_j^* u_k^*}{\partial r_{1k}}] + \frac{\tau \Delta t}{2} [C_{1ij} + C_{1ji}] \\
 &\approx -\tau \Delta t [(u_j^* F_{1i} + u_i^* F_{1j}) + \rho c_s^2 (\frac{\partial u_i^*}{\partial r_{1j}} + \frac{\partial u_j^*}{\partial r_{1i}})] + \frac{\tau \Delta t}{2} [C_{1ij} + C_{1ji}] \quad (\text{C-17})
 \end{aligned}$$

where the third-order velocity term $O(u^3)$ and higher order term are ignored. If we take $\mathbf{C} = 2\mathbf{u}\mathbf{F}_1$ or $\mathbf{C} = \mathbf{u}\mathbf{F}_1 + \mathbf{F}_1\mathbf{u}$, then additional terms related to force could be canceled and the first-order momentum flux tensor reduces to the viscous stress term in the Navier-Stokes equation $-\Pi_{ij}^{(1)} = \nu [\rho \nabla_{1i} \mathbf{u}_j^* + \rho \nabla_{1j} \mathbf{u}_i^*]$, where the kinetical viscosity is given as $\nu = c_s^2 \tau \Delta t$, which is the same as that in the continuous Boltzmann equation.

Now take the zero-order and first-order moments of Eq. (C-9c), we can have

$$\frac{\partial \rho}{\partial t_2} = \Delta t (m - \frac{1}{2}) \nabla_1 \cdot \mathbf{F}_1 \quad (\text{C-18a})$$

$$\frac{\partial \rho \mathbf{u}^*}{\partial t_2} = \Delta t (m - \frac{1}{2}) \frac{\partial \mathbf{F}}{\partial t_1} + \nabla_1 \cdot \boldsymbol{\sigma}_1 \quad (\text{C-18b})$$

where the stress tensor is given as

$$\begin{aligned}
 \sigma_1 &= -(1 - \frac{1}{2\tau}) \Pi_{ij}^{(1)} - \frac{\Delta t}{4} (C_{1ij} + C_{1ji}) \\
 &= (\tau - \frac{1}{2}) c_s^2 \Delta t \rho (\nabla_{1i} \mathbf{u}_j^* + \nabla_{1j} \mathbf{u}_i^*) + \Delta t [(\tau - \frac{1}{2}) (\mathbf{u}_j^* \mathbf{F}_{1i} + \mathbf{u}_i^* \mathbf{F}_{1j}) - \frac{\tau}{2} (C_{1ij} + C_{1ji})]
 \end{aligned} \tag{C-19}$$

Obviously it can be found that there exist extra terms related to the body force. Similar to the treatment in Appendix B, the lattice discrete effect can be incorporated into the viscous term as

$$\nu = c_s^2 (\tau - \frac{1}{2}) \Delta t \tag{C-20}$$

In terms of the additional terms related to the body force, we can construct suitable $\mathbf{C}$ to eliminate these unexpected terms

$$\mathbf{C} = (1 - \frac{1}{2\tau}) 2\mathbf{u}^* \mathbf{F} \tag{C-21}$$

From Eq. (C-18 a, b) it can be seen that the temporal and spatial derivatives of the body force could affect the density and momentum, which violates the conservation of mass and momentum. To eliminate this incorrectness, we can choose the coefficient as

$$m = \frac{1}{2} \quad \text{and} \quad \rho \mathbf{u}^* = \sum_i \mathbf{e}_i f_i + \frac{\Delta t}{2} \mathbf{F} \tag{C-22}$$

Thus another coefficient can be determined considering Eq. (C-12)

$$n = 1 - \frac{1}{2\tau} \tag{C-23}$$

Combine the coefficients resolved in Eq. (C-12), Eq. (C-21) and Eq. (C-23), the force term constructed in Eq. (C-5) can be finally obtained as

$$F_i = (1 - \frac{1}{2\tau}) \omega_i [\frac{\mathbf{e}_i - \mathbf{v}}{c_s^2} + \frac{\mathbf{e}_i \cdot \mathbf{v}}{c_s^4} \mathbf{e}_i] \cdot \mathbf{F} \tag{C-24}$$

where the flow velocity is defined as

Appendix

$$\rho \mathbf{v} = \sum_i \mathbf{e}_i f_i + \frac{\Delta t}{2} \mathbf{F} \quad (\text{C-25})$$

Combining the results in Eq. (C-11) and Eq. (C-18) at two time scales, we can obtain the final macroscopic equations with body force

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0 \quad (\text{C-26a})$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla p + \nu \nabla \cdot [\rho (\nabla \mathbf{v} + (\nabla \mathbf{v})^T)] + \mathbf{F} \quad (\text{C-26b})$$

It is noteworthy to be addressed that the correct fluid velocity counting for the body force must be defined to get the correct continuity equation and the effect of the body force on the viscous stress should be eliminated to obtain the correct momentum equation

Bibliography

A. Caiazzo . Analysis of lattice Boltzmann nodes initialization in moving boundary problems. Progress in Computational Fluid Dynamics, 2008; 8:1-4.

A. Caiazzo and M. Junk. Boundary forces in lattice Boltzmann: Analysis of Momentum Exchange algorithm. Computers and Mathematics with Applications, 2008; 55: 1415-1423.

A. J. C. Ladd¹ and R. Verberg, Lattice-Boltzmann Simulations of Particle-Fluid Suspensions, Journal of Statistical Physics, Vol. 104, Nos. 5/6, (2001);

A. Tiwari, S. P. Vanka. A ghost fluid Lattice Boltzmann method for complex geometries. Int. J. Numer. Meth. Fluids 2012; 69:481-498.

A.K. Gunstensen, D.H. Rothman, S. Zaleski, G. Zanetti, Lattice Boltzmann model of immiscible fluids, Phys. Rev. A 43 (1991) 4320–4327;

Abbas Fakhari, Martin Geier, Taehun Lee, A mass-conserving lattice Boltzmann method with dynamic grid refinement for immiscible two-phase flows, Journal of Computational Physics 315(2016), 434-457;

Andreas Kolke, Modellierung und Diskretisierung bewegter Diskontinuitäten in randgekoppelten Mehrfeldsystemen. PhD thesis, Technische Universität Braunschweig, 2005;

Anthony J. C. Ladd. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. Journal of Fluid Mechanics 1994; 271:285- 309.

Anthony J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results. Journal of Fluid Mechanics 1994; 271:311- 339.

B. Wen et al., Lattice-type-dependent momentum-exchange method for moving boundaries. Phys Rev E 2012; 85: 016704.

B. Wen et al., Galilean invariant fluid–solid interfacial dynamics in lattice Boltzmann simulations, J Comput Phys 2014; 266:161–170.

Bibliography

Chen & Doolen, Lattice Boltzmann method for fluid flows, *Annual Review Fluid Mechanics*, 1998, 30:329-364.

C. Janssen et al. (2010) Hybrid volume-of-fluid-based algorithm for the simulation of free surface flow with non-uniform block-structured grids, *Computers and Mathematics with Applications* 59(2010) 2215-2235.

C. Janssen et. al. (2013) Enhanced non-linear free surface flow simulations with hybrid LBM-VOF model, *Computers and Mathematics with Applications* 65(2013) 211-229.

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

C.G. Koh et al., Modelling of liquid sloshing with constrained floating baffle, *Computers and Structures* 122 (2013) 270–279;

Changhong Hu & Masashi Kashiwagi, A CIP-based method for numerical simulations of violent free-surface flows, *Journal of Marine Science and Technology* (2004),9: 143-157;

Changhong Hu & Masashi Kashiwagi, Two-dimensional numerical simulation and experiment on strongly nonlinear wave–body interactions (2009), 14:200–213;

Changhong Hu et al., Application of CIP Method for Strongly Nonlinear Marine Hydrodynamics, *Ship Technology Research* (2006), 53:2, 74-87;

Chen S Y, Martinez D, Mei R W. On boundary conditions in lattice Boltzmann methods, *Physics of Fluids*, 1996, 8(9):2257-2536;

Cheng Liu and C.H. Hu. An efficient immersed boundary treatment for complex moving object. *J Comput Phys* 2014; 274:654–680.

Christian Janßen & Manfred Krafczyk, Free surface flow simulations on GPGPUs using the LBM, *Computers and Mathematics with Applications* 61 (2011) 3549–3563;

Christian Obrecht et al., A new approach to the lattice Boltzmann method for graphics processing units, *Computers and Mathematics with Applications* 61 (2011): 3628–3638;

Cyrus K. Aidun *et.al.* Lattice Boltzmann method for complex flows. Annual Review of Fluid Mechanics 2010; 42:439-472.

Cyrus K. Aidun, Yannan Lu and E.-Jiang Ding. Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation. Journal of Fluid Mechanics 1998; 373:287- 311.

D. BHAGAT AND M. E. WEBER, Bubbles in viscous liquids: shapes, wakes and velocities, J. Fluid Mech. (1981), vol. 105, pp. 61-85;

D. d'Humieres, In Rarefied Gas Dynamics: Theory and Simulations, Vol. 159 of Progress in Astronautics and Aeronautics, edited by B. D. Shizgal and D. P. Weave (AIAA, Washington, DC, 1992), pp. 450–458;

D. Korner *et.al.* ,Lattice Boltzmann model for free surface flow for modeling foaming, Journal of statistical physics, 2005.

d'Humieres.D, Ginzburg.I, Krafczyk.M, Lallemand.P, Luo.L, Multiple-relaxation-time lattice Boltzmann models in three dimensions. Philosophical Transactions of the Royal Society A, 360(2002):437–451;

Dupuis A, Chatelain P, Koumoutsakos P. An immersed boundary-lattice Boltzmann method for the simulation of the flow past an impulsively started cylinder. J Comput Phys 2008; 227:4486–4498.

Eric Lorenz, Alfonso Caiazzo, and Alfons G. Hoekstra. Corrected momentum exchange method for lattice Boltzmann simulations of suspension flow. Phys Rev E 2009;**79**:036705.

Feng ZG, Michaelides EE. Proteus: a direct forcing method in the simulation of particulate flows. J Comput Phys 2005;202:20–51.

Feng Z-G, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. J Comput Phys 2004;195:602–628.

F.J. Higuera, S. Succi, R. Benzi, Lattice gas dynamics with enhanced collisions, EPL (Europhysics Letters), 9(4):345, 1989.

G. Le, JF Zhang. Boundary slip from the immersed boundary lattice Boltzmann models. Phys Rev E 2009;79:026701.

Bibliography

G. Thömmes et al., A lattice Boltzmann method for immiscible multiphase flow simulations using the level set method, *Journal of Computational Physics* 228 (2009) 1139–1156;

G.F.Zhou,L.M.Wang,X.W.Wang,W.Ge. Galilean-invariant algorithm coupling immersed moving boundary conditions and Lees-Edwards boundary conditions. *Phys Rev E* 2011;84:066701.

Guo Z L, Zheng C G, Shi B C, Non-equilibrium extrapolation method for velocity and boundary conditions in the lattice Boltzmann method, *Chinese Physics*,2002,11(4):0366-0374;

H.H. Hu, D.D. Joseph, Direct simulation of fluid particle motions, *Theoretical and Computational Fluid Dynamics* 1992, 3:285:306

Harold Grad. On the kinetic theory of rarefied gases. *Communications on Pure and Applied Mathematics* 1949; 2:331-407.

<https://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf>;

Hu, C.H., Sueyoshi, M., 2010, Numerical simulation and experiment on dam break problem. *Journal of Marine Science and Application*, 9, 109-114;

Huabing Li et.al.. Force evaluations in lattice Boltzmann simulations with moving boundaries in two dimensions. *Phys. Rev. E* 2004,70:026701.

I. Ginzburg and D. d’Humières. Multireflection boundary conditions for lattice Boltzmann models. *Phys. Rev. E* 2003,68:066614.

Irina Ginzburg, Frederik Verhaeghe, and d’Humières.D, Two-Relaxation-Time Lattice Boltzmann Scheme: About Parametrization, Velocity, Pressure and Mixed Boundary Conditions. *Communications in computational physics*, 3:427–478, 2008;

Irina Ginzburg, Konrad Steiner, Lattice Boltzmann model for free-surface flow and its application to filling process in casting. *Journal of computational physics* 185(2003) 61-99.

J. Feng, H.H. Hu, D.D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid, Part 2 Couette and Poiseuille Flows, *Journal of Fluid*

Mechanics 1994, 277:271-301.

J. Lu, H. Han, B. Shi, Z. Guo. Immersed boundary lattice Boltzmann model based on multiple relaxation times. Phys Rev E 2012;85:016711.

J. M. Buick and C. A. Greated, Gravity in a lattice Boltzmann model, PHYSICAL REVIEW E, VOLUME 61, NUMBER 5 (2000);

J.R.Clausen,C.K.Aidun. Galilean invariance in the lattice Boltzmann method and its effect on the calculation of rheological properties in suspensions. Int.J.Multiph.Flow 2009;35:307–311.

Jinsong Hua, Jing Lou, Numerical simulation of bubble rising in viscous liquid, Journal of Computational Physics 222 (2007) 769–795;

Jonas Latt, Choice of units in lattice Boltzmann simulations, LBMethod.org documentation project (2008);

Jonas Tölke, Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA, Computing and Visualization in Science (2008), 13-1:29-39;

Koshizuka S. & Oka Y., Moving particle semi-implicit method for fragmentation of incompressible fluid, Nuclear Science and Engineering (1996), 123:421-434;

L. Chen, Y. Yu, G. Hou. Sharp-interface immersed boundary lattice Boltzmann method with reduced spurious-pressure oscillations for moving boundaries. Phys Rev E 2013; 87:053306.

L.S. Luo, Unified theory of the lattice Boltzmann models for nonideal gases, Phys.Rev.Lett. 81(1998), 1618-1621;

Latva-Kokko M & Rothman D, Static contact angle in lattice Boltzmann models of immiscible fluids. Physical Review E (2005) 72(4), 046701;

Li-Shi Luo et al., Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations, PHYSICAL REVIEW E 83, 056710 (2011);

Luca Scarbolo et al., Unified framework for a side-by-side comparison of different multicomponent algorithms: Lattice Boltzmann vs. phase field model, Journal of

Bibliography

Computational Physics 234 (2013) 263–279;

Lucy L.B., A numerical approach to the testing of the fission hypothesis (1977), *Astron. J.*, 82:1013-1024;

M. Bouzidi *et.al.* Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids* 2001; 13:3452-3459.

M. Schafer et al., Benchmark computations of laminar flow around a cylinder, *Flow Simulations with High-Performance Computers II* (1996), 48:547-566

M. Swift, E. Orlandini, W.R. Osborn, J.M. Yeomans, Lattice Boltzmann simulations of liquid–gas and binary fluid systems, *Phys. Rev. E* 54 (1996) 5041–5052.

Manfred Krafczyk, Lishi Luo, Large-eddy simulations with a Multiple-Relaxation-Time LBE model, *International Journal of Modern Physics B*, Volume 17, Issue 01n02, 20 January 2003;

Martin Geier, Abbas Fakhari and Taehun Lee, Conservative phase-field lattice Boltzmann model for interface tracking equation, *PHYSICAL REVIEW E* 91, 063309 (2015);

Mei RW, Luo LS, Shyy W. An accurate curved boundary treatment in the lattice Boltzmann method. *J Comput Phys* 1999; 155:307-333.

MIAO, G. , Hydrodynamic forces and dynamic responses of circular cylinders in wave zones. Doctor thesis, Dept. of Marine Hydrodynamics, NTH, Trondheim, Norway (1989);

Ming Cheng, J. Lou, L.S. Luo, Numerical study of a vortex ring impacting a flat wall, *Journal of Fluid Mechanics*, 660,430 (2010);

Ming Cheng, Jinsong Hua, Jing Lou, Simulation of bubble–bubble interaction using a lattice Boltzmann method, *Computers & Fluids* 39 (2010) 260–270;

N. Thurey (2007) Physically based animation of LBM free surface flows with the lattice Boltzmann method, PhD thesis, 2007.

N. Thurey, Optimization and stabilization of LBM free surface flow simulations using adaptive parameterization, 2005.

N. Thurey, U.Rude, Free surface lattice-Boltzmann fluid simulations with and without level

sets, 2004.

Naoyuki Onodera and Takayuki Aoki: Peta-scale large-eddy simulation using lattice Boltzmann method on the TSUBAME supercomputer, KSME-JSME Joint Symposium on Computational Mechanics & CAE 2014 at JEJU, JEJU, Korea, May 1, 2014;

Naoyuki Onodera et al., GPU implementation of lattice Boltzmann method for free-surface flow, 29th Computational Fluid Dynamics Symposium in Japan (2015), B10-2;

Naoyuki Onodera, Takayuki Aoki, Takashi Shimokawabe, Tatsuji Miyashita, and Hiromichi Kobayashi, Large-Eddy Simulation of Fluid-Structure Interaction using Lattice Boltzmann Method on Multi-GPU Clusters, Proceedings of the 5th Asia Pacific Congress on Computational Mechanics and 4th International Symposium on Computational, Singapore, Dec.11,2013;

Naoyuki Onodera, Takayuki Aoki: Large-eddy simulation of turbulent flow around a car body using lattice Boltzmann method on the TSUBAME supercomputer, 11th. World Congress on Computational Mechanics (WCCM XI), Barcelona, Spain, July 20-25, 2014;

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

Noble D, Torczynski J.. A lattice Boltzmann method for partially saturated cells. International Journal of Modern Physics C 1998;9:1189-1201.

Nvidia Toolkit CUDA v7.0;

O. Filippova and D. Hanel. Grid Refinement for Lattice-BGK Models. J Comput Phys 1998;147:219-228.

P. Lallemand et.al. (2007) A lattice Boltzmann front-tracking method for interface dynamics with surface tension in two dimensions, Journal of computational physics ,226(2007) 1367-1384

P. Lallemand, Luo LS. Lattice Boltzmann method for moving boundaries. J Comput Phys 2003;184:406-421.

Bibliography

- P.A. Skordos, Initial and Boundary Conditions for the Lattice Boltzmann Method, *Phys. Rev. E* 48, 4823-4842, 1993;
- Pao-Hsiung Chiu, Yan-Ting Lin, A conservative phase field method for solving incompressible two-phase flows, *Journal of Computational Physics* 230 (2011) 185–204;
- Peskin CS. Flow patterns around heart valves: a numerical method. *J Comput Phys* 1972; 10:251–271.
- Peter Bailey et al., *International Conference on Parallel Processing* (2009), 550-557;
- Pierre Lallemand and L.S.Luo. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys Rev E* 2000; Vol 61, NUMBER 6.
- Pierre Lallemand and Li-Shi Luo, Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability, *Phys. Rev. E* 61, 6546 (2000)
- Qisu Zou & Xiaoyi He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model, *Physics of Fluids* 9, 1591 (1997);
- R. Eymard, R. Herbin, The finite volume method, *Handbook of Numerical Analysis* (2000), Vol. VII:713–1020;
- R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous, A pseudocompressibility method for the numerical simulation of incompressible multifluid flows, *J. Comput. Phys.* 30 (2004) 901–937;
- Rainald Löhner et al., An unstructured grid-based, parallel free surface solver (1999), *Applied Numerical Mathematics*, 31-3:271-293;
- Renwei Mei, Dazhi Yu, Wei Shyy, and Li-Shi Luo. Force evaluation in the lattice Boltzmann method involving curved geometry. *Phys. Rev. E* 2002,65:041203.
- Renwei Mei, Li-Shi Luo, Pierre Lallemand, d’Humières.D, Consistent initial conditions for lattice Boltzmann simulations, *Computers & Fluids* 35 (2006) 855–862;
- S. Krithivasan et.al.. Diffused bounce-back condition and refill algorithm for the lattice Boltzmann method. *Phys Rev E* 2014; 89:033313
- S.K. Kang, Y.A. Hassan. A comparative study of direct-forcing immersed boundary-lattice

Boltzmann method for stationary complex boundaries. *Int. J. Numer. Meth. Fluids* 2011; 66:1132-1158.

Samuel M. Allen, John W. Cahn, Mechanisms of phase transformations within the miscibility gap of Fe-rich Fe-Al alloys, *Acta Metallurgica*, Volume 24, Issue 5, Pages 425-437 (1976);

Shiyi Chen and Gary D. Doolen, Lattice Boltzmann method for fluid flows, *Annual Review of Fluid Mechanics* 1998;30: 329-364.

Simon Bogner, Ulrich Rüde, Simulation of floating bodies with the lattice Boltzmann method, *Computers and Mathematics with Applications* 65 (2013):901–913;

T. Inamuro et al., A lattice Boltzmann method for incompressible two-phase flows with large density differences, *Journal of Computational Physics* 198 (2004) 628 – 644;

T. Lee, L. Liu, Lattice Boltzmann simulations of micron-scale drop impact on dry surfaces, *J. Comput. Phys.* 229 (2010) 8045–8063;

T. Seta, R. Rojas, K. Hayashi, A. Tomiyama. Implicit-correction-based immersed boundary-lattice Boltzmann method with two relaxation times. *Phys Rev E* 2014; 89:023307.

Taehun Lee & Ching-Long Lin, Pressure evolution lattice-Boltzmann-equation method for two-phase flow with phase change, *PHYSICAL REVIEW E* 67(2003), 056703;

Taehun Lee and Paul F. Fischer, Eliminating parasitic currents in the lattice Boltzmann equation method for nonideal gases, *PHYSICAL REVIEW E* 74, 046709 2006;

Takaji Inamuro et al., Flow between parallel walls containing the lines of neutrally buoyant circular cylinders, *International Journal of Multiphase Flow* 2000; 26:1981-2004.

Thomas Pohl (2007) High performance simulation of LBM free surface flows using the lattice Boltzmann method, PhD thesis, 2007.

WAGNER, H., Über Stoss- und Gleitvorgänge an der Oberfläche von Flüssigkeiten. *Zeitschr. f. Angewandte Mathematik und Mechanik*, 12, 4, 193-235 (1932);

Wang Xian & Aoki Takayuki, Multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster, *Parallel Computing* 37 (2011) 521–535;

Bibliography

Victor W Lee et al., Debunking the 100X GPU vs. CPU Myth: An Evaluation of Throughput Computing on CPU and GPU, ISCA'10, June 19–23, 2010;

Wu J, Shu C. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *J Comput Phys* 2009; 228:1963–1979.

X. He, S. Chen, R. Zhang, A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulations of Rayleigh–Taylor instability, *J. Comput. Phys.* 152 (1999) 652–663;

X. Shan, H. Chen, Lattice Boltzmann model for simulating flows with multiple phases and components, *Phys. Rev. E* 47 (1993) 1815–1819;

X.He, X.Shan, G.D.Doolen, Discrete Boltzmann equation model for nonideal gases, *Phys.Rev.E*, 57(1998),13-16;

Xiaoyi He & Li-Shi Luo, Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation, *PHYSICAL REVIEW E*, VOLUME 56, NUMBER 6, 1997;

Xiaoyi He, Xiaowen Shan, Gary D. Doolen, Discrete Boltzmann equation model for nonideal gases, *PHYSICAL REVIEW E*, VOLUME 57, NUMBER 1 (1998);

Xinying Zhu, Application of the CIP method to strongly nonlinear wave-body interaction problems, Doctor thesis, Department of Marine Technology, Norwegian University of Science and Technology (2006);

Xuhui Li, Fei Jiang, Changhong Hu, Analysis of the accuracy and pressure oscillation of the lattice Boltzmann method for fluid-solid interactions, *Computers and Fluids* 129 (2016), 33-52;

Yaling He, Yong Wang, Qing Li, Lattice Boltzmann method: Theory and Applications, 2008, Peking, Science Press.

Y. Chen et.al. Momentum-exchange method in lattice Boltzmann simulations of particle-fluid interactions. *Phys Rev E* 2013;88:013303.

Y. Sun, C. Beckermann, Sharp interface tracking using the phase-field equation, *Journal of*

Computational Physics 220 (2007) 626–653;

Y. Wang, C. Shu, H.B. Huang, C.J. Teo, Multiphase lattice Boltzmann flux solver for incompressible multiphase flows with large density ratio, *Journal of Computational Physics* 280 (2015) 404–423;

Y.H. Tseng, J.H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry, *J Comput Phys* 2003;192:593–623.

Yan P., L.S. Luo. A comparative study of immersed-boundary and interpolated bounce-back methods in LBE. *Progress in Computational Fluid Dynamics* 2008;8:1–4.

Z. Guo et.al.. An extrapolation method for boundary conditions in lattice Boltzmann method. *Physics of Fluids* 2002; 14:6.

Z.L. Guo et.al.. Non-equilibrium extrapolation method for velocity and boundary conditions in the lattice Boltzmann method. *Chinese Physics* 2002;11(4): 0366-0374.

Zhaoli Guo, Chuguang Zheng and Baochang Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, *PHYSICAL REVIEW E*, VOLUME 65, 046308 (2002);

Zhaoli Guo, Chuguang Zheng, Baochang Shi, Force imbalance in lattice Boltzmann equation for two-phase flows, *PHYSICAL REVIEW E* 83, 036707 (2011);

Zhuangming Zhao et al., *International Journal for Numerical Methods in Fluids*, 71(2013):223–248.