An Efficient Computation for Tsunami Modelling by the Lattice Boltzmann Method

Doctoral Thesis

Thesis Title

An Efficient Computation for Tsunami Modelling

by the Lattice Boltzmann Method

Department of Civil Engineering
Graduate School of Engineering,
TOHOKU UNIVERSITY

KENTA SATO

(ID No. B6TD6005)

Acknowledgements

I would like to express my sincere gratitude to Professor Shunichi Koshimura for his guidance and the genuine concern. Thanks are also extended to Dr. Christian F. Janßen, Hamburg University of Technology, for his valuable comments.

Once again, grateful acknowledgements are due to Professor Shunichi Koshimura for his financial assistance to accomplish this goal. This study was partly supported by Japan Society for the Promotion of Science.

Abstract

Free surface flow problems occur in numerous disaster simulations, such as tsunamis inland penetration in urban area. Simulation models for these problems have to be non-hydrostatic, three-dimensional and highly resolved because of the strong non-linearity and higher-order physical phenomena. Despite all the progress in the modern computational fluid dynamics, such simulations still present formidable challenges both from numerical and computational cost point of view due to the pressure Poisson equation in the incompressible Navier-Stokes equations' fluid modellings.

Emerging as a powerful numerical technique in recent years, the lattice Boltzmann method exhibits unique numerical and computational features in specific problems for its ability to simulate large scale flow fields as well as its inherent advantage to deliver favourable computational efficiencies on massively parallel processors such as multi-core processors or graphical processing units. The lattice Boltzmann method is one of computational fluid dynamics solvers. It starts with a fully discrete model rather than discretising a set of particle differential equations and solving them, directly. In this regard, the lattice Boltzmann method is suitable for a large-scaled fluid simulation and gains substantial expectation as an efficient three-dimensional tsunami simulation approach.

In this thesis, two classes of the lattice Boltzmann method for tsunami flow simulations are developed which are coupled with the piecewise linear interface calculation with the Volume-of-Fluid technique and based on the non-linear shallow water theory. The first model is for an efficient three-dimensional tsunami simulation by a one-fluid formulation, where the lattice Boltzmann equation is assigned to solve for a single virtual fluid and the gas-liquid interface is captured through convection of fluid fraction values by solving the advection equation with the finite volume method.

The most famous model generated with the lattice Boltzmann method is based on the Bhatnagar-Gross-Krook collision model, using the single relaxation time coefficient. This approach, however, has obvious defects; for example, the Prandtl number is fixed for its simplicity. The collision model additionally tends to encounter stability problems in high-Reynolds number flows. An advanced multiple-relaxation-time collision model has been developed simultaneously with the Bhatnagar-Gross-Krook model. One attractive advantage of the model is the improvement in the numerical stability in high-Reynolds number flows. The multiple-relaxation-time model, thus, can be seen as an alternative approach to overcome the defects of the Bhatnagar-Gross-Krook model.

Researchers have developed an epoch-making fluid model and confirmed its accuracy by the multiple-relaxation-time model; however, few studies have validated the collision model in three-dimensional free surface flows, especially in spacing fluid density. It is well known that pseudo-compressibility appears in the lattice Boltzmann method because of an explicit method, which means the lattice Boltzmann method does not assume the sound speed to be infinity, and the compressibility has to be controlled and reduced by the calculation parameters to simulate incompressible flow fields as accurately as possible in coastal engineering. In this thesis, some benchmark problems were carried out with the multiple-relaxation-time model to conduct a usability assessment of proposed free surface model and to parameterise the model for the compressibility to achieve the validation of the lattice Boltzmann method as a three-dimensional tsunami simulation method.

The second model provides a hydrostatic two-dimensional tsunami simulation based on the non-linear shallow water theory. It is well known that this approach gives a good approximation of large-scaled tsunami simulation within a hydrostatic pressure can be assumed. Various benchmark problems are also carried out to validate the utility of the proposed models in term of turbulent flow problems. We concretely applied the two-dimensional model for the tsunami by the 2011 great east Japan earthquake. To carry out the simulation, we also added a moving shoreline boundary condition based on simple linear extrapolation approach and modified coupling technique for multiple scale grid analysis. The proposed two-dimensional tsunami simulation model by the lattice Boltzmann method was validated by comparison with the conventional finite difference method, which is solving the non-linear shallow water equations, directly.

A hybrid technique for coupling two and three-dimensional fluid models, moreover, is developed to achieve a consistent tsunami simulation from its occurrence to inland penetration. Considering tsunami simulation, it is necessary to correspond the flow situation, in which a large amount of fluid flows into three-dimensional computing area. In addition to the requirement, the coupling algorithm between two and three-dimensional fluid model has to be taken the compressibility effects. In this thesis, a coupling algorithm was proposed considering the compressibility and applied the scheme in the benchmark problem.

From the theoretical point of view, we have successfully developed a general free surface fluid model, which is expected to offer a more flexible and highly resolved tool in the investigation of fluid structure, in details. The main practical contribution offered by this study is to enable large-scaled tsunami simulation more efficiently. It allows to understand the interaction behaviour such as fluid transport in soils coupling with the discrete element method, which can be considered as starting points for future researches.

Contents

Acknowledgements **ii**

Abstract **iii**

1 Introduction **1**

1.1 Motivation 1

1.2 Contributions of this thesis 2

1.3 Outline 2

References **3**

2 Lattice Boltzmann method **4**

2.1 Introduction 4

2.2 Statistical mechanics and Boltzmann equation 4

2.3 From the Boltzmann-BGK equation to the Navier-Stokes equations 7

2.4 The Chapman-Enskog expansion 10

2.4.1 The lattice Boltzmann-BGK model 10

2.4.2 The Chapman-Enskog expansion 11

2.5 Chapman-Enskog ansatz 17

2.6 The basic calculation algorithm 22

2.7 Compressibility effect of the lattice Boltzmann method 24

2.8 The multiple-relaxation-time collision model 25

2.8.1 The multiple-relaxation-time lattice Boltzmann equation 26

2.8.2 The multiple-relaxation-time model for D3Q19 model 27

2.9 A Smagorinsky subgrid-scale model 30

2.10 Boundary condition for the lattice Boltzmann method 31

2.10.1 Overview of boundary conditions 31

2.10.2 Solid wall boundary conditions 32

2.11 Parametrisation 32

2.12 The calculation procedure 33
2.13 Primitive model verification 34
2.13.1 Performance of parallel computing 35
2.13.2 Comparison between the BGK-LBM and MRT-LBM 35
2.13.3 Detailed verification of the MRT-LBM 36
2.13.4 Periodic solutions of high-Reynolds number flow 37
2.13.5 Three-dimensional lid-driven cavity flow 37
2.14 Conclusion 38

References **51**

3 Free surface simulation by the Volume-of-Fluid method **56**
3.1 Introduction 56
3.2 Notations 57
3.3 Advection of the colour function 57
3.4 The volume of fluid (VOF) method 58
3.4.1 Introduction 58
3.4.2 Basic properties 59
3.5 Interface reconstruction 61
3.6 Convergence order of a reconstruction method 61
3.7 Evaluation of the interface unit normal 62
3.7.1 Youngs' finite-difference method 62
3.7.2 Centred-columns difference method 63
3.7.3 Parker and Youngs' method 64
3.7.4 The ELVIRA method 67
3.8 Determination of the interface parameter 68
3.9 Geometrical one-dimensional linear-mapping method 72
3.9.1 Operator splitting 72
3.9.2 The out-of-cell explicit linear mapping 72
3.9.3 The onto-cell implicit linear mapping 74
3.9.4 Combined linear mapping 75
3.10 Related one-dimensional advection methods 76
3.10.1 One-dimensional advection equation for the volume fraction 77
3.10.2 The Lagrangian explicit scheme 77
3.11 The volume-of-fluid in the lattice Boltzmann method 78
3.11.1 Free surface boundary condition of the lattice Boltzmann method 78
3.11.2 Introduction of a control volume 79
3.11.3 Choice of the control volume 79

3.11.4 Discretisation of the advection equation 79
3.11.5 Averaging of cell fill levels 80
3.12 Implementation the PLIC-VOF approach into the lattice Boltzmann method 81
3.12.1 Calculation of cell face velocities 81
3.12.2 Algorithm and implementation 81
3.13 Verification and validation 82
3.13.1 Analysis of a standing waves in rectangular tank 82
3.13.2 Verification for the weak-compressibility in dam-breaking flows 83
3.13.3 Validation for the spacing resolution 84
3.13.4 A dam-breaking flow with a single obstacle 86
3.13.5 Breaking wave in a rectangular tank 86
3.13.6 Tsunami inundation in realistic bathymetry 87
3.14 Conclusion 87

References **110**

4 Lattice Boltzmann method for shallow water flows **115**
4.1 Introduction 115
4.2 A non-linear shallow water equation 116
4.3 A Subgrid-scale stress model 122
4.4 The lattice Boltzmann method for a non-linear shallow water equation 123
4.4.1 Derivation of the lattice Boltzmann equation 123
4.5 The lattice Boltzmann equation 125
4.6 Lattice pattern 126
4.6.1 The equilibrium distribution function 127
4.7 Macroscopic properties 130
4.8 Recovery of the shallow water equations 131
4.9 Stability conditions 133
4.10 Improved force term and treatment of bed slope 134
4.11 Turbulence modelling 135
4.12 The multiple-relaxation-time collision model in the D2Q9 square lattice 136
4.13 Recovery of the MRT-LBM for shallow water equations 137
4.14 Boundary and initial conditions 141
4.14.1 No-slip boundary condition 141
4.14.2 Slip boundary condition 141
4.14.3 Semi-slip boundary conditions 141
4.14.4 Initial condition 142
4.15 Implementation for tsunami modellings 142

4.15.1 Moving shoreline algorithm 142
4.15.2 Multiple spacing grid coupled model 143
4.15.3 The calculation procedure 144
4.16 Primitive benchmarks 144
4.16.1 Simple dam-breaking flows 144
4.16.2 Flows around a square cylinder 145
4.16.3 Plane-driven cavity flows 146
4.17 Verification with tsunami by the 2011 Great East Japan Earthquake 147
4.18 Conclusion 148

References **164**

5 A 2D/3D hybrid coupling tsunami modelling **169**
5.1 Introduction 169
5.2 2D/3D hybrid tsunami simulation 170
5.3 Coupling 2D to 3D algorithm 170
5.4 Coupling 3D to 2D algorithm 171
5.5 Calculation algorithm 172
5.6 Validation 173
5.6.1 A dam-breaking flow with obstacles 173
5.7 Conclusion 174

References **180**

6 Conclusion **182**
6.1 Summary 182
6.2 Discussion and future work 183
6.2.1 Parallel adaptive mesh refinement implementations 183
6.2.2 Multiple GPGPU implementations 183
6.2.3 Coupling with the discrete element method 184

References **185**

List of Tables

3.1 Calculation parameters of the dam breaking flow [KO96]. 109

3.2 Calculation parameters of the dam breaking flow [MM52]. 109

3.3 Calculation parameters of dam breaking flow [AK09]. 109

4.1 Parameters used in the dam-breaking problem. 163

4.2 Parameters used in the square cylinder benchmark problem. 163

4.3 Parameters used in the plane-driven cavity problem. 163

4.4 Parameters tsunami simulation in Onagawa. 163

5.1 Parameters used in the dam-breaking benchmark [FKM10]. 179

List of Figures

2.1 Three-dimensional 19-speed square lattice (D3Q19). 39

2.2 Collision and propagation. 39

2.3 The general calculation procedure for the lattice Boltzmann method. 40

2.4 The computational domain of the two-dimensional lid-driven cavity flow. 41

2.5 MPI communication before streaming step for the benchmark [GGS82]. 41

2.6 Performance of MPI parallel computing: (a) elapsed time, (b) speed. 42

2.7 Dimensionless velocity profiles: (a) Re = 100, (b) Re = 400, (c) Re = 1000, (d) Re = 3200, (e) Re = 5000, (f) Re = 7500. 43

2.8 Dimensionless velocity profiles of the MRT-LBM: (a) Re = 1000, (b) Re = 3200, (c) Re = 5000, (d) Re = 7500. 44

2.9 Relative error values for u_x, (a) Re = 1000, (b) Re = 3200, (c) Re = 5000, (d) Re = 7500. 45

2.10 Relative error values for u_y, (a) Re = 1000, (b) Re = 3200, (c) Re = 5000, (d) Re = 7500. 46

2.11 (a) Horizontal velocity history, (b) the power spectrum. 47

2.12 Phase portrait at the monitoring point. 48

2.13 The computational domain and calculation parameters for the three-dimensional lid-driven cavity flow. 49

2.14 Dimensionless velocity profiles of the MRT-LBM along x-axis u_x: (a) Re = 100, (b) Re = 400, (c) Re = 1000, along z-axis u_z: (d) Re = 100, (e) Re = 400, (f) Re = 1000. 50

3.1 One-dimensional advection by the VOF method. Given the value of the colour function in the interface cell j, and the side where the full cell is, the location of the interface can be found and the fluxes computed exactly. 89

3.2 The basic principle of the VOF methods: (a) a portion of the interface line and the colour function value in each cell; (b) the SLIC reconstruction, where each segment in parallel to one coordinate axis; (c) a PLIC reconstruction with unconnected segments across each cell; the normal vector $\boldsymbol{m} = -\nabla C$ is pointing outwards from the reference phase. 89

3.3 Representing the colour function as shaded squares of size proportional to the fractional volumes helps to understand the reconstruction problem. (a) A well-behaved distribution of the colour function; it corresponds to a smooth circular arc. (b) A more confused distribution. 90

3.4 A line cutting two grid cells (thick solid line). In the trapezoidal rule the piecewise linear approximation connects consecutive points on the line and it is continuous across the cell boundary (dashed lines). The PLIC-VOF reconstruction is not continuous across the cell boundary but satisfies the area conservation constraint (thin solid lines). The dotted regions represent the area error for a PLIC-VOF reconstruction (left) and the trapezoidal rule (right). 90

3.5 In the centred-columns scheme, volume fractions are added column-wise, heights y_{i-1} and y_{i+1}, for case (a) and row-wise, widths x_{j-1} and x_{j+1}, for case (b), to get the correct slope of the linear interface. In case (c), an off-centred scheme, with heights y_i and y_{i+1} should be used. 91

3.6 The stencil that Parker and Youngs use to determine ∇f in two dimension. 91

3.7 (a) Parker and Youngs' method will reconstruct this line exactly only if $\alpha = 2$. (b) The volume fractions associated with the line shown in (a). (c) Parker and Youngs' method does not reconstruct this line exactly for $\alpha = 2$. Thus it does not reproduce all linear interface exactly, and so we conclude it is at best first-order method. (d) The volume fractions associated with the line shown in (c). 92

3.8 The cut volume refers to the region within the rectangle $ABCD$ which also lies below the straight line EH. 93

3.9 The cut volume is the region inside the parallelepiped $ABCEFGH$ and below the plane IJK. 93

3.10 (a) The horizontal one-dimensional mapping T_x^E transforms the square cell $\sum$ onto the rectangle Γ_x, by transporting the cell edges with the one-dimensional flows. (b) Three consecutive cells are transformed by the piecewise linear mapping T_x^E onto three rectangles, which are projected back to the original grid to calculate the three contributions D, E and F to the square cell $\sum$. 94

3.11 The images of the square cells by the mapping T_x^E form a tessellation of the plane by rectangles (thick solid lines). These may be either expanded or compressed and shifted to the left or to the right with respect to the original square cells (thin solid lines), depending on the local value of the velocity field. 95

3.12 (a) The horizontal one-dimensional mapping T_x^I transforms the rectangle Γ_x onto the square cell $\sum$ by using the velocity at the cell sides. The velocity field u is the same as Figure 3.10. (b) The contributions to Γ_x of three consecutive cells are transformed by the linear mapping T_x^I onto D, E and F of the central cell $\sum$. 96

3.13 (a) In the Lagrangian method, the end points A and B of the interface segment are advected by the flow. The local velocity may be interpolated linearly between the velocities u^- and u^+ at the cell faces. (b) In the Eulerian method, the total area flux through the right side of the cell is the area of the rectangle with width $u^+\Delta t$, while the reference phase flux is the shaded portion of this rectangle. 97

3.14 The missing distribution functions f_i on the interface cells, (red functions) functions from empty gas cells, (green functions) functions along the interface normal $\boldsymbol{n}$ determined by the VOF algorithms (broken line). 98

3.15 Here the different cell types required for the free surface algorithm can be seen. (yellow) Gas-phase which means cells without fluid. (light blue) Interface-phase which means cells filled with fluid partly. (deep bule) Fluid-phase which means cells completely filled with Fluid. 98

3.16 The calculation domain and initial settings [WT94] (left), Calculation parameters (right). 99

3.17 The calculated interface shapes with MRT-LBM (Ma = 0.01), (a) $t = 0.3$s, (b) $t = 0.6$s, (c) $t = 0.9$s, (d) $t = 1.1$s, (e) $t = 1.4$s, (f) $t = 1.7$s. 99

3.18 The comparison of the spacing density profiles at $t = 0.5$s, (a) MRT-LBM, (b) BGK-LBM. 100

3.19 Timeseries of the water level at the centre of the numerical tank. 100

3.20 The calculation domain of the Koshizuka and Oka [KO96] dam breaking flows. . . . 101

3.21 Fluid density profiles of the MRT-LBM due to the difference in Mach numbers at $t = 0.3$s. 101

3.22 Fluid density profile of the MRT-LBM and BGK-LBM at at $t = 0.3$s. 101

3.23 The calculation domain of the Martin and Moyce [MM52] dam breaking flows. . . . 102

3.24 Timeseries of the dimensionless position of the surge front. 102

3.25 The calculation domain of the Araki and Koshimura [AK09] dam breaking flows. . . 103

3.26 Free surface shapes with case2 simulation in three-dimensional view, (a) $t = 0.50$s, (b) $t = 1.0$s, (c) $t = 1.5$s, (d) $t = 2.0$s, (e) $t = 2.5$s, (f) $t = 3.0$s. 103

3.27 Free surface shapes in two-dimensional view, (top) Case1, (middle) Case2, (bottom) Experimental data [AK09]. 104
3.28 Timeseries water depth at the observed points (Figure 3.25), (a) Point1, (b) Point2. 104
3.29 The calculation domain of the Kölke [Köl05] dam breaking flows. 105
3.30 The comparison of the interface shapes in two-dimensional view, (top) our results, (bottom) experimental data [Köl05]. 105
3.31 Initial velocity profiles for three-dimensional breaking waves benchmark [LVAC06], (a) initial phase profile, (b) velocity along x-axis, (c) velocity along y-axis. 106
3.32 Time evolution of the three-dimensional breaking wave. (left) Lubin *et. al.*'s three-dimensional VOF results [LVAC06], (right) our results. 106
3.33 The bathymetry data of the three-dimensional Onagawa benchmark problem (from Disaster prevention research institute, Kyoto university). 107
3.34 The incident wave from right-hand boundary in Figure 3.33. 107
3.35 Free surface in three-dimensional view, (a) $t = 0.0$s, (b) $t = 30.0$s, (c) $t = 40.0$s, (d) $t = 50.0$s, (e) $t = 60.0$s, (f) $t = 70.0$s. (g) $t = 80.0$s, (h) $t = 90.0$s. 108

4.1 Shallow water flows sketch. 149
4.2 Two-dimensional 9-speed square lattice (D2Q9) in the horizontal plane. 150
4.3 Layout of wall boundary and lattice nodes. 150
4.4 Layout of moving shoreline algorithm in this thesis. 151
4.5 Layout of connecting algorithm for different spacing grid sizes. 151
4.6 Geometry of the dam-breaking problem. 152
4.7 Comparison of the total water depth between the BGK and MRT models $t = 1.0$s: (a) Case1, (b) Case2, (c) Case3. 153
4.8 Geometry of the square cylinder benchmark problem. 154
4.9 The x-axis velocity profiles after 10 minutes iterations: (a) Case1, (b) Case2, (c) Case3, (d) Case4. 155
4.10 The y-axis velocity profiles after 10 minutes iterations: (a) Case1, (b) Case2, (c) Case3, (d) Case4. 155
4.11 Timeseries of velocities along the x-axis at $(x, y) = (10.0\text{m}, 0.0\text{m})$, (a) velocity along x-axis, (b) velocity along y-axis in the calculation case 4. 156
4.12 The time-averaged dimensionless velocity along the y-axis at $x = 1$m. 156
4.13 Geometry and boundary conditions of the plane-driven cavity flow. 157
4.14 The velocity profiles along the centre line after 30 minutes iteration: (a) Re = 10000, (b) Re = 15000, (c) Re = 20000, (d) Re = 30000. 158
4.15 The calculation domains: (a) sub-region1, (b) sub-region2, (c) sub-region3, (d) sub-region4, (e) sub-region5. Sub-region5 is the inundation area (Onagawa). 159

4.16 Comparisons of calculated tsunami waveforms between LBM and FDM in sub-region1: (a) TM-1, (b) TM-2, (c) Iwate N, (d) Iwate M, (e) Iwate S, (f) Miyagi M, (g) Mutsuogawara Port, (h) Kamaishi, (i) Sendai NewPort, (j) Choshi. 160
4.17 The observed and calculated tsunami waveforms in sub-region3 (Onagawa). 161
4.18 Maximum inundation depth in sub-region5: (a) LBM, (b) FDM. 162

5.1 This schematic illustration gives an overview of the 2D/3D hybrid tsunami simulation method. The full three-dimensional fluid flow is solved in a given region of interest (illustrated by a 2D rectangle), and coupled to a two-dimensional shallow water simulation (shown as a one-dimensional line in the picture). 175
5.2 Detail of the double layer boundary conditions in the overlapping interface region. . 176
5.3 Here the effect of the different time step sizes for the 2D/3D hybrid fluid simulation is shown. The numbers indicate the order in which the steps are performed. Dashed arrows indicate interpolation, while straight arrows from one circle to another represent lattice Boltzmann steps with the indicated time step length. 176
5.4 The calculation domain of the Fukui *et. al.* [FKM10] dam-breaking flows. 177
5.5 Free surface shapes at $t = 1.34$s: (a) two-dimensional view of hybrid simulation (b) two-dimensional view of only two-dimensional simulation, (c) three-dimensional view, (d) experimental data [FKM10]. 177
5.6 Timeseries water depth at: (a) $(x, y) = (6.0\text{m}, 0.1125\text{m})$, (b) $(x, y) = (6.0\text{m}, 0.3375\text{m})$. 178

Chapter 1

Introduction

1.1 Motivation

Hydrodynamic free surface flows have received an ever-increasing interest over the past few years. Applications of such flows range from coastal architecture to offshore, marine and tsunami engineering. Free surface flow problems involve an air and a water phase, so that, technically, such flows can be classified as multi-phase flows. However, as for most problems, the flow behaviour is dominated by the water phase, which means that it is not necessary to resolve the full aerodynamics inside the air phase. Instead, the latter can be represented by proper kinematic and dynamic boundary conditions at the free surface by a one-fluid flow formulation, and the sharp interface is allowed to move freely [DD91, Her07].

Typical free surface flow problems occur in many different fields of civil and coastal engineering. Moreover, large-scaled free surface flows, such as breaking waves or tidal currents, are fascinating, as they give us indications of the enormous amount of energy that water can contain, for both good and bad reasons. Natural disasters have, recently, attracted public attention to hazardous free surface flow events. Particularly, the recent 2011 Tohoku tsunami [DMM^{+}11] have increased the awareness of the potential threat posed by tsunamis and, consequently, the responsibility of civil engineers to minimise the consequences of such major impacts. Here, numerical tools can help to establish early warning systems and to provide accurate and fast predictions for tsunami severeness and the details of coastal impact [SB06, Jos11].

Despite offering a powerful tool to investigate tsunamis, three-dimensional tsunami simulations have been successfully applied only in limited computing environment. The overall implicit scheme of most Navier-Stokes solvers has to solve one or more linear system of equations in each time step (i.e., the pressure Poisson equation), this implies that the capillary effects could acts as a formidable computational barrier. Implicit inclusion of force terms would then be considered as a solution [Hys07], the subsequent computational complications, however, are yet to be questioned.

Consequently, one may think of computationally less sophisticated kinetic-based methods which basically march in time in an explicit fashion with naturally slow paces, enabling them to detect rapid temporal changes without imposing an extra cost. This is where the lattice Boltzmann method, as the most evolved variant of kinetic schemes [MZ88, CCM92].

The aim of this thesis is development of a consistent and efficient tsunami simulation model by the advanced lattice Boltzmann method in order to solve the above-mentioned problems.

1.2 Contributions of this thesis

The goal of this thesis is the efficient and large-scaled tsunami simulation with the lattice Boltzmann method. The main contributions of this thesis to achieve this are:

- A non-hydrostatic free surface model to enable an efficient tsunami simulation in three dimension by the advanced one-fluid Volume-of-Fluid approach.
- Parameterisations of the lattice Boltzmann method to reduce the pseudo-compressibility. It gives an incompressible flow field similar to the conventional approaches.
- A hydrostatic non-linear shallow water model to achieve a consistent tsunami simulation from its occurrence to inland penetration.
- A hybrid algorithm to enable simulations of large-scaled tsunami flows by coupling two and three-dimensional fluid model.

1.3 Outline

This thesis is organised as follows. The theoretical background for the lattice Boltzmann method developed as a very good alternative solver for the conventional Navier-Stokes equations is investigated in Chapter 2. Chapter 3 outlines the implementation of free surface model based on the higher-order volume of fluid approach for an efficient three-dimensional tsunami simulation. Chapter 4 briefly describes the lattice Boltzmann method for non-linear shallow water equations for hydrostatic two-dimensional tsunami simulation. Chapter 5 presents a hybrid fluid simulation algorithm coupling hydrostatic shallow water flow model and non-hydrostatic free surface model to achieve a consistent tsunami simulation from its occurrence to inland penetration. A summary of the work and the outlook for future studies is eventually presented as the closure of the thesis in Chapter 6.

References

[CCM92] Chen, H., Chen, S., and Matthaeus, W. H. Recovery of the Navier-Stokes equations using a lattice-gas Soltzmann method. *Phys. Rev. A*, 45(8):5339–5342, 1992.

[DD91] Dean, R. G. and Dalrymple, R. A. *Water Wave Mechanics for Engineers and Scientists.* World Scientific, 1991.

[DMM$^+$11] Dunbar, P., McCullough, H., Mungov, G., Varner, J., and Stroker, K. 2011 Tohoku earthquake and tsunami data available from the National Oceanic and Atmospheric Administration/National Geophysical Data Center. *Geomatics, Nat. Hazards Risk*, 2(4):305–323, 2011.

[Her07] Hervouet, J. M. *Hydrodynamics of Free Surface Flows.* Wiley, 2007.

[Hys07] Hysing, S. *Numerical simulation of immiscible fluids with FEM level set techniques.* PhD thesis, Technische Universität Dortmund, 2007.

[Jos11] Joseph, A. *Tsunamis: detection, monitoring, and early-warning technologies.* Elsevier, 2011.

[MZ88] McNamara, G. R. and Zanetti, G. Use of the boltzmann equation to simulate lattice-gas automata. *Phys. Rev. Lett.*, 61(20):2332–2335, 1988.

[SB06] Synolakis, C. E. and Bernard, E. N. Tsunami science before and beyond boxing day 2004. *Philos. Trans. A Math. Phys. Eng. Sci.*, 364(1845):2231–2265, 2006.

Chapter 2

Lattice Boltzmann method

2.1 Introduction

Historically, the lattice Boltzmann method (LBM) evolved from methods for the simulation of gases that computed the motion of each molecule in the gas purely integer operations. In [HDP76], there was a first attempt to perform fluid simulations with this approach. It took ten years to discover that the isotropy of the lattice vectors is crucial for correct approximation of the Navier-Stokes equations [FDH$^+$87]. Motivated by this improvement [MZ88], they developed the algorithm that was actually called LBM by performing simulations with averaged floating point values instead of single fluid molecules. The important contribution to the basic LBM was the simplified collision operator with a single relaxation parameter. This collision model is well-known as the Bhatnagar-Gross-Krook (BGK) approximation [BGK54], and was derived independently by [CCM92], and [QDL92]. Since then, the LBM has been applied to many classes of fluid mechanics problems: the direct numerical simulation of turbulence [YGL05], Eulerian-Lagrangian simulations [MC98], the single-phase free surface simulations analogous to the volume-of-fluid (VOF) approach [KTH$^+$05, TPR$^+$06, JK11], and two-phase fluid simulation coupling with the phase field modelling approaches, e.g., the Cahn-Hilliard equations [IOTK04, IYTT16, BJG14, BJGK14] or the level-set method [BJK$^+$09, ST16, SPT17], among others. The primitive background of LBM will be described in detail as below.

2.2 Statistical mechanics and Boltzmann equation

Statistical mechanics was pioneered by the Austrian physicist Ludwig Boltzmann from the 1870s. It provides a framework for the interpretation and prediction of the macroscopic bulk properties of materials by the macroscopic properties of individual atoms and molecules. The atomistic dynamics that describes the motion of individual molecules by the deterministic Newton equations or Hamiltonian system and the thermodynamics that describes thermodynamic quantities such an pressure, momentum and energy by continuity, momentum and energy quantities are bridged by the statistical mechanics in a probabilistic approach. The Boltzmann equation, established by Ludwig Boltzmann in 1872 is the cornerstone of the development of statistical mechanics.

The atomistic dynamics for the molecules of the air governed by Newton equations are:

$$\frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} = \frac{\boldsymbol{p}_i}{m} \tag{2.1}$$

$$\frac{\mathrm{d}\boldsymbol{p}_i}{\mathrm{d}t} = \boldsymbol{F}_i \tag{2.2}$$

where $i = 1, \ldots, N$, $\boldsymbol{x}_i$ is the position coordinate of the i^{th} molecule, $\boldsymbol{p}_i = m\boldsymbol{\xi}_i$ the linear momentum and $\boldsymbol{F}_i$ the external force from intermolecular interactions or external fields such as gravity. In three-dimensional space, there are totally $6N$ functions of time $\left(\boldsymbol{x}_i\left(t\right), \boldsymbol{p}_i\left(t\right)\right), i = 1, \ldots, N$. The above equations provide a detailed description of all the N molecules, from which there is the possibility to capture the behaviour of the whole system. However, N is generally too big for the ability of any foreseeable computer.

This problem is taken into account by a higher level of description with a system of N particles by distribution functions $f_N\left(\boldsymbol{x}_1, \boldsymbol{p}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{p}_N, t\right)$, where N is much smaller than the real number of molecules. $f_N\left(\boldsymbol{x}_1, \boldsymbol{p}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{p}_N, t\right) \mathrm{d}\boldsymbol{x}_1 \mathrm{d}\boldsymbol{p}_1 \cdots \mathrm{d}\boldsymbol{x}_N \mathrm{d}\boldsymbol{p}_N$ is the probability to find a particle within the interval $[\boldsymbol{x}_1, \boldsymbol{x}_1 + \mathrm{d}\boldsymbol{x}_1] \times [\boldsymbol{p}_1, \boldsymbol{p}_1 + \mathrm{d}\boldsymbol{p}_1] \times \cdots \times [\boldsymbol{x}_N, \boldsymbol{x}_N + \mathrm{d}\boldsymbol{x}_N] \times [\boldsymbol{p}_N, \boldsymbol{p}_N + \mathrm{d}\boldsymbol{p}_N]$. f_N encompasses all statistical information of the dynamical process, and evolves according to the Liouville equation:

$$\frac{\partial f_N}{\partial t} - \sum_{j=1}^{3N} \left(\frac{\partial H_N}{\partial x_j} \frac{\partial f_N}{\partial p_j} - \frac{\partial H_N}{\partial p_j} \frac{\partial f_N}{\partial x_j} \right) = 0 \tag{2.3}$$

where H_N is the Hamiltonian of the system. Integrating over part of the phase space, we define the reduced densities as:

$$F_s\left(\boldsymbol{x}_1, \boldsymbol{p}_1, \ldots, \boldsymbol{x}_s, \boldsymbol{p}_s, t\right) = V^s \int f_N\left(\boldsymbol{x}_1, \boldsymbol{p}_1, \ldots, \boldsymbol{x}_s, \boldsymbol{p}_s, t\right) \mathrm{d}\boldsymbol{x}_{s+1} \mathrm{d}\boldsymbol{p}_{s+1} \cdots \mathrm{d}\boldsymbol{x}_N \mathrm{d}\boldsymbol{p}_N \tag{2.4}$$

where V^s is a normalisation factor. A coupled system of differential equations for $F_s\left(1 \leq s \leq N\right)$ with the name of BBGKY after Bogoljubov, Born, Green, Kirkwood and Yvon has been shown to be equivalent to the Liouville equation. The Boltzmann equation has been derived from BBGKY by the assumption of two-partial local collisions with uncorrelated velocities before collision and free of external forces. The integro-differential Boltzmann equation describes a single partial distribution function $f\left(\boldsymbol{x}, \boldsymbol{\xi}, t\right) \propto F_1\left(\boldsymbol{x}_1, \boldsymbol{p}_1, t\right)$ as:

$$\left(\partial_t + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} + \boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}}\right) f\left(\boldsymbol{x}, \boldsymbol{\xi}, t\right) = Q\left(f, f\right) \tag{2.5}$$

where $\boldsymbol{g}\left(\boldsymbol{x}, t\right)$ is the acceleration depending only on space and time, with $F\left(\boldsymbol{\xi}\right) = -\boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}} f$ as the body force. The collision integral $Q(f,f)$ with $\sigma(\Omega)$ the differential collision cross section for the two-particle collision which transforms from the incoming velocity $(\boldsymbol{\xi},\boldsymbol{\xi}_1)$ to the outgoing velocity $(\boldsymbol{\xi}',\boldsymbol{\xi}_1')$ is defined as:

$$Q(f,f)=\int \mathrm{d}\boldsymbol{\xi}_1 \int \mathrm{d}\Omega\sigma(\Omega)\left|\boldsymbol{\xi}-\boldsymbol{\xi}_1\right|\left(f(\boldsymbol{\xi}')\,f(\boldsymbol{\xi}_1')-f(\boldsymbol{\xi})\,f(\boldsymbol{\xi}_1)\right) \tag{2.6}$$

it can be shown that the collision integral has five elementary collision invariants $\psi_k(\boldsymbol{\xi}), k = 0,1,2,3,4$ as:

$$\int Q(f,f)\,\psi_k(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}=0 \tag{2.7}$$

where $\psi_0=1,(\psi_1,\psi_2,\psi_3)=\boldsymbol{\xi},\psi_4=\boldsymbol{\xi}^2$ related to mass, momentum, and kinetic energy. Written in a linear combination, the general collision invariants $\phi(\boldsymbol{\xi})$ is as:

$$\phi(\boldsymbol{\xi})=a+\boldsymbol{b}\cdot\boldsymbol{\xi}+c\boldsymbol{\xi}^2 \tag{2.8}$$

it can also be shown that only the functions with the form:

$$f(\boldsymbol{\xi})=\exp\left(a+\boldsymbol{b}\cdot\boldsymbol{\xi}+c\boldsymbol{\xi}^2\right) \tag{2.9}$$

can satisfy the null collision integral $Q(f,f)=0$. The special case of the Maxwell-Boltzmann equilibrium distribution function is defined as:

$$f^{(0)}(\boldsymbol{x},\boldsymbol{\xi},t)=\frac{\rho(\boldsymbol{x},t)}{\left(2\pi\theta(\boldsymbol{x},t)\right)^{D/2}}\exp\left(-\frac{(\boldsymbol{\xi}-\boldsymbol{u})^2}{2\theta(\boldsymbol{x},t)}\right) \tag{2.10}$$

where $\rho(\boldsymbol{x},t)$ and $\boldsymbol{u}(\boldsymbol{x},t)$ are the density and velocity of the fluid at spatial position $\boldsymbol{x}$ and time t; θ is defined as k_BT/m with k_B the Boltzmann constant, m the molecular mass of gas particles and T the absolute temperature under our consideration (one notices that for isothermal fluid T is constant). As can be seen from Equation 2.6, the complicated collision integral is a big obstacle when dealing with Boltzmann equation. In order to simply the collision integral $Q(f,f)$ to a convenient integral $J(f)$, two rules have to be obeyed. The first one is that $J(f)$ conserves all the collision invariants $\psi_k(\boldsymbol{\xi})$ of $Q(f,f)$. While the other one is that the collision term should bear the tendency of distribution function to the Maxwell-Boltzmann distribution function by H-theorem. Both of the two rules are fulfilled by the widely used BGK model developed by Bhatnagar, Gross and Krook [BGK54, QDL92].

2.3 From the Boltzmann-BGK equation to the Navier-Stokes equations

The Boltzmann-BGK equation in D-dimensional space is defined as:

$$\left(\partial_t + \boldsymbol{\xi} \cdot \nabla_{\boldsymbol{x}} + \boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}}\right) f\left(\boldsymbol{x}, \boldsymbol{\xi}, t\right) = -\frac{1}{\tau}\left(f - f^{(0)}\right) \tag{2.11}$$

where τ is the relaxation time and $f^{(0)}$ is the Maxwell-Boltzmann distribution function in equilibrium state as defined in Equation 2.10. Here and throughout the following context, we rescale all the velocity $\boldsymbol{\xi}$, $\boldsymbol{u}$ by the sound speed $c_s = \sqrt{k_B T/m}$ and θ by c_s^2 so that all the variables become dimensionless. Especially, $\theta = 1$ for isothermal fluid. The rescaled equilibrium distribution function is still given by Equation 2.10.

Compared to the macroscopic thermodynamic variables, the kinetic distribution function provides much more detailed information. In fact, the macroscopic thermodynamic variables can be expressed as the moments of the distribution function with different order with respect to velocity. Specifically, we have mass density:

$$\rho\left(\boldsymbol{x}, t\right) = \int f\left(\boldsymbol{x}, \boldsymbol{\xi}, t\right) \mathrm{d}\boldsymbol{\xi} \tag{2.12}$$

momentum density:

$$\rho\boldsymbol{u}\left(\boldsymbol{x}, t\right) = \int \boldsymbol{\xi} f\left(\boldsymbol{x}, \boldsymbol{\xi}, t\right) \mathrm{d}\boldsymbol{\xi} \tag{2.13}$$

kinetic energy density:

$$\rho\epsilon\left(\boldsymbol{x}, t\right) = \frac{1}{2}\int \boldsymbol{c}^2 f\left(\boldsymbol{x}, \boldsymbol{\xi}, t\right) \mathrm{d}\boldsymbol{c} \tag{2.14}$$

pressure or stress tensor:

$$P = \int \boldsymbol{c}\boldsymbol{c} f\left(\boldsymbol{x}, \boldsymbol{\xi}, t\right) \mathrm{d}\boldsymbol{c} \tag{2.15}$$

energy or heat flux:

$$\boldsymbol{q} = \frac{1}{2}\int \boldsymbol{c}\boldsymbol{c}^2 f\left(\boldsymbol{x}, \boldsymbol{\xi}, t\right) \mathrm{d}\boldsymbol{c} \tag{2.16}$$

where $\boldsymbol{c} = \boldsymbol{\xi} - \boldsymbol{u}$ is defined as the intrinsic velocity or mean velocity. The form of $\boldsymbol{c}^2$ stands for the inner product $\boldsymbol{c}^2 = c_i c_i$ while $\boldsymbol{c}\boldsymbol{c}$ represents the tensor product such that $\left(\boldsymbol{c}\boldsymbol{c}\right)_{ij} = c_i c_j$, where the subscripts i, j denote Cartesian components and Einstein summation convention is used for repeated indices in the component notation, with the meaning of $c_i c_i = \sum_i c_i^2$.

If stories for fluid in the mesoscopic world are all about distribution function f subjected to Boltzmann-BGK equation, then the bridge to go to the macroscopic world from the mesoscopic one are velocity average (integral) in different order of moments with respect to the distribution function. When we are in the macroscopic world with its own variables such as density ρ and velocity $\boldsymbol{u}$, what we would play should be submitted to Newton's laws, including mass conservation, momentum conservation and energy conservation in the realm of classical fluid mechanics.

Define an integral operator $I\left[\cdot\right]$ - as the rule to cross the bridge as follows:

$$I\left[h\right]\left(\boldsymbol{x},t\right)=\frac{1}{\rho\left(\boldsymbol{x},t\right)}\int_{R^D}=h\left(\boldsymbol{\xi}\right)f\left(\boldsymbol{x},\boldsymbol{\xi},t\right)\mathrm{d}\boldsymbol{\xi} \tag{2.17}$$

where $h\left(\boldsymbol{\xi}\right)$ is an integrable function dependent only on the velocity $\boldsymbol{\xi}$. In particular, 1, $\boldsymbol{\xi}$, $\boldsymbol{\xi}^2/2$, corresponding to mass, momentum and energy are chosen among the moments of velocity. In general, multiplying by $h\left(\boldsymbol{\xi}\right)$ and integrating both sides of the Boltzmann-BGK equation over the velocity space, we can obtain:

$$\int\left(\partial_t f+\boldsymbol{\xi}\cdot\nabla_{\boldsymbol{x}}f+\boldsymbol{g}\cdot\nabla_{\boldsymbol{\xi}}f\right)h\mathrm{d}\boldsymbol{\xi}=-\frac{1}{\tau}\int\left(f-f^{(0)}\right)h\mathrm{d}\boldsymbol{\xi} \tag{2.18}$$

where $h\left(\boldsymbol{\xi}\right)=1,\boldsymbol{\xi},\boldsymbol{\xi}^2/2$ respectively, the right hand side of Equation 2.18 vanishes thanks to the conservation laws of mass, momentum and kinetic energy for ideal collision. While for the left hand side, term by term we have:

$$\int\left(\partial_t f\right)h\left(\boldsymbol{\xi}\right)\mathrm{d}\boldsymbol{\xi}=\partial_t\left(\frac{\rho}{\rho}\int fh\left(\boldsymbol{\xi}\right)\mathrm{d}\boldsymbol{\xi}\right)=\partial_t\left(\rho I\left[h\left(\boldsymbol{\xi}\right)\right]\right) \tag{2.19}$$

$$\begin{aligned}\int\left(\boldsymbol{\xi}\cdot\nabla_{\boldsymbol{x}}f\right)h\left(\boldsymbol{\xi}\right)\mathrm{d}\boldsymbol{\xi}&=\nabla_{\boldsymbol{x}}\cdot\left(\int h\left(\boldsymbol{\xi}\right)\boldsymbol{\xi}f\mathrm{d}\boldsymbol{\xi}\right)-\int f\nabla_{\boldsymbol{x}}\cdot\left(\boldsymbol{\xi}h\left(\boldsymbol{\xi}\right)\right)\mathrm{d}\boldsymbol{\xi}\\&=\nabla_{\boldsymbol{x}}\cdot\left(\frac{\rho}{\rho}\int h\left(\boldsymbol{\xi}\right)\boldsymbol{\xi}f\mathrm{d}\boldsymbol{\xi}\right)=\nabla_{\boldsymbol{x}}\cdot\left(\rho I\left[h\left(\boldsymbol{\xi}\right)\boldsymbol{\xi}\right]\right)\end{aligned} \tag{2.20}$$

$$\begin{aligned}\int\left(\boldsymbol{g}\left(\boldsymbol{x},t\right)\cdot\nabla_{\boldsymbol{\xi}}f\right)h\left(\boldsymbol{\xi}\right)\mathrm{d}\boldsymbol{\xi}&=\int\left(h\left(\boldsymbol{\xi}\right)\boldsymbol{g}\left(\boldsymbol{x},t\right)\right)\cdot\nabla_{\boldsymbol{\xi}}f\mathrm{d}\boldsymbol{\xi}\\&=\int\left(h\left(\boldsymbol{\xi}\right)\boldsymbol{g}\left(\boldsymbol{x},t\right)\right)\cdot\left(f\boldsymbol{n}\right)\mathrm{d}\gamma-\int\nabla_{\boldsymbol{\xi}}\cdot\left(h\left(\boldsymbol{\xi}\right)\boldsymbol{g}\left(\boldsymbol{x},t\right)\right)f\mathrm{d}\boldsymbol{\xi}\\&=-\frac{\rho}{\rho}\int\boldsymbol{g}\left(\boldsymbol{x},t\right)\cdot\nabla_{\boldsymbol{\xi}}h\left(\boldsymbol{\xi}\right)f\mathrm{d}\boldsymbol{\xi}=-\rho I\left[\boldsymbol{g}\left(\boldsymbol{x},t\right)\cdot\nabla_{\boldsymbol{\xi}}h\left(\boldsymbol{\xi}\right)\right]\end{aligned} \tag{2.21}$$

substituting the above three terms into Equation 2.18 and noticing that the right hand side vanishes, the following integral-differential equation holds for $h\left(\boldsymbol{\xi}\right)=1,\boldsymbol{\xi},\boldsymbol{\xi}^2/2$:

$$\partial_t\left(\rho I\left[h\left(\boldsymbol{\xi}\right)\right]\right)+\nabla_{\boldsymbol{x}}\cdot\left(\rho I\left[h\left(\boldsymbol{\xi}\right)\boldsymbol{\xi}\right]\right)-\rho I\left[\boldsymbol{g}\left(\boldsymbol{x},t\right)\cdot\nabla_{\boldsymbol{\xi}}h\left(\boldsymbol{\xi}\right)\right]=0 \tag{2.22}$$

In addition, we can easily verify that the following equalities for $h\left(\boldsymbol{\xi}\right)=1,\boldsymbol{\xi},\boldsymbol{\xi}^2/2$,

$$\begin{cases} I\left[1\right]=1,\ I\left[\boldsymbol{\xi}\right]=\boldsymbol{u},\ I\left[\boldsymbol{c}\right]=I\left[\boldsymbol{\xi}\right]-I\left[\boldsymbol{u}\right]=\boldsymbol{u}-\boldsymbol{u}=0 \\ I\left[\boldsymbol{\xi}^2\right]=I\left[\left(\boldsymbol{c}+\boldsymbol{u}\right)^2\right]=I\left[\boldsymbol{c}^2\right]-2I\left[\boldsymbol{c}\right]\cdot\boldsymbol{u}+\boldsymbol{u}^2=2\epsilon+\boldsymbol{u}^2 \\ I\left[\boldsymbol{\xi}\boldsymbol{\xi}\right]=I\left[\left(\boldsymbol{c}+\boldsymbol{u}\right)\left(\boldsymbol{c}+\boldsymbol{u}\right)\right]=I\left[\boldsymbol{c}^2\boldsymbol{c}\right]+2I\left[\boldsymbol{c}\boldsymbol{c}\right]\cdot\boldsymbol{u}+I\left[\boldsymbol{c}\right]\boldsymbol{u}^2+I\left[\boldsymbol{\xi}^2\right]\boldsymbol{u} \\ \qquad\qquad\qquad\qquad\qquad\qquad =2\boldsymbol{q}/\rho+2P/\rho\cdot\boldsymbol{u}+\left(2\epsilon+\boldsymbol{u}^2\right)\boldsymbol{u} \end{cases} \tag{2.23}$$

With Equation 2.23 at hand, Equation 2.22 for $h\left(\boldsymbol{\xi}\right)=1,\boldsymbol{\xi}$ leads to the continuity equation and momentum equation respectively:

$$\partial_t\rho+\nabla\cdot\left(\rho\boldsymbol{u}\right)=0 \tag{2.24}$$

$$\partial_t\left(\rho\boldsymbol{u}\right)+\nabla\cdot\left(\rho\boldsymbol{u}\boldsymbol{u}\right)=-\nabla\cdot P+\rho\boldsymbol{g} \tag{2.25}$$

with the constraint of the continuity equation (2.24), the momentum equation (2.25) can be simplified in the Eulerian frame as:

$$\rho\partial_t\boldsymbol{u}+\rho\left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u}=-\nabla\cdot P+\rho\boldsymbol{g} \tag{2.26}$$

Equations 2.24 and 2.25 or 2.26 define the full compressible Navier-Stokes equations. Notice that stress tensor P is unknown. When $h\left(\boldsymbol{\xi}\right)=\boldsymbol{\xi}^2/2$, the energy equation is obtained from Equation 2.22 as follows:

$$0=\frac{1}{2}\partial_t\left(\rho I\left[\boldsymbol{\xi}^2\right]\right)+\frac{1}{2}\nabla_{\boldsymbol{x}}\cdot\left(\rho I\left[\boldsymbol{\xi}^2\boldsymbol{\xi}\right]\right)-\frac{1}{2}\rho I\left[\boldsymbol{g}\left(\boldsymbol{x},t\right)\cdot\nabla_{\boldsymbol{\xi}}\boldsymbol{\xi}^2\right] \tag{2.27}$$

$$\text{(by Equation 2.23)}=\partial_t\left(\rho\epsilon+\frac{1}{2}\rho\boldsymbol{u}^2\right)+\nabla\cdot\left(\left(\rho\epsilon+\frac{1}{2}\rho\boldsymbol{u}^2\right)\boldsymbol{u}+P\cdot\boldsymbol{u}+\boldsymbol{q}\right)-\rho\boldsymbol{g}\cdot\boldsymbol{u} \tag{2.28}$$

$$\begin{aligned}\text{(reorder)}&=\underbrace{\left(\partial_t\left(\rho\epsilon\right)+\nabla\cdot\left(\rho\boldsymbol{u}\epsilon\right)\right)}_{\text{Term 1}}+\underbrace{\left(\frac{1}{2}\partial_t\left(\rho\boldsymbol{u}^2\right)+\frac{1}{2}\nabla\cdot\left(\left(\rho\boldsymbol{u}^2\right)\boldsymbol{u}\right)+P\cdot\boldsymbol{u}\right)}_{\text{Term 2}}\\&+\nabla\cdot\boldsymbol{q}-\rho\boldsymbol{g}\cdot\boldsymbol{u}\end{aligned} \tag{2.29}$$

$$\text{(Term 1)}=\left(\partial_t\rho\right)\epsilon+\rho\partial_t\epsilon+\left(\nabla\cdot\left(\rho\boldsymbol{u}\right)\right)\epsilon+\rho\boldsymbol{u}\cdot\left(\nabla\epsilon\right) \tag{2.30}$$

$$\begin{aligned}\text{(by Equation 2.24)}&=-\left(\nabla\cdot\left(\rho\boldsymbol{u}\right)\right)\epsilon+\left(\nabla\cdot\left(\rho\boldsymbol{u}\right)\right)\epsilon+\left(\rho\partial_t\epsilon+\rho\boldsymbol{u}\cdot\left(\nabla\epsilon\right)\right)\\&=\rho\left(\partial_t+\left(\nabla\epsilon\right)\cdot\boldsymbol{u}\right)=\rho\frac{\mathrm{d}\epsilon}{\mathrm{d}t}\end{aligned} \tag{2.31}$$

$$\text{(Term 2)}=\frac{1}{2}\left(\left(\partial_t\rho\right)\boldsymbol{u}^2+\rho\left(\partial_t\boldsymbol{u}^2\right)+\left(\nabla\cdot\left(\rho\boldsymbol{u}\right)\right)\boldsymbol{u}^2+\rho\boldsymbol{u}\cdot\left(\nabla\boldsymbol{u}^2\right)\right)+\left(\boldsymbol{u}\cdot\left(\nabla\cdot P\right)+P:\left(\nabla\boldsymbol{u}\right)\right) \tag{2.32}$$

where $P:\left(\nabla\boldsymbol{u}\right)=\sum_{i,j}P_{i,j}\left(\nabla\boldsymbol{u}\right)_{i,j}$.

$$\begin{aligned}\text{(by Equation 2.24)}&=\frac{1}{2}\left(-\left(\nabla\cdot\left(\rho\boldsymbol{u}\right)\right)\boldsymbol{u}^2+\left(\nabla\cdot\left(\rho\boldsymbol{u}\right)\right)\boldsymbol{u}^2+2\rho\boldsymbol{u}\cdot\left(\partial_t\boldsymbol{u}+\left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u}\right)\right)\\&+\boldsymbol{u}\cdot\left(\nabla\cdot P\right)+P:\left(\nabla\boldsymbol{u}\right)\end{aligned} \tag{2.33}$$

$$\text{(by Equation 2.26)} = P : (\nabla \boldsymbol{u}) + \rho \boldsymbol{g} \cdot \boldsymbol{u} \tag{2.34}$$

substituting Term 2.30, 2.31 and Term 2.32-2.34 into Equation 2.29, we can get the energy equation in the Lagrangian formulation as:

$$\rho \frac{\mathrm{d}\epsilon}{\mathrm{d}t} + P : (\nabla \boldsymbol{u}) + \nabla \cdot \boldsymbol{q} = 0 \tag{2.35}$$

Briefly, Equations 2.24, 2.25 and 2.35 are the hydrodynamic equations for a fluid, which take both thermal and compressible effects into account. From the definition of stress tensor and kinetic energy density we have the following relation:

$$P_{ii} = 2\rho\epsilon \tag{2.36}$$

while the average of the trace of the stress tensor is borrowed to define the hydrostatic pressure as:

$$p = \frac{P_{ii}}{D} \tag{2.37}$$

Meanwhile, by the perfect gas law, the kinetic energy density is related to temperature according to the equipartition theorem as:

$$p = \rho\theta = \rho c_s^2 \tag{2.38}$$

2.4 The Chapman-Enskog expansion

There are several approaches to investigate the property of the lattice Boltzmann method mathematically, such as the Grad's approach [Gra49], S-expansion [IYO97] and Chapman-Enskog expansion. We will use the Chapman-Enskog approach, which is the most popular approach in this thsis.

In the Chapman-Enskog expansion approach, the derivation from the Boltzmann-BGK equation to the Navier-Stokes equations can find its roots in lattice gas automata, in which field many results are ready to be used. Moreover, the Chapman-Enskog expansion approach could also be used to analyse the numerical accuracy of initial condition and boundary conditions. Roughly speaking, the Chapman-Enskog expansion is based on the multi-scale expansion of the distribution function, spatial and temporal variables as well as the projection of the lattice Boltzmann equation on the lattice symmetries in different scales so as to get the corresponding macroscopic equations.

2.4.1 The lattice Boltzmann-BGK model

Unlike the Grad's representation approach, where the derivation from the Boltzmann-BGK equation to the Navier-Stokes equations is carried out in continuous system and then discretisation is applied to the continuous system to obtain the numerical model, the Chapman-Enskog expansion starts from the discrete system - lattice Boltzmann system directly. The lattice structure of the discrete system is originated from the Lattice-Gas Cellular Automata model, while it can also be obtained from the Gauss-Hermite quadrature for the weights as well as discrete velocity - the abscissae of the quadrature.

The lattice Boltzmann - BGK model is given in the lattice Boltzmann system (all variables have lattice units, in particular $\delta x = 1$, $\delta t = 1$) as:

$$f_i\left(\boldsymbol{x} + \boldsymbol{\xi}_i, t+1\right) - f_i\left(\boldsymbol{x}, t\right) = -\frac{1}{\tau}\left(f_i\left(\boldsymbol{x}, t\right) - f_i^{(0)}\left(\boldsymbol{x}, t\right)\right) + F_i\left(\boldsymbol{x}, t\right) \tag{2.39}$$

where $f_i\left(\boldsymbol{x} + \boldsymbol{\xi}_i, t+1\right)$, $f_i\left(\boldsymbol{x}, t\right)$ are the discrete distribution functions and $F_i\left(\boldsymbol{x}, t\right)$ is the unknown discrete body force along the spatial direction $\boldsymbol{\xi}_i$ for $i = 0, 1, \ldots, q-1$. τ is the single relaxation time. And the lattice is constructed such that the two adjacent nodes id the lattice are connected by the spatial discrete velocity $\boldsymbol{\xi}_i$.

The equilibrium distribution functions $f^{(0)}\left(\boldsymbol{x}, t\right)$ represent a stationary state of the fluid. The Maxwell-Boltzmann distribution is used as the equilibrium distribution function in the lattice Boltzmann method. This function will be Taylor expanded in $\boldsymbol{u}$ up to the second order, which is a valid approximation for small velocities, and low Mach numbers. The following formula will be used as the local equilibrium distribution for the following derivations assuming an athermal fluid:

$$f_i^{(0)}\left(\boldsymbol{x}, t\right) = w_i \rho \left(1 + \frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{c_s^2} + \frac{\left(\boldsymbol{\xi}_i \cdot \boldsymbol{u}\right)^2}{2c_s^4} - \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2c_s^2}\right) \tag{2.40}$$

where w_i is the particle weighting parameter, which can be determined in the course of the Chapman-Enskog expansion for each square lattice model.

The force term $F_i\left(\boldsymbol{x}, t\right)$ will be specified as a modification in deriving Navier-Stokes equations. The macroscopic density and momentum are defined under the conservation laws as:

$$\rho\left(\boldsymbol{x}, t\right) = \sum_{i=0}^{q-1} f_i\left(\boldsymbol{x}, t\right) = \sum_{i=0}^{q-1} f_i^{(0)}\left(\boldsymbol{x}, t\right) \tag{2.41}$$

$$\boldsymbol{j}\left(\boldsymbol{x}, t\right) = \rho\left(\boldsymbol{x}, t\right) \boldsymbol{u}\left(\boldsymbol{x}, t\right) = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i\left(\boldsymbol{x}, t\right) = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)}\left(\boldsymbol{x}, t\right) \tag{2.42}$$

2.4.2 The Chapman-Enskog expansion

The basic idea behind Chapman-Enskog expansion is to separate the physical time and space as well as distribution function into multiple scales with respect to the order of Knudsen number ϵ. Physical properties of the macroscopic variables, including the density and momentum, are automatically separated into the corresponding different scales. By applying Taylor expansion for the convection term of the lattice Boltzmann-BGK equation and analysing it in different scales, the physical properties of mass conservation and momentum conservation are formulated into continuity equation and momentum equation respectively, ending up with the Navier-Stokes equations.

Recall the lattice Boltzmann-BGK equation with the fictitious lattice collision Ω_i defined as:

$$\Omega_i\left(\boldsymbol{x},t\right) = -\frac{1}{\tau}\left(f_i\left(\boldsymbol{x},t\right) - f_i^{(0)}\left(\boldsymbol{x},t\right)\right) + F_i\left(\boldsymbol{x},t\right) = f_i\left(\boldsymbol{x}+\boldsymbol{\xi}_i,t+1\right) - f_i\left(\boldsymbol{x},t\right) \tag{2.43}$$

the Chapman-Enskog expansion for time t, space $\boldsymbol{x}$, distribution function f_i together with collision Ω_i in different scales of Knudsen number, in two different scales are given as:

$$\begin{cases} \partial_t = \epsilon\partial_{t1} + \epsilon^2\partial_{t2} + O\left(\epsilon^3\right) \\ \nabla = \epsilon\nabla_1 + \epsilon^2\nabla_2 + O\left(\epsilon^3\right) \\ f_i = f_i^{(0)} + \epsilon f_i^{(1)} + +O\left(\epsilon^2\right) \\ \Omega_i = \epsilon\Omega_i^{(1)} + \epsilon^2\Omega_i^{(2)} + O\left(\epsilon^3\right) \end{cases} \tag{2.44}$$

in most of the literatures, the spatial derivative is expanded only up to the first order $\nabla = \epsilon\nabla_1 + O\left(\epsilon^2\right)$. The reason for this could be intuitively explained as that in order to achieve the Navier-Stokes equations, involving both the second order derivative - diffusion term $\nabla^2\boldsymbol{u}$ and the first order derivative - convection term $\left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u}$ while the temporal derivative is only the first order $\partial_t\left(\rho\boldsymbol{u}\right)$, we need to parallelise the spatial derivative and temporal derivative to the same order, i.e., second order in time and first order in space. In the following multi-scale analysis, we first stick to the second order expansion for space as in Equation 2.44.

Applying Taylor expansion with respect to $f_i\left(\boldsymbol{x},t\right)$ for the lattice Boltzmann equation 2.43 up to the second order, we have:

$$\begin{aligned} \Omega_i\left(\boldsymbol{x},t\right) &= f_i\left(\boldsymbol{x}+\boldsymbol{\xi}_i,t+1\right) - f_i\left(\boldsymbol{x},t\right) \\ &= \left(\partial_t + \nabla\cdot\boldsymbol{\xi}_i\right)f_i + \frac{1}{2}\left(\partial_t^2 + 2\partial_t\nabla\cdot\boldsymbol{\xi} + \nabla\nabla:\boldsymbol{\xi}_i\boldsymbol{\xi}_i\right)f_i + O\left(\mathrm{d}_t^3\right)f_i \end{aligned} \tag{2.45}$$

where $O\left(\mathrm{d}_t^3\right)f_i$ denote higher order of derivative. Substituting Equation 2.44 into Equation 2.45, we obtain the multi-scale expansion as:

$$\begin{aligned} \epsilon\Omega_i^{(1)} + \epsilon^2\Omega_i^{(2)} + O\left(\epsilon^3\right) = &\Big(\epsilon\left(\partial_{t1} + \nabla_1\cdot\boldsymbol{\xi}_i\right) + \epsilon^2\left(\partial_{t2} + \nabla_2\cdot\boldsymbol{\xi}_i\right) \\ &+\frac{1}{2}\epsilon^2\left(\partial_{t1}^2 + 2\partial_{t1}\nabla_1\cdot\boldsymbol{\xi}_i + \nabla_1\nabla_1:\boldsymbol{\xi}_i\boldsymbol{\xi}_i\right) + O\left(\epsilon^3\right)\Big)\left(f_i^{(0)} + \epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)} + O\left(\epsilon^3\right)\right) \end{aligned} \tag{2.46}$$

Separating the multi-scale expansion Equation 2.46 on different scales of ϵ, we have the following two equations on the scales of ϵ and ϵ^2:

$$\Omega_i^{(1)} = \left(\partial_{t1} + \nabla_1\cdot\boldsymbol{\xi}_i\right)f_i^{(0)} \tag{2.47}$$

for ϵ, and:

$$\begin{aligned}\Omega_i^{(2)} &= \left(\partial_{t1} + \nabla_1 \cdot \boldsymbol{\xi}_i\right) f_i^{(1)} + \left(\partial_{t2} + \nabla_2 \cdot \boldsymbol{\xi}_i\right) f_i^{(0)} \\ &+ \frac{1}{2}\left(\partial_{t1}^2 + 2\partial_{t1}\nabla_1 \cdot \boldsymbol{\xi}_i + \nabla_1\nabla_1 : \boldsymbol{\xi}_i\boldsymbol{\xi}_i\right) f_i^{(0)}\end{aligned} \tag{2.48}$$

for ϵ^2. Corresponding to the expansion of distribution function, from the definition of the macroscopic variables, we have the expansion for momentum as:

$$\boldsymbol{j} = \boldsymbol{j}^{(0)} + \epsilon \boldsymbol{j}^{(1)} + O\left(\epsilon^2\right) \tag{2.49}$$

The force term, which is responsible for the change of momentum is located on the first order scale as:

$$\boldsymbol{F} = \epsilon \boldsymbol{F}^{(1)} + O\left(\epsilon^2\right) \tag{2.50}$$

and suppose:

$$\boldsymbol{j}^{(1)} = -\boldsymbol{F}^{(1)}/2 \tag{2.51}$$

which will be highly depended for derivation later on. Therefore, the zeroth order and first order collision invariants can be written as:

$$\begin{cases} 0 = \sum_{i=0}^{q-1} \Omega_i = \sum_{i=0}^{q-1} \left(\epsilon \Omega_i^{(1)} + \epsilon^2 \Omega_i^{(2)} + O\left(\epsilon^3\right)\right) \\ \boldsymbol{F} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \Omega_i = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \left(\epsilon \Omega_i^{(1)} + \epsilon^2 \Omega_i^{(2)} + O\left(\epsilon^3\right)\right) \end{cases} \tag{2.52}$$

Hence, the zeroth order (with respect to velocity $\boldsymbol{\xi}$) macroscopic equations are given on the scales of ϵ and ϵ^2 respectively by collision invariant from Equation 2.47 and 2.48 as:

$$\begin{aligned} 0 = \sum_{i=0}^{q-1} \Omega_i^{(1)} &= \sum_{i=0}^{q-1} \left(\partial_{t1} + \nabla_1 \cdot \boldsymbol{\xi}_i\right) f_i^{(0)} \\ &= \partial_{t1}\left(\sum_{i=0}^{q-1} f_i^{(0)}\right) + \nabla_1 \cdot \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)}\right) \\ &= \partial_{t1}\rho + \nabla_1 \cdot \boldsymbol{j}^{(0)} \end{aligned} \tag{2.53}$$

for ϵ. Noticing $\boldsymbol{\xi}_i\boldsymbol{\xi}_i = Q_i + c_s^2\boldsymbol{I}$ and defining $\Pi = \sum_{i=0}^{q-1} Q_i f_i$, we have:

$$\begin{aligned}
0 &= \sum_{i=0}^{q-1} \Omega_i^{(2)} \\
&= \sum_{i=0}^{q-1}\Big(\left(\partial_{t1} + \nabla_1 \cdot \boldsymbol{\xi}_i\right) f_i^{(1)} + \left(\partial_{t2} + \nabla_2 \cdot \boldsymbol{\xi}_i\right) f_i^{(0)} + \frac{1}{2}\left(\partial_{t1}^2 + 2\partial_{t1}\nabla_1 \cdot \boldsymbol{\xi}_i + \nabla_1\nabla_1 : \boldsymbol{\xi}_i\boldsymbol{\xi}_i\right) f_i^{(0)}\Big) \\
&= \partial_{t1}\left(\sum_{i=0}^{q-1} f_i^{(0)}\right) + \nabla_1 \cdot \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(1)}\right) + \partial_{t2}\left(\sum_{i=0}^{q-1} f_i^{(0)}\right) + \nabla_2 \cdot \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)}\right) \\
&+ \frac{1}{2}\partial_{t1}^2\left(\sum_{i=0}^{q-1} f_i^{(0)}\right) + \partial_{t1}\nabla_1 \cdot \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)}\right) + \frac{1}{2}\nabla_1\nabla_1 : \left(\sum_{i=0}^{q-1}\left(Q_i + C_s^2\boldsymbol{I}\right) f_i^{(0)}\right) \\
&= \nabla_1 \cdot \boldsymbol{j}^{(1)} + \partial_{t2}\rho + \nabla_2 \cdot \boldsymbol{j}^{(0)} + \frac{1}{2}\partial_{t1}^2\rho + \partial_{t1}\nabla_1 \cdot \boldsymbol{j}^{(0)} + \frac{1}{2}\nabla_1\nabla_1 : \Pi^{(0)} + \frac{1}{2}c_s^2\nabla_1^2\rho \\
&= \nabla_1 \cdot \boldsymbol{j}^{(1)} + \partial_{t2}\rho + \nabla_2 \cdot \boldsymbol{j}^{(0)} + \frac{1}{2}\partial_{t1}\nabla_1 \cdot \boldsymbol{j}^{(0)} + \frac{1}{2}\nabla_1\nabla_1 : \Pi^{(0)} + \frac{1}{2}c_s^2\nabla_1^2\rho
\end{aligned} \tag{2.54}$$

for ϵ^2. While the first order (with respect to velocity $\boldsymbol{\xi}$) macroscopic equations are given on the scales of ϵ from Equations 2.46 as:

$$\begin{aligned}
\boldsymbol{F}^{(1)} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \Omega_i^{(1)} &= \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \left(\partial_{t1} + \nabla_1 \cdot \boldsymbol{\xi}_i\right) f_i^{(0)} \\
&= \partial_{t1}\left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)}\right) + \nabla_1 \cdot \left(\sum_{i=0}^{q-1}\left(Q_i + c_s^2\boldsymbol{I}\right) f_i^{(0)}\right) \\
&= \partial_{t1}\boldsymbol{j}^{(0)} + \nabla_1 \cdot \Pi^{(0)} + c_s^2\nabla_1\rho
\end{aligned} \tag{2.55}$$

for ϵ. From the equation we have:

$$\begin{cases}
\partial_{t1}\nabla_1 \cdot \boldsymbol{j}^{(0)} = \nabla_1 \cdot \boldsymbol{F}^{(1)} - \nabla_1\nabla_1 : \Pi^{(0)} - c_s^2\nabla_1^2\rho \\
\partial_{t1}^2\boldsymbol{j}^{(0)} = \partial_{t1}\boldsymbol{F}^{(1)} - \partial_{t1}\nabla_1 \cdot \Pi^{(0)} - c_s^2\partial_{t1}\nabla_1\rho
\end{cases} \tag{2.56}$$

substituting Equation 2.56 and Equation 2.51 into Equation 2.54, we obtain the simplified macroscopic equation with zeroth order of moment on the second scale Equation 2.54 as:

$$\begin{aligned}
0 = &-\frac{1}{2}\nabla_1 \cdot \boldsymbol{F}^{(1)} + \partial_{t2}\rho + \nabla_2 \cdot \boldsymbol{j}^{(0)} \\
&+ \frac{1}{2}\left(\nabla_1 \cdot \boldsymbol{F}^{(1)} - \nabla_1\nabla_1 : \Pi^{(0)} - c_s^2\nabla_1^2\rho\right) \\
&+ \frac{1}{2}\nabla_1\nabla_1 : \Pi^{(0)} + \frac{1}{2}c_s^2\nabla_1^2\rho = \partial_{t2}\rho + \nabla_2 \cdot \boldsymbol{j}^{(0)}
\end{aligned} \tag{2.57}$$

Equations 2.53 and 2.57 lead to the continuity equation as (noticing $\boldsymbol{j}^{(0)} = \rho\boldsymbol{u}$):

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{2.58}$$

defining

$$R = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i f_i \tag{2.59}$$

we have the first order (with respect to velocity $\boldsymbol{\xi}$) macroscopic equations given on the scales of ϵ^2 from Equations 2.47 as:

$$\begin{aligned}
0 &= \sum_{i=0}^{q-1} \boldsymbol{\xi}_i \Omega_i^{(2)} \\
&= \sum_{i=0}^{q-1} \left(\boldsymbol{\xi}_i \left(\partial_{t1} + \nabla_1 \cdot \boldsymbol{\xi}_i\right) f_i^{(1)} + \boldsymbol{\xi}_i \left(\partial_{t2} + \nabla_2 \cdot \boldsymbol{\xi}_i\right) f_i^{(0)} + \frac{1}{2} \boldsymbol{\xi}_i \left(\partial_{t1}^2 + 2\partial_{t1} \nabla_1 \cdot \boldsymbol{\xi}_i + \nabla_1 \nabla_1 : \boldsymbol{\xi}_i \boldsymbol{\xi}_i\right) f_i^{(0)} \right) \\
&= \partial_{t1} \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(1)} \right) + \nabla_1 \cdot \left(\sum_{i=0}^{q-1} \left(\Omega_i + c_s^2 \boldsymbol{I}\right) f_i^{(1)} \right) + \partial_{t2} \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)} \right) + \nabla_2 \cdot \left(\sum_{i=0}^{q-1} \left(\Omega_i + c_s^2 \boldsymbol{I}\right) f_i^{(0)} \right) \\
&+ \frac{1}{2} \partial_{t1}^2 \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{(0)} \right) + \partial_{t1} \nabla_1 \cdot \left(\sum_{i=0}^{q-1} \left(\Omega_i + c_s^2 \boldsymbol{I}\right) f_i^{(0)} \right) + \frac{1}{2} \nabla_1 \nabla_1 : \left(\sum_{i=0}^{q-1} \boldsymbol{\xi}_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i f_i^{(0)} \right) \\
&= \partial_{t1} \boldsymbol{j}^{(1)} + \nabla_1 \cdot \Pi^{(1)} + \partial_{t2} \boldsymbol{j}^{(0)} + \nabla_2 \cdot \Pi^{(0)} + c_s^2 \nabla_2 \rho \\
&+ \frac{1}{2} \partial_{t1}^2 \boldsymbol{j}^{(0)} + \partial_{t1} \nabla_1 \cdot \Pi^{(0)} + c_s^2 \partial_{t1} \nabla_1 \rho + \frac{1}{2} \nabla_1 \nabla_1 : R^{(0)} \\
&= -\frac{1}{2} \partial_{t1} \boldsymbol{F}^{(1)} + \nabla_1 \cdot \Pi^{(1)} + \partial_{t2} \boldsymbol{j}^{(0)} + \nabla_2 \cdot \Pi^{(0)} \\
&+ c_s^2 \nabla_2 \rho + \frac{1}{2} \left(\partial_{t1} \boldsymbol{F}^{(1)} - \partial_{t1} \nabla_1 \cdot \Pi^{(0)} - c_s^2 \partial_{t1} \nabla_1 \rho \right) \\
&+ \partial_{t1} \nabla_1 \cdot \Pi^{(0)} + c_s^2 \partial_{t1} \nabla_1 \rho + \frac{1}{2} \nabla_1 \nabla_1 : R^{(0)} \\
&= \nabla_1 \cdot \Pi^{(1)} + \partial_{t2} \boldsymbol{j}^{(0)} + \nabla_2 \cdot \Pi^{(0)} + c_s^2 \nabla_2 \rho \\
&+ \frac{1}{2} \left(\partial_{t1} \nabla_1 \cdot \Pi^{(0)} + c_s^2 \partial_{t1} \nabla_1 \rho + \nabla_1 \nabla_1 : R^{(0)} \right)
\end{aligned} \tag{2.60}$$

Therefore, for the first order of moment we have the full macroscopic equation by taking Equations (2.55) + $\epsilon^2 \times$ (2.60) as:

$$\begin{aligned}
\epsilon\boldsymbol{F}^{(1)} &= \left(\epsilon\partial_{t1} + \epsilon^2\partial_{t2}\right)\boldsymbol{j}^{(0)} + \frac{1}{2}\left(\epsilon\nabla_1 + \epsilon^2\nabla_2\right)\cdot\left(\epsilon\partial_{t1}\Pi^{(0)} + c_s^2\epsilon\partial_{t1}\rho\boldsymbol{I} + \epsilon\nabla_1\cdot R^{(0)}\right) \\
&- \frac{1}{2}\epsilon^2\nabla_2\cdot\left(\epsilon\partial_{t1}\Pi^{(0)} + c_s^2\epsilon\partial_{t1}\rho\boldsymbol{I} + \epsilon\nabla_1\cdot R^{(0)}\right) + \left(c_s^2\left(\epsilon\nabla_1 + \epsilon^2\nabla_2\right)\rho\right) \\
&+ \left(\epsilon\nabla_1\cdot\left(\Pi^{(0)} + \epsilon\Pi^{(1)}\right) + \epsilon^2\nabla_2\cdot\Pi^{(0)} + \epsilon^2\nabla_2\cdot\left(\epsilon\Pi^{(1)}\right) - \epsilon^2\nabla_2\cdot\left(\epsilon\Pi^{(1)}\right)\right) \\
&= \partial_t\boldsymbol{j}^{(0)} + \frac{1}{2}\nabla\cdot\left(\epsilon\partial_{t1}\Pi^{(0)} + c_s^2\epsilon\partial_{t1}\rho\boldsymbol{I} + \epsilon\nabla_1\cdot R^{(0)}\right) + c_s^2\nabla\rho + \nabla\cdot\Pi + O\left(\epsilon^3\right)
\end{aligned} \tag{2.61}$$

from the above equation, we can see that if the spatial derivative is expanded up to the first order $\nabla = \epsilon\nabla_1$, all of the terms with ∇_2 vanish, and we can also have the accuracy up to the second order of Knudsen number. Moreover, in order to have the second order accuracy, temporal derivative up to the second order is necessary. This multi-scale derivation could explain theoretically why we may only provide expansion of the spatial derivative to the first order of Knudsen number while the second order is needed for temporal expansion.

By the definition of equilibrium distribution function, we can compute $\boldsymbol{j}^{(0)}$, $\Pi^{(0)}$, $R^{(0)}$ explicitly provided the following lattice symmetries (which are related to the orthonormal properties of Hermite polynomials):

$$\begin{cases}
\sum_{i=0}^{q-1} w_i = 1, \ \sum_{i=0}^{q-1} w_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i = c_s^2\boldsymbol{I} = c_s^2\delta_{\alpha_1\alpha_2} \\
\sum_{i=0}^{q-1} w_i\boldsymbol{\xi}_i = 0, \ \sum_{i=0}^{q-1} w_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i = 0, \ \sum_{i=0}^{q-1} w_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i = 0 \\
\sum_{i=0}^{q-1} w_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i = 3c_s^4\boldsymbol{I}\boldsymbol{I} = c_s^4\left(\delta_{\alpha_1\alpha_2}\delta_{\alpha_3\alpha_4} + \delta_{\alpha_1\alpha_3}\delta_{\alpha_2\alpha_4} + \delta_{\alpha_1\alpha_4}\delta_{\alpha_2\alpha_3}\right)
\end{cases} \tag{2.62}$$

where δ is a Kronecker delta. Therefore $\boldsymbol{j}^{(0)}$, $\Pi^{(0)}$ and $R^{(0)}$ are given by:

$$\begin{cases}
\boldsymbol{j}^{(0)} = \sum_{i=0}^{q-1}\boldsymbol{\xi}_i f^{(0)} = \rho\boldsymbol{u} \\
\Pi^{(0)} = \sum_{i=0}^{q-1} Q_i f_i^{(0)} = \rho\boldsymbol{u}\boldsymbol{u} \\
\nabla_1\cdot R^{(0)} = \sum_{i=0}^{q-1}\boldsymbol{\xi}_i\boldsymbol{\xi}_i\boldsymbol{\xi}_i f_i^{(0)} = c_s^2\left(\nabla_1\left(\rho\boldsymbol{u}\right) + \left(\nabla_1\left(\rho\boldsymbol{u}\right)\right)^{\mathrm{T}} + \nabla_1\cdot\left(\rho\boldsymbol{u}\right)\boldsymbol{I}\right)
\end{cases} \tag{2.63}$$

Equation 2.61 is simplified with accuracy up to the second order of Knudsen number as:

$$\begin{aligned}
\boldsymbol{F} &= \partial_t \boldsymbol{j}^{(0)} + \nabla \cdot \left(\Pi^{(0)} + \epsilon \Pi^{(1)} + c_s^2 \rho \boldsymbol{I} + \frac{\epsilon}{2} \left(\partial_{t1} \left(\Pi^{(0)} + c_s^2 \rho \boldsymbol{I} \right) \right) + \nabla_1 \cdot R^{(0)} \right) \\
&= \partial_t \left(\rho \boldsymbol{u} \right) + \nabla \cdot \left(\rho \boldsymbol{u} \boldsymbol{u} + c_s^2 \rho \boldsymbol{I} + \left(\epsilon \Pi^{(0)} + \frac{\epsilon}{2} \left(\Pi^{(0)} \right) \right) \right) \\
&+ \nabla \cdot \left(\frac{1}{2} c_s^2 \left(\nabla \left(\rho \boldsymbol{u} \right) + \left(\nabla \left(\rho \boldsymbol{u} \right) \right)^{\mathrm{T}} \right) + \frac{1}{2} c_s^2 \left(\epsilon \partial_{t1} \rho + \nabla \cdot \left(\rho \boldsymbol{u} \right) \right) \boldsymbol{I} \right)
\end{aligned} \tag{2.64}$$

which is explicit expressed except for the first order stress tensor $\Pi^{(1)}$, left to be approximated later on. Recall the full Navier-Stokes equations in the following form:

$$\begin{cases} \partial_t \rho + \nabla \cdot \left(\rho \boldsymbol{u} \right) = 0 \\ \partial_t \left(\rho \boldsymbol{u} \right) + \nabla \cdot \left(\rho \boldsymbol{u} \boldsymbol{u} + \rho \boldsymbol{I} - \boldsymbol{\tau} \right) = \rho \boldsymbol{g} \end{cases} \tag{2.65}$$

where the deviatoric stresses $\boldsymbol{\tau}$ is defined as:

$$\boldsymbol{\tau} = \rho \nu \left(\nabla \boldsymbol{u} + \left(\nabla \boldsymbol{u} \right)^{\mathrm{T}} \right) \tag{2.66}$$

where ν is the dynamic shear viscosity, or with an additional term:

$$\boldsymbol{\tau} = -\rho \nu' \left(\nabla \cdot \boldsymbol{u} \right) \boldsymbol{I} + \rho \nu \left(\nabla \boldsymbol{u} + \left(\nabla \boldsymbol{u} \right)^{\mathrm{T}} \right) \tag{2.67}$$

where ν' is the dynamic bulk viscosity. Comparatively, the continuity equation is the same as we can see from Equation 2.58 and Equation 2.67, while the momentum equations are the same in Equation 2.64 and Equation 2.67 except for the item of deviatoric stresses. In order to retrieve the exact Navier-Stokes equations, the so called Chapman-Enskog ansatz is applied to the collision operator to achieve the term with dynamic shear viscosity and an additional term with dynamic bulk viscosity respectively.

2.5 Chapman-Enskog ansatz

Suppose that the collision operator is given in the form of a BGK relaxation term as:

$$\Omega_i \left(\boldsymbol{x}, t \right) = -\frac{1}{\tau} \left(f_i \left(\boldsymbol{x}, t \right) - f_i^{(0)} \left(\boldsymbol{x}, t \right) \right) + F_i \left(\boldsymbol{x}, t \right) + C_i \left(\boldsymbol{x}, t \right) \tag{2.68}$$

where F_i is the force term and C_i is the correction term and they are given by:

$$\begin{cases} F_i\left(\boldsymbol{x},t\right) = \dfrac{a}{c_s^2} w_i \boldsymbol{\xi}_i \cdot \boldsymbol{F} + \dfrac{b}{2c_s^4} w_i Q_i : \left(\boldsymbol{F}\boldsymbol{u} + \boldsymbol{u}\boldsymbol{F}\right) \\ C_i\left(\boldsymbol{x},t\right) = \dfrac{c}{2c_s^4} w_i Q_i : \delta\Pi \end{cases} \tag{2.69}$$

where the coefficients a, b and c are left to be determined. We will show in several steps that the force term F_i is responsible for the dynamic shear viscosity ν while the additional correction term C_i is responsible for the dynamic bulk viscosity ν'. Firstly, corresponding to the scale expansion of force $\boldsymbol{F}$ and second order tensor $\delta\Pi$, we have on the first scale of the expansion for force term and correction term that:

$$\begin{cases} F_i^{(1)}\left(\boldsymbol{x},t\right) = \dfrac{a}{c_s^2} w_i \boldsymbol{\xi}_i \cdot \boldsymbol{F}^{(1)} + \dfrac{b}{2c_s^4} w_i Q_i : \left(\boldsymbol{F}^{(1)}\boldsymbol{u} + \boldsymbol{u}\boldsymbol{F}^{(1)}\right) \\ C_i^{(1)}\left(\boldsymbol{x},t\right) = \dfrac{c}{2c_s^4} w_i Q_i : \delta\Pi^{(1)} \end{cases} \tag{2.70}$$

in order to determine what coefficients should be, we will examine the mass conservation law, the momentum conservation law and beyond for the higher order moments.

For the zeroth order moment $\boldsymbol{\xi}^0$, we have:

$$\begin{aligned} 0 = \sum_{i=0}^{q-1} \Omega_i &= -\frac{1}{\tau} \sum_{i=0}^{q-1} \left(f_i\left(\boldsymbol{x},t\right) - f_i^{(0)}\left(\boldsymbol{x},t\right) \right) + \sum_{i=0}^{q-1} F_i\left(\boldsymbol{x},t\right) + \sum_{i=0}^{q-1} C_i\left(\boldsymbol{x},t\right) \\ &= -\frac{1}{\tau}\left(\rho - \rho\right) + \frac{a}{c_s^2} \underbrace{\sum_{i=0}^{q-1} w_i \boldsymbol{\xi}_i}_{\text{Term 1}} \cdot \boldsymbol{F} + \frac{b}{2c_s^4} \underbrace{\sum_{i=0}^{q-1} w_i Q_i}_{\text{Term 2}} : \left(\boldsymbol{F}\boldsymbol{u} + \boldsymbol{u}\boldsymbol{F}\right) + \frac{c}{2c_s^4} \underbrace{\sum_{i=0}^{q-1} w_i Q_i}_{\text{Term 3}} : \delta\Pi = 0 \end{aligned} \tag{2.71}$$

where Term 1 = Term 2 = Term 3 = 0, because of the lattice symmetries Equation 2.62.

For the zeroth order moment $\boldsymbol{\xi}^1$, we have:

$$\begin{aligned} \boldsymbol{F} = \sum_{i=0}^{q-1} \boldsymbol{\xi}\Omega_i &= -\frac{1}{\tau} \sum_{i=0}^{q-1} \left(\boldsymbol{\xi} f_i\left(\boldsymbol{x},t\right) - \boldsymbol{\xi} f_i^{(0)}\left(\boldsymbol{x},t\right) \right) + \sum_{i=0}^{q-1} \boldsymbol{\xi} F_i\left(\boldsymbol{x},t\right) + \sum_{i=0}^{q-1} \boldsymbol{\xi} C_i\left(\boldsymbol{x},t\right) \\ &= -\frac{1}{\tau} \underbrace{\left(\boldsymbol{j} - \boldsymbol{j}^{(0)}\right)}_{=-\boldsymbol{F}/2} + \frac{a}{c_s^2} \underbrace{\sum_{i=0}^{q-1} w_i \boldsymbol{\xi}\boldsymbol{\xi}_i}_{\text{Term 1}} \cdot \boldsymbol{F} + \frac{b}{2c_s^4} \underbrace{\sum_{i=0}^{q-1} w_i \boldsymbol{\xi} Q_i}_{\text{Term 2}} : \left(\boldsymbol{F}\boldsymbol{u} + \boldsymbol{u}\boldsymbol{F}\right) \\ &+ \frac{c}{2c_s^4} \underbrace{\sum_{i=0}^{q-1} w_i \boldsymbol{\xi} Q_i}_{\text{Term 3}} : \delta\Pi = \left(a + \frac{1}{2\tau}\right)\boldsymbol{F} \end{aligned} \tag{2.72}$$

where Term $1 = c_s^2 \boldsymbol{I},$ Term $2 =$ Term $3 = 0$, because of the lattice symmetries Equation 2.62. From the above equation, we have for the first coefficient $a = 1 - 1\left(2\tau\right)$.

With the definition $\Pi = \sum_{i=0}^{q-1} Q_i f_i$, we have that $\Pi^{(1)} = \sum_{i=0}^{q-1} Q_i f_i^{(1)}$ holds on the first order of expansion. It is left to compute $f_i^{(1)}$ in order to obtain $\Pi^{(1)}$ for Equation 2.64. By the expansion of distribution function and operator, we have:

$$\epsilon \Omega_i^{(1)} + O\left(\epsilon^2\right) = \Omega_i = -\frac{1}{\tau} \epsilon f_i^{(1)} + \epsilon F_i^{(1)} + \epsilon C_i^{(1)} + O\left(\epsilon^2\right) \tag{2.73}$$

together with Equation 2.47 and equilibrium distribution function, we have:

$$\begin{aligned} f_i^{(1)} &= -\tau \Omega_i^{(1)} + \tau \left(F^{(1)_i} + C_i^{(1)}\right) = -\tau \left(\partial_{t1} + \nabla_1 \cdot \boldsymbol{\xi}_i\right) f_i^{(0)} + \tau \left(F_i^{(1)} + C_i^{(1)}\right) \\ &= -w_i \tau \left(\partial_{t1} + \nabla_1 \cdot \boldsymbol{\xi}_i\right) \left(\rho + \frac{\boldsymbol{\xi}_i \cdot (\rho \boldsymbol{u})}{c_s^2} + \frac{1}{2c_s^4} Q_i : \rho \boldsymbol{u}\boldsymbol{u}\right) + \tau \left(F_i^{(1)} + C_i^{(1)}\right) \\ &= -w_i \tau \left(\partial_{t1} \rho + \frac{1}{c_s^2} \partial_{t1} \left(\boldsymbol{\xi}_i \cdot \rho \boldsymbol{u}\right) + \frac{1}{2c_s^4} \partial_{t1} \left(Q_i : \rho \boldsymbol{u}\boldsymbol{u}\right)\right. \\ &\left. + \boldsymbol{\xi}_i \cdot \nabla_1 \rho + \frac{1}{c_s^2} \boldsymbol{\xi}_i \boldsymbol{\xi}_i : \nabla_1 \left(\rho \boldsymbol{u}\right) + \frac{1}{2c_s^4} \left(\boldsymbol{\xi}_i \cdot \nabla_1\right) \left(Q_i : \rho \boldsymbol{u}\boldsymbol{u}\right)\right) + \tau \left(F_i^{(1)} + C_i^{(1)}\right) \end{aligned} \tag{2.74}$$

for the three temporal derivatives, by virtue of continuity equation and momentum equation as well as assumption of weak compressibility, i.e. $\partial_t \rho = O\left(\mathrm{Ma}\right)$, we have:

$$\partial_{t1} \rho = -\nabla_1 \cdot \left(\rho \boldsymbol{u}\right) \tag{2.75}$$

by Equation 2.53,

$$\frac{1}{c_s^2} \partial_{t1} \left(\boldsymbol{\xi}_i \cdot \rho \boldsymbol{u}\right) = \frac{1}{c_s^2} \boldsymbol{\xi}_i \cdot F^{(1)} - \frac{1}{c_s^2} \boldsymbol{\xi}_i \nabla_1 : \rho \boldsymbol{u}\boldsymbol{u} - \boldsymbol{\xi}_i \cdot \nabla_1 \rho \tag{2.76}$$

$$\begin{aligned} \frac{1}{2c_s^4} \partial_{t1} \left(Q_i : \rho \boldsymbol{u}\boldsymbol{u}\right) &= \frac{1}{2c_s^4} Q_i : \left(\partial_{t1} \left(\rho \boldsymbol{u}\right) \boldsymbol{u} + \boldsymbol{u} \partial_{t1} \left(\rho \boldsymbol{u}\right) - \underbrace{\partial_{t1} \left(\rho\right) \boldsymbol{u}\boldsymbol{u}}_{O(\mathrm{Ma}^{1+2})}\right) \\ &= \frac{1}{c_s^4} Q_i : \partial_{t1} \left(\rho \boldsymbol{u}\right) \boldsymbol{u} \qquad \left(Q_i \text{ is symmetric}\right) \\ &= \frac{1}{c_s^4} Q_i : \left(F^{(1)} - \underbrace{\nabla_1 \cdot \left(\rho \boldsymbol{u}\boldsymbol{u}\right)}_{O(\mathrm{Ma}^2)} - c_s^2 \nabla_1 \rho\right) \underbrace{\boldsymbol{u}}_{O(\mathrm{Ma})} \\ &= \frac{1}{c_s^4} Q_i : \left(F^{(1)} - c_s^2 \nabla_1 \rho\right) \boldsymbol{u} \end{aligned} \tag{2.77}$$

by Equation 2.55. By substituting the three temporal derivatives into Equation 2.74, neglecting the terms of $O\left(\mathrm{Ma}^3\right)$, the distribution function on the scale of first order of Knudsen number becomes:

$$\begin{aligned} f_i^{(1)} = & -\frac{w_i\tau}{c_s^2}\left(Q_i : \rho\nabla_1\boldsymbol{u} - \boldsymbol{\xi}_i\nabla_1 : \rho\boldsymbol{u}\boldsymbol{u} + \frac{1}{2c_s^2}\left(\boldsymbol{\xi}_i\cdot\nabla_1\right)\left(Q_i : \rho\boldsymbol{u}\boldsymbol{u}\right)\right) \\ & - \frac{3w_i\tau}{2c_s^2}\boldsymbol{\xi}_i\cdot\boldsymbol{F}^{(1)} + \frac{(b-1)\,w_i\tau}{2c_s^4}Q_i : \left(\boldsymbol{F}^{(1)}\boldsymbol{u} + \boldsymbol{u}\boldsymbol{F}^{(1)}\right) + \frac{cw_i\tau}{2c_s^4\epsilon}Q_i : \delta\Pi \end{aligned} \tag{2.78}$$

Therefore, with the lattice symmetries Equation 2.62 we obtain the explicit expression for $\Pi^{(1)}$ as:

$$\Pi^{(1)} = \sum_{i=0}^{q-1} Q_i f_i^{(1)} = -\frac{c_s^2\tau}{\epsilon}\rho\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{(\mathrm{T})}\right) + (b-1)\,\tau\left(\boldsymbol{F}^{(1)}\boldsymbol{u} + \boldsymbol{u}\boldsymbol{F}^{(1)}\right) + \frac{c\tau}{\epsilon}\delta\Pi \tag{2.79}$$

From the calculation of Equation 2.77, we can also obtain by Equation 2.53:

$$\partial_{t1}\Pi^{(0)} = \partial_{t1}\left(\rho\boldsymbol{u}\boldsymbol{u}\right) = \boldsymbol{F}^{(1)}\boldsymbol{u} + \boldsymbol{u}\boldsymbol{F}^{(1)} - c_s^2\left(\left(\nabla_1\rho\right)\boldsymbol{u} + \boldsymbol{u}\left(\nabla_1\rho\right)\right) \tag{2.80}$$

$\epsilon \times (2.79) + \epsilon \times (2.80)\,/2$ leads to:

$$\begin{aligned} \epsilon\Pi^{(1)} + \frac{\epsilon}{2}\partial_{t1}\Pi^{(0)} = & \left(\underbrace{\tau\left(b-1\right) + \frac{1}{2}}_{\text{Term 1}}\right)\left(\boldsymbol{F}\boldsymbol{u} + \boldsymbol{u}\boldsymbol{F}\right) - c_s^2\tau\rho\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\mathrm{T}}\right) \\ & - \frac{1}{2}c_s^2\left(\left(\nabla\rho\right)\boldsymbol{u} + \boldsymbol{u}\left(\nabla\rho\right)\right) + c\tau\delta\Pi \end{aligned} \tag{2.81}$$

taking $b = 1 - 1\left(2\tau\right)$ so as to eliminate Term 1, substituting the above equation into Equation 2.64, we get:

$$\begin{aligned} \boldsymbol{F} = & \,\partial_t\left(\rho\boldsymbol{u}\right) + \left(\rho\boldsymbol{u}\boldsymbol{u} + c_s^2\rho\boldsymbol{I} - c_s^2\tau\rho\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\mathrm{T}}\right) + c\tau\delta\Pi\right) \\ & + \nabla\cdot\left(\frac{1}{2}c_s^2\left(\nabla\left(\rho\boldsymbol{u}\right) + \left(\nabla\left(\rho\boldsymbol{u}\right)\right)^{\mathrm{T}}\right) - \frac{1}{2}c_s^2\left(\left(\nabla\rho\right)\boldsymbol{u} + \boldsymbol{u}\left(\nabla\rho\right)\right)\right) \\ & + \nabla\cdot\left(\frac{1}{2}c_s^2\left(\underbrace{\partial_t\rho + \nabla\cdot\left(\rho\boldsymbol{u}\right)}_{\text{continuity equation=0}} - \epsilon^2\underbrace{\partial_{t2}\rho}_{O(\mathrm{Ma})}\right)\boldsymbol{I}\right) \\ = & \,\partial_t\left(\rho\boldsymbol{u}\right) + \nabla\cdot\left(\rho\boldsymbol{u}\boldsymbol{u} + c_s^2\rho\boldsymbol{I} - c_s^2\left(\tau - \frac{1}{2}\right)\rho\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\mathrm{T}}\right) + c\tau\delta\Pi\right) + O\left(\epsilon^2\mathrm{Ma}\right) \\ & \,\partial_t\left(\rho\boldsymbol{u}\right) + \nabla\cdot\left(\rho\boldsymbol{u}\boldsymbol{u} + c_s^2\rho\boldsymbol{I} - \boldsymbol{\tau}\right) \end{aligned} \tag{2.82}$$

with accuracy order up to $\epsilon^2 \times \mathrm{Ma}$, where we denote:

$$\boldsymbol{\tau} = 2\nu\rho S - c\tau\delta\Pi, \quad S = \frac{1}{2}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\mathrm{T}}\right), \quad \nu = c_s^2\left(\tau - \frac{1}{2}\right) \tag{2.83}$$

If $c = 0$, we recover by Equations 2.82 and 2.83 the same Navier-Stokes equations. As the side product, we obtain the force term as:

$$F_i = w_i \left(1 - \frac{1}{2\tau}\right)\left(\frac{\boldsymbol{\xi}_i \cdot \boldsymbol{F}}{c_s^2} + \frac{Q_i : (\boldsymbol{F}\boldsymbol{u})}{c_s^4}\right) \tag{2.84}$$

Equation 2.82 is the Navier-Stokes equations with dynamic share viscosity but not bulk viscosity. To retrieve the dynamic bulk viscosity, the correction term C_i has to be taken into account. Suppose the correction term as:

$$\delta\Pi = \frac{\epsilon}{D}\left(\mathrm{Tr}\left(\Pi^{(1)}\right) + k\boldsymbol{F}^{(1)} \cdot \boldsymbol{u}\right)\boldsymbol{I} \tag{2.85}$$

where D is the dimension of the system, $\mathrm{Tr}\left(\Pi^{(1)}\right)$ is the trace of the tensor $\Pi^{(1)}$ and k is a coefficient to be determined. In order to have $\mathrm{Tr}\left(\Pi^{(1)}\right)$, taking the trace of Equation 2.79 yields:

$$\begin{aligned}\mathrm{Tr}\left(\Pi^{(1)}\right) &= \frac{c\tau}{\epsilon}\mathrm{Tr}\left(\delta\Pi\right) - 2c_s^2\tau\rho\nabla_1 \cdot \boldsymbol{u} - \boldsymbol{F}^{(1)} \cdot \boldsymbol{u} \\ &= \frac{c\tau}{\epsilon}\left(\mathrm{Tr}\left(\Pi^{(1)}\right) + k\boldsymbol{F}^{(1)} \cdot \boldsymbol{u}\right) - 2c_s^2\tau\rho\nabla_1 \cdot \boldsymbol{u} - \boldsymbol{F}^{(1)} \cdot \boldsymbol{u}\end{aligned} \tag{2.86}$$

from the above equation, we have:

$$\mathrm{Tr}\left(\Pi^{(1)}\right) = \frac{1}{1 - c\tau}\left(\left(c\tau k - 1\right)\boldsymbol{F}^{(1)} \cdot \boldsymbol{u} - 2c_s^2\tau\rho\nabla_1 \cdot \boldsymbol{u}\right) \tag{2.87}$$

hence, the partial correction term turns out to be:

$$c\tau\delta\Pi = \frac{\epsilon}{1 - c\tau}\frac{c\tau}{D}\left(\underbrace{\left(\left(1 + c\tau\right)k - 1\right)}_{\text{Term 1}}\boldsymbol{F}^{(1)} \cdot \boldsymbol{u} - 2c_s^2\tau\rho\nabla_1 \cdot \boldsymbol{u}\right)\boldsymbol{I} \tag{2.88}$$

When

$$k = 1/\left(1 + c\tau\right) \tag{2.89}$$

Term 1 vanishes, and choose c as:

$$c = \frac{1}{\tau - 2c_s^2\tau^2/\left(\mathrm{d}\nu'\right)} \tag{2.90}$$

we have eventually:

$$c\tau\delta\Pi = \rho\nu'\nabla \cdot \boldsymbol{u}\boldsymbol{I} \tag{2.91}$$

which is responsible for the dynamic bulk viscosity in Equation 2.82 such that:

$$\boldsymbol{\tau} = -\rho\nu'\left(\nabla \cdot \boldsymbol{u}\right)\boldsymbol{I} + 2\rho\nu S \tag{2.92}$$

As a result, the full correction term is:

$$
\begin{aligned}
C_i &= \frac{cw_i}{2Dc_s^4}\left(\mathrm{Tr}\left(\epsilon\Pi^{(1)}\right) + k\epsilon\boldsymbol{F}^{(1)}\cdot\boldsymbol{u}\right)\left(Q_i : \boldsymbol{I}\right) \\
&= \frac{cw_i}{2Dc_s^4}\left(\mathrm{Tr}\left(\Pi - \Pi^{(0)}\right) + k\boldsymbol{F}\cdot\boldsymbol{u}\right)\left(\xi_i^2 - c_s^2 D\right) + O\left(\epsilon^2\right) \\
&= \frac{cw_i}{2Dc_s^4}\left(\mathrm{Tr}\left(\sum_{i=0}^{q-1}\boldsymbol{\xi}_i\boldsymbol{\xi}_i f_i - \rho\boldsymbol{u}\boldsymbol{u} - \rho c_s^2\boldsymbol{I}\right) + k\boldsymbol{F}\cdot\boldsymbol{u}\right)\left(\boldsymbol{\xi}_i^2 - c_s^2 D\right) + O\left(\epsilon^2\right) \\
&= \frac{cw_i}{2Dc_s^4}\left(\sum_{i=0}^{q-1}\boldsymbol{\xi}_i^2 f_i - \rho\boldsymbol{u}^2 - \rho c_s^2 d + k\boldsymbol{F}\cdot\boldsymbol{u}\right)\left(\boldsymbol{\xi}_i^2 - c_s^2 D\right) + O\left(\epsilon^2\right)
\end{aligned}
\tag{2.93}
$$

with k and c given in Equations 2.89 and 2.90. Equation 2.93 shows that the correction term finds itself up to the scale of the first order of the Knudsen number.

In summary, the Chapman-Enskog approach is based on the asymptotic multi-scale analysis of the lattice Boltzmann equation and sheds light on the second order accuracy with respect to Mach number of retrieving Navier-Stokes equations up to the scale of the first order of Knudsen number. In order to obtain the continuity equation for mass conservation and the momentum equation for momentum conservation as in the full Navier-Stokes equations, the following three main steps are carried our. First of all, expand the temporal and spatial variables up to the second order of Knudsen number while the distribution function, density as well as momentum up to the first order of Knudsen number. Secondly, by examining the lattice Boltzmann - BGK equation on the scale of zeroth order, first order and second order of Knudsen number with the truncated equilibrium distribution to the second order of Mach number, we have the exact continuity equation corresponding to mass conversation and momentum conservation for the Navier-Stokes equations. However, the stress tensor for Navier-Stokes equations is unknown on the scale of first order of Knudsen number. Finally, by ansatz of the collision operator, including the force term and the correction term for dynamic bulk viscosity, we succeed in closing the full Navier-Stokes equations with not only the shear viscosity but also the desirable bulk viscosity from appropriate choice of force term and correction term.

2.6 The basic calculation algorithm

The basic calculation algorithm of the lattice Boltzmann method consists of two steps, the streaming step and the collision step as described above sections. These are usually combined with solid wall [IYO95, ZH97, MBG96, BFL01] or free surface boundary conditions [KTH^{+}05] for an obstacle and gas-liquid interface in calculation domain. This simplicity of the algorithm is especially evident when implementing it, which, for the basic algorithm, requires roughly a single page of fortran or C-code. Using the LBM, the virtual particle movement is restricted into a limited number of directions. In this thesis, We will use the three-dimensional 19-speed square lattice (D3Q19) model for three-dimensional fluid simulation. Alternatives are models with 15 or 27 velocities. However, the latter one has no apparent advantages over the 19 velocity model, while the model with 15 velocities has a decreased stability. The D3Q19 model is thus usually preferable as it requires less memory than the 27 velocity model. The D3Q19 model with its lattice velocity vectors $\boldsymbol{e}_{0,\ldots,18}$ is shown in Figure 2.1 The velocity vectors take the following values:

$$\boldsymbol{e}_i = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{bmatrix} \tag{2.94}$$

For each of the velocities, a floating point number $f_{0,\ldots,18}$, representing the fraction of particles moving with this velocity, needs to be stored. In the D3Q19 model there are particles not moving at all (f_0), moving with speed 1 ($f_{1,\ldots,6}$) and moving with speed $\sqrt{2}$ ($f_{7,\ldots,18}$). In the following, a subscript of I will denote the value from the inverse direction of a value with subscript i. Thus, f_i and f_I are opposite distribution functions with inverse velocity vectors $\boldsymbol{e}_I = -\boldsymbol{e}_i$. During the first part of the algorithm (the streaming step), all distribution functions are advected with their respective velocities. This propagation results in a movement of the floating point values to the neighboring cells, as shown in Figure 2.2. Formulated in terms of distribution functions the streaming step can be written as:

$$f_i^* \left(\boldsymbol{x}, t + \Delta t\right) = f_i \left(\boldsymbol{x} + \boldsymbol{e}_I \Delta t, t\right) \tag{2.95}$$

Here, Δx denotes the size of a cell and Δt the time step interval. Both are normalised by the condition $\Delta x / \Delta t = 1$, which makes it possible to handle the advection by a simple copying operation. These post-streaming distribution functions f_i^* have to be distinguished from the standard distribution functions f_i, and are never really stored in the grid. The streaming step alone is clearly not enough to simulate the behaviour of incompressible fluids, which is governed by the ongoing collisions of the particles with each other. The second step of the lattice Boltzmann method, the collision step, amounts for this by weighting the distribution functions of a grid with the so called the equilibrium distribution functions, denoted by f_i^{eq}. These depend solely on the macroscopic density ρ and velocity $\boldsymbol{u}$ of the fluid. The density and velocity can be calculated by the summation of all the distribution functions for one grid:

$$\rho = \sum_i f_i \tag{2.96}$$

$$\rho \boldsymbol{u} = \sum_i \boldsymbol{e}_i f_i \tag{2.97}$$

for a single direction i, the equilibrium distribution functions f_i^{eq} are determined as:

$$f_i^{eq} = w_i \rho \left[1 + 3 \boldsymbol{e}_i \cdot \boldsymbol{u} + \frac{9}{2} \left(\boldsymbol{e}_i \cdot \boldsymbol{u} \right)^2 - \frac{3}{2} \left(\boldsymbol{u} \cdot \boldsymbol{u} \right) \right] \tag{2.98}$$

where w_i is a weighting parameter given as:

$$w_i = \begin{cases} 1/3 & (i = 0) \\ 1/18 & (i = 1, \ldots, 6) \\ 1/36 & (i = 7, \ldots, 18) \end{cases} \tag{2.99}$$

for D3Q19 square lattice model.

The equilibrium distribution functions represent a stationary state of the fluid. However, this does not mean that the fluid in not moving, but only that the values of the distribution functions would not change, if the whole fluid was at an equilibrium state. For very viscous flows, such an equilibrium state can be globally reached. In this case, the distribution functions will converge towards constant values. The collision of the molecules in a real fluid are approximated by linear relaxing the distribution functions of a cell towards their equilibrium state. Thus, each distribution function f_i is weighted with the corresponding f_i^{eq} using the lattice Boltzmann equation based on Boltzmann-BGK approach:

$$f_i\left(\boldsymbol{x}, t+\Delta t\right) = f_i^*\left(\boldsymbol{x}, t+\Delta t\right) - \frac{1}{\tau}\left[f_i^*\left(\boldsymbol{x}, t+\Delta t\right) - f_i^{eq}\right] \tag{2.100}$$

Here, τ is the single relaxation time that controls the viscosity of the fluid. Often, $\omega = 1/\tau$ is also used to denote the lattice viscosity. This parameter is in the range of $(0, 2]$, where values close to 0 result in very viscous fluids, while values near 2 result in more turbulent flows. However, for values close to 2, the lattice Boltzmann method can become numerical instability.

2.7 Compressibility effect of the lattice Boltzmann method

As shown in the above sections, the continuity equation in the simplified lattice Boltzmann method introduces a compressibility effect which become critical for the simulation of the incompressible flows. In order to reduce the compressibility effect for the time dependent flows, we need the following incompressible lattice Boltzmann approach [HL97], a slight modification of the compressible one with respect to the density.

It is well understood that in incompressible flows the density is a constant, say ρ_0, and the density fluctuation $\delta\rho$, should be of the order $O\left(\mathrm{Ma}^2\right)$ in the limit of Ma → 0. If we explicitly substitute $\rho = \rho_0 + \delta\rho$ into the equilibrium distribution function, and neglect the terms proportional to $\delta\rho\left(\boldsymbol{u}/c\right)$ and $\delta\rho\left(\boldsymbol{u}/c\right)^2$, which are of the order $O\left(\mathrm{Ma}^3\right)$ or higher, then the equilibrium distribution function becomes:

$$f_i^{eq} = f_i^{(0)}\left(\boldsymbol{x}, t\right) = w_i\left[\rho + \rho_0\left(\frac{(\boldsymbol{e}_i \cdot \boldsymbol{u})}{c_s^2} + \frac{(\boldsymbol{e}_i \cdot \boldsymbol{u})^2}{2c_s^4} - \frac{(\boldsymbol{u} \cdot \boldsymbol{u})}{2c_s^2}\right)\right] \tag{2.101}$$

while to recover the velocity from distribution function, we have:

$$\rho_0 \boldsymbol{u} = \sum_i \boldsymbol{e}_i f_i \tag{2.102}$$

The above distribution function is the equilibrium distribution function of the incompressible lattice Boltzmann model. By the same Chapman-Enskog expansion procedure, we can obtain the incompressible Navier-Stokes equations accurate up to the second order of Mach number $O\left(\mathrm{Ma}^2\right)$ for continuity equation and third order of Mach number $O\left(\mathrm{Ma}^3\right)$ for momentum equations as:

$$\begin{cases} \nabla \cdot \boldsymbol{u} = 0 + O\left(\mathrm{Ma}^2\right) \\ \partial_t \boldsymbol{u} + \left(\boldsymbol{u} \cdot \nabla\right) \boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{g} + O\left(\mathrm{Ma}^3\right) \end{cases} \tag{2.103}$$

Notice that the above system is still biased from standard Navier-Stokes equations by the compressibility effect related to Mach number. In order to reduce this compressibility effect, small Mach number is preferable, which means that the time step Δt has to be small with respect to the space step Δx.

2.8 The multiple-relaxation-time collision model

The simplest collision model - single relaxation BGK approach easily becomes numerical unstable in high-Reynolds-number flow. To overcome the defects of the lattice BGK model, the multiple-relaxation-time (MRT) collision approach is receiving a lot of attention in recent years.

The MRT lattice Boltzmann collision approach was also developed at the same time with the BGK model. The MRT lattice Boltzmann equation overcomes some obvious defects of the BGK model, such as fixed Prandtl number (Pr = 1 for the BGK model) and fixed ratio between the kinetic an bulk viscosities. The MRT lattice Boltzmann equation has been persistently pursued, and much progress has been made. Successes include formulation of optimal boundary conditions [GA94, Lad94], interface conditions in multiphase flows [GA95, SPT17] and free surfaces [GS02, JK11], thermal [MGA95] and viscoelastic models [GDL97, GDL98], models with reduced lattice symmetries [DBL01, BDLL02], and improvement of numerical stability [LL00]. As of today, stabilised D3Q27 lattice MRT collision model has been proposed by [SKTC15], and successfully applied in a grid-refinement method for more efficient fluid simulation [KS16]. It should be stressed that most of the above results cannot be obtained with the BGK models.

2.8.1 The multiple-relaxation-time lattice Boltzmann equation

The MRT collision approach defines the collision term $\Omega_i\left[f_i\left(\boldsymbol{x},t\right)\right]$ as:

$$\Omega_i\left[f_i\left(\boldsymbol{x},t\right)\right]=-S\left[\left|f\left(\boldsymbol{x},t\right)\right\rangle-\left|f^{eq}\left(\boldsymbol{x},t\right)\right\rangle\right] \tag{2.104}$$

in this equation, we use the following notation for column vectors in q-dimensional space as:

$$\left|f\left(\boldsymbol{x},t\right)\right\rangle=\left(f_0\left(\boldsymbol{x},t\right),f_1\left(\boldsymbol{x},t\right),\ldots,f_{q-1}\left(\boldsymbol{x},t\right)\right)^{\mathrm{T}} \tag{2.105}$$

$$\left|f^{eq}\left(\boldsymbol{x},t\right)\right\rangle=\left(f_0^{eq}\left(\boldsymbol{x},t\right),f_1^{eq}\left(\boldsymbol{x},t\right),\ldots,f_{q-1}^{eq}\left(\boldsymbol{x},t\right)\right)^{\mathrm{T}} \tag{2.106}$$

where the superscript T denotes the transpose operator and we always assume that $\boldsymbol{e}_0=\boldsymbol{0}$. From here on the Dirac notations of bra $\langle\cdot|$ and ket $|\cdot\rangle$ vectors are used to denote, respectively, the row and column vectors. The q eigenvalues of S are all between 0 and 2 so as to maintain linear stability and the separation of scales, which means that the relaxation times of non-conserved quantities are much faster that the hydrodynamic time-scales. The BGK collision models are special cases in which the q relaxation times are all equal, and the collision matrix $S=\omega I$, where I is the identity matrix, $\omega=1/\tau$, and $\tau\,(>1/2)$ is the single relaxation time of the model.

To simulate athermal fluids, a necessary criterion is that the discrete velocity set must be sufficient to represent a scalar (fluid density ρ), a vector (momentum j), another scalar (pressure P), and a symmetric traceless second-rank tensor (viscous stress tensor σ_{ij}). More generally, the velocity set must possess sufficient symmetries for the hydrodynamic equations to hold; the conserved quantities and their fluxes must transform properly so that they can approximate their continuous counterparts in an appropriate limit. The local conserved quantities must be the fluid density ρ and the momentum $\boldsymbol{j}$ for athermal fluids.

Given a chosen set of discrete velocities $\boldsymbol{e}_i\,(i=0,1,\ldots,q-1)$ and corresponding distribution functions $f_i\,(i=0,1,\ldots,q-1)$, an equal number of moments $m_\beta\,(\beta=0,1,\ldots,q-1)$ of the distribution functions f_i can be obtained as:

$$m_\beta=\langle\phi_\beta|f\rangle=\langle f|\phi_\beta\rangle\,,\quad\langle f|=\left(f_0,f_1,\ldots,f_{q-1}\right) \tag{2.107}$$

where $|\phi_\beta\rangle\,(\beta=0,1,\ldots,q-1)$ is an orthogonal dual basis set constructed by the Gram-Schmidt orthogonalisation procedure [BDLL02] from polynomials of the column vectors $|e_{x_i}\rangle$ in space. Vector $|e_{x_i}\rangle$ is built from the components of the $\boldsymbol{e}_i$ (e.g., $|e_x\rangle\,,|e_y\rangle\,,|e_z\rangle$ in three dimensions).

The set $|\phi_\beta\rangle$ is analogous to the Hermite tensor polynomials in continuous velocity space. It should be stressed that the orthogonal functions defined on a finite set of discrete velocity $\boldsymbol{e}_i$ has some degeneracies which do not exist in the Hermite tensor polynomials in continuous space. Obviously, the moments are simply linear combinations of the f_i, therefore velocity space, spanned by $|f\rangle=$

$(f_0, f_1, \ldots, f_{q-1})^{\mathrm{T}}$, and the moment space, spanned by $|m\rangle = (m_0, m_1, \ldots, m_{q-1})^{\mathrm{T}}$, are related by a linear mapping M: $|m\rangle = \mathrm{M}\,|f\rangle$ and $|f\rangle = \mathrm{M}^{-1}\,|m\rangle$. The transformation matrix M would be an orthogonal transformation if the basis vector $|\phi_\beta\rangle$ are normalised.

If the matrix S is chosen such that the $|\phi\rangle_\beta$ are its eigenvectors, the linear relaxation of the kinetic modes in moment space naturally accomplishes the collision process. Then the collision term of the MRT lattice Boltzmann equation becomes [LL00, DGK$^+$02]:

$$\Omega_i\left[f_i\left(\boldsymbol{x}, t\right)\right] = -\mathrm{M}^{-1}\hat{S}\left[\left|m\left(\boldsymbol{x}, t\right)\right\rangle - \left|m^{eq}\left(\boldsymbol{x}, t\right)\right\rangle\right] \tag{2.108}$$

where the collision matrix $\hat{S} = \mathrm{M}\cdot S\cdot \mathrm{M}^{-1}$ is diagonal; $\hat{S} = \mathrm{diag}\left(s_0, s_1, \ldots, s_{q-1}\right)$, m_i^{eq} is the equilibrium value of the moment m_i. The q moments can be separated into two groups: the 'hydrodynamic' (conserved) moments and the 'kinetic' (non-conserved) moments. The first group consists of the moments locally conserved in the collision process, so that in general $m_\beta^{eq} = m_\beta$. The second group consists of the moments not conserved in the collision process so that $m_\beta^{eq} \neq m_\beta$. The equilibrium distribution functions m_β^{eq} are functions of the conserved moments and are invariant under the symmetry group of the underlying lattice. For models designed to simulate athermal fluids, the only hydrodynamic moments are fluid density ρ (a scalar value) and momentum $\boldsymbol{j}$ (a vector value); energy is not a conserved quantity for athermal fluids. Equilibrium values of kinetic moments are functions of ρ and $\|\boldsymbol{j}\|^2$ for scalars, and $\boldsymbol{j}$ times some functions of ρ and $\|\boldsymbol{j}\|^2$ (eventually a constant) for vectors, and so on.

2.8.2 The multiple-relaxation-time model for D3Q19 model

In D3Q19 square lattice model, the 19 discrete velocities are given by Equation 2.94, and the components of the 19 orthogonal basis vectors are given by [TFK06]:

$$\begin{cases} \left|\phi_0\right\rangle_i = \boldsymbol{e}_i^0 \\ \left|\phi_1\right\rangle_i = \boldsymbol{e}_i^2 - e^2 \\ \left|\phi_2\right\rangle_i = 3\left(\boldsymbol{e}_i^2\right)^2 - 6\boldsymbol{e}_i^2 e^2 + e^4 \end{cases} \tag{2.109}$$

$$\begin{cases} \left|\phi_3\right\rangle_i = e_{ix} \\ \left|\phi_5\right\rangle_i = e_{iy} \\ \left|\phi_7\right\rangle_i = e_{iz} \end{cases} \tag{2.110}$$

$$\begin{cases} \left|\phi_4\right\rangle_i = \left(3\boldsymbol{e}_i^2 - 5e^2\right) e_{ix} \\ \left|\phi_6\right\rangle_i = \left(3\boldsymbol{e}_i^2 - 5e^2\right) e_{iy} \\ \left|\phi_8\right\rangle_i = \left(3\boldsymbol{e}_i^2 - 5e^2\right) e_{iz} \end{cases} \tag{2.111}$$

$$\begin{cases} \left|\phi_9\right\rangle_i = 3e_{ix}^2 - \boldsymbol{e}_i^2 \\ \left|\phi_{11}\right\rangle_i = e_{iy}^2 - e_{iz}^2 \end{cases} \tag{2.112}$$

$$\begin{cases} \left|\phi_{13}\right\rangle_i = e_{ix}e_{iy} \\ \left|\phi_{14}\right\rangle_i = e_{iy}e_{iz} \\ \left|\phi_{15}\right\rangle_i = e_{iz}e_{ix} \end{cases} \tag{2.113}$$

$$\begin{cases} \left|\phi_{10}\right\rangle_i = \left(2\boldsymbol{e}_i^2 - 3e^2\right)\left(3e_{ix}^2 - \boldsymbol{e}_i^2\right) \\ \left|\phi_{12}\right\rangle_i = \left(2\boldsymbol{e}_i^2 - 3e^2\right)\left(e_{iy}^2 - e_{iz}^2\right) \end{cases} \tag{2.114}$$

$$\begin{cases} \left|\phi_{16}\right\rangle_i = \left(e_{iy}^2 - e_{iz}^2\right) e_{ix} \\ \left|\phi_{17}\right\rangle_i = \left(e_{iz}^2 - e_{ix}^2\right) e_{iy} \\ \left|\phi_{18}\right\rangle_i = \left(e_{ix}^2 - e_{iy}^2\right) e_{iz} \end{cases} \tag{2.115}$$

where $i \in (0, 1, \ldots, 18)$. The corresponding 19 moments $m_\beta\,(\beta = 0, 1, \ldots, 18)$ are arranged in the following order:

$$|m\rangle = \left(\rho, e, \epsilon, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{zx}, m_x, m_y, m_z\right)^{\mathrm{T}} \tag{2.116}$$

The diagonal collision matrix $\hat{S}$ is:

$$\hat{S} = \operatorname{diag}\left(0, s_1, s_2, 0, s_4, 0, s_4, 0, s_4, s_9, s_{10}, s_9, s_{10}, s_{13}, s_{13}, s_{13}, s_{16}, s_{16}, s_{16}\right) \tag{2.117}$$

and the transformation matrix M is given by:

$$\left[\begin{array}{ccccccccccccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -2 & -2 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & -2 & 2 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
0 & 2 & 2 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\
0 & -2 & -2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\
0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{array}\right] \tag{2.118}$$

the 4th, 6th and 8th row vectors of M (corresponding to j_x, j_y and j_z, respectively) uniquely define the ordering (or labelling) of the velocity $\boldsymbol{e}_i$ with respect to subscript i.

With $c_s^2 = 1/3$ and $s_9 = s_{13}$, the non-zero equilibria of the non-conserved moments are given as functions up to second-order in ρ and $\boldsymbol{u}$ with an incompressible flow as follows:

$$\begin{cases} m_0^{eq} = \rho, \ \ m_3^{eq} = \rho_0 u_x, \ \ m_5^{eq} = \rho_0 u_y, \ \ m_7^{eq} = \rho_0 u_z \\ m_1^{eq} = e^{eq} = \rho_0 \left(u_x^2 + u_y^2 + u_z^2\right) \\ m_9^{eq} = 3p_{xx}^{eq} = \rho_0 \left(2u_x^2 - u_y^2 - u_z^2\right) \\ m_{11}^{eq} = p_{ww}^{eq} = \rho_0 \left(u_y^2 - u_z^2\right) \\ m_{13}^{eq} = p_{xy}^{eq} = \rho_0 u_x u_y, \ \ m_{14}^{eq} = p_{yz}^{eq} = \rho_0 u_y u_z, \ \ m_{15}^{eq} = p_{zx}^{eq} = \rho_0 u_z u_x \end{cases} \tag{2.119}$$

where ρ_0 is a constant fluid density.

In D3Q19 square lattice model, the bulk viscosity ζ is given by:

$$\zeta = \frac{\left(5 - 9c_s^2\right)}{9}\left(\frac{1}{s_1} - \frac{1}{2}\right) = \frac{2}{9}\left(\frac{1}{s_1} - \frac{1}{2}\right) \tag{2.120}$$

and its kinematic viscosity ν is given by:

$$\nu = \frac{1}{3}\left(\frac{1}{s_9} - \frac{1}{2}\right) = \left(\frac{1}{s_{13}} - \frac{1}{2}\right) \tag{2.121}$$

The non-zero values of the collision matrix S are determined as follows:

$$\begin{cases} s_{l,1,1} = s_{l,a} \\ s_{l,2,2} = s_{l,b} \\ s_{l,4,4} = s_{l,6,6} = s_{l,8,8} = s_{l,c} \\ s_{l,10,10} = s_{l,12,12} = s_{l,d} \\ s_{l,9,9} = s_{l,11,11} = s_{l,13,13} = s_{l,14,14} = s_{l,15,15} = -\dfrac{1}{\tau} = s_{l,\omega} \\ s_{l,16,16} = s_{l,17,17} = s_{l,18,18} = s_{l,e} \end{cases} \tag{2.122}$$

the parameters $s_{l,a}$, $s_{l,b}$, $s_{l,c}$, $s_{l,d}$ and $s_{l,e}$ are freely chosen in the range $[-2, 0]$. These values depend of the flow conditions, such as the initial and boundary conditions, and cannot be determined generically. We chose $s_{l,a} = s_{l,b} = s_{l,c} = s_{l,d} = s_{l,e} = -1.0$ as the moments as the relaxation rate approaches its equilibrium state, which is a reasonable way to determine these parameters.

2.9 A Smagorinsky subgrid-scale model

In order to simulate turbulent flows by the lattice Boltzmann method, the basic calculation procedure has to be extended, as its stability is limited once the relaxation rate τ approaches 1/2. Here, the Smagorinsky sub-grid-scale model [Sma63] is applied in this thesis. Its primary use is to stabilise the fluid simulation. Compared to the small slowdown due to the increased complexity of the collision operator, this usually results in large improvement of efficiency, as the simulations would otherwise require considerably finer grid resolutions.

The Smagorinsky sub-grid-scale turbulence model applies the calculation of the local stress tensor as described in [Sma63] to the lattice Boltzmann method. The computation of this tensor is relatively easy for the lattice Boltzmann method, as each lattice already contains information about the derivatives of the hydrodynamic variables in each distribution function f_i. The magnitude of the stress tensor is then used in each lattice to modify the relaxation time according to the eddy viscosity. For the calculation of the modified relaxation time, the Smagorinsky constant C_S is used. For the simulations in the following, C_S will be set to 0.162. Values in this range are commonly used for the lattice Boltzmann simulations, and were shown to yield good modelling of the subgrid vortices [GFN02]. The turculence model is integrated into the basic algorithm by adding the calculation of the modified relaxation time.

The modified relaxation time τ_s is calculated by performing the steps that are described in the following. First the non-equilibrium stress tensor $\Pi_{\alpha,\beta}$ is calculated for each lattice as:

$$\Pi_{\alpha,\beta} = \sum_i \boldsymbol{e}_{i_\alpha} \boldsymbol{e}_{i_\beta} \left(f_i - f_i^{eq}\right) \tag{2.123}$$

using the notation from [HSCD96, KTL03]. Thus, α and β each run over the three spatial dimensions, while i is the index of the respective velocity vector and distribution function for the D3Q19 model. The intensity of the local stress tensor S is then computed as:

$$S = \frac{1}{6C_S\Delta^2}\left(\sqrt{\nu^2 + 18C_S\Delta^2\sqrt{\Pi_{\alpha,\beta}\Pi_{\alpha,\beta}}} - \nu\right) \tag{2.124}$$

now the modified relaxation time is given by:

$$\tau_s = 3\left(\nu + C_S\Delta^2 S\right) + \frac{1}{2} \tag{2.125}$$

where $\Delta = 1$ is a spatial filter.

From Equation 2.124 it can be seen that S will always have a positive value - thus the local viscosity will be increased depending on the size of the stress tensor calculated from the non-equilibrium parts of the distribution functions of the cell to be relaxed.

2.10 Boundary condition for the lattice Boltzmann method

2.10.1 Overview of boundary conditions

Boundary conditions play a very important role for numerical methods, such as a finite element method, to solve any differential system. In a similar way, imposing boundary conditions accurately is crucial for the lattice Boltzmann method. In general, there are three distinct kinds of boundary conditions for macroscopic differential equations, namely Dirichlet boundary conditions which specify the value of the function on the boundary, Neumann boundary conditions that provide the normal derivative of the function on the boundary, and the combination of prescribed function and it's normal derivative with the name of Robin boundary conditions. When the boundary conditions are approximated on the boundary the same way as that within the domain by a certain conventional numerical method, the problem ends up with solving a linear nor non-linear algebraic system.

As for the conventional method, the macroscopic function imposed on the boundary is usually the same as the function directly treated by the method. Take velocity in Navier-Stokes equations for example, the prescribed velocity or pressure on the boundary is the unknown macroscopic variables to be solved by 'macroscopic methods'. For some aspects, the lattice Boltzmann method is quite different from the conventional methods. More in detail, the leading role playing on the stage is not macroscopic variables but rather microscopic function. Since there is no explicit distribution function streaming from the boundary into the bulk, we have to construct the unknown distribution functions based on the macroscopic imposed boundary conditions, which is also one of the significant difficulties for the lattice Boltzmann method. In this thesis, we will concentrate on how to impose solid wall boundary conditions for the lattice Boltzmann method in term of a tsunami modellings.

2.10.2 Solid wall boundary conditions

For no-slip solid wall boundaries, we impose a second-order bounce-back scheme [BFL01]. It slightly violates the conservation of mass, but because it is second order in space, an overall scheme of second order can be achieved. The modified bounce-back scheme is given by:

$$f_{IA}^{t+1} = \begin{cases} (1-2q)\, f_{iF}^{t} + 2q f_{iA}^{t} & (0.0 \leq q < 0.5) \\ \dfrac{(2q-1)}{2q} f_{IA}^{t} + \dfrac{1}{2q} f_{iA}^{t} & (0.5 \leq q \leq 1.0) \end{cases} \tag{2.126}$$

where f_I and f_i are anti-parallel incoming (missing) and outgoing distribution functions, respectively, and where q denotes the subgrid wall distance for direction i. f_{iF} and f_{iA} are pre- and post-collision distribution functions.

At slip boundaries, the macroscopic velocity $\boldsymbol{u}$ at the boundary has to determined, then projected onto the surface normal vector $\hat{\boldsymbol{n}}$ as:

$$\boldsymbol{u}_{||} = \boldsymbol{u} - (\boldsymbol{u} \cdot \hat{\boldsymbol{n}})\, \hat{\boldsymbol{n}} \tag{2.127}$$

and finally applied as boundary velocity $\boldsymbol{u}_{||}$ using the following modified version of the second-order bounce-back scheme:

$$f_{IA}^{t+1} = \begin{cases} (1-2q)\, f_{iF}^{t} + 2q f_{iA}^{t} - 2\rho w_i \dfrac{\boldsymbol{e}_i \cdot \boldsymbol{u}_{||}}{c_s^2} & (0.0 \leq q < 0.5) \\ \dfrac{(2q-1)}{2q} f_{IA}^{t} + \dfrac{1}{2q} f_{iA}^{t} - \rho w_i \dfrac{\boldsymbol{e}_i \cdot \boldsymbol{u}_{||}}{q c_s^2} & (0.5 \leq q \leq 1.0) \end{cases} \tag{2.128}$$

2.11 Parametrisation

Below, the conversion of dimensional quantities, denoted by primed symbols, into dimensionless quantities used in the lattice Boltzmann method will be described. Given the real-world values for kinematic viscosity $\nu' \left(\mathrm{m}^2/\mathrm{s}\right)$, domain size $S'\,(\mathrm{m})$, a desired grid resolution r, and a gravitational force $\boldsymbol{g}' \left(\mathrm{m}/\mathrm{s}^2\right)$, the corresponding lattice quantities are computed as described in the following. Let S' be the length of one side of the domain that should be resolved with r cells. The cell size used by the lattice Boltzmann method can then be computed as $\Delta x' = S'/r$.

The dimensional time step interval $\Delta t'$ should be determined by limiting the weak compressibility of the lattice Boltzmann method. In the lattice Boltzmann method, the Mach number which determine the order of the weak compressibility is defined by the following equation:

$$\mathrm{Ma} \sim \frac{\|u_{\max}\|}{c_s} \tag{2.129}$$

where $\|u_{\max}\|$ is a maximum fluid velocity in a simulation domain. This means that the weak compressibility of the lattice Boltzmann method can controlled by the sound velocity (Equation 2.103) if the maximum fluid velocity $\|u_{\max}\|$ can be estimated.

Given $\Delta x'$ and $\Delta t'$, the dimensionless lattice viscosity ν, and relaxation rate τ can be determined as:

$$\nu = \nu' \frac{\Delta t'}{\Delta x'^2} \tag{2.130}$$

and

$$\tau = 3\nu + \frac{1}{2} \tag{2.131}$$

Likewise, the lattice acceleration $\boldsymbol{g}$ (e.g., gravity acceleration) is determined as:

$$\boldsymbol{g} = \boldsymbol{g}' \frac{\Delta t'^2}{\Delta x'} \tag{2.132}$$

The gravity acceleration effect is calculated by adding a term directly to the collision step in every time step in this thesis as:

$$\Delta f_i = 3\omega_i \rho \boldsymbol{e}_i \cdot \boldsymbol{g} \tag{2.133}$$

In conclusion, a valid parametrisation for the lattice Boltzmann fluid simulation is given by the physical scale, the desired kinematic viscosity and the compressibility factor. For a given grid resolution, the values of τ and Δt can then be calculated. Note that when the grid resolution is small in combination with a low viscosity, the resulting value of τ will be close to 1/2. Due to the turbulence mode, the simulation will remain stable, but effectively increase the viscosity to a value that can be handled with the chosen grid resolution.

2.12 The calculation procedure

Once the lattice structure is specified, the lattice Boltzmann method is determined. And the general algorithm can be given as follows: (flow chart is shown in Figure 2.3).

1. Set initial macro variables: density ρ and velocity $\boldsymbol{u}$. The initial distribution function $f_i(\boldsymbol{x}, 0)$ is given by the equilibrium distribution function $f_i^{eq}(\boldsymbol{x}, 0)$.

2. Calculate collision step using the BGK or MRT collision model.

3. Calculate streaming step.

4. Impose boundary conditions.

5. Calculate macroscopic variables: density ρ and velocity $\boldsymbol{u}$.

6. Stop and go to the step of post processing if certain stopping condition is satisfied. Otherwise go back to step 2.

2.13 Primitive model verification

Two-dimensional lid-driven cavity flows are investigated for a primitive model verification of the lattice Boltzmann method. Figure 2.4 shows the computing domain and boundary conditions, where x and y are the components of the domain, and $U_{\max}$ is the dimensionless velocity of the top boundary, which is also one of the characteristic value. $U_{\max} = 0.1$ was set and fixed by varying the Reynolds number Re. The characteristic velocity $U_{\max}$ should be properly set in the lattice Boltzmann simulation to reduce its the weak-compressibility (i.e., $\|U_{\max}\|/c_s < 0.3$). L is the length of the domain, which is also the characteristic length.

The boundary conditions on the top lid are obtained in velocity space by assigning distributions to [Lad94]:

$$f_i = w_i \rho_0 \frac{\boldsymbol{e}_i \cdot U_{\max}}{c_s^2} \tag{2.134}$$

it should be stressed that this particular implementation of a sliding boundary imposes a constant pressure $p_0 = c_s^2 \rho_0$ at the boundary, which is incorrect; and the momentum transfer in the direction perpendicular to the moving lid is significantly weakened. The correct boundary conditions consistent with the bounce-back conditions should be [Luo98, Luo00, BFL01]:

$$f_I = f_i + 2 w_i \rho_0 \frac{\boldsymbol{e}_I \cdot U_{\max}}{c_s^2} = f_i - 2 w_i \rho_0 \frac{\boldsymbol{e}_i \cdot U_{\max}}{c_s^2} \tag{2.135}$$

The bounce-back boundary condition given in Equation 2.126 are applied to the rest three walls as no-slip boundary conditions.

The criterion for steady state of this simulation is defined as the following condition:

$$\sqrt{\frac{\sum^{N} \left[\left(u_x^{n+1} - u_x^n \right)^2 + \left(u_y^{n+1} - u_y^n \right)^2 \right]}{\sum^{N} \left[u_x^n + u_y^n \right]}} < 10^{-10} \tag{2.136}$$

where N is the total number of computing cells in the domain. $\left(u_x^n, u_x^{n+1}\right)$ and $\left(u_y^n, u_y^{n+1}\right)$ are the velocity of each component for the previous and current time steps, respectively.

2.13.1 Performance of parallel computing

The lattice Boltzmann method has a high-potential to utilise high-performance parallel computing due to the data locality [RDVC12, OKTR13, CGK$^+$16, DSGS17]. In this thesis, the Message Passing Interface (MPI), which is one of the distributed memory parallel programming API is used. To divide a spacing computing domain by MPI, the data of the distribution functions f_i are required for the streaming step at the boundary layer as shown in Figure 2.5. The other macroscopic variables (i.e., density ρ and velocity $\boldsymbol{u}$) do not have to communicated. These are all local in computing cells.

Figure 2.6 shows the strong scaling of the benchmark. The measurement conditions are as followings:

1. Domain size: 256×256 cells.

2. Iteration: 1000 time step.

3. Collision model: The MRT-LBM with the Smagorinsky sub-grid-scale model.

The computing speed increases monotonically without saturating with the 64 MPI process computing. In addition, we achieved an effective parallelisation ratio of 99.7%. It appears that the processes that have to apply the lid-driven or bounce-back boundary condition decreased the performance of the other processes. We should derive a way to apply the boundary condition to bring the effective parallelisation ratio much closer to 100%.

2.13.2 Comparison between the BGK-LBM and MRT-LBM

This investigation was conducted to clarify the model accuracy between the BGK-LBM and MRT-LBM. According to the study of [BS06], the critical Reynolds number of the benchmark problem is from 8000 to 8050 within less than 1% of error. The simulation, thus, has been performed until Re = 7500 to focus on the steady solutions. The number of computing cells is 256×256. This parameter is common to both collision models.

Figure 2.7 compare the calculated dimensionless velocity components of the u_x and u_y profiles in the steady state along vertical and horizontal lines at $y/L = 1/2$ and $x/L = 1/2$.

In the cases of Re = 100 and Re = 400, both the BGK-LBM and MRT-LBM are in good agreement with the results of [GGS82]. In the extremely low Reynolds number flow, the BGK-LBM can be applicable and satisfy the accuracy requirement with a 256×256 grid calculation. The BGK-LBM model velocity profile, however, disagrees with the references data. This tendency is particularly strong near the no-slip boundary. We confirmed that the bounce-back scheme with the BGK-LBM is difficult for conducting high accuracy simulations even when the Reynolds number is smaller than the critical number with 256×256 cells.

In contrast, the results of the MRT-LBM are in good agreement with the references data regardless of the increase of the Reynolds number. The results near the no-slip boundary do not decrease in accuracy; It is, therefore, shown that the bounce-back scheme with the MRT-LBM model can satisfy the accurate no-slip condition, more accurately. It is assumed that the parameters that affect the accuracy boundary condition are set correctly in the MRT-LBM relaxation rates. Accordingly, the relaxation rates must be determined by each flow field.

Through the model comparison, it can be concluded that the MRT-LBM is capable of higher accuracy calculations compared to the BGK-LBM in lid-driven cavity flows, particularly near the no-slip boundary.

2.13.3 Detailed verification of the MRT-LBM

We verified the accuracy of the MRT-LBM depending on the spacing grid sizes. The Reynolds numbers are 1000, 3200, 5000 and 7500 to examine the steady state. The numbers of grids are set as 128 × 128, 256 × 256, 512 × 512, 1024 × 1024 and 2024 × 2024. The criterion for steady state is defined by Equation 2.136.

Figure 2.8 show the u_x and u_y velocity profiles in the steady state. The results of the MRT-LBM are is in good agreement with the reference data in all calculation cases. In addition, we calculated the relative error defined as:

$$\text{Error} = \left| \frac{u_{\text{LBM}} - u_{\text{Ghia}}}{u_{\text{Ghia}}} \right| \times 100 \tag{2.137}$$

to evaluate the model accuracy, quantitatively; where Error is the relative error, u_{LBM} is the velocity calculated by the MRT-LBM, and u_{Ghia} is the reference data. We did, however, not take the points where the reference velocity is near 0 into account to avoid a dividing-by-zero.

Figures 2.9 and 2.10 show the relative error for each Reynolds number. The results of the MRT-LBM become closer to the reference data as the number of calculation grids is larger, particularly near the solid wall boundary. It is required to set at least 512×512 grid points in less than 5% error calculation to obtain the accuracy results especially near the solid wall boundary in this benchmark by the MRT-LBM.

In the case of Re = 7500, the accuracy of the results decreases as the number of computing points increases, which is in contrast to the other calculation cases. The absolute value of velocity calculated by the MRT-LBM in the case of Re = 7500 decreases as the number of computing grids increases, similar to the other Reynolds number cases. In addition, the other Reynolds number cases increase in accuracy as the number of the computing grids increases; therefore, we hypothesise that this problem is caused by the coarse grid computing (129 × 129 computing grid) in the data by [GGS82]. Except for the Re = 7500 case, the MRT-LBM model increases in accuracy as the number of computing grids increases. It can be concluded that the MRT-LBM has a convergence of calculation capability that depends on the grid size similar to the other computational fluid dynamics.

2.13.4 Periodic solutions of high-Reynolds number flow

In the case of Re = 10000, this Reynolds number is larger than the critical Reynolds number. A numerical simulation was performed on the MRT-LBM with the Smagorinsky sub-grid-scale model. The number of computing grids is from 128×128 to 1024×1024. We validated the MRT-LBM using the time series velocity calculated at the monitoring point, which is set referring to [BS06] because of difficulty to examine the steady state.

Figure 2.11 shows the time evolution of the dimensionless velocity along the x-axis at monitoring point $(14/16, 13/16)$, and Fourier transform analysis is performed for the time series velocity signal. Time, which is the horizontal axis, is a non-dimensional value calculated by the characteristic velocity U and length L. The time series velocity along the horizontal axis is periodic regardless of the number of computing grids. Furthermore, the horizontal velocity is smaller as the number of computing points increases. The diminution converges in the case of 1024×1024 computing. The power spectrum shows one large peak around a frequency of 0.4. The results converge in the 1024×1024 computing case. Bruneau *et al.* [BS06] reported that a plurality of peaks appears in Re = 10000 lid-driven cavity flow simulation; we showed that there is one strong peak through the MRT-LBM Smagorinsky sub-grid-scale model simulation. In addition, because the results of the 512×512 case are in good agreement with the 1024×1024 case, the 512×512 calculation can ensure accuracy in Re = 10000 lid-driven cavity flow with the MRT-LBM simulation.

Figure 2.12 also shows the time series velocity profile at the monitoring point. The horizontal axis is dimensionless velocity along the y-axis, and the vertical axis is dimensionless velocity along the x-axis. This figure cannot describe a circle if both axis velocities are not periodic. In the MRT-LBM results, the plotted lines perfectly overlap from 10000 to 10040 times. According to the above discussion, we conclude the the MRT-LBM Smagorinsky sub-grid-scale model flow field is perfectly periodic in Re = 10000 lid-driven cavity flow. Sufficient calculation accuracy can be achieved in more than 512×512 simulations.

2.13.5 Three-dimensional lid-driven cavity flow

We also carried out a primitive benchmark test for the three-dimensional lattice Boltzmann method with the multiple-relaxation-time model. Figure 2.13 shows initial setting and calculation parameters. The critical Reynolds number of the three-dimensional problem is around 1000. We, thus, simulated the benchmark until Re = 1000 to investigate the velocity profile in steady state.

Figure 2.14 shows the calculated dimensionless velocity components of the u_x and u_z profiles in th steady state along the vertical and horizontal line. All the simulated results are in good agreement with the reference data [KHT87]. Through the primitive model verification, it can be concluded that the two and three-dimensional MRT-LBM are capable of high accuracy fluid simulation in low Mach number. We will expand the three-dimensional MRT-LBM with a free surface flows to enable it a tsunami modelling in the next chapter.

2.14 Conclusion

In this chapter, the Euler equations and Navier-Stokes equations were derived from the kinetic Boltzmann - BGK equation by the Chapman-Enskog expansion. The Chapman-Enskog expansion and ansatz can also be used to analyse the accuracy of the approximation up to a certain order for the derivation from the lattice Boltzmann - BGK equation to the Navier-Stokes equations. What's more, by the Chapman-Enskog anasatz we can also obtain independent bulk and shear viscosity for the Navier-Stokes equations. The advanced multiple-relaxation-time (MRT) collision approach was, moreover, shown to overcome the numerical defects of the conventional Boltzmann - BGK approach.

Finally, we performed a detailed verification of the BGK-LBM and MRT-LBM to investigate the accuracy in the two and three-dimensional lid-driven cavity flows which is one of primitive benchmark problems for computational fluid dynamics. From the benchmark tests, the following conclusions can be drawn:

- Compared to the BGK-LBM, results of the MRT-LBM are in good agreement with the reference data, particularly near the solid wall boundary. Hence, it appears that no special treatments are required for the simple bounce-back scheme with the MRT-LBM in the benchmark problem.

- A convergence of calculation capability of the MRT-LBM that depends on the spacing grids were demonstrated as a computational fluid dynamics through the detailed numerical simulations.

It is observed that, the lattice Boltzmann method should require more computing cells compared with the conventional incompressible Navier-Stokes solvers such as the finite difference method. It is assumed that the grid-refinement approach [TFK06, YF09, FL14, KS16] enables the lattice Boltzmann method to perform a more efficient fluid simulation.

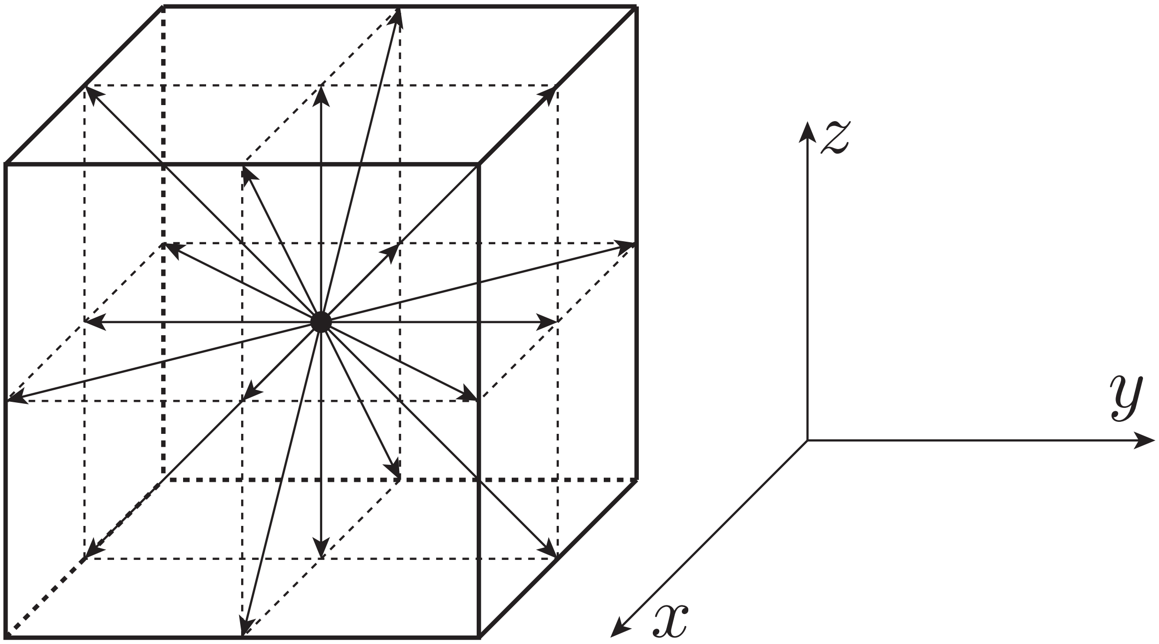


Figure 2.1: Three-dimensional 19-speed square lattice (D3Q19).

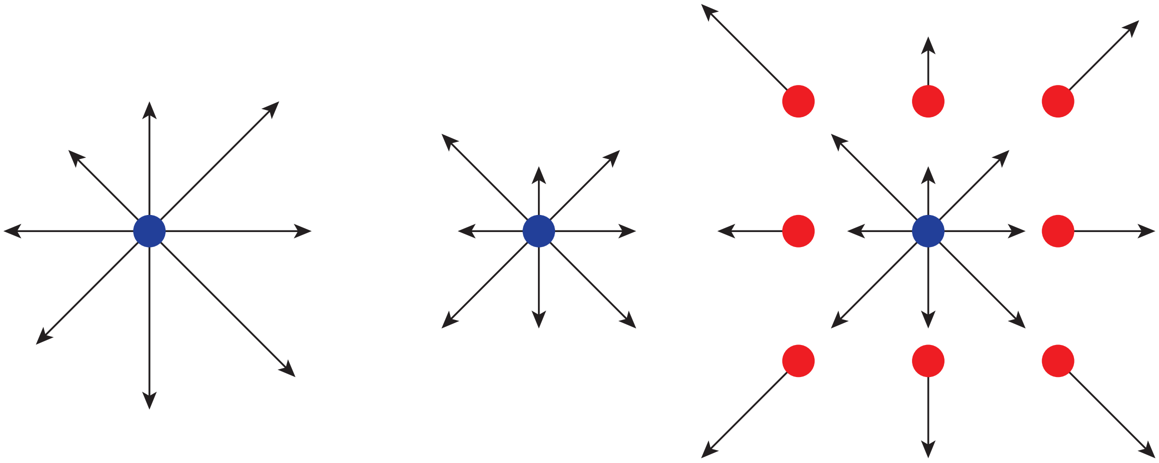


Figure 2.2: Collision and propagation.

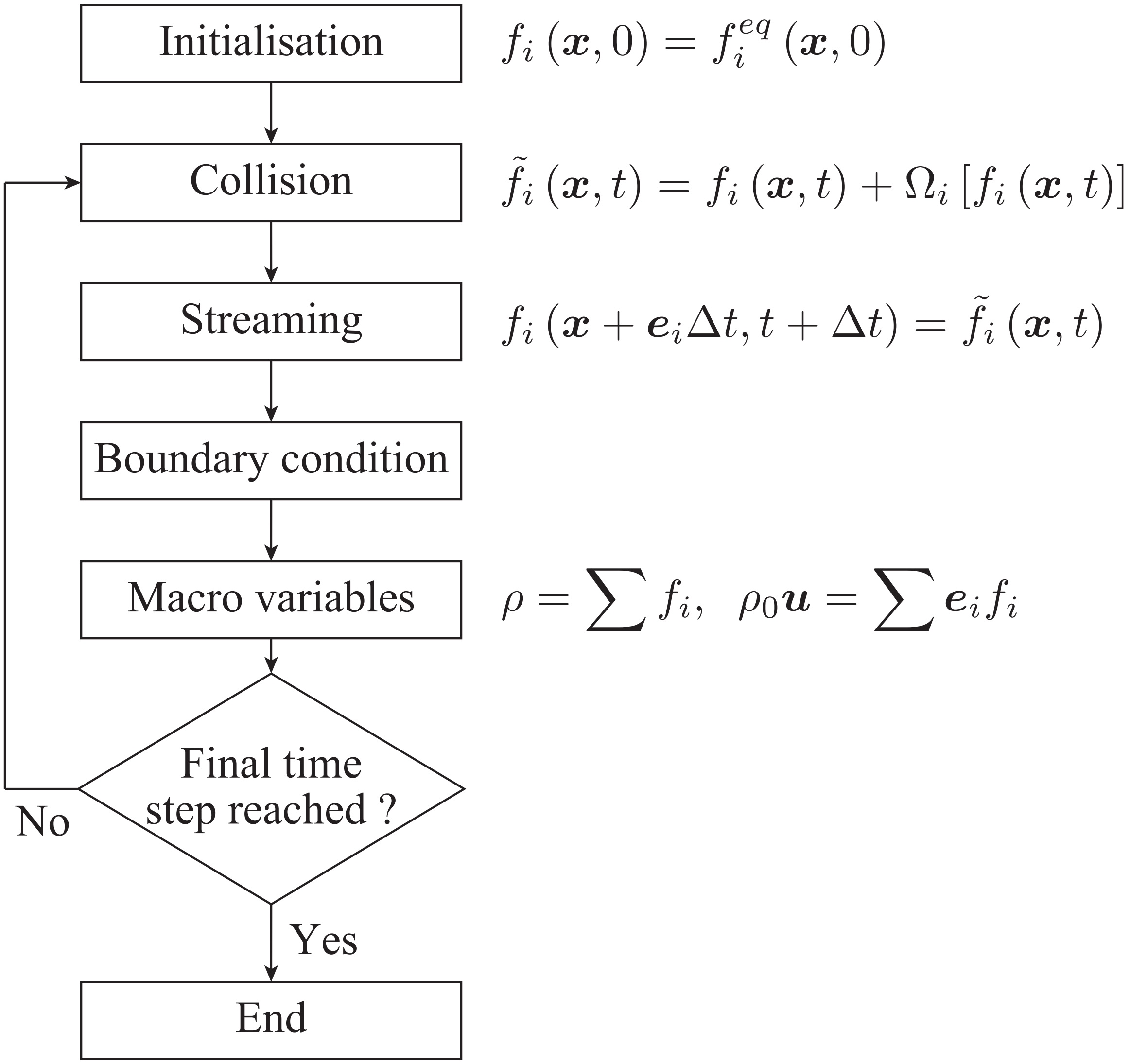

Initialisation
$f_i(\boldsymbol{x},0)=f_i^{eq}(\boldsymbol{x},0)$
Collision
$\tilde{f}_i(\boldsymbol{x},t)=f_i(\boldsymbol{x},t)+\Omega_i\left[f_i(\boldsymbol{x},t)\right]$
Streaming
$f_i(\boldsymbol{x}+\boldsymbol{e}_i\Delta t,t+\Delta t)=\tilde{f}_i(\boldsymbol{x},t)$
Boundary condition
Macro variables
$\rho=\sum f_i,\quad \rho_0\boldsymbol{u}=\sum \boldsymbol{e}_i f_i$
Final time step reached ?
No
Yes
End

Figure 2.3: The general calculation procedure for the lattice Boltzmann method.

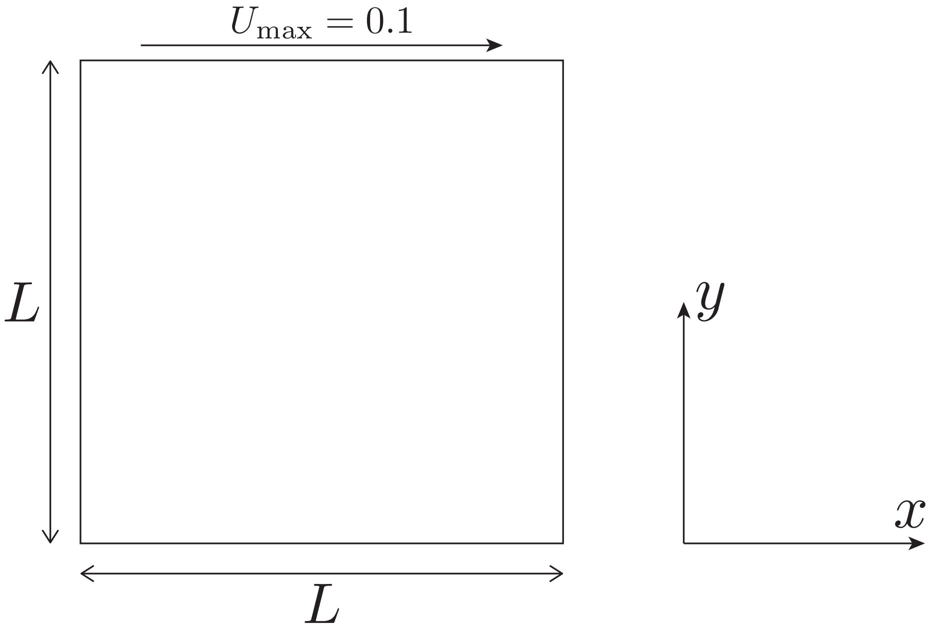


Figure 2.4: The computational domain of the two-dimensional lid-driven cavity flow.

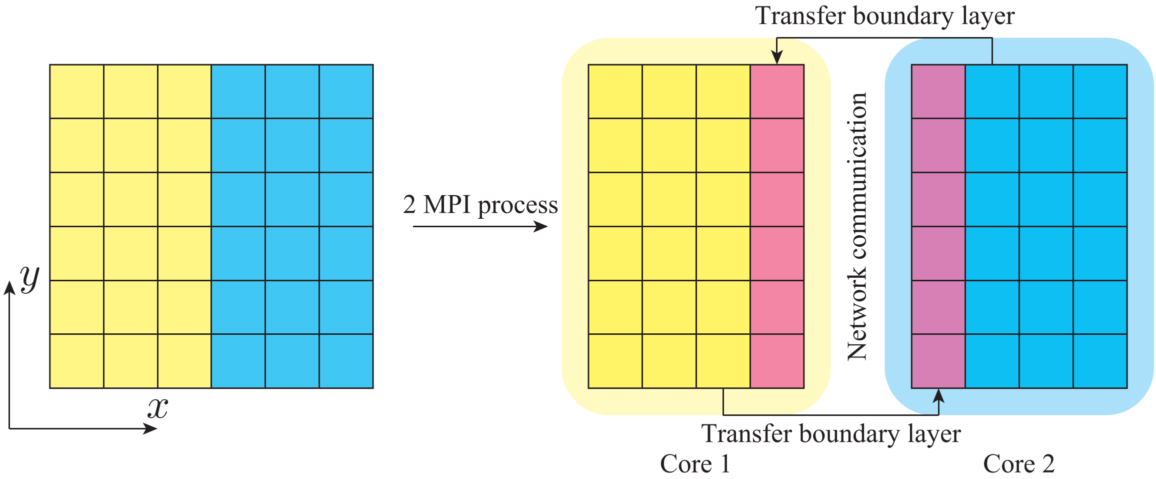


Figure 2.5: MPI communication before streaming step for the benchmark [GGS82].

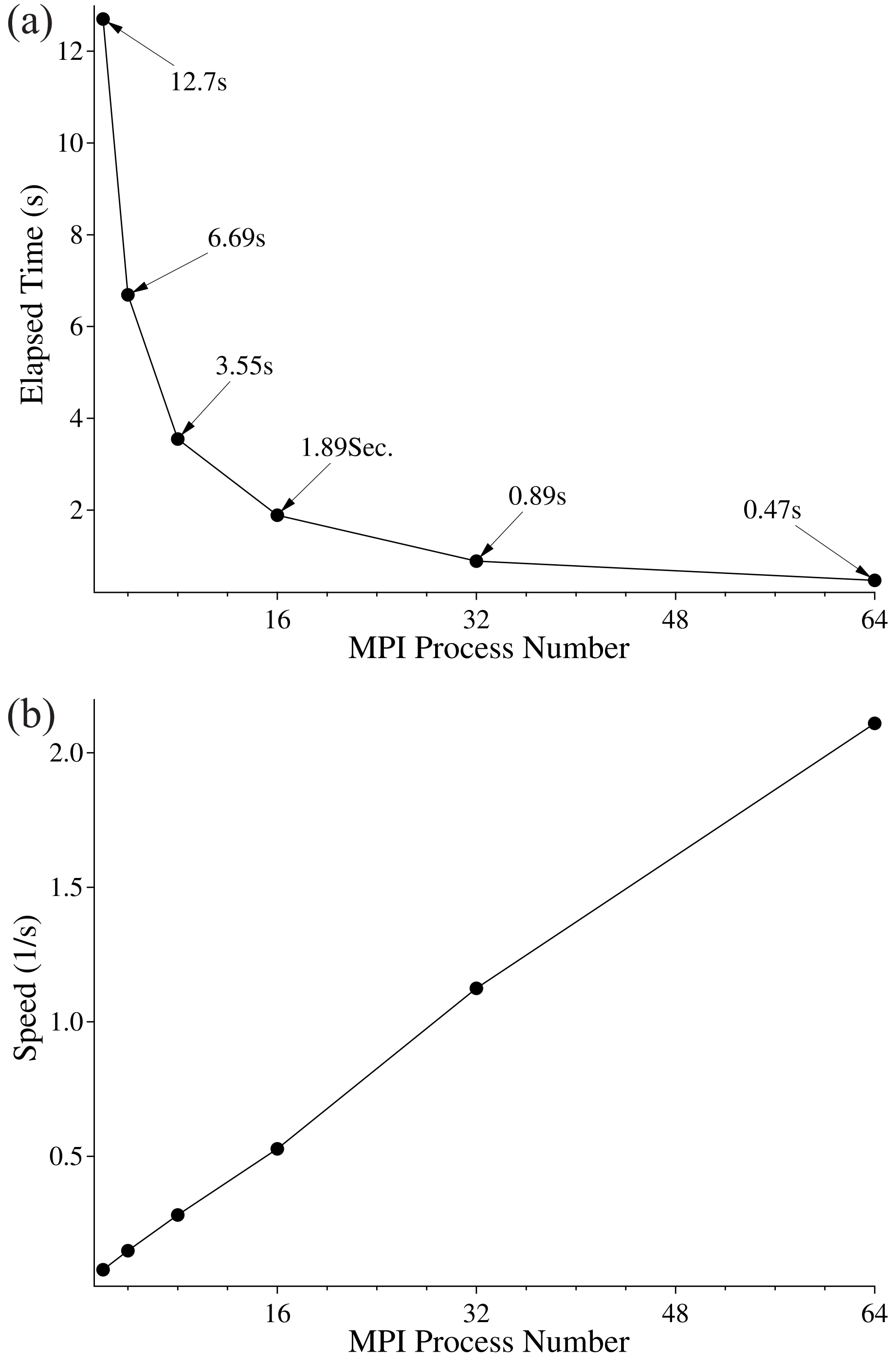


Figure 2.6: Performance of MPI parallel computing: (a) elapsed time, (b) speed.

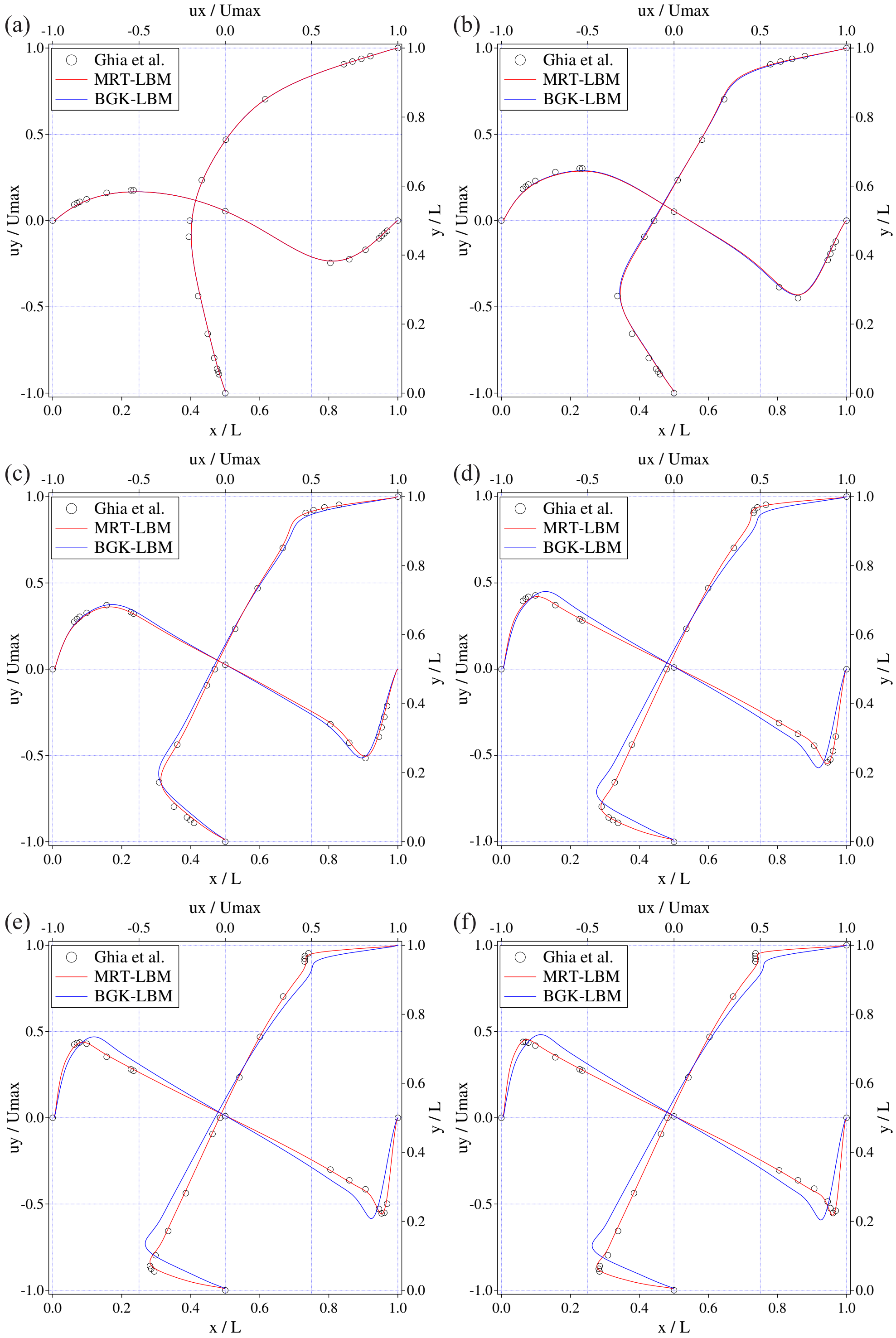


Figure 2.7: Dimensionless velocity profiles: (a) Re = 100, (b) Re = 400, (c) Re = 1000, (d) Re = 3200, (e) Re = 5000, (f) Re = 7500.

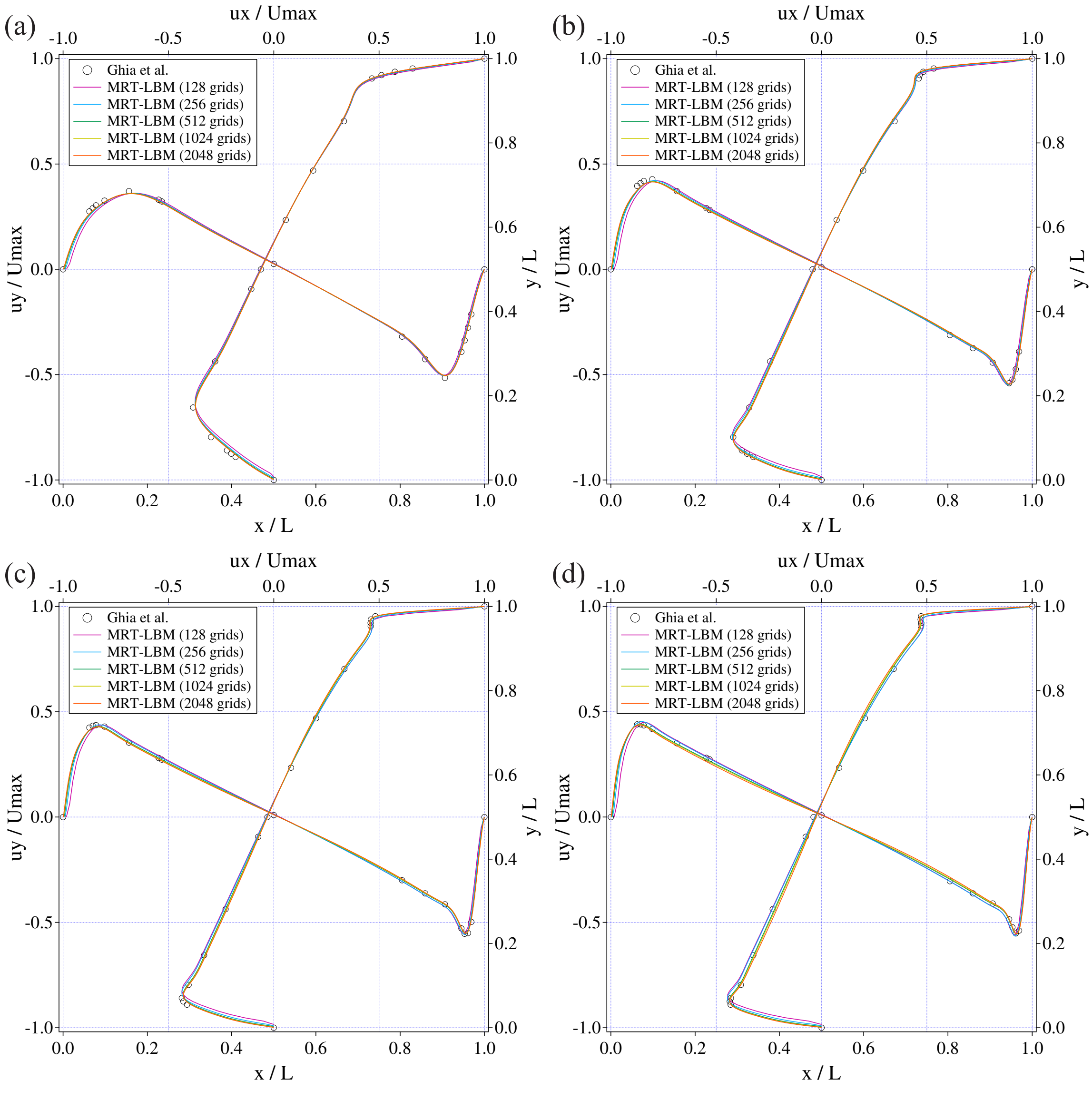


Figure 2.8: Dimensionless velocity profiles of the MRT-LBM: (a) Re = 1000, (b) Re = 3200, (c) Re = 5000, (d) Re = 7500.

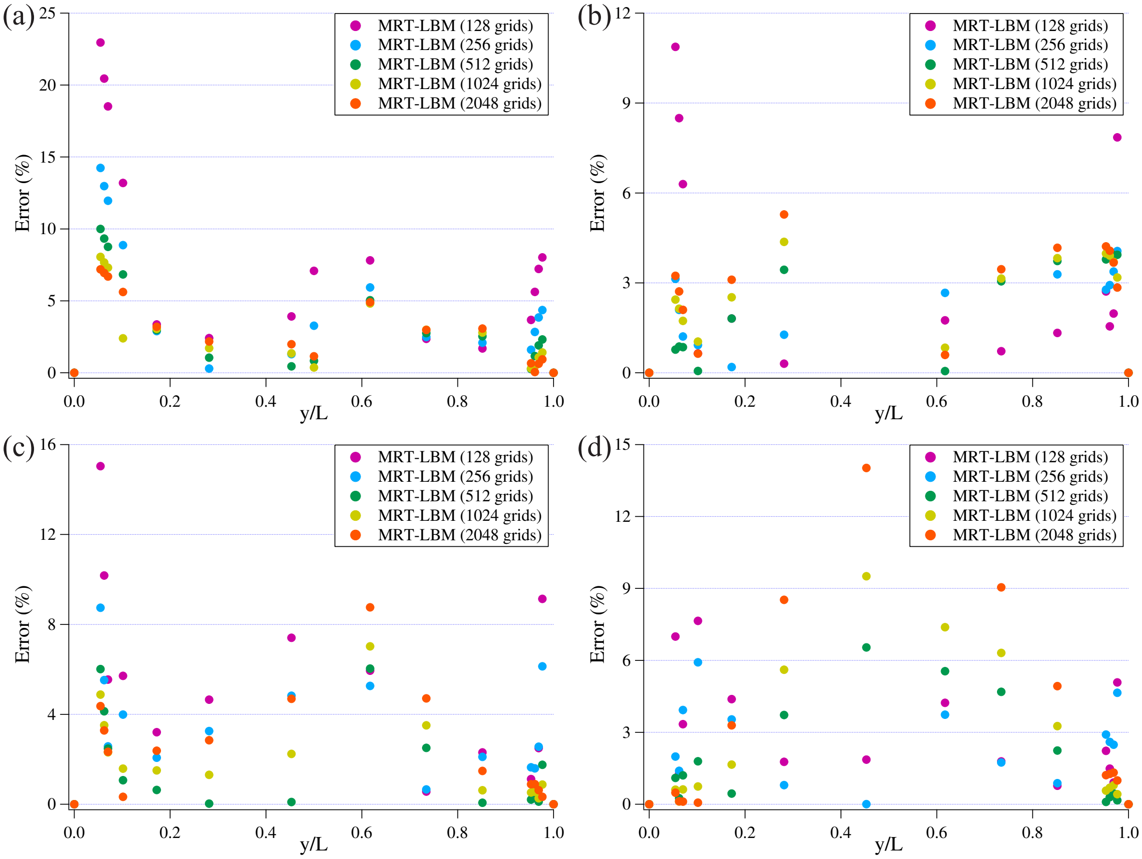


Figure 2.9: Relative error values for u_x, (a) Re = 1000, (b) Re = 3200, (c) Re = 5000, (d) Re = 7500.

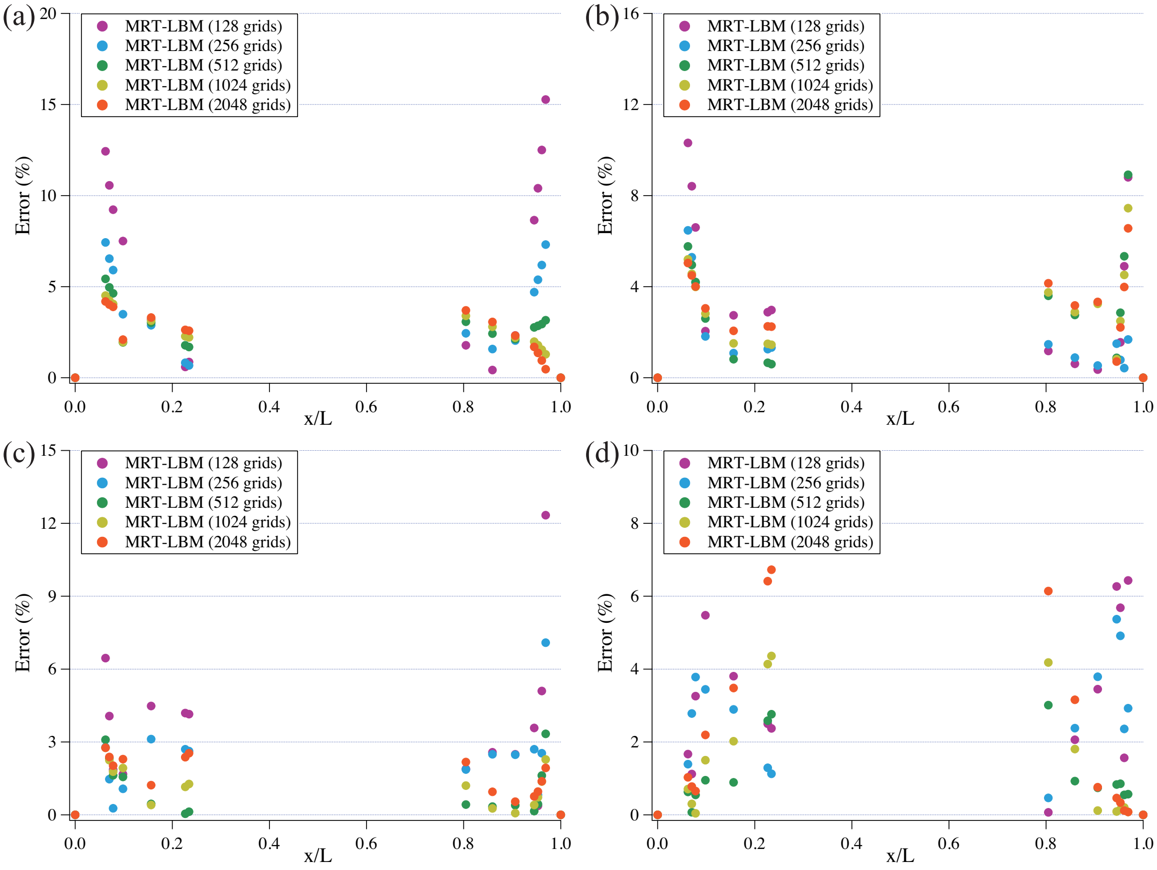


Figure 2.10: Relative error values for u_y, (a) Re = 1000, (b) Re = 3200, (c) Re = 5000, (d) Re = 7500.

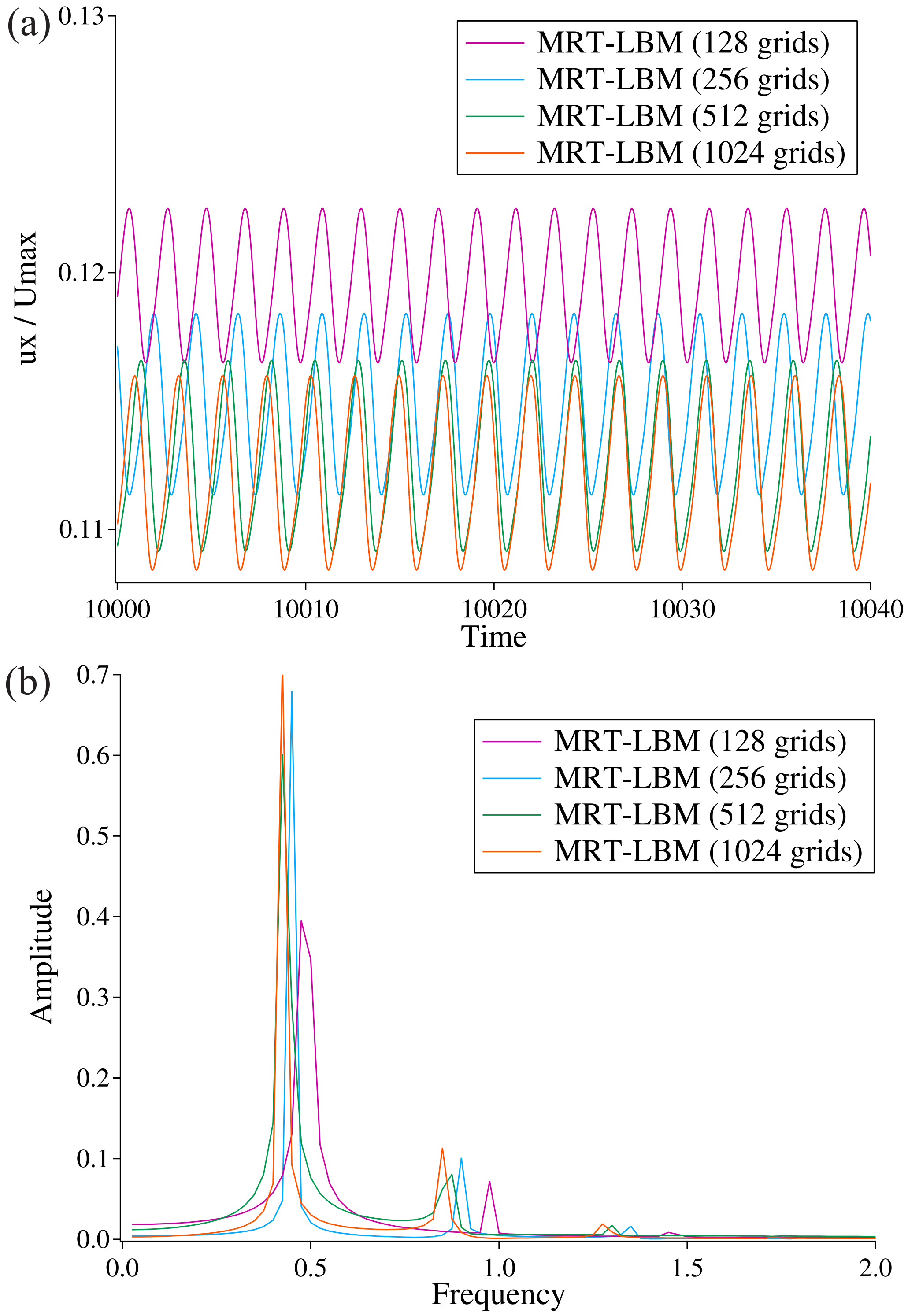


Figure 2.11: (a) Horizontal velocity history, (b) the power spectrum.

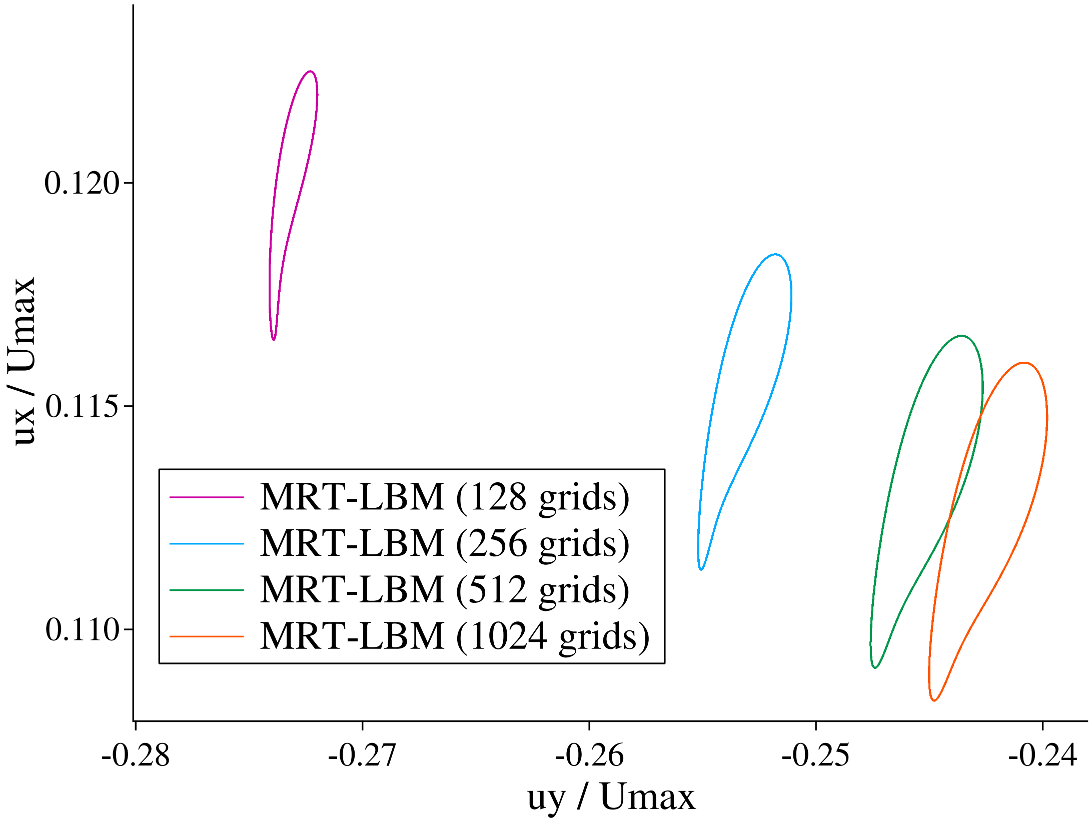
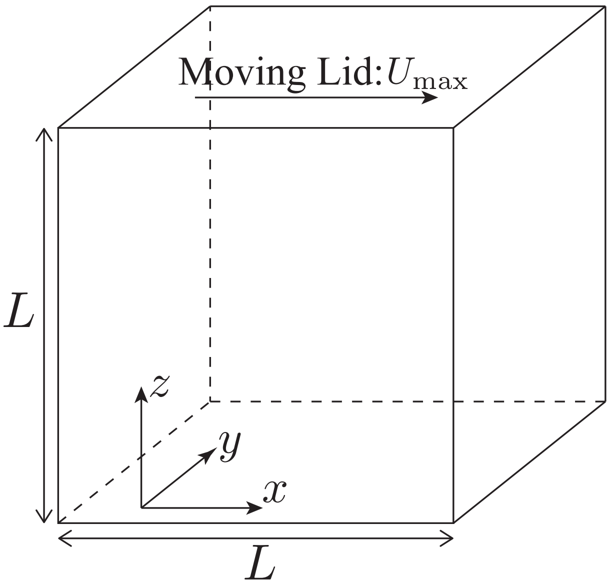


Figure 2.12: Phase portrait at the monitoring point.



Calculation parameters

Parameter	Value
Resolution	128×128 × 128
L	1.0
$U_{\max}$	0.1
Re	100, 400, 1000

Relaxation rate is given as:

$$\tau = 3\frac{U_{\max}L}{\mathrm{Re}} + \frac{1}{2}$$

Figure 2.13: The computational domain and calculation parameters for the three-dimensional lid-driven cavity flow.

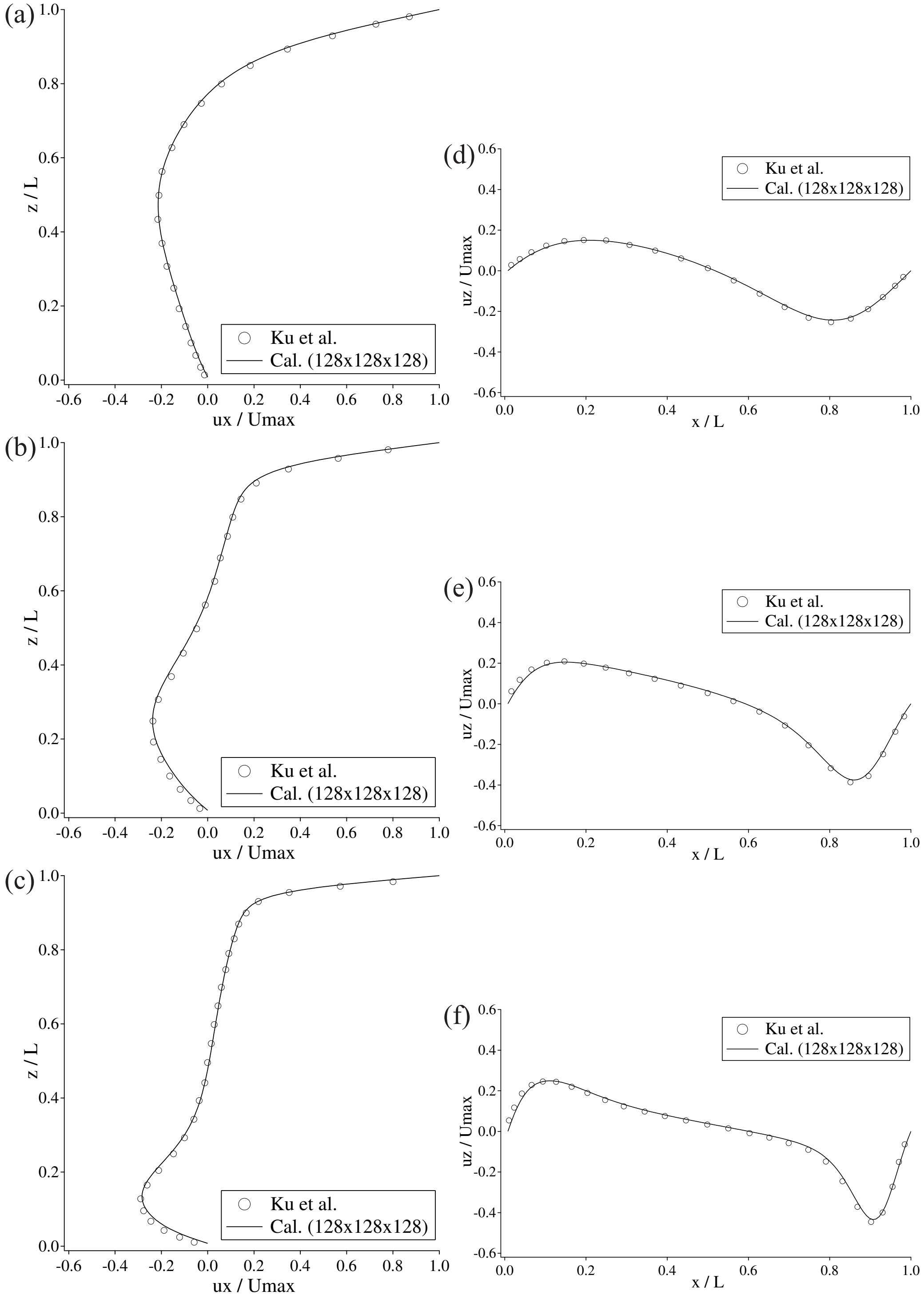


Figure 2.14: Dimensionless velocity profiles of the MRT-LBM along x-axis u_x: (a) Re = 100, (b) Re = 400, (c) Re = 1000, along z-axis u_z: (d) Re = 100, (e) Re = 400, (f) Re = 1000.

References

[BDLL02] Bouzidi, M., D'Humières, D., Lallemand, P., and Luo, L. S. Lattice boltzmann equation on a two-dimensional rectangular grid. *J. Comput. Phys.*, 172:704–717, 2002.

[BFL01] Bouzidi, M., Firdaouss, M., and Lallemand, P. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Phys. Fluids*, 13(11):3452–3459, 2001.

[BGK54] Bhatnagar, P. L., Gross, E. P., and Krook, M. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94(3):511–525, 1954.

[BJG14] Banari, A., Janßen, C. F., and Grilli, S. T. An efficient lattice Boltzmann multiphase model for 3D flows with large density ratios at high Reynolds numbers. *Comput. Math. with Appl.*, 68(12):1819–1843, 2014.

[BJGK14] Banari, A., Janßen, C., Grilli, S. T., and Krafczyk, M. Efficient GPGPU implementation of a lattice Boltzmann model for multiphase flows with high density ratios. *Comput. Fluids*, 93:1–17, 2014.

[BJK$^+$09] Becker, J., Junk, M., Kehrwald, D., Thömmes, G., and Yang, Z. A combined lattice BGK/level set method for immiscible two-phase flows. *Comput. Math. with Appl.*, 58(5):950–964, 2009.

[BS06] Bruneau, C. H. and Saad, M. The 2D lid-driven cavity problem revisited. *Comput. Fluids*, 35(3):326–348, 2006.

[CCM92] Chen, H., Chen, S., and Matthaeus, W. H. Recovery of the Navier-Stokes equations using a lattice-gas Soltzmann method. *Phys. Rev. A*, 45(8):5339–5342, 1992.

[CGK$^+$16] Calore, E., Gabbana, A., Kraus, J., Pellegrini, E., Schifano, S. F., and Tripiccione, R. Massively parallel lattice-Boltzmann codes on large GPU clusters. *Parallel Comput.*, 58:1–24, 2016.

[DBL01] D'Humières, D., Bouzidi, M., and Lallemand, P. Thirteen-velocity three-dimensional lattice Boltzmann model. *Phys. Rev. E*, 63:1–7, 2001.

[DGK$^+$02] D'Humières, D., Ginzburg, I., Krafczyk, M., Lallemand, P., and Luo, L. S. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Philos. Trans. A. Math. Phys. Eng. Sci.*, 360:437–451, 2002.

[DSGS17] Dietrich, R., Schmitt, F., Grund, A., and Stolle, J. Critical-blame analysis for OpenMP 4.0 offloading on Intel Xeon Phi. *J. Syst. Softw.*, 125:381–388, 2017.

[FDH$^+$87] Frisch, U., D'Humières, D., Hasslacher, B., Lallemand, P., Pomeau, Y., and Rivet, J. P. Lattice gas as hydrodynamics in two and three dimensions. *Complex Syst.*, 1:649–707, 1987.

[FL14] Fakhari, A. and Lee, T. Finite-difference lattice Boltzmann method with a block-structured adaptive-mesh-refinement technique. *Phys. Rev. E*, 89(3):1–12, 2014.

[GA94] Ginzbourg, I. and Adler, P. M. Boundary flow condition analysis for the three-dimensional lattice Boltzmann model. *J. Phys. II*, 4:191–214, 1994.

[GA95] Ginzbourg, I. and Adler, P. M. Surface tension models with different viscosities. *Transp. Porous Media*, 20(1-2):37–76, 1995.

[GDL97] Giraud, L., D'Humières, D., and Lallemand, P. A lattice Boltzmann model for visco-elasticity. *Int. J. Mod. Phys. C*, 8(4):805–815, 1997.

[GDL98] Giraud, L., D'Humières, D., and Lallemand, P. A lattice Boltzmann model for Jeffreys viscoelastic fluid. *Europhys. Lett.*, 42(6):625–630, 1998.

[GFN02] Gotoh, T., Fukayama, D., and Nakano, T. Velocity field statistics in homogeneous steady turbulence obtained using a high-resolution direct numerical simulation. *Phys. Fluids*, 14(3):1065–1081, 2002.

[GGS82] Ghia, U., Ghia, K. N., and Shin, C. T. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.*, 48(3):387–411, 1982.

[Gra49] Grad, H. Note on N-dimensional hermite polynomials. *Commun. Pure Appl. Math.*, 2(4):325–330, 1949.

[GS02] Ginzburg, I. and Steiner, K. A free-surface lattice Boltzmann method for modelling the filling of expanding cavities by Bingham fluids. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, 360(1792):453–466, 2002.

[HDP76] Hardy, J., De Pazzis, O., and Pomeau, Y. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Phys. Rev. A*, 13(5):1949–1961, 1976.

[HL97] He, X. and Luo, L. S. Lattice boltzmann model for the incompressible Navier-Stokes equation. *J. Stat. Phys.*, 88(3/4):927–944, 1997.

[HSCD96] Hou, S., Sterling, J., Chen, S., and Doolen, G. D. A lattice boltzmann subgrid model for high Reynolds number flows. *Fields Inst. Commun.*, 6:151–166, 1996.

[IOTK04] Inamuro, T., Ogata, T., Tajima, S., and Konishi, N. A lattice Boltzmann method for incompressible two-phase flows with large density differences. *J. Comput. Phys.*, 198:628–644, 2004.

[IYO95] Inamuro, T., Yoshino, M., and Ogino, F. A non-slip boundary condition for lattice Boltzmann simulations. *Phys. Fluids*, 7(12):2928–2930, 1995.

[IYO97] Inamuro, T., Yoshino, M., and Ogino, F. Accuracy of the lattice Boltzmann method for small Knudsen number with finite Reynolds number. *Phys. Fluids*, 9(11):3535–3542, 1997.

[IYTT16] Inamuro, T., Yokoyama, T., Tanaka, K., and Taniguchi, M. An improved lattice Boltzmann method for incompressible two-phase flows with large density differences. *Comput. Fluids*, 137:55–69, 2016.

[JK11] Janßen, C. and Krafczyk, M. Free surface flow simulations on GPGPUs using the LBM. *Comput. Math. with Appl.*, 61(12):3549–3563, 2011.

[KHT87] Ku, H. C., Hirsh, R. S., and Taylor, T. D. A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations. *J. Comput. Phys.*, 70(2):439–462, 1987.

[KS16] Kuwata, Y. and Suga, K. Imbalance-correction grid-refinement method for lattice Boltzmann flow simulations. *J. Comput. Phys.*, 311:348–362, 2016.

[KTH$^+$05] Körner, C., Thies, M., Hofmann, T., Thürey, N., and Rüde, U. Lattice Boltzmann model for free surface flow for modeling foaming. *J. Stat. Phys.*, 121(1/2):179–196, 2005.

[KTL03] Krafczyk, M., Tölke, J., and Luo, L. S. Large-eddy simulations with a multiple-relaxation-time LBE model. *Int. J. Mod. Phys. B*, 17(1–2):33–39, 2003.

[Lad94] Ladd, A. J. C. Numerical simulations of particulate suspensions via a discretized boltzmann equation. part 1. theoretical foundation. *J. Fluid Mech.*, 271:285–309, 1994.

[LL00] Lallemand, P. and Luo, L. S. Theory of the lattice boltzmann method: dispersion dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E*, 61(6):6546–6562, 2000.

[Luo98] Luo, L. S. Unified theory of lattice Boltzmann models for nonideal gases. *Phys. Rev. Lett.*, 81(8):1618–1621, 1998.

[Luo00] Luo, L. S. Theory of the lattice Boltzmann method: Lattice Boltzmann models for nonideal gases. *Phys. Rev. E*, 62(4):4982–4996, 2000.

[MBG96] Maier, R. S., Bernard, R. S., and Grunau, D. W. Boundary conditions for the lattice Boltzmann method. *Phys. Fluids*, 8(7):1788–1801, 1996.

[MC98] Masselot, A. and Chopard, B. A lattice Boltzmann model for particle transport and deposition. *Europhys. Lett.*, 42(3):259–264, 1998.

[MGA95] McNamara, G. R., Garcia, A. L., and Alder, B. J. Stabilization of thermal lattice Boltzmann models. *J. Stat. Phys.*, 81(1):395–408, 1995.

[MZ88] McNamara, G. R. and Zanetti, G. Use of the boltzmann equation to simulate lattice-gas automata. *Phys. Rev. Lett.*, 61(20):2332–2335, 1988.

[OKTR13] Obrecht, C., Kuznik, F., Tourancheau, B., and Roux, J. J. Scalable lattice Boltzmann solvers for CUDA GPU clusters. *Parallel Comput.*, 39(6-7):259–270, 2013.

[QDL92] Qian, Y. H., D'Humières, D., and Lallemand, P. Lattice BGK models for Navier-Stokes Equation. *Europhys. Lett.*, 6(17):479–484, 1992.

[RDVC12] Rinaldi, P. R., Dari, E. A., Vénere, M. J., and Clausse, A. A Lattice-Boltzmann solver for 3D fluid simulation on GPU. *Simul. Model. Pract. Theory*, 25:163–171, 2012.

[SKTC15] Suga, K., Kuwata, Y., Takashima, K., and Chikasue, R. A D3Q27 multiple-relaxation-time lattice Boltzmann method for turbulent flows. *Comput. Math. with Appl.*, 69(6):518–529, 2015.

[Sma63] Smagorinsky, J. General circulation experiments with the primitive equations. *Mon. Weather Rev.*, 91(3):99–164, 1963.

[SPT17] Safi, M. A., Prasianakis, N., and Turek, S. Benchmark computations for 3D two-phase flows: A coupled lattice Boltzmann-level set study. *Comput. Math. with Appl.*, 73(3):520–536, 2017.

[ST16] Safi, M. A. and Turek, S. GPGPU-based rising bubble simulations using a MRT lattice Boltzmann method coupled with level set interface capturing. *Comput. Fluids*, 124:170–184, 2016.

[TFK06] Tölke, J., Freudiger, S., and Krafczyk, M. An adaptive scheme using hierarchical grids for lattice Boltzmann multi-phase flow simulations. *Comput. Fluids*, 35(8-9):820–830, 2006.

[TPR+06] Thürey, Nils, Pohl, Thomas, Rüde, Ulrich, Öchsner, Markus, and Körner, Carolin. Optimization and stabilization of LBM free surface flow simulations using adaptive parameterization. *Comput. Fluids*, 35(8-9):934–939, 2006.

[YF09] Yu, Z. and Fan, L. S. An interaction potential based lattice Boltzmann method with adaptive mesh refinement (AMR) for two-phase flow simulation. *J. Comput. Phys.*, 228(17):6456–6478, 2009.

[YGL05] Yu, H., Girimaji, S. S., and Luo, L. S. DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *J. Comput. Phys.*, 209(2):599–616, 2005.

[ZH97] Zou, Q. and He, X. On pressure and velocity flow boundary conditions and bounceback for the lattice Boltzmann BGK model. *Phys. Fluids*, 9(6):1591–1598, 1997.

Chapter 3

Free surface simulation by the Volume-of-Fluid method

3.1 Introduction

When the governing equations are solved on a fixed grid such as the lattice Boltzmann method, using one set of equations for the whole flow field, the different fluids have to be identified in some way. This is generally done by using a marker function that takes different values in the different fluids. Sometimes a material property, such as the fluid density for incompressible fluids, can serve as a marker function, but here we shall assume that the role of the marker function is only to identify the different fluids. As the fluids move, and the boundary between the different fluids changes location, the marker function must be updated. Updating the marker function accurately is both critical for the success of simulations of multiphase flows and also surprisingly difficult. In this chapter, we discuss the difficulties with advecting the marker function directly and the various methods that have been developed to overcome these difficulties. Next, we implement the free surface algorithm in the lattice Boltzmann method for three-dimensional tsunami modellings.

The volume-of-fluid (VOF) method [HN81, You82, Rud97] is the oldest approach to advect a marker function and - after many improvements and innovations - continues to be widely used. Other marker function methods include the level-set method [OS88, SSO94, SFSO98, OF01, DSG17], advanced hybrid VOF/Level-set method [SP00, Sus03], simplified hybrid VOF/Level-set method for interFoam with OpenFOAM [ADR$^+$13, YOD17], the phase-field method [Jac99, YFLS06, DSS07], and the cubic interpolated profile (CIP) method [TY87, NY99, YXU01]. Instead of advecting the marker function directly, the boundary between the different fluids can also be tracked using marker points, and the marker function then reconstructed from the location of the interface. Methods using marker points are generally referred to as front-tracking methods to distinguish them from front-capturing methods, where the marker function is advected directly.

3.2 Notations

To identify whether a given fluid i is present at a particular location x, we use a Heaviside function H, defined by:

$$H_i\left(x\right) = \begin{cases} 1 & \text{(if } x \text{ is in fluid } i\text{)} \\ 0 & \text{(if } x \text{ is not in fluid } i\text{)} \end{cases} \tag{3.1}$$

for a two-fluid system, $i = 1, 2$ and $H_2 = 1 - H_1$, so it is sufficient to work with $H = H_1$. As the interface moves, the shape of the region occupied by each fluid changes, but each fluid particle retains its identity. Thus, the material derivative of H is zero, or:

$$\frac{DH}{Dt} = \frac{\partial H}{\partial t} + \boldsymbol{u} \cdot \nabla H = 0 \tag{3.2}$$

once H is known, the material properties of each fluid can be found, and the velocity updated.

The colour function C is defined as the average value of H in each computational cell. For a rectangular two-dimensional cell:

$$C_{i,j} = \frac{1}{\Delta x \Delta y} \int_V H\left(x, y\right) \mathrm{d}x\mathrm{d}y \tag{3.3}$$

cells away from the interface are either full, $C = 1$, or empty, $C = 0$, but if an interface is located somewhere in a given cell, C for that cell has a fractional value.

3.3 Advection of the colour function

It is perhaps somewhat counterintuitive that the seemingly simple problem of pushing a piecewise-constant function around by a prescribed velocity field, using the linear 1-st order advection equation, should be one of the hard problems in computational science. The difficulty is even more surprising given the deceptive simplicity of the one-dimensional problem. Yet, the one-dimensional problem can serve as both an introduction to the difficulties and as a motivation for the various techniques that have been designed to overcome the difficulties. We shall therefore examine the one-dimensional problem in some detail here assuming the flow is incompressible so that the velocity is constant, $u = U > 0$.

The advection of the colour function is governed by:

$$\frac{\partial C}{\partial t} + \frac{\partial F}{\partial x} = 0 \tag{3.4}$$

where $F = UC$ is the flux function. As the interface moves to the right. C flows into cell j through the left boundary and our through the right boundary. For cell j we denote the left and the right boundaries by $j - 1/2$ and $j + 1/2$, respectively, and the fluxes by $F_{j-1/2}$ and $F_{j+1/2}$. If the value of C in cell j at time level n is denoted by C_j^n and the value at the end of a time step Δt by C_j^{n+1}, then we can write as following:

$$C_j^{n+1} = C_j^n - \frac{1}{\Delta x} \int_t^{t+\Delta t} \left(F_{j+1/2} - F_{j-1/2}\right) \mathrm{d}t \tag{3.5}$$

or, the change in C_j is the difference between what flows in and what flows out of cell j. Here, $F_{j-1/s} = UC_{j-1/2}$ and $F_{j+1/2} = UC_{j+1/2}$, and the key challenge is to accurately estimate $C_{j-1/2}$ and $C_{j+1/2}$.

Assuming C in each cell to be a constant, then the integration of the fluxes over time is simple. The value of C that crosses the $j - 1/2$ boundary is simply the value in the cell to the left, C_{j-1}, and the value of C crossing the $j + 1/2$ boundary is C_j. Thus, as long as limiting the size of each time step to $\Delta tU \leq \Delta$, we can update C_j by:

$$C_j^{n+1} = C_j^n - \frac{U\Delta t}{\Delta x} \left(C_j - C_{j-1}\right) \tag{3.6}$$

3.4 The volume of fluid (VOF) method

3.4.1 Introduction

In one dimension, as depicted in Figure 3.1, the answer is embarrassingly simple. C_j is either 0 or 1, except in an interface cell, so the value of C_j in the interface cell yields immediately the exact location of the interface. If $C_{j-1} = 1$, $C_j = 1/3$ and $C_{j+1} = 0$ then the interface is $\Delta x/3$ from the left boundary. If $C_j = 1/2$, then the interface is in the middle of the cell, and so forth. If $U \geq 0$, then the flux through the left boundary is always U and we can compute exactly the flux through the right boundary, since we know the exact location of the interface. If $U\Delta t < (1 - C_j)\,\Delta x$, then the flux is zero; and if $U\Delta t > (1 - C_j)\,\Delta x$, then the flux is first zero and then U after the interface reaches the right boundary:

$$\int_t^{t+\Delta t} F_{j+1/2} \mathrm{d}t = \begin{cases} 0 & (\Delta t \leq (1 - C_j)\,\Delta x/U) \\ (C_j - 1)\,\Delta x + U\Delta t & (\Delta t > (1 - C_j)\,\Delta x/U) \end{cases} \tag{3.7}$$

thus, for one-dimensional problems the advection of the marker function is essentially trivial. The first attempt to extend the advection described by the above equation to higher dimensions was the simple line interface calculation, or simple linear interface calculation (SLIC) method [NW76]. In this approach the marker function is advected by time splitting, where we advect first in one coordinate direction and then in the other (assuming two-dimensional flow). For advection in the horizontal direction, an interface cell is divided by a vertical line into a full part and an empty part, with the decision of which side is empty and which is full depending on the volume fraction in the cells to the left and the right. Once the location of the interface has been determined, the time integration of the fluxes is done using the above equation. The interface is then advected in the vertical direction by dividing the cell by a horizontal line into a full and an empty part. In three dimensions this process obviously must also be repeated for the third coordinate direction. Hirt and Nichols [HN81] proposed a slightly different method where the interface was still approximated by straight lines, parallel to the coordinate directions. To determine whether the interface should be horizontal or vertical, Hirt and Nichols [HN81] have found the normal to the interface, using values of C in the neighboring cells, and selected the orientation of the interface depending on whether the normal was more closely aligned with the horizontal or the vertical axis. Although perhaps more appealing than the original SLIC method, tests by [Rud97] suggest that the [HN81] method is not significantly more accurate. In addition to distorting the interfaces, both methods generally generate considerable amount of ”floatsam” and ”jetsam”, where pieces of the interface break away in an unphysical way.

Although the method of [HN81] perhaps did not improve significantly on the SLIC approach, it nevertheless suggested that the key to improving the behaviour of the advection scheme was the reconstruction of the interface in each cell, using the value of the marker function in each cell, along with the value in the neighbouring cells. In the piecewise linear interface calculation (PLIC) method introduced by Youngs [You82], the interface is approximated by a straight-line segment in each cell, but the line can be oriented arbitrarily with respect to the coordinate axis. As of today, Multi-interface advection and reconstruction solver (MARS) has been proposed by [KK05] as an advanced PLIC algorithm for multiphase fluid analysis. The orientation of the line is determined by the normal to the interface, which is found by considering the value C in both the cell under consideration and in the adjacent cells. Once the interface in each cell has been constructed, the fluxes from one cell to another are computed by geometric considerations. The result of the advection generally depends on the accuracy of the interface reconstruction; finding the normal accurately, therefore, becomes critical for PLIC methods. Several methods have been proposed to do so. Given C in each cell and the normals, the exact location of the interface can be determined. In two dimensions the line segment can cross any of two adjacent or opposite cell faces, so there are two basic interface configurations. In three dimensions there are many more possible configurations, adding considerably to the complexity of the method.

3.4.2 Basic properties

The volume fraction or colour function C is the discrete version of the characteristic function H. We will be considering only free surface flows in this thesis, so that the C data represent the fraction of each grid cell occupied by reference phase. Furthermore, we restrict our analysis to Cartesian grids with square cells of side $h = \Delta x = \Delta y$ for simplicity.

The function C varies between the constant value one in full cells to zero in empty cells, while mixed cells with an intermediate value of C define the transition region where the interface is localised.

Low-order VOF methods do not need to specify the location of the interface in the transition region, but a geometrical interpretation of these methods shows that in two dimensions the interface line in each mixed cell is represented by a segment parallel to one of the two coordinate axes. The interface is clearly not continuous across the cell boundary, and the jump usually is of order h, $O(h)$, as seen in Figure 3.2(b). Higher order methods reconstruct the interface in various ways. The standard one is the PLIC reconstruction, where the interface in each mixed cell is represented by a segment perpendicular to the local gradient, $\boldsymbol{m} = -\nabla C$, of the scalar function C. There is, however, still no requirement that the interface be continuous at the cell boundary, but now the interface discontinuity is usually much smaller and in general it is a function of the grid spacing h.

The VOF method proceeds in two steps:

1. Reconstruction of the interface shape: from the knowledge of the volume fraction in each cell, one has to build an approximation of the interface. The problem is illustrated in Figure 3.3. The key issue for the PLIC reconstruction is to find the local normal vector.

2. Advection of the reconstructed interface in a given velocity field. This amounts to exchanging reference phase volumes across the boundary of neighbouring cells.

We describe these two steps in the following sections using in this thesis. Several qualities are expected from a VOF method. It preserves mass in a natural way provided the advection method is adequate. Topology changes, such as those occurring during reconstruction or breakup, are implicit in the formulation. The extension from two-dimensional to three-dimensional Cartesian geometry is relatively straightforward. The fraction C data structure is a static matrix on a fixed grid and does not require any dynamical adjustment, as opposed to a marker approach. Furthermore, the algorithms are local, in the sense that only the C values of the neighbouring cells are needed to update the C value in a given cell. For this reason, it is relatively simple to implement these algorithms in parallel, in the framework of domain decomposition techniques.

Of all these qualities, the mass conservation is specific to the VOF technique. Let us consider the multidimensional version of the advection equation for the marker function H for an incompressible flow:

$$\frac{\partial H}{\partial t} + \nabla \cdot (\boldsymbol{u} H) = 0 \tag{3.8}$$

we first integrate this equation over the square cell (i, j) of side h of a Cartesian two-dimensional grid and use the definition of the colour function C:

$$h^2 \frac{\partial C_{i,j}(t)}{\partial t} + \int_{\Gamma} \boldsymbol{u} \cdot \boldsymbol{n} H(\boldsymbol{x}, t)\, \mathrm{d}l = 0 \tag{3.9}$$

where Γ is the cell boundary line and $\boldsymbol{n}$ the outgoing unit normal. Finally, we integrate the above equation in the time $\Delta t = t^{n+1} - t^n$ to get:

$$h^2 \left(C_{i,j}^{n+1} - C_{i,j}^{n}\right) = -\left(\Phi_{x:i+1/2,j}^{n} - \Phi_{x:i-1/2,j}^{n}\right) - \left(\Phi_{y:i,j+1/2}^{n} - \Phi_{y:i,j-1/2}^{n}\right) \tag{3.10}$$

where the flux $\Phi_{x:i+1/2,j}^{n}$ denotes the reference phase area crossing the right side of the cell in the time Δt. By summing the above equation over all the grid cells with appropriate boundary conditions, the internal fluxes cancel out in pairs and we get:

$$\sum_{ij} C_{i,j}^{n+1} = \sum_{ij} C_{i,j}^{n} \tag{3.11}$$

which leads to conservation of the total area. Notice that in writing expression the Equation 3.10 we have implicitly assumed that the numerical algorithm does not generate unphysical overshoots, $C > 1$, or undershoots, $C < 0$, in the volume fraction and that it dose not flux some area twice across the cell boundary.

3.5 Interface reconstruction

For PLIC methods the reconstruction is basically a two-step procedure. In any given cell the normal $\boldsymbol{m}$ (the notation $\boldsymbol{n}$ is used for the unit normal, i.e. $\boldsymbol{n} = \boldsymbol{m}/\|\boldsymbol{m}\|$) is first determined from the knowledge of the colour function in this cell and in the neighbouring ones. The equation of the interface segment is then written as:

$$\boldsymbol{m} \cdot \boldsymbol{x} = m_x x + m_y y = \alpha \tag{3.12}$$

Geometrically, the interface line given as the above equations is moved along the normal direction, i.e. the parameter α is adjusted, until the area under the interface equals $h^2 C_{i,j}$.

3.6 Convergence order of a reconstruction method

The numerical estimate of the integral of a given function $f(x)$ geometrically requires the evaluation of the area comprised between the graph of the function and the coordinate x-axis. When we reconstruct an interface from the volume fraction data we are actually solving the inverse problem: we know the area under the interface line and we want to find the function.

To integrate a function $f(x)$ we can subdivide the range $a \leq x \leq b$ into n equal subintervals of size $h = \Delta x = (b-a)/n$. In particular, in the trapezoidal rule we approximate the function with a piecewise linear interpolation and in each subinterval the area under the function is given by the area of a right trapezoid, as shown in Figure 3.4. The truncation error in the evaluation of the area spanned by the function, which is supposed to have continuous second derivatives, is proportional to $f''(\xi)\, h^2$, with $a \leq \xi \leq b$. Hence, the trapezoidal rule is a second-order method, $O\left(h^2\right)$, and a straight line, by having a zero second derivative, is reproduced correctly. In Figure 3.4 we see that the line arc is approximated differently by the trapezoidal rule and a typical PLIC-VOF reconstruction. However, it is reasonable to require that a second-order reconstruction algorithm should have a similar truncation error. Reconstruction techniques that do not reproduce an arbitrary straight line exactly are lower than second order, i.e. they are $O(h^n)$ with $n < 2$. A method that approximates interfaces as circle arcs could be third order, which is very high for a VOF method. In our opinion it is not practical to attempt to develop VOF methods with polynomials of such an order, in particular in three dimension.

3.7 Evaluation of the interface unit normal

The choice of the method used to calculate the interface normal is independent of the other steps: it can be based on a finite-difference approximation of the volume fraction gradient ∇C or satisfy some other minimizing criteria. All the algorithms described in this section use a 3×3 block of cells to determine the approximate interface in the central cell of the block. In Figure 3.2(c) this central cell is denoted by the two indices (i, j).

3.7.1 Youngs' finite-difference method

In this method, developed by [You84], the normal $\boldsymbol{m}$ of the equations of the interface segment is estimated as a gradient:

$$\boldsymbol{m} = -\nabla_h C \tag{3.13}$$

with finite differences. We first evaluate the normal vector $\boldsymbol{m}$ at the four corners of the central cell (i, j); for example, the components of $\boldsymbol{m}$ in two dimension on the top-right corner are given by:

$$m_{x:i+1/2,j+1/2} = -\frac{1}{2h}\left(C_{i+1,j+1} + C_{i+1,j} - C_{i,j+1} - C_{i,j}\right) \tag{3.14}$$

$$m_{y:i+1/2,j+1/2} = -\frac{1}{2h}\left(C_{i+1,j+1} - C_{i+1,j} + C_{i,j+1} - C_{i,j}\right) \tag{3.15}$$

and similarly for the other three corners. The cell-centred vector is finally obtained by averaging the four cell-corner values:

$$\boldsymbol{m}_{i,j} = \frac{1}{4}\left(\boldsymbol{m}_{i+1/2,j+1/2} + \boldsymbol{m}_{i+1/2,j-1/2} + \boldsymbol{m}_{i-1/2,j+1/2} + \boldsymbol{m}_{i-1/2,j-1/2}\right) \tag{3.16}$$

This finite-difference scheme does not reproduce any straight line exactly. This may by shown by working out the expressions in a particular case, for instance by considering the line $y = 2x/3 + h$ in the 3×3 block of cells of side h with the origin in the bottom-left corner.

3.7.2 Centred-columns difference method

In the same block of cells the volume fractions can be added column-wise along the vertical direction to define the height function $y = f(x)$, or row-wise for the width function $x = g(y)$. For example, the height y_{i-1} at the abscissa x_{i-1}, placed in the centre of the column as shown in Figure 3.5(a), is given by the expression $hy_{i-1} = h^2 \sum_{k=-1}^{1} C_{i-1,j+k}$, while in Figure 3.5(b) the width x_{j+1} at the ordinate y_{j+1} is $hx_{i-1} = h^2 \sum_{k=-1}^{1} C_{i+k,j+1}$. The height y_{i-1} is placed exactly on a linear interface only if the straight line cuts the two vertical sides of the column (Figure 3.5(a) and (c) for two different cases). We approximate the height function $y = f(x)$ in the central cell of the block with the linear equation:

$$\operatorname{sgn}(m_y)\, y = -m_x x + \alpha' \tag{3.17}$$

and compute the slope of the straight line with a centred scheme, $m_x = m_{xc}$,

$$m_{xc} = -\frac{1}{2h}\left(y_{i+1} - y_{i-1}\right) = -\frac{1}{2}\sum_{k=-1}^{1}\left(C_{i+1,j+k} - C_{i-1,j+k}\right) \tag{3.18}$$

since $m_y = -\partial C/\partial y$, we compute the sign of the variation of C along the y-direction with centred finite differences. We have to calculate explicitly the sing of m_y, because by adding C column-wise we lose information about which phase is on the top or on the bottom of the block of cells.

We can also describe the interface line with the width function $x = g(y)$ and approximate it linearly with:

$$\operatorname{sgn}(m_x)\, x = -m_y y + \alpha'' \tag{3.19}$$

and similarly compute m_y as:

$$m_{yc} = -\frac{1}{2h}\left(x_{j+1} - x_{j-1}\right) = -\frac{1}{2}\sum_{k=-1}^{1}\left(C_{i+k,j+1} - C_{i+k,j-1}\right) \tag{3.20}$$

it is evident from Figure 3.5 that when a linear interface cuts two opposite sides of the block of cells, one of the two representations gives the correct slope, m_{xc} in case (a) and m_{yc} in case (b). In case (c), where the interface cuts two consecutive sides, the centred-columns scheme does not compute the correct slope. As a matter of fact, a wider stencil with more columns or rows should be used; however, backward and forward difference schemes can also be considered. Since the centred-columns scheme cannot reconstruct any straight line exactly, it is not second order. Furthermore, the interface shape is not actually known, so we need a strategy to select one between the two numerically computed values m_{yc} and m_{xc}. To this aim we define a simple criterion based on the reconstruction of a linear interface. We consider the two linear equations $y = m_x x + \alpha'$ and $x = m_y y + \alpha''$ and notice the following conditions between the two coefficients m_x and m_y: $|m_x| = |m_y| = 1$ for a straight line with a 45-degree slope, $|m_x| = 1/|m_y| < 1$ for the line shown in Figure 3.5(a) and $|m_y| = 1/|m_x| < 1$ in Figure 3.5(b). Therefore, to reconstruct exactly the straight line in the two case (a) and (b), we have to select the angular coefficient with the minimum absolute value:

$$|m^*| = \min\left(|m_{xc}|, |m_{yc}|\right) \tag{3.21}$$

in the centred-columns method, we have considered a centred scheme to compute the angular coefficient, but one can also use forward or backward finite differences, as in the efficient least-squares VOF interface reconstruction algorithm (ELVIRA) method. A different criterion should be applied when we have two or more estimates for the same angular coefficient, say m_x. In Figure 3.5(c), y_{i-1} overestimates the local value of the height function and the absolute value of m_{xc}, based on the centred scheme, is smaller than the angular coefficient of the straight line. Hence, if we have two different estimates for the same angular coefficient, say m_{x1} and m_{x2}, we now choose:

$$|m^*| = \max\left(|m_{x1}|, |m_{x2}|\right) \tag{3.22}$$

with these two simple criteria we can reconstruct any linear interface, but they may not be very efficient in the case of a curved interface.

3.7.3 Parker and Youngs' method

This method is the most efficient and high-precision approach to calculate an interface normal by an explicitly way. In this method, due to [PP04], one calculates an approximation to ∇C, which is taken to point in the direction normal to the approximate interface. One calculates ∇C with the following difference scheme:

$$\frac{\partial C}{\partial x} = \frac{C_E - C_W}{2h} \tag{3.23}$$

$$\frac{\partial C}{\partial y} = \frac{C_N - C_S}{2h} \tag{3.24}$$

in two dimension; where the variables C_E, C_W, C_N and C_S are centred in the cells as shown in Figure 3.6 and are given by:

$$C_E = \frac{1}{2+\alpha}\left(C_{i+1,j-1} + \alpha C_{i+1,j} + C_{i+1,j+1}\right) \tag{3.25}$$

$$C_W = \frac{1}{2+\alpha}\left(C_{i-1,j-1} + \alpha C_{i-1,j} + C_{i-1,j+1}\right) \tag{3.26}$$

$$C_N = \frac{1}{2+\alpha}\left(C_{i-1,j+1} + \alpha C_{i,j+1} + C_{i+1,j+1}\right) \tag{3.27}$$

$$C_S = \frac{1}{2+\alpha}\left(C_{i-1,j-1} + \alpha C_{i,j-1} + C_{i+1,j-1}\right) \tag{3.28}$$

where α is a free parameter. Parker and Youngs has repeated that $\alpha = 2$ seems to give the best results. In order to determine how well Parker and Youngs' method approximates straight lines, we consider the line $y = x/3 + h$ in Figure 3.7(a) The volume fractions due to this line are shown in Figure 3.7(b). The values of C_E, C_W, C_N and C_S can be calculated as:

$$C_E = \frac{1}{\alpha+2}\left(\frac{5\alpha}{6} + 1\right) \tag{3.29}$$

$$C_W = \frac{1}{\alpha+2}\left(\frac{\alpha}{6} + 1\right) \tag{3.30}$$

$$C_N = 0 \tag{3.31}$$

$$C_S = \frac{1}{\alpha+2}\left(1 + \alpha + 1\right) = 1 \tag{3.32}$$

and hence:

$$\frac{\partial C}{\partial x} = \frac{C_E - C_W}{2h} = \frac{\alpha}{3\left(\alpha+2\right)h} \tag{3.33}$$

$$\frac{\partial C}{\partial y} = \frac{C_N - C_S}{2h} = -\frac{1}{2h} \tag{3.34}$$

the slope of the appropriate interface is therefore:

$$\bar{m} = -\frac{\partial C/\partial x}{\partial C/\partial y} = \frac{2\alpha}{3\left(\alpha+2\right)} \tag{3.35}$$

the correct slope of the line is $m = 1/3$. Thus if we wish to choose α so that:

$$m = \frac{2\alpha}{3\left(\alpha+2\right)} = \frac{1}{3} \tag{3.36}$$

We must have $\alpha = 2$. In other words, only the value of $\alpha = 2$ will yield the correct linear interface $y = x/3 + h$, We now show that choosing $\alpha = 2$ does not result in an algorithm that reconstructs all lines exactly. Consider the line $y = 2x/3 + h$ shown in Figure 3.7(c). The volume fractions due to this line are shown in Figure 3.7(d). When $\alpha = 2$ the values C_E, C_W, C_N and C_S can be calculated as:

$$C_E = \frac{1}{4}\left(1 + 2 + \frac{2}{3}\right) = \frac{11}{12} \tag{3.37}$$

$$C_W = \frac{1}{4}\left(1 + \frac{2}{3} + 0\right) = \frac{5}{12} \tag{3.38}$$

$$C_N = \frac{1}{4}\left(0 + \frac{2}{12} + \frac{2}{3}\right) = \frac{5}{24} \tag{3.39}$$

$$C_S = \frac{1}{4} = (1 + 2 + 1) = 1 \tag{3.40}$$

and hence:

$$\frac{\partial C}{\partial x} = \frac{C_E - C_W}{2h} = \frac{1}{2h}\left(\frac{11}{12} - \frac{5}{12}\right) = \frac{1}{4h} \tag{3.41}$$

$$\frac{\partial C}{\partial y} = \frac{C_N - C_S}{2h} = \frac{1}{2h}\left(\frac{5}{24} - 1\right) = -\frac{19}{48h} \tag{3.42}$$

the slope of the appropriate interface is therefore:

$$\bar{m} = -\frac{\partial C/\partial x}{\partial C/\partial y} = \frac{1}{4}\frac{48}{19} = \frac{12}{19} \tag{3.43}$$

Since the correct slope is $m = 2/3$, we conclude that [PP04] algorithm does not reconstruct all linear interfaces exactly as same as [You84] method.

Note that the quantity in the above result is independent of the grid with h. This implies that the approximation to the slope does not improve as $h \to 0$; i.e., in general this algorithm makes an $O(1)$ error in the slope of the interface. We therefore conclude that [PP04] algorithm is at best first-order accurate and have used the scheme to evaluate the interface normal in this thesis.

In three dimension, the gradient $\nabla_h C$ is obtained from the surrounding cell fluid fraction values:

$$\nabla_h C = \frac{1}{2h}\begin{pmatrix} \bar{C}_x(x+1, y, z) - \bar{C}_x(x-1, y, z) \\ \bar{C}_y(x, y+1, z) - \bar{C}_y(x, y-1, z) \\ \bar{C}_z(x, y, z+1) - \bar{C}_z(x, y, z-1) \end{pmatrix} \tag{3.44}$$

where $\bar{C}_{x,y,z}(\boldsymbol{x})$ are averaged values of the neighbouring cells as:

$$\bar{C}_x(x, y, z) = \sum_{i=-1}^{1}\sum_{j=-1}^{1} C(x, y+i, z+j) \cdot w_{i,j} \tag{3.45}$$

$$\bar{C}_y(x, y, z) = \sum_{i=-1}^{1}\sum_{j=-1}^{1} C(x+i, y, z+j) \cdot w_{i,j} \tag{3.46}$$

$$\bar{C}_z\left(x,y,z\right)=\sum_{i=-1}^{1}\sum_{j=-1}^{1}C\left(x+i,y+j,z\right)\cdot w_{i,j} \tag{3.47}$$

where $w_{i,j}$ is the weighting factor similar to α in two dimension. In three dimension, the following weights produce the best results:

$$w_{0,0}=4,\;\; w_{\pm1,0}=w_{0,\pm1}=2 \;\text{ and }\; w_{\pm1,\pm1}=1 \tag{3.48}$$

3.7.4 The ELVIRA method

Other high order schemes can be used to determine the interface normal $\boldsymbol{m}$. One of them is the efficient least squares volume-of-fluid interface reconstruction algorithm (ELVIRA) introduced by [PP04]. We consider the height function $y=f\left(x\right)$, and for the slope m_x we compute the centred scheme and also the backward and forward estimates m_{xc}, m_{xb} and m_{xf}, which are given by the following expressions:

$$m_{xc}=-\frac{1}{2h}\left(y_{i+1}-y_{i-1}\right)=-\frac{1}{2}\sum_{k=-1}^{1}\left(C_{i+1,j+k}-C_{i-1,j+k}\right) \tag{3.49}$$

$$m_{xb}=-\frac{1}{h}\left(y_{i}-y_{i-1}\right)=-\sum_{k=-1}^{1}\left(C_{i,j+k}-C_{i-1,j+k}\right) \tag{3.50}$$

$$m_{xf}=-\frac{1}{h}\left(y_{i+1}-y_{i}\right)=-\sum_{k=-1}^{1}\left(C_{i+1,j+k}-C_{i,j+k}\right) \tag{3.51}$$

and similarly we compute m_{yc}, m_{yb} and m_{yf} for the angular coefficient m_y of the width function. There are now six different choices and a criterion has to be designed in order to select the best linear approximation. In the ELVIRA strategy the choice is done by minimizing a measure of the error between the volume fractions given by true and approximate interfaces. For each of the six normal vectors $\boldsymbol{m}_n$, with $1\le n\le 6$, the corresponding value of the line constant α_n is first determined by area conservation in the central cell. Then the approximate linear interface is drawn across the whole 3×3 block of cells, defining a new volume fraction value $\tilde{C}$ in each of the surrounding eight cells. The normalised area in the cell (k,l) under the line corresponding to the normal $\boldsymbol{m}_n$ is $h^2\tilde{C}_{k,l}\left(\boldsymbol{m}_n\right)$, and the area error in L_2 is:

$$E\left(n\right)=h^2\sum_{k=i-1}^{i+1}\sum_{l=j-1}^{j+1}\left(\tilde{C}_{k,l}\left(\boldsymbol{m}_n\right)-C_{k,l}\right)^2 \tag{3.52}$$

and in L_∞ it is:

$$E\left(n\right)=h^2\max\left|\bar{C}_{k,l}\left(\boldsymbol{m}_n\right)-C_{k,l}\right| \tag{3.53}$$

The selected value of $\boldsymbol{m}_n$ among the six angular coefficients is the one that minimises E. It is easy to verify that there is always at least one selection that reproduces correctly a straight line even when it cuts two consecutive sides of the block of cells, as in Figure 3.5(c), since the line is constrained to go across two consecutive rows or columns at least. For a tessellation of the plane with rectangular cells of equal size, this algorithm requires a 3×3 block of cells to reconstruct any linear interface; however, a wider stencil is required when the cell size is not constant across the computational domain. The ELVIRA method has the drawback of being relatively slow, especially in three dimensions, where the number of evaluations of the predicted volume fractions $\bar{C}_{k,l,p}\left(\boldsymbol{m}_n\right)$ becomes large.

3.8 Determination of the interface parameter

Once the normal $\boldsymbol{m}$ has been calculated, the non-homogeneous term α of the interface segment equation is determined by enforcing area conservation. With reference to Figure 3.8, for a more general rectangular cell, of sides Δx and Δy in two dimension, it can be computed with an iterative procedure that requires an estimate of α, the determination of the area $A\left(\alpha\right)$ of the polygon ABFGD, and its comparison with the value $\Delta x \Delta y C$, until the difference between the two areas is below some prescribed tolerance. In other terms, we have to find a zero of the nonlinear function:

$$h\left(\alpha\right) = A\left(\alpha\right) - \Delta x \Delta y C \tag{3.54}$$

with a root-finding algorithm (e.g., Brent method). The area of ABFGD can be found by collecting in counterclockwise order the coordinates (x_i, y_i) of its vertices and by using the following formula for the area A of a n-sided polygon:

$$A = \frac{1}{2}\sum_{k=1}^{n}\left(x_k y_{k+1} - x_{k+1} y_k\right) \tag{3.55}$$

where vertex $n+1$ coincides with vertex 1 We describe an alternative approach [SZ00] which is not iterative and that relies heavily on the symmetry of Cartesian cell so that, with proper mirror reflections about the coordinate axes, the interface segment equation has both coefficients m_x and m_y positive, as shown in Figure 3.8. In this case the area V of the polygon ABFGD is given by the expression:

$$V = \frac{1}{2m_x m_y}\left[\alpha^2 - F_2\left(\alpha - m_x \Delta x\right) - F_2\left(\alpha - m_y \Delta y\right)\right] \tag{3.56}$$

where $F_n\left(z\right) = z^n$ when $z > 0$ and zero otherwise. the three contributions to V represent the areas of the similar triangles AEH, BEF and DGH. The expression the about equation is a quadratic function of α when the interface line cuts two consecutive cell sides, then one of the two cut figures is a triangle, and it is linear when the two intersections with the cell boundary are on opposite sides.

Furthermore, it is a continuous, strictly monotonically increasing function of α and it can be easily inverted.

In three-dimensional space with Cartesian coordinates (x_1, x_2, x_3) we consider a rectangular parallelepiped of sides Δx_1, Δx_2, Δx_3 and a plane with normal vector $\boldsymbol{m} = (m_1, m_2, m_3)$ given by:

$$\boldsymbol{m} \cdot \boldsymbol{x} = m_1 x_1 + m_2 x_2 + m_3 x_3 = \alpha \tag{3.57}$$

In the standard forward problem we assume that the three coefficient m_i are all positive and we need to determine the cut volume ABGHKNML of the rectangular cell which is also below the given plane shown in Figure 3.9. Then the function $f(x_1, x_2, x_3) = m_1 x_1 + m_2 x_2 + m_3 x_3 - \alpha$ is negative at point A and positive at point F, while the vector $\boldsymbol{m}$ is pointing toward the region where f is positive. If we interchange the two species the normal vector reverses its orientation, the coefficients m_i are now negative, and the volume of interest becomes DCFEKNML. The volume ABGHKNML in three dimension is given by:

$$V = \frac{1}{6 m_1 m_2 m_3} \left[\alpha^3 - \sum_{i=1}^{3} F_3 \left(\alpha - m_i \Delta x_i \right) + \sum_{i=1}^{3} F_3 \left(\alpha - \alpha_{\max} + m_i \Delta x_i \right) \right] \tag{3.58}$$

where $\alpha_{\max} = \sum_i m_i \Delta x_i$.

For the moment, we restrict an analysis to a unitary cube, i.e. $\Delta x_i = 1$; also normalise the plane equation by dividing it by $(\sum_i m_i)$; then $\alpha_{\max} = \sum_i m_i = 1$. The normalisation of the two-dimensional problem is similar. Later we will generalise this procedure to negative m_i and to a rectangular parallelepiped. We can now summarise some useful properties:

1. V is a continuous, one-to-one, monotonically increasing function of α with continuous first derivative,

2. Both V and α vary in the range $[0, 1]$,

3. The expression for V is invariant with respect to a permutation of the indices, so we need to consider only one case, say $m_1 \leq m_2 \leq m_3$ in three dimension and $m_1 \leq m_2$ in two dimension,

4. The graph of V has odd symmetry with respect to the point $(V, \alpha) = (1/2, 1/2)$, so we can restrict the analysis to the range $0 \leq \alpha \leq 1/2$,

5. In three dimension we let $m_{12} = m_1 + m_2$ and $m = \min(m_{12}, m_3)$; then V varies cubically in the region $0 \leq \alpha \leq m_1$, quadratically in $m_1 \leq \alpha \leq m_2$, again cubically in $m_2 \leq \alpha \leq m$, and finally in $m \leq \alpha \leq 1/2$ cubically if $m = m_3$ and otherwise linearly if $m = m_{12}$; in two dimension with $m = m_1$, V varies quadratically in the region $0 \leq \alpha \leq m$ and linearly in $m \leq \alpha \leq 1/2$,

6. In the interval $0 \leq \alpha \leq 1/2$ and for arbitrary m_i there is a lower bound for $V(\alpha)$. In three dimension this line is realised for $m_1 = m_2 = m_3 = 1/3$, corresponding to a plane cutting each coordinate plane with a 45°-angle line, and it is a quadratic function of α: $V = 2\alpha^2$,

7. In the same region there is also an upper bound. In three dimension this is the line with $m_1 = m_2 = 0$ and it represents a plane parallel to one of the three coordinates planes. The function $V(\alpha)$ is linear: $V = \alpha$. In two dimension the same linear function is obtained with $m_1 = 0$, representing a line parallel to one of the two coordinate axes.

8. In three dimension, the limit $m_1 \to 0$ is smooth: the lines become those of the two-dimensional problem. In particular, the first cubic region collapses into the origin, the quadratic one extends itself to $0 \leq \alpha \leq m_2 = m$, the next cubic region collapses to the point of the line at $\alpha = m$, and finally the straight line, since now $m_3 = (1-m) \geq m_{12} = m$, extends itself to the interval $m \leq \alpha \leq 1/2$. The further limit $m_2 \to 0$ is also smooth with the quadratic region collapsing into the origin.

From the above discussion, in two-dimensional forward problem we have:

$$V = \begin{cases} \dfrac{\alpha^2}{2m\,(1-m)} & (0 \leq \alpha < m) \\ \dfrac{\alpha}{1-m} - V_1 & (m \leq \alpha \leq 1/2) \end{cases} \tag{3.59}$$

the two-dimensional inverse problem is specified by:

$$\alpha = \begin{cases} \sqrt{2m\,(1-m)\,V} & (0 \leq V < V_1) \\ V\,(1-m) + m/2 & (V_1 \leq V \leq 1/2) \end{cases} \tag{3.60}$$

where $V_1 = m/2\,(1-m)$. Notice that the limit $m = 0$ is correctly described by these expressions.

Similarly, in three dimension forward problem is:

$$V = \frac{\alpha^3}{6m_1m_2m_3} \quad \text{for } 0 \leq \alpha < m_1 \tag{3.61}$$

$$V = \frac{\alpha\,(\alpha - m_1)}{2m_2m_3} + V_1 \quad \text{for } m_1 \leq \alpha < m_2 \tag{3.62}$$

$$V = \frac{\alpha^2\,(3m_{12} - \alpha) + m_1^2\,(m_1 - 3\alpha) + m_2^2\,(m_2 - 3\alpha)}{6m_1m_2m_3} \quad \text{for } m_2 \leq \alpha < m \tag{3.63}$$

for the fourth interval there are two possible cases, one for $m = m_3 < m_{12}$ and the other for $m = m_{12} < m_3$:

$$V = \frac{\alpha^2\,(3 - 2\alpha) + m_1^2\,(m_1 - 3\alpha) + m_2^2\,(m_2 - 3\alpha) + m_3^2\,(m_3 - 3\alpha)}{6m_1m_2m_3} \quad \text{for } m_3 \leq \alpha < 1/2 \tag{3.64}$$

$$V = \frac{2\alpha - m_{12}}{2m_3} \quad \text{for } m_{12} \le \alpha < 1/2 \tag{3.65}$$

the three-dimensional inverse problem is given by:

$$\alpha = \sqrt[3]{6m_1m_2m_3V} \quad \text{for } 0 \le V < V_1 \tag{3.66}$$

$$\alpha = \frac{1}{2}\left(m_1 + \sqrt{m_1^2 + 8m_2m_3\left(V - V_1\right)}\right) \quad \text{for } V_1 \le V < V_2 \tag{3.67}$$

$$P\left(\alpha\right) = a_3'\alpha^3 + a_2'\alpha^2 + a_1'\alpha + \alpha_0' = 0 \quad \text{for } V_2 \le V < V_3 \tag{3.68}$$

again there are two cases in the fourth interval, one for $V_3 = V_{31} < V_{32}$ and the other for $V_3 = V_{32} < V_{31}$:

$$P\left(\alpha\right) = a_3''\alpha^3 + a_2''\alpha^2 + a_1''\alpha + \alpha_0'' = 0 \quad \text{for } V_{31} \le V < 1/2 \tag{3.69}$$

$$\alpha = m_3V + \frac{m_{12}}{2} \quad \text{for } V_{32} \le V < 1/2 \tag{3.70}$$

in the previous relations $V_1 = m_1^2/\left(\max\left(6m_2m_3, \epsilon\right)\right)$, which is an approximation of the value $m_1^2/6m_2m_3$. This approximation is needed because the limit for V_1 as m_1, $m_2 \to 0$ is zero; however, numerically we must prevent the denominator of V_1 from becoming zero, so ϵ is an arbitrary small number. Aside from that, the above set of expressions for V and α is well behaved numerically for all possible m_i. The other limiting values of the range of validity of each relation are given by the following expressions:

$$V_2 = V_1 + \left(m_2 - m_1\right)/2m_3 \tag{3.71}$$

$$V_3 = V_{31} = \left[m_3^2\left(3m_{12} - m_3\right) + m_1^2\left(m_1 - 3m_3\right) + m_2^2\left(m_2 - 3m_3\right)\right]/\left(6m_1m_2m_3\right) \tag{3.72}$$

when $m = m_3$ or $V_3 = V_{32} = m_{12}/2m_3$ when $m = m_{12}$. For the coefficients of the two cubic polynomials we have $a_3' = -1$, $a_2' = 3m_{12}$, $a_1' = -3\left(m_1^2 + m_2^2\right)$, $a_0' = m_1^3 + m_2^3 - 6m_1m_2m_3V$, $a_3'' = -2$, $a_2'' = 3$, $a_1'' = -3\left(m_1^2 + m_2^2 + m_3^2\right)$, $a_0'' = m_1^3 + m_2^3 + m_3^3 - 6m_1m_2m_3V$.

In the third and fourth regions, when $V_{31} \le V \le 1/2$, there is the need to find the roots of the cubic polynomial $P\left(\alpha\right)$. An analytical solution can be easily found following [AS64]. With $a_3 = 1$, the discriminant $\Delta = p_0^3 + q_0^3$ is negative, and $P\left(\alpha\right)$ has three real roots $\left(\alpha_1, \alpha_2, \alpha_3\right)$. However, only α_2 satisfies the requirement that $P\left(\alpha\right)$ is an increasing function of α:

$$\alpha = \sqrt{-p_0}\left(\sqrt{3}\sin\theta - \cos\theta\right) - \frac{a_2}{3} \tag{3.73}$$

where we have set $p_0 = a_1/3 - \left(a_2/3\right)^2$, $\left(a_1a_2 - 3a_0\right)/6 - \left(a_2/3\right)^3$, and $\cos\left(3\theta\right) = q_0/\sqrt{-p_0^3}$.

As of today, a simple algorithm has been proposed by [Kaw16] to determine the interface parameter α. This approach is much more suitable for high-performance computing in single instruction multiple data (SIMD). We, even through, used the original algorithm in this thesis, because the bottleneck of the fluid simulation is not the advection of the free surface but the iterative computing for the Poisson equation in the conventional incompressible Navier-Stokes simulation.

3.9 Geometrical one-dimensional linear-mapping method

In PLIC methods the interface is first reconstructed by a discontinuous piecewise linear line and then advected in a given velocity field up to the next discrete time. This evolution may be approximated with geometrical methods, by computing the reference phase fluxes across the cell boundary.

In the geometrical approach [PAB^{+}97], a deep insight on interface advection can be gained by expressing one-dimensional advection schemes along the x-direction as different mappings of the plane onto itself. The mappings possess the remarkable property that the consistency condition $0 \leq C \leq 1$ is satisfied after advection in each grid cell.

We discuss two different linear mappings. In the first one we advect in a given one-dimensional velocity field each grid cell and then we mark out the deformed cells in the original grid to update the volume fraction data; hence the name *out-of-cell*. By contrast, in the *onto-cell* mapping we compute the area that will be advected into each grid cell.

3.9.1 Operator splitting

In one-dimensional advection we follow the interface as it is advected by a discretised velocity fields $u\left(x\right)$. In two dimensions, one-dimensional advections in the two directions are performed in sequence. To reduce possible asymmetries induced by the splitting we can consider first a motion along the x-direction and then along y on odd time steps and vice versa on even time steps. For three-dimensional Cartesian grids we need three one-dimensional sweeps and a more complex sequence of sweep direction orders.

3.9.2 The out-of-cell explicit linear mapping

Consider the case in which we perform the advection along the x-axis first. In the linear mapping method the reconstructed interface at time step n is mapped on the interface at time t^{n+1} by a linearised velocity field. Each end point of the interface segments is moved with and *interpolated velocity*. Consider the motion $x\left(t\right)$ of a particle in the velocity field $u\left(x\right)$. The equation of motion is $\mathrm{d}x/\mathrm{d}t = u\left(x\right)$. An explicit first-order scheme for its integration is:

$$x\left(t^{n+1}\right) = x\left(t^{n}\right) + u\left[x\left(t^{n}\right)\right]\left(t^{n+1} - t^{n}\right) \tag{3.74}$$

Before proceeding further, it is useful to rescale space and time in grid spacing and time step units. Using dimensions of h and $\Delta t = t^{n+1} - t^{n}$, the new non-dimensional physical variables are $x' = x/h$, $t' = t/\Delta t$, and $u' = u\Delta t/h$. Notice that the x-component of the velocity vector is now a CFL number. The grid cells are mapped to unit squares and the origin of the local coordinate system is in the lower-left corner. From now on, we drop the primes when using the cell non-dimensional variables; then:

$$x\left(t^{n+1}\right) = x\left(t^{n}\right) + u\left[x\left(t^{n}\right)\right] \tag{3.75}$$

To complete the integration of the equation of motion we need an approximation of u. We label with $\sum$ the unit area of the cell (i,j) and consider a linear interpolation of the u velocity between the two values $u_{i-1/2,j}$, on the left edge of the MAC cell, and $u_{i+1/2,j}$, on its right edge:

$$u\left(x\right) = u_{i-1/2,j}\left(1-x\right) + u_{i+1/2,j}x \tag{3.76}$$

combining the above equations we see that the updated position of the point $\boldsymbol{x'} = \boldsymbol{x}\left(t^{n+1}\right)$ is a function of the old coordinates:

$$\begin{cases} x' = bx + u_{i-1/2,j} \\ y' = y \end{cases} \tag{3.77}$$

where $b = 1 + u_{i+1/2,j} - u_{i-1/2,j}$. This equation describes a linear mapping $\boldsymbol{x'} = T_x^E\left(\boldsymbol{x}\right)$. The mapping of the whole computational domain is piecewise linear, each piece transforming a square cell $\sum$ of the grid onto a rectangular Γ_x:

$$\Gamma_x = T_x^E\left(\sum\right) \tag{3.78}$$

In Figure 3.10 we show the case where the velocity field compresses the square, but depending on the velocity values the square may also be expanded and shifted to the left or to the right as well. Because the mapping is linear, straight lines are transformed into straight lines and the reconstructed interface is easily advected and then remapped onto the original mesh. In the case depicted in Figure 3.10 the new volume fraction in the central cell $\sum$ is obtained by adding the three area contributions from the central and the two adjacent cells after their transformation by T_x^E:

$$C_{i,j}^{n+1} = D + E + F \tag{3.79}$$

The three-dimensional extension is straightforward, except that D, E and F are now polyhedra obtained by cutting right hexahedra by the transported planar interface elements. Their volume is computed by the technique described in the above section. It is important to stress that the entire plane is partitioned in a tessellation made of image rectangles such as Γ_x (Figure 3.11). In other words, the Γ_x rectangles do not overlap or leave empty space, which ensures that area is properly advected. The method has the important feature that the updated volume fraction C satisfies $0 \leq C \leq 1$. This is because the volume fraction is computed as the sum of the three areas D, E, and F.

It is now possible to construct a two-dimensional advection method with two consecutive out-of-cell explicit linear mappings:

1. reconstruct the interface at time step t^n,
2. perform the advection along x-axis by using the mapping T_x^E to move the two interface end points in each cut cell,
3. from the updated position of the end points, compute the areas with the expressions given in the above section and update an intermediate volume fraction field C^*:

$$C^*_{i,j} = D_x + E_x + F_x \tag{3.80}$$

4. reconstruct the interface from the C^* data,
5. perform the advection along the y-axis by using the T_y^E to compute in a similar way:

$$C^{n+1}_{i,j} = D_y + E_y + F_y \tag{3.81}$$

the extension to three dimensions requires three reconstructions and three one-dimensional advections. This method has a major disadvantage: it does not preserve volume and mass exactly. To achieve mass conservation it must be combined with a different one-dimensional advection scheme.

3.9.3 The onto-cell implicit linear mapping

To construct this new mapping instead of the explicit method, we consider an implicit first-order scheme to integrate the equation of motion:

$$x\left(t^{n+1}\right) = x\left(t^n\right) + u\left[x\left(t^{n+1}\right)\right] \tag{3.82}$$

but notice that the x-position at time t^{n+1} is calculated with the velocity field u at time t^n. By using the above equation we find that the action of the mapping $\boldsymbol{x}' = T_x^I\left(\boldsymbol{x}\right)$ reads in components:

$$\begin{cases} x' = ax + au_{i-1/2,j} \\ y' = y \end{cases} \tag{3.83}$$

where $a = 1/\left(1 - u_{i+1/2,j} + u_{i-1/2,j}\right)$. Consider now the pre-image Γ_x of the central square $\sum$ of Figure 3.12 by T_x^I; then

$$\sum = T_x^I\left(\Gamma_x\right) \tag{3.84}$$

We compute the new volume fraction in the central cell by considering the area of the reference phase in Γ_x sent onto $\sum$:

$$C_{i,j}^{n+1} = D + E + F \tag{3.85}$$

where the three area contributions are defined in Figure 3.12. For clarity, we point out a few differences between the two mappings T_x^E and T_x^I. In particular, the width of the area D is $u_{i-1/2,j}$ in Figure 3.10 and $au_{i-1/2,j}$ in Figure 3.12. Furthermore, the slope of the advected interface changes in a different way, since the coefficient of the linear mapping is b for T_x^E and a for T_x^I. Finally, it is important to stress that, as in the explicit mapping, the whole computational domain is partitioned in a tessellation made of pre-image rectangles such as Γ_x. This ensures that no area is lost or fluxed twice; however, the sequence of two consecutive onto-cell implicit mappings in the x and y-directions does not conserve the area.

3.9.4 Combined linear mapping

A remarkable conservation property is achieved when we combine the two mappings. Indeed, the mapping sequence T_x^I followed by T_y^E conserves area/mass exactly. When the initial rectangle Γ_x is mapped onto $\sum$, the reference phase area is compressed (or expanded) by the factor $a = 1/\left(1 - u_{i+1/2,j} + u_{i-1/2,j}\right)$. Then, when $\sum$ is mapped onto a new rectangle Γ_x, it is expanded (or compressed) by the mapping T_y^E, which reads:

$$\begin{cases} x' = x \\ y' = by + v_{i,j-1/2} \end{cases} \tag{3.86}$$

with $b = 1 + v_{i,j+1/2} - v_{i,j-1/2}$. The combination of the two mappings leads to the following two-dimensional advection scheme:

1. reconstruct the interface at time step t^n,
2. perform the advection along x-axis by using the onto-cell mapping T_x^I and get the provisional scalar field C^*:

$$C_{i,j}^{*} = D_x + E_x + F_x \tag{3.87}$$

3. reconstruct the interface from the C^* data,
4. perform the advection along the y-axis by using the out-of-cell mapping T_y^E and get:

$$C_{i,j}^{n+1} = D_y + E_y + F_y \tag{3.88}$$

The area of $\Gamma_y = T_y^E \left(\sum\right) = T_y^E T_x^I \left(\Gamma_x\right)$ is the area of Γ_x multiplied by:

$$ab = \frac{1 + v_{i,j+1/2} - v_{i,j-1/2}}{1 - u_{i+1/2,j} + u_{i-1/2,j}} \tag{3.89}$$

but for an incompressible flow the divergence-free condition on a staggered MAC grid with square cells reads:

$$u_{i+1/2,j} - u_{i-1/2,j} + v_{i,j+1/2} - v_{i,j-1/2} = 0 \tag{3.90}$$

then $ab = 1$ and $\Gamma_y = \Gamma_x$. Thus, for an incompressible flow, the area is conserved exactly and the consistency condition $0 \leq C \leq 1$ is always satisfied. To minimise asymmetries, the combined mapping $T_y^E T_x^I$ at one time step should be alternated with $T_x^E T_y^I$ at the next time step. Finally, notice that if we consider the mapping $T_y^I T_x^E \; T_y^E T_x^I$, the velocity components of different cells appear in the two coefficients a and b; then the above equation does not apply and the area is not conserved.

An extension to three dimensions has been proposed by [AMSZ03, AMSZ07]. The velocity field $\boldsymbol{u} = (u, v, w)$ may be split in the three fields $\boldsymbol{u}_1 = (u_1, v_1, 0)$, $\boldsymbol{u}_2 = (u_2, 0, w_2)$, $\boldsymbol{u}_3 = (0, v_3, w_3)$, where each field $\boldsymbol{u}_i$ is assumed to be incompressible, $\nabla \cdot \boldsymbol{u}_i = 0$, and $\boldsymbol{u}_1 + \boldsymbol{u}_2 + \boldsymbol{u}_3 = \boldsymbol{u}$. There are six scalar equations for the six unknowns $(u_1, v_1, u_2, w_2, v_3, w_3)$, but only five of them are independent. For example, if $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ satisfy $\nabla \cdot \boldsymbol{u}_1 = \nabla \cdot \boldsymbol{u}_2 = 0$, then $\boldsymbol{u}_3 = \boldsymbol{u} - \boldsymbol{u}_1 - \boldsymbol{u}_2$, but in this case $\nabla \cdot \boldsymbol{u}_3 = \nabla \cdot (\boldsymbol{u} - \boldsymbol{u}_1 - \boldsymbol{u}_2) = 0$ is automatically satisfied. Therefore, it is possible to exploit this degree of freedom to specify one of the six unknowns and, starting from the boundary where the velocity field is given, to compute on a staggered MAC grid the velocity field components which are still unknown by solving the divergence-free equations.

3.10 Related one-dimensional advection methods

The terminology we have employed in the previous section is not the most widespread one. Instead, for historical reasons one often refers to Eulerian and Lagrangian methods. The two linear mappings have been naturally derived from an explicit or implicit Lagrangian point of view. In Lagrangian methods, particles or markers are transported in space by the velocity field. The velocity is defined at the grid nodes and needs to by interpolated to the points where markers are located. If a fixed grid is used, then a Lagrangian method requires the interface after its advection to be projected back to the underlying grid to calculate local field values. On the other hand, in Eulerian methods the field values are updated at the local grid points or cells. For the volume fraction field this is performed by exchanging area fluxes between adjacent cells. The difference between the two methods is illustrated in Figure 3.13.

3.10.1 One-dimensional advection equation for the volume fraction

We consider again the advection equation, for the marker function for an incompressible flow:

$$\frac{\partial H}{\partial t} + \nabla \cdot (\boldsymbol{u}H) = H \nabla \cdot \boldsymbol{u} = 0 \tag{3.91}$$

For operator split schemes we consider the one-dimensional version of the above equation, integrate it over the square cell (i, j), and consider for the term $\partial u / \partial x$ and average value across the cell:

$$h^2 \frac{\partial C_{i,j}(t)}{\partial t} + \int_{\Gamma} \boldsymbol{u} \cdot \boldsymbol{n} H(\boldsymbol{x}, t)\, \mathrm{d}l = h^2 C_{i,j}(t) \frac{\partial u}{\partial x} \tag{3.92}$$

where the term on the right-hand side represents a one-dimensional compression or expansion that may differ from zero even if the multidimensional flow is incompressible, as pointed out by [RK98]. We integrate this equation in time, approximate the velocity derivative with a centred finite-difference scheme, and use again non-dimensional variables:

$$C_{i,j}^{n+1} = C_{i,j}^{n} - \bar{\Phi}_{i+1/2,j} + \bar{\Phi}_{i-1/2,j} + \bar{C}_{i,j}\left(u_{i+1/2,j} - u_{i-1/2,j}\right) \tag{3.93}$$

there are different choices for $\bar{C}_{i,j}$, but once we have selected one we have to compute properly the fluxes $\bar{\Phi}_{i+1/2,j}$ and $\bar{\Phi}_{i-1/2,j}$. The geometrical method will help us to calculate correctly the fluxes.

3.10.2 The Lagrangian explicit scheme

We have used the scheme for advection calculation in this thesis. We set $\bar{C}_{i,j} = C_{i,j}^{n}$ in the above equation, then the scheme is explicit and we get:

$$C_{i,j}^{n+1} = bC_{i,j}^{n} - \bar{\Phi}_{i+1/2,j} + \bar{\Phi}_{i-1/2,j} \tag{3.94}$$

where $b = 1 + u_{i+1/2,j} - u_{i-1/2,j}$ is the contraction/expansion coefficient of the out-of-cell linear mapping. With reference to Figure 3.10, where the cell area $\sum$ is mapped onto Γ_x, the term $bC_{i,j}^{n}$ is the value of the area E. Therefore, the two fluxes $\bar{\Phi}$ of the above equation must be equal to the two areas D and F of Figure 3.10. This Lagrangian scheme for interface advection was first introduced by Li. If the two fluxes are not calculated after the interface line has been advected by the mapping., minor inconsistencies may arise for example, [RK98] and [PAB$^+$97]. They are usually seen as small unphysical overshoots, $C > 1$, or undershoots, $C < 0$, in the volume fraction, but also as little holes generated in the bulk of the reference phase or as some level of small debris outside it, which are usually called wisps. These minor inconsistencies should not be confused with the so-called floatsam and jetsam, which are bigger in nature and mainly due to low-order reconstruction methods, such as SLIC, where the interface discontinuity at the cell boundary is $O(h)$. The easiest way to get around the problem is simply to ignore it. This is conveniently done when the over/undershoots are rather small by letting:

$$C_{i,j} = \min\left(1, \max\left(C_{i,j}, 0\right)\right) \tag{3.95}$$

we have used this technique in this thesis.

3.11 The volume-of-fluid in the lattice Boltzmann method

3.11.1 Free surface boundary condition of the lattice Boltzmann method

When we apply the single-phase free surface model into the lattice Boltzmann method, another problem be occurred in the interface cells. As described above, the distribution functions of empty gas cells are never accessed. Interface cells, however, always have empty cell neighbours (Figure 3.15). Therefore, during the streaming step only distribution functions from fluid cells or other interface cells are streamed normally, while the distribution functions that would be read from empty cells need to be reconstructed with corresponding free surface boundary conditions at the interface cell [KTH$^+$05, TR09]. This boundary condition does not require additional constructs, such as ghost layers around the interface. Thus, they can be treated locally for each cell. An atmospheric pressure of $\rho_A = 1.0$ is used, as this is also the reference density and pressure of the fluid. Moreover, it is assumed that the viscosity of the fluid is significantly lower than that of the gas phase, while having a higher density. Hence, the gas follows the fluid motion at the interface. In terms of distribution functions, this means that if at $(\boldsymbol{x} + \Delta t \boldsymbol{e}_i)$ there is an empty cell:

$$f_I'\left(\boldsymbol{x}, t + \Delta t\right) = f_i^{eq}\left(\rho_A, \boldsymbol{u}\right) + f_I^{eq}\left(\rho_A, \boldsymbol{u}\right) - f_i\left(\boldsymbol{x}, t\right) \tag{3.96}$$

where I is the index parameter that satisfies $\boldsymbol{e}_I = -\boldsymbol{e}_i$; $\boldsymbol{u}$ is the velocity of the interface cell at position $(\boldsymbol{x})$ and time t. The pressure of the atmosphere onto the fluid interface is introduced by using ρ_A for the density of the equilibrium distribution functions. Applying Equation 3.96 to all directions with empty neighbour cells would result in a full set of distribution functions for interface cells. This boundary condition, thus, can be used for a multiphase flow [ABR$^+$14] taking proper density instead of density for atmosphere. However, to balance the forces on each side of the interface, the distribution functions coming from the direction of the interface normal $\boldsymbol{n}$ are also reconstructed. Thus, as shown in Figure 3.14, if the distribution function f_i would be streamed from an empty gas cell, or if $\boldsymbol{n} \cdot \boldsymbol{e}_I > 0$ holds, f_i is reconstructed using Equation 3.96. Hence, the interface normal $\boldsymbol{n}$ is approximated with the Parker and Youngs' method [PP04] in the PLIC-VOF approach by the fluid fraction value C in each spatial direction.

3.11.2 Introduction of a control volume

A fluid fraction variable C is introduced to describe the fill level of a control volume V_{CV}, i.e., the volume fraction being filled with fluid, similar to the conventional volume-of-fluid (VOF) method as described above as:

$$C = \frac{V_{\mathrm{fluid}}}{V_{\mathrm{CV}}} \tag{3.97}$$

for a unit cell ($\Delta x_i = 1.0$) filled with a fluid of density ρ_f and mass m_f, we can state:

$$C = \frac{m_f/\rho_f}{V_{\mathrm{CV}}} = \frac{m_f}{\rho_f \Delta x_1 \Delta x_2 \Delta x_3} = \frac{m_f}{\rho_j} \tag{3.98}$$

Thus, a fill level of $C = 0.0$ indicates a completely empty cell, while a fill level of $C = 1.0$ indicates a completely filled cell (Figure 3.15).

We will develop not a two-phase multiphase flow model but a single-phase free surface model based on the volume-of-fluid approach in this thesis, because the dynamics of gas-phase can be ignored in large-scale three-dimensional tsunami simulations. It should be noted that gas cells has any physical variables in this model.

3.11.3 Choice of the control volume

The layout of the control volume V_{CV} has not been specified. In the conventional free-surface volume-of-fluid approaches, the control volumes are directly assigned to the lattice Boltzmann grids. In contrast to the approaches, we use a staggered grid layout to apply to PLIC-VOF method as described above sections into the lattice Boltzmann method. In a staggered grid layout, every calculation cell (i.e., control volume) is spanned by eight the lattice Boltzmann grids.

The new layout leads to additional requirements for the consistency between cell and node states. A cell is considered fluid if and only if all its vertices are fluid nodes. A cell considered gas if and only if all its vertices are gas nodes. All other cells are interface cells. With this layout, the exchanges of the fluid fluxes will take place between six neighbouring cells in three-dimensional problems (cf. Equation 3.99).

$$\hat{\boldsymbol{n}}_i \, (i = 0, \ldots, 5) = \begin{Bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{Bmatrix} \tag{3.99}$$

3.11.4 Discretisation of the advection equation

The fluid fluxes exchange between neighbouring cells can no longer be determined in terms of the lattice Boltzmann distribution functions f_i, but must be evaluated from macroscopic velocity and density information on the participating the lattice Boltzmann nodes. For the weak-compressible lattice Boltzmann method, the sum of fill levels is not conserved perfectly, so a recourse to the continuity equation and the principle of conservation of mass is used to derive the advection algorithm:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = \frac{\partial \rho}{\partial t} + \nabla \cdot (\boldsymbol{u}\rho) = 0 \tag{3.100}$$

We discrete the above equation with a conventional finite volume method by integrating the equation over the control volume and applying the divergence theorem to obtain a surface integral for the convective term:

$$\int_{\Omega} \frac{\partial \rho}{\partial t} \mathrm{d}\Omega + \int_{\Omega} \nabla \cdot (\boldsymbol{u}\rho) = \frac{\partial}{\partial t} \int_{\Omega} \rho \mathrm{d}\Omega + \int_{\Gamma} (\boldsymbol{u}\rho) \cdot \hat{\boldsymbol{n}} \mathrm{d}\Gamma \tag{3.101}$$

where $\hat{\boldsymbol{n}}$ is the unit out-normal vector on the corresponding face of the control volume. Discretising in time with an explicit Euler finite difference scheme leads to:

$$m^{t+1} = m^t - \sum_{i=0}^{5} \Phi_i \tag{3.102}$$

where Φ_i denotes the flux through the i-th face of the control volume. The new cell fill level of an interface cell is calculated by:

$$C^* = \frac{C^t \rho^t - \Phi_0 - \Phi_1}{\rho^{t+1}}, \quad C^{**} = \frac{C^* \rho^t - \Phi_2 - \Phi_3}{\rho^{t+1}}, \quad \text{and} \quad C^{t+1} = \frac{C^{**} \rho^t - \Phi_4 - \Phi_5}{\rho^{t+1}} \tag{3.103}$$

similar to the split method. Where C^* and C^{**} are the intermediate values for fill level.

3.11.5 Averaging of cell fill levels

The fluxes Φ_i are determined locally for every interface cell, and the mass flux can immediately be balanced. One important criterion for the quality of a numerical model for transport problems is that it preserves the conservation of mass. Below we present two different surface reconstruction schemes: a straightforward scheme without detailed surface reconstruction and a classical PLIC-VOF scheme. Both schemes lead to consistent flux terms.

In the simplest averaged cell fill level approach [JK11], the face fill level is determined without geometrical surface reconstruction by averaging the fill level of the two cells between which the mass exchange takes place. This approach is mass-conserving for the transfer between two interface cells but not for the transfer between interface and fluid cells. For this latter case, an effective face fill level below one is obtained for the interface cell, yet for an adjacent (completely filled) fluid cell, the whole edge is moistened. To fix this problem, a case distinction in introduced, which sets the face fill level to one if either of the two participating cells is a fluid cell:

$$C_i = \begin{cases} 1.0 & C\left(\boldsymbol{x}+\boldsymbol{e}_i\right)=1.0 \\ 0.0 & C\left(\boldsymbol{x}+\boldsymbol{e}_i\right)=0.0 \\ \dfrac{C\left(\boldsymbol{x}+\boldsymbol{e}_i\right)-C\left(\boldsymbol{x}\right)}{2} & \text{Otherwise} \end{cases} \tag{3.104}$$

subsequently, the resulting value for C_i is used to calculate the fluid flux Φ_i. This simple scheme leads to good results with minimum effort. No detailed surface reconstruction needs to be done, and even the normal vector information is not necessary. This approach is similar to the one in Thürey approach, but instead of using all 18 lattice directions for the mass exchange, only the six orthogonal directions are used. This may be the reason why, although the global shape and advection behaviour is represented fairly accurately, oscillations can still occur on the interface cells.

3.12 Implementation the PLIC-VOF approach into the lattice Boltzmann method

3.12.1 Calculation of cell face velocities

The treatment and calculation of free surface velocities has to be done with great care. In standard VOF methods, the advection velocities are defined on each face centroids of the control volume. In most cases, the same staggered grid layout is also used for the solution of the Navier-Stokes equations, so that inter or extrapolation is not needed. By contrast, in the present free surface method, the fluid velocities are known on the eight lattice nodes in the centre of each cell. To determine the cell face velocity $\boldsymbol{u}_i$, we use the 1-st order inter- or extrapolation method from the cell centred velocity in this thesis:

$$\boldsymbol{u}_i\left(\boldsymbol{x}\right)=\boldsymbol{u}\left(\boldsymbol{x}\right)+0.5\left(\boldsymbol{u}\left(\boldsymbol{x}\right)-\boldsymbol{u}\left(\boldsymbol{x}-\hat{\boldsymbol{n}}_i\right)\right) \tag{3.105}$$

accurate velocity extrapolation is crucial to ensuring accurate free surface results, especially in regions with high gradients and high interface curvature as, e.g., in breaking wave jets. A higher-order extrapolation might further improve the results.

3.12.2 Algorithm and implementation

After the lattice Boltzmann collision and streaming steps, the position of the interface is updated on the basis of the new, valid velocity fields (Algorithm 1).

The advection algorithm loops over whole domain. The first part (Algorithm 2) evaluates the fluid fluxes for fluid and interface phase as described above sections. If an interface cell changes its

Algorithm 1 Calculation loop of three-dimensional free surface flows

```
collision with the BGK-LBM or MRT-LBM
forcing gravity effect
streaming
apply the interface and solid wall boundary conditions
–
evaluate fluid fluxes and determine cell status changes (Algorithm 2)
apply cell changes and update status (Algorithm 3)
```

Algorithm 2 Evaluate fluid fluxes and determine cell changes

```
if cell type == interface then
    determine cell face velocity $\boldsymbol{u}_i$
    determine interface normal vector $\boldsymbol{n}$
    reconstruct surface (i.e., determine plane parameter $\alpha$)
end if
calculate fluid flux and evaluate new fill level $C^{t+1}$
```

state (i.e., becomes filled or void), it is added to a list of new fluid or new gas cells.

In the second part of the interface update (Algorithm 3), these changes are stored in the state matrix, and consistency between the nodal state and cell fill level is ensured. In the loop over the new cells, the eight nodes of the cell are checked and, in case they change their state, are stored in a list of new gas or fluid nodes.

Afterwards, the new fluid nodes have to be initialised because they do not have any physical variables. The macroscopic values of density and velocity from neighbouring old fluid and interface cells are interpolated by an ensemble average. Based on this information, the distribution functions f_i at new interface cells are initialised with the equilibrium distribution functions.

3.13 Verification and validation

3.13.1 Analysis of a standing waves in rectangular tank

Standing waves in rectangular tank were simulated to verify our three-dimensional model. The purpose of the simulation is to investigate the weak-compressibility of the lattice Boltzmann method by its theory. In this benchmark problem, Wu and Taylor [WT94] have calculated an analytical solution by non-linear water wave theory in second order accuracy. Figure 3.16 shows calculation domain and initial settings. The simulation was carried out for 5 seconds. The maximum magnitude

Algorithm 3 Apply cell changes and update nodal states

```
set new cell status
initialise new fluid nodes
check and adjust interface cell status
```

of the velocity was calculated by the wave speed $\sqrt{gH}$, where g is the gravity acceleration and H is the still water depth. The spacing resolution is fixed with $(x, y, z) = (400, 100, 300)$.

Figure 3.17 shows the calculated interface shapes by the MRT-LBM in three-dimensional view. It looks like that our approach is free srom the non-physical oscillations of the interface cells unlike the conventional free surface model for the lattice Boltzmann method [TR09]. In the simplest free surface model by [TR09], free surface shapes becomes discontinuity when the large deformation occurred because the approach does not consider the shape of the interface. Our proposed approach can easily consider the interface shapes by the PLIC-VOF approach. This difference is critical in term of coastal engineering because the breaking wave may often occur in tsunamis.

Figure 3.18 compares the spacing density profile at $t = 0.5$s between the MRT-LBM and BGK-LBM. The effect of the weak-compressibility in both the collision models appear in the Mach numbers square $O\left(\mathrm{Ma}^2\right)$ based on the theory of lattice Boltzmann method. The BGK-LBM, however, caused artificial oscillations. In the lattice Boltzmann method, the gradient of fluid pressure ∇P in Navier-Stokes equation is directly determined by the gradient of density $\nabla \rho$. It is, thus, important to calculate the spacing density profile in high accuracy. MRT-LBM has successfully become the defect of BGK-LBM well. It can calculate the density profile naturally.

Figure 3.19 shows the Timeseries of water level at the centre of the tank. The key feature of 2nd order solution is that the 1st and 3rd crests are higher that 2nd one. This is caused by the non-linear effects of waves. The phenomena, hence, can not seen in first order solution because that is based on linear theory. Our model has successfully reproduced non-linear effects well. On the other hand, our model leave room for improvement to enhance a energy conservation because the water level was damped after $t = 3.0$s.

3.13.2 Verification for the weak-compressibility in dam-breaking flows

We simulated the classical dam-breaking flow [KO96] to verify the weak-compressibility of the lattice Boltzmann method. Figure 3.20 shows the initial settings of the flow domain.

The purpose of this benchmark test was also to investigate whether our proposed model can control the weak-compressibility by setting the time step Δt in realistic tsunami modellings, if the maximum velocity magnitude is known. We estimated the maximum flow velocity as follows [SCB98]:

$$u_{\max} = 2\sqrt{gH} \tag{3.106}$$

where H is the initial water depth.

Table 3.1 shows the calculation parameters. We determined the spacing grid size $\Delta x = L/100$ and fixed it in this verification, where $L = 0.146$m in this experiment [KO96]. He and Luo [HL97] reported that the incompressible limit is $\mathrm{Ma} < 0.15$, although $\mathrm{Ma} < 0.3$ in general. We assessed additional smaller Mach numbers $\mathrm{Ma} = 0.1$ and $\mathrm{Ma} = 0.01$, considering their report as a criterion.

Figure 3.21 compares the density profiles at $t = 0.3$s with the MRT-LBM. The weak-compressibility decreases at smaller Mach numbers, and the density decreases in $O\left(\mathrm{Ma}^2\right)$. Our model controls the weak-compressibility and approximates the incompressible flows well, setting the time step interval Δt by the Mach number with the required accuracy. The calculation costs, however, increase as the Mach number decreases. We, thus, should determine Δt in view of the trade-off between the calculation cost and the required accuracy.

Figure 3.22 compares the density profiles of the BGK-LBM and MRT-LBM as a preliminary investigation. Both MRT-LBM and BGK-LBM exhibited the same behaviour in terms of free surface movements. The density profile of the BGK-LBM, however, became non-physical especially at solid wall boundaries. We must calculate the fluid density with high accuracy because the pressure gradient is simulated by the density gradient. The calculated pressure, therefore, was disturbed in the BGK-LBM. On the other hand, the MRT-LBM could accurately simulate natural density profiles in the flow field.

3.13.3 Validation for the spacing resolution

We also simulated some dam-breaking benchmarks to validate the accuracy depending on the its spacing resolution. Figure 3.23 shows the initial condition of the dam-breaking flow [MM52]. In this benchmark, we validated the convergence behaviour of the model accuracy depending on the competing grid size, which is an important factor in determining the applicability of the model to other problems from an engineering perspective. The model accuracy was evaluated based on the dimensionless position of the surge front in the time evolution.

Table 3.2 shows the calculation parameters. In this benchmark test, the dimensionless time T and distance from the left side of the competing domain Z were defined using the realistic $t\,(\mathrm{s})$, $z\,(\mathrm{m})$, and $a\,(\mathrm{m})$ shown in Figure 3.23 and gravity $g\,\left(\mathrm{m/s^2}\right)$ as follows [MM52]:

$$T = t\sqrt{\frac{2g}{a}} \tag{3.107}$$

$$Z = \frac{z}{a} \tag{3.108}$$

The effect of gate opening was considered to shift the plots [NH71] instead of directly solving, for example, by the immersed boundary method. We applied modified bounce back rule as slip boundary conditions for the edge boundaries in the domain.

Figure 3.24 shows the dimensionless time series position of the surge front. Experimental data from [KO96] were also plotted. The results of our model were in good agreement with the experimental data from the Case2 and 3 simulations. Our model has a convergence of the calculation capability that depends on the resolution through the simulations. We found that our model achieves results that match the experimental data well, and we can resolve at least 80 computing cells per the characteristic length a in the benchmark.

On the other hand, the underestimations in the Case1 simulation are due to the difference in physical properties caused by the coarse grid size. The Reynolds number in the flow field will be smaller in the LBM if we cannot ensure a sufficient spacing resolution. To overcome this problem, it can be considered that another parameterisation approach for the relaxation time τ, determined by not the viscosity ν but the Reynolds number in the flow. This method might fix the Reynolds number as turbulence. However, we believe that a practical solution for the problem is to ensure high resolution in space because we could not determine the Reynolds number in unsteady flows, such a free-surface flow. The solid wall boundary condition also should be treated carefully. Nishi and Doan [ND13] have pointed out that three is room improvement for the modified bounce-back scheme especially in free surface model because the accuracy of the bounce-back scheme in the lattice Boltzmann method depends on the relaxation time τ.

Finally, we simulated another dam-breaking benchmark [AK09] to validate a two-dimensional free-surface model with the lattice Boltzmann method according to [TR09] in a strong current or flow, such as a splash along the vertical wall with bubbles. We calculated this benchmark in three dimension. Figure 3.25 shows the calculation domain, where Points 1 and 2 were the measurement points of the time series water depth.

Table 3.3 shows the calculation parameters. Case2 was considered to have sufficient resolution based on the above discussion. We used $\mathrm{Ma} = 0.01$ to determine the time step interval Δt, assuming incompressible flows.

Figure 3.26 shows the free-surface shapes in a three-dimensional view with Case2. Non-physical oscillation and diffusion near the interfaces did not occur in our results because we reconstructed the shapes via PLIC. Our model succeeded in reproducing non-hydrostatic three-dimensional free-surface flows. On the other hand, in some cells, especially on the wall boundary, the fluid fraction was not fully filled. We believe that this problem is caused by the modified bounce-back condition in the corner cells. We should use a high-order scheme [SFH89, MBG96] for the boundary condition for a more precise simulation. In addition, some bubbles remained in the fluid domain and did not rise. Our model cannot address the bubble dynamics directly because of the single-phase volume-of-fluid model. The remaining bubbles, however, negatively affected the stability of the model and the calculation accuracy of the interface normal $\boldsymbol{n}$. Numerical treatments are required to address the bubbles.

Figure 3.27 compares the free surface shapes between the experimental data and simulated results. Our model well reproduced the realistic dam breaking flows such as the breaking waves or involving the fluid after collision with the solid wall. Figure 3.28 also shows the time series water depth at Points 1 and 2 in Figure 3.25. Our model results approached the experimental data. Accordingly, the resolution increases and the Case2 simulation results were in good agreement with the experimental results. We conclude that our model can simulate realistic tsunami flows analogous to dam breaks by choosing an appropriate spacing resolution and time step interval.

3.13.4 A dam-breaking flow with a single obstacle

From the above sections, the standard dam breaking bottlenecks were successfully simulated by our proposed free surface model. Here, this bottleneck is extended with the addition of a solid obstacle in the centre of the flow domain by [Köl05] (Figure 3.29).

Upon removal of gate, the collapsing of the water column is initiated and the front evolves and finally impacts the solid obstacle. At the moment, the flow pattern changes drastically, similar to what occurs during wave impact on structures. As the water height decreases, highly curved jet evolves, and the water is deflected to the top of the obstacle.

Figure 3.30 compares between the calculated results with the experimental data by [Köl05]. We fixed $\Delta x = 1.25 \times 10^{-3}$ and $\Delta t = 3.12 \times 10^{-6}$ to reduce the weak-compressibility. Very good agreement can be seen, at least for this qualitative comparison. The jet evolves slightly faster in computations tan in the experiment. Hence, we conclude that a proper grid resolution is essential, to capture at least the large-scale sprays and the resulting energy dissipation, and to accurately represent the solid body in conjunction with the above discussions. We also verified that, in the lattice Boltzmann method, a careful treatment of the corner boundaries is crucial to achieve good results. For example modified slip boundary conditions in the lattice Boltzmann method, an accurate normal vector is required in the corner cells, which is discontinuous points. If a smoothed geometry [SS13, SS15] is used, where the corner is rounded out and the corner node uses a linear interpolation of the neighbouring surface normal vectors, the jet shape is affected. For this reason, the normal vector at the obstacle corners is set to the normal vector of each respectively participating face.

3.13.5 Breaking wave in a rectangular tank

We simulated a three-dimensional breaking ocean wave at an air-water interface. Numerical simulations based on fully non-linear potential flow theory [LHC76, GSS89] showed that a periodic sinusoidal wave of large amplitude, with initial velocities specified from linear wave theory, is not stable in a fully non-linear model and rapidly overturns and breaks.

The initial velocity field and interface shape for a two-dimensional linear wave of height H in depth h, specified in a three-dimensional, are given by [LVAC06] with z denoting the vertical direction and $z = 0$ at the undisturbed air-water interface:

$$\eta = \frac{H}{2} \cos(kx) \tag{3.109}$$

$$u_x = \frac{H}{2} \sigma \frac{\cosh k(h+z)}{\sinh(kh^*)} \cos(kx) \tag{3.110}$$

$$u_z = \frac{H}{2}\sigma\frac{\sinh k\left(h+z\right)}{\sinh\left(kh^*\right)}\sin\left(kx\right) \tag{3.111}$$

$$u_y = 0 \tag{3.112}$$

where $\sigma = 2\pi/T$ is the wave angular frequency, $k = 2\pi/L$ is the wave number, and $h^* = h+\epsilon$, where $\epsilon\left(y\right) = 0.08y$ is a small linear perturbation of the seafloor geometry. In this thesis, we set $L = 0.1$m and $T = 0.308$s. As a result, the initial velocity profiles is shown in Figure 3.31.

Figure 3.32 compares the calculated interface shapes between our results and the reference data [LVAC06]. We fixed the spacing resolution as $(x, y, z) = (1024, 128, 256)$ in this simulation. As expected, the wave quickly overturns and develops a plunging jet. The agreement of our results with [LVAC06] results is good, although the shape after breaking wave was a little weakened. This is caused by the averaging extrapolation approach for empty cells, which become interface cells at the next time step. The extrapolation velocities for the empty cell crucial affects a numerical accuracy in the VOF approach. However, using high order schemes easily causes a numerical instability. We, thus, used averaging extrapolation approach in this thesis.

3.13.6 Tsunami inundation in realistic bathymetry

We finally simulated three-dimensional tsunami inundation in realistic urban area. Figure 3.33 shows the 10mm bathymetry data (Scale: 1/250) of Onagawa benchmark test [Pra17, PYMM18] carried out by the Disaster prevention research institute, Kyoto university. The purpose of this benchmark is to bring back the tsunami by the 2011 Great East Japan Earthquake in Onagawa (Miyagi prefecture) with three-dimensional realistic bathymetry. The spacing resolution is $(x, y, z) = (383, 439, 75)$ from two-dimensional bathymetry data. The time step interval was fixed $\Delta t = 2.0 \times 10^{-4}$s to reduce the weak-compressibility of LBM. Figure 3.34 shows the incident wave (input wave) from right-hand boundary in Figure 3.33.

Figure 3.35 shows timeseries free surface in three-dimensional view. Our model qualitatively reproduced the tsunami inundation in Onagawa town. The Onagawa hospital was actually inundated in the event as shown in Figure 3.35 (f). We, thus, conclude that our model can simulate realistic tsunami flows in three dimension. On the other hand, a detailed validation of our model in urban area such as estimating flow velocity and pressure acting an obstacle in still remain as future challenges.

3.14 Conclusion

In this chapter, we presented the implementation the single-phase free surface model based on PLIC approach to the lattice Boltzmann method to enable three-dimensional tsunami modellings more efficient. Moreover, we performed the verification and validation of the BGK-LBM and MRT-LBM in classic dam-break benchmark problems. The following conclusions were drawn from the discussions above:

- We have developed a fully explicit three-dimensional free-surface model by the lattice Boltzmann method. Our model reduces interface oscillations and diffusions using the PLIC approach.
- We found that the MRT-LBM can overcome the defects of the BGK-LBM, such as the disturbance of the fluid density, and can simulate the flow field naturally.
- We determined the appropriate spacing resolution by the characteristic length of the dam-break benchmark tests. Moreover, we demonstrated the application to other problems.
- We substantiated that the weak-compressibility in the MRT-LBM can be controlled within $O\left(\mathrm{Ma}^2\right)$ by choosing a time step interval based on the maximum velocity magnitude in the flow field.

The MRT-LBM might require finer resolution than that of other computational fluid dynamic. As described in the last chapter, the adaptive mesh refinement [ZAK15, OI18] technique will be needed to execute more efficient simulations. In addition to model development, we will investigate more detailed reconstruction tests [RK98, AMSZ07] and the wave force acting on structures [KFV$^+$05] by our model, which allows the weak-compressibility to appear from the viewpoint of coastal engineering.

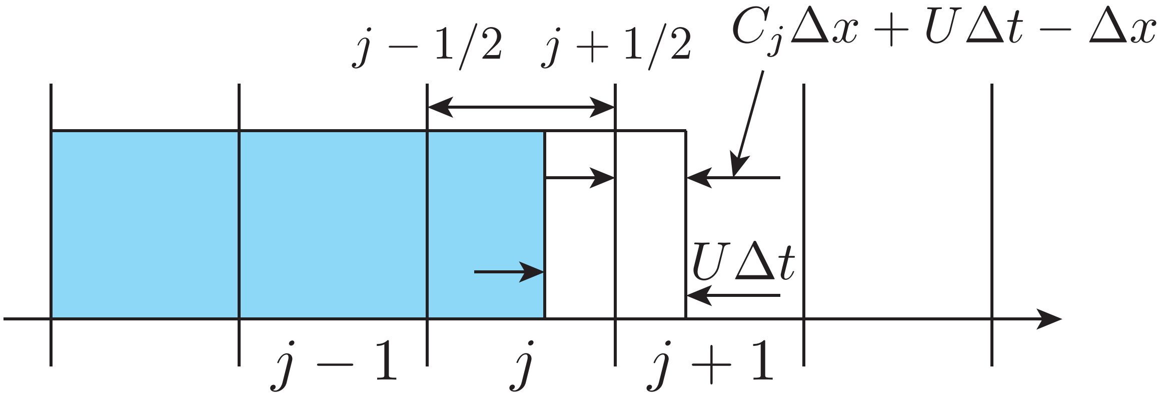


Figure 3.1: One-dimensional advection by the VOF method. Given the value of the colour function in the interface cell j, and the side where the full cell is, the location of the interface can be found and the fluxes computed exactly.

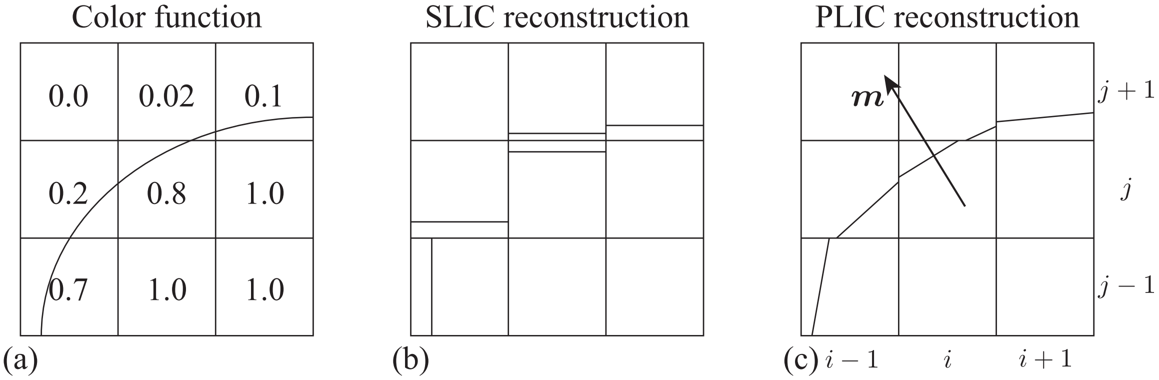


Figure 3.2: The basic principle of the VOF methods: (a) a portion of the interface line and the colour function value in each cell; (b) the SLIC reconstruction, where each segment in parallel to one coordinate axis; (c) a PLIC reconstruction with unconnected segments across each cell; the normal vector $\boldsymbol{m} = -\nabla C$ is pointing outwards from the reference phase.

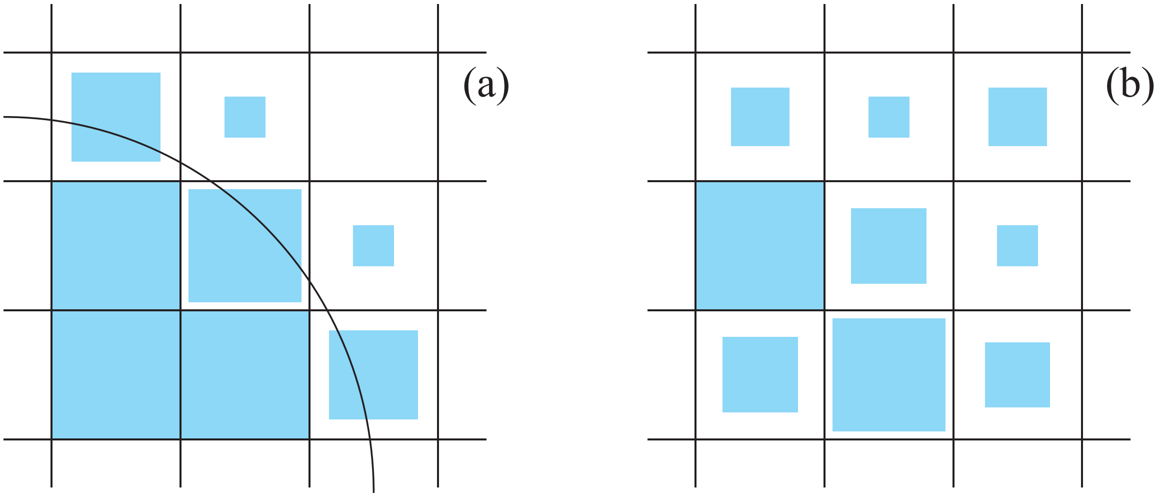


Figure 3.3: Representing the colour function as shaded squares of size proportional to the fractional volumes helps to understand the reconstruction problem. (a) A well-behaved distribution of the colour function; it corresponds to a smooth circular arc. (b) A more confused distribution.

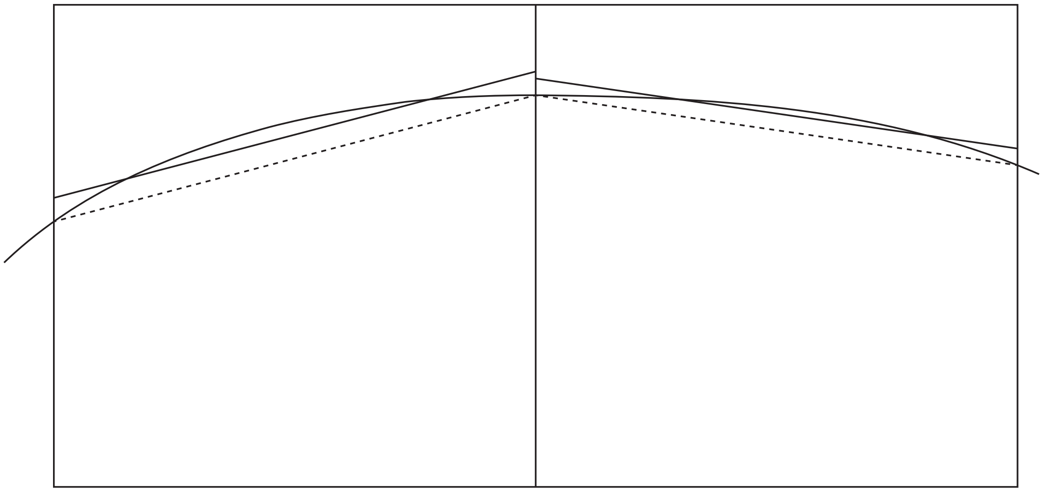


Figure 3.4: A line cutting two grid cells (thick solid line). In the trapezoidal rule the piecewise linear approximation connects consecutive points on the line and it is continuous across the cell boundary (dashed lines). The PLIC-VOF reconstruction is not continuous across the cell boundary but satisfies the area conservation constraint (thin solid lines). The dotted regions represent the area error for a PLIC-VOF reconstruction (left) and the trapezoidal rule (right).

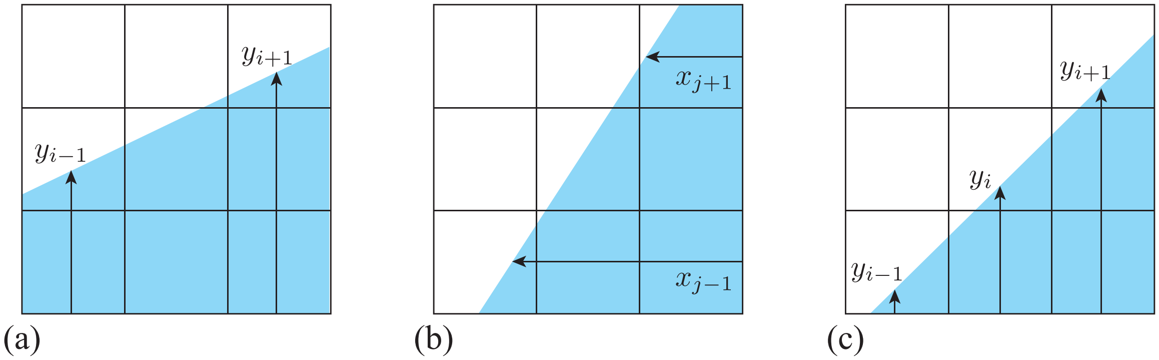


Figure 3.5: In the centred-columns scheme, volume fractions are added column-wise, heights y_{i-1} and y_{i+1}, for case (a) and row-wise, widths x_{j-1} and x_{j+1}, for case (b), to get the correct slope of the linear interface. In case (c), an off-centred scheme, with heights y_i and y_{i+1} should be used.

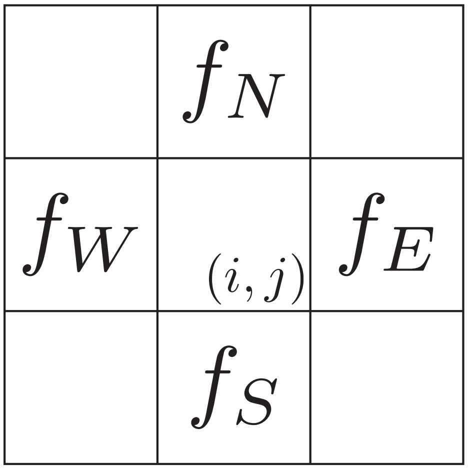
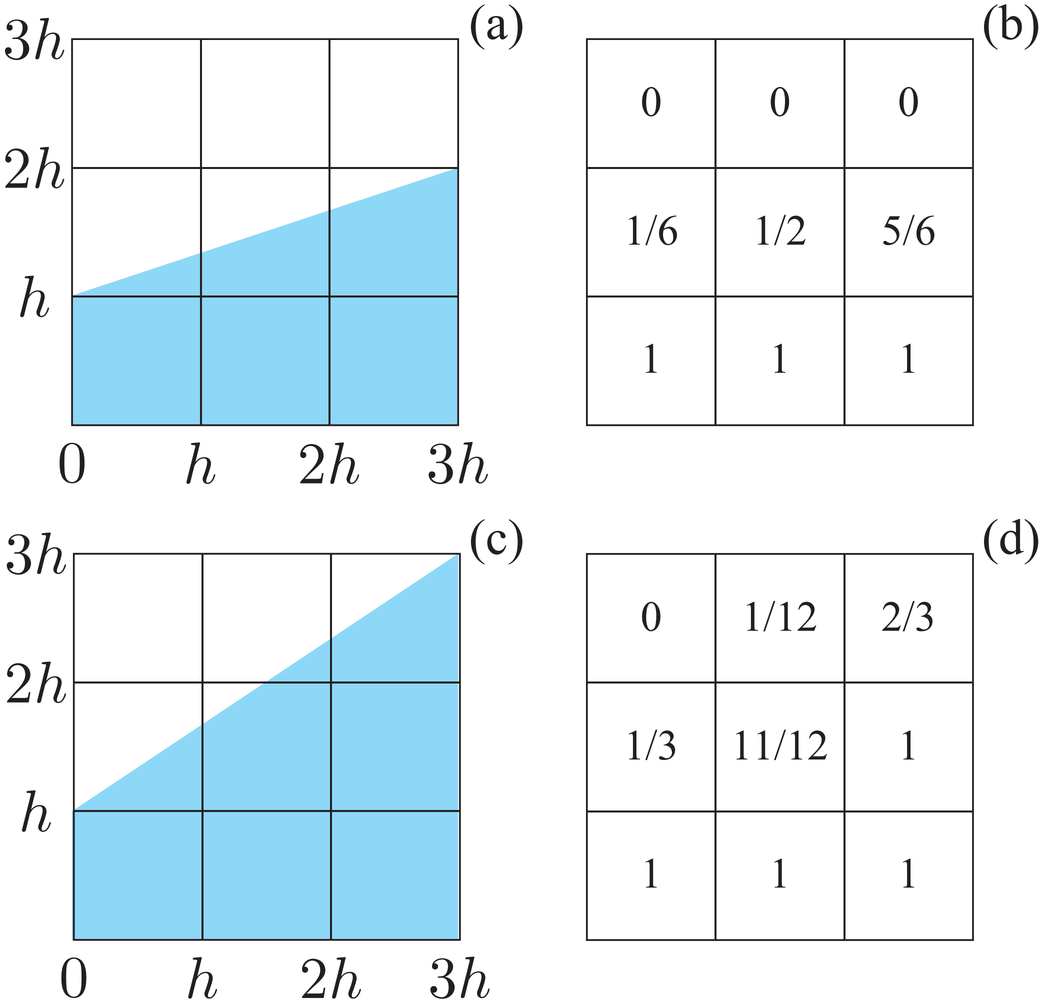


Figure 3.6: The stencil that Parker and Youngs use to determine ∇f in two dimension.



0	0	0
1/6	1/2	5/6
1	1	1

0	1/12	2/3
1/3	11/12	1
1	1	1

Figure 3.7: (a) Parker and Youngs' method will reconstruct this line exactly only if $\alpha = 2$. (b) The volume fractions associated with the line shown in (a). (c) Parker and Youngs' method does not reconstruct this line exactly for $\alpha = 2$. Thus it does not reproduce all linear interface exactly, and so we conclude it is at best first-order method. (d) The volume fractions associated with the line shown in (c).

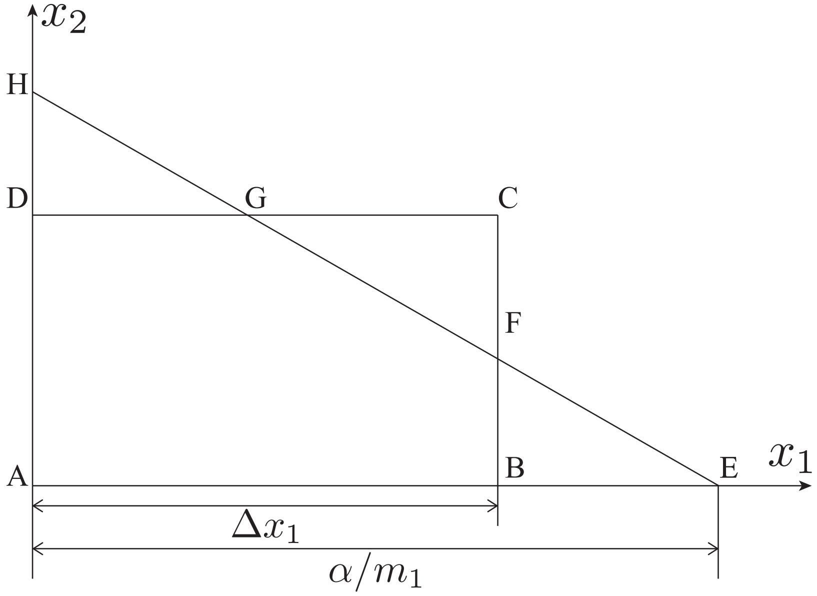


Figure 3.8: The cut volume refers to the region within the rectangle $ABCD$ which also lies below the straight line EH.

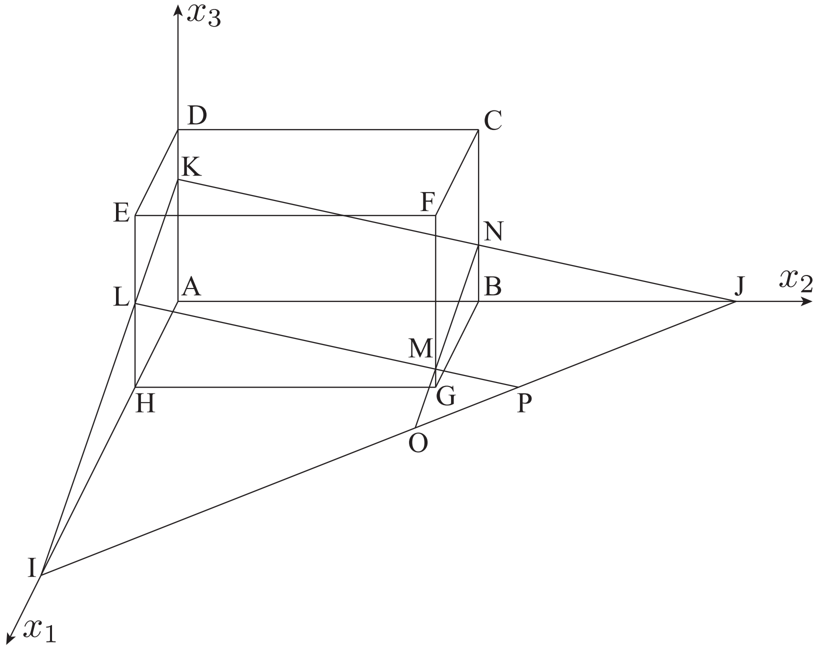


Figure 3.9: The cut volume is the region inside the parallelepiped $ABCEFGH$ and below the plane IJK.

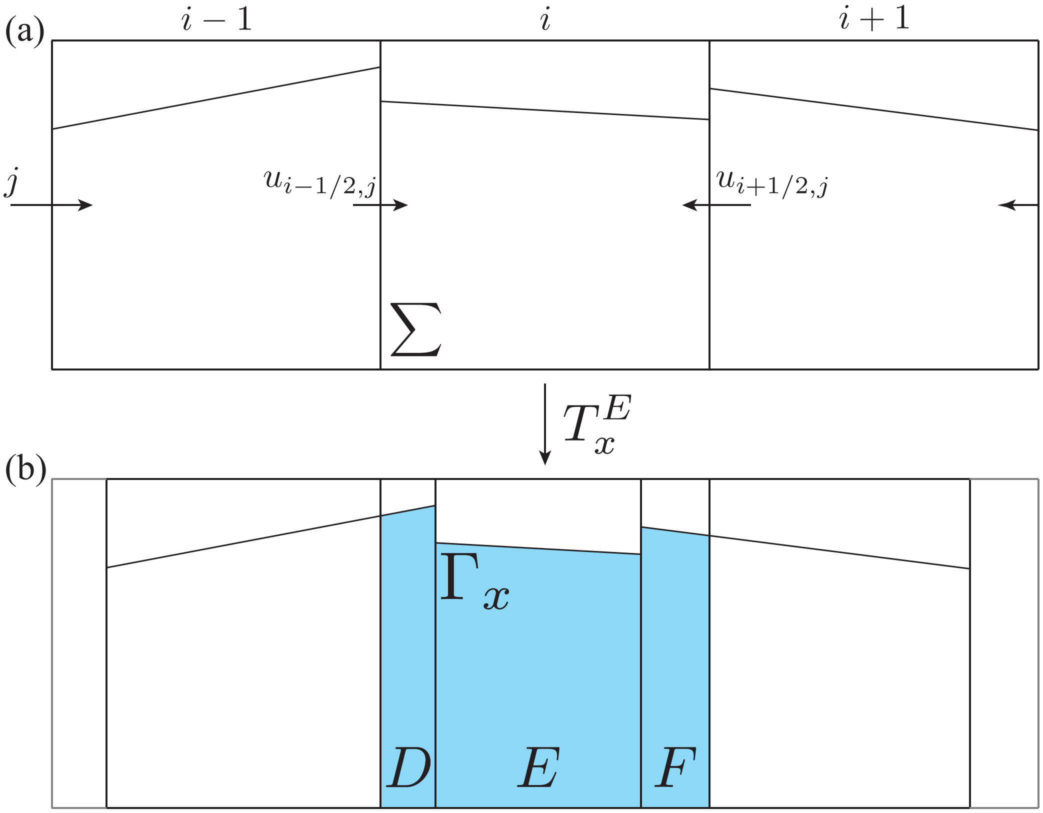


Figure 3.10: (a) The horizontal one-dimensional mapping T_x^E transforms the square cell $\sum$ onto the rectangle Γ_x, by transporting the cell edges with the one-dimensional flows. (b) Three consecutive cells are transformed by the piecewise linear mapping T_x^E onto three rectangles, which are projected back to the original grid to calculate the three contributions D, E and F to the square cell $\sum$.

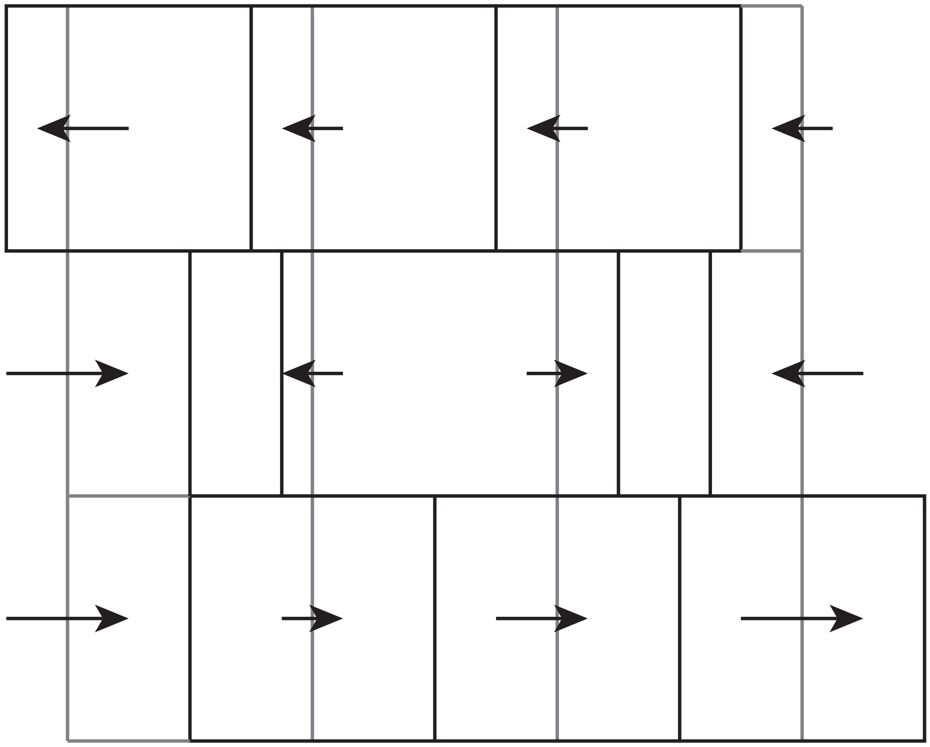


Figure 3.11: The images of the square cells by the mapping T_x^E form a tessellation of the plane by rectangles (thick solid lines). These may be either expanded or compressed and shifted to the left or to the right with respect to the original square cells (thin solid lines), depending on the local value of the velocity field.

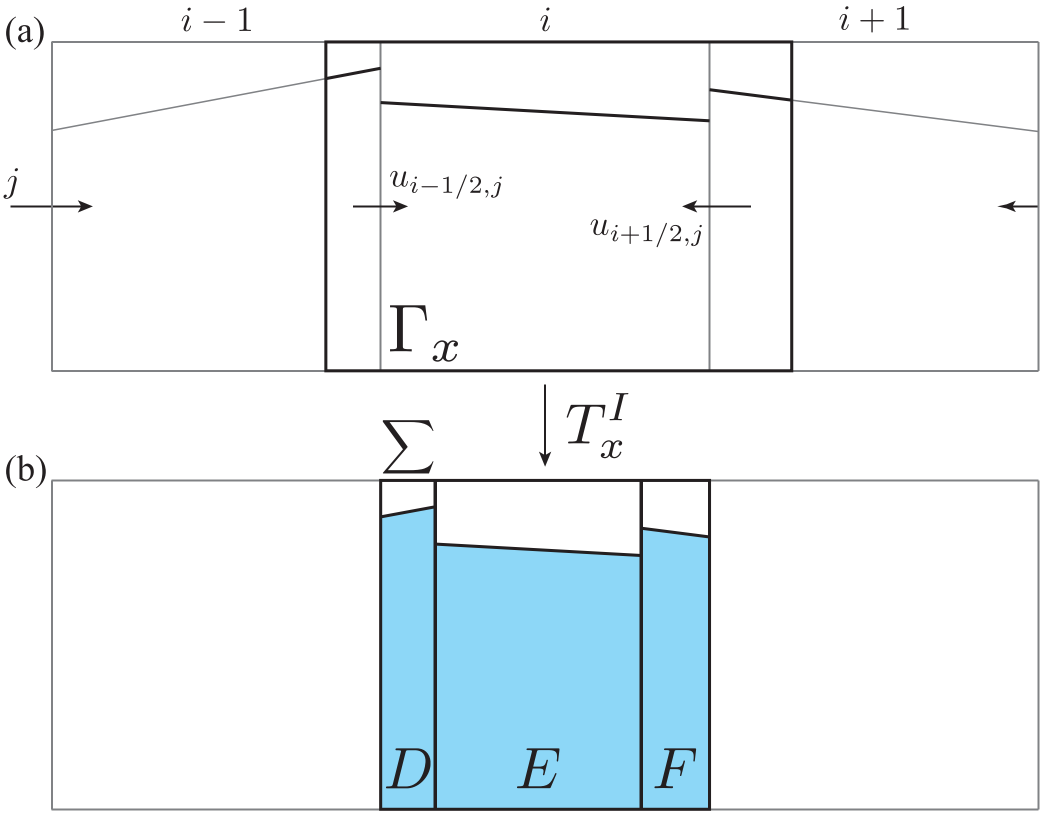


Figure 3.12: (a) The horizontal one-dimensional mapping T_x^I transforms the rectangle Γ_x onto the square cell $\sum$ by using the velocity at the cell sides. The velocity field u is the same as Figure 3.10. (b) The contributions to Γ_x of three consecutive cells are transformed by the linear mapping T_x^I onto D, E and F of the central cell $\sum$.

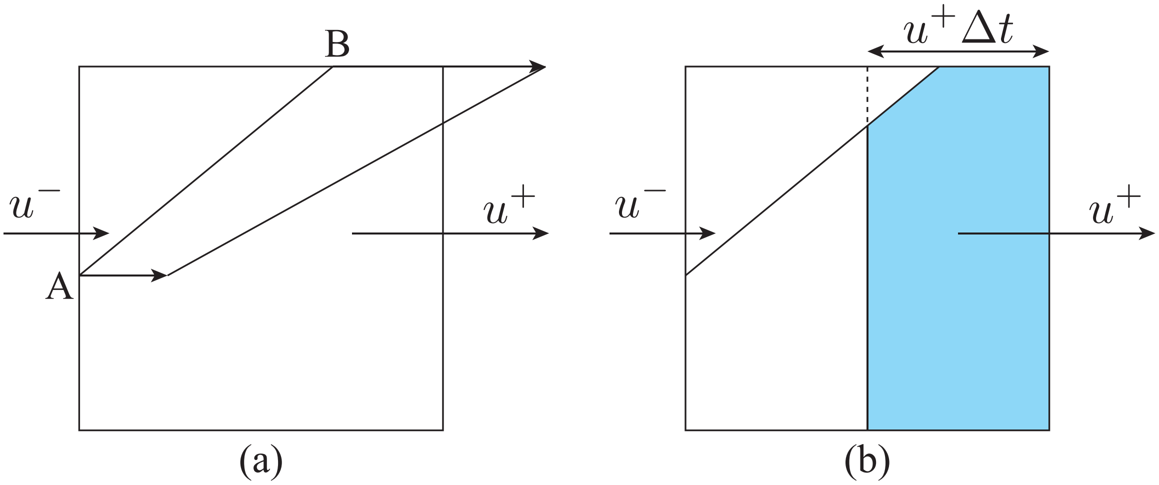


Figure 3.13: (a) In the Lagrangian method, the end points A and B of the interface segment are advected by the flow. The local velocity may be interpolated linearly between the velocities u^- and u^+ at the cell faces. (b) In the Eulerian method, the total area flux through the right side of the cell is the area of the rectangle with width $u^+\Delta t$, while the reference phase flux is the shaded portion of this rectangle.

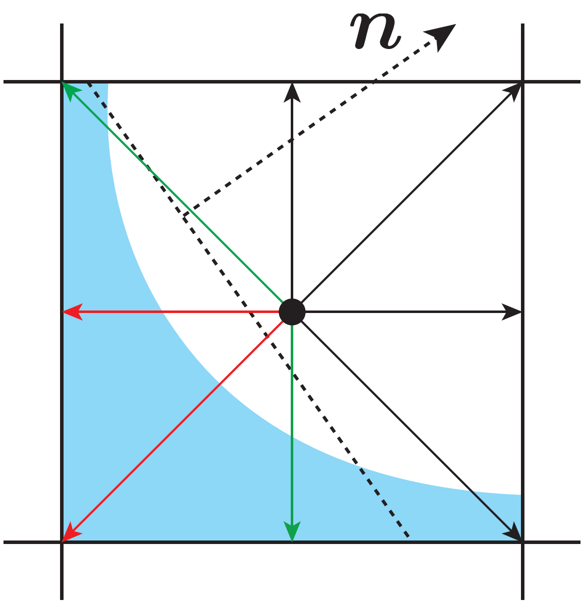
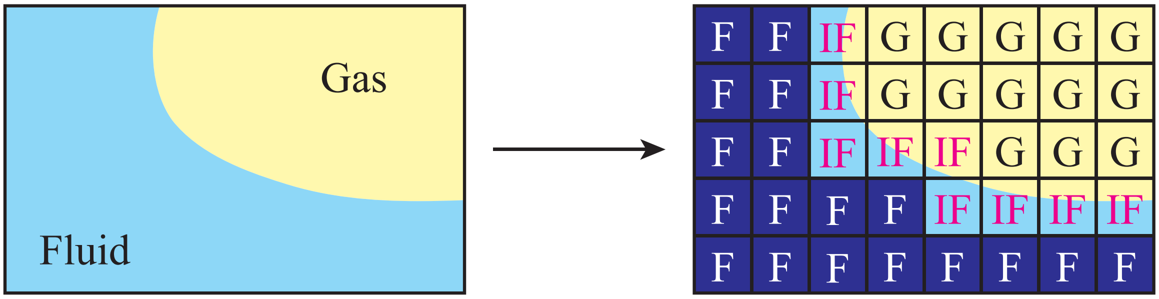
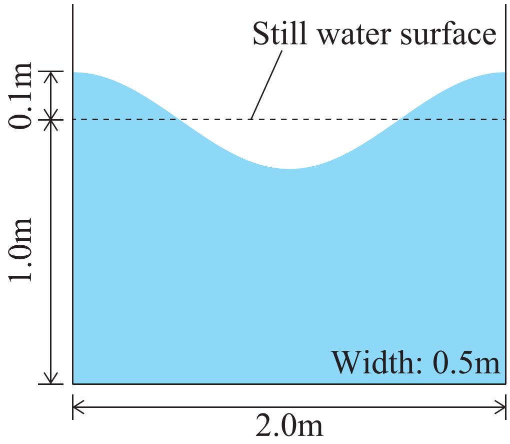


Figure 3.14: The missing distribution functions f_i on the interface cells, (red functions) functions from empty gas cells, (green functions) functions along the interface normal $\boldsymbol{n}$ determined by the VOF algorithms (broken line).



F	F	IF	G	G	G	G	G
F	F	IF	G	G	G	G	G
F	F	IF	IF	IF	G	G	G
F	F	F	F	IF	IF	IF	IF
F	F	F	F	F	F	F	F

Figure 3.15: Here the different cell types required for the free surface algorithm can be seen. (yellow) Gas-phase which means cells without fluid. (light blue) Interface-phase which means cells filled with fluid partly. (deep bule) Fluid-phase which means cells completely filled with Fluid.



Calculation parameters

Case No.	Mach	$\Delta t\,(\mathrm{s})$	stability
Case 1	0.30	2.77×10^{-4}	unstable
Case 2	0.15	1.38×10^{-4}	stable
Case 3	0.10	9.22×10^{-5}	stable
Case 4	0.010	9.22×10^{-6}	stable
Case 5	0.0010	9.22×10^{-7}	unstable

Figure 3.16: The calculation domain and initial settings [WT94] (left), Calculation parameters (right).

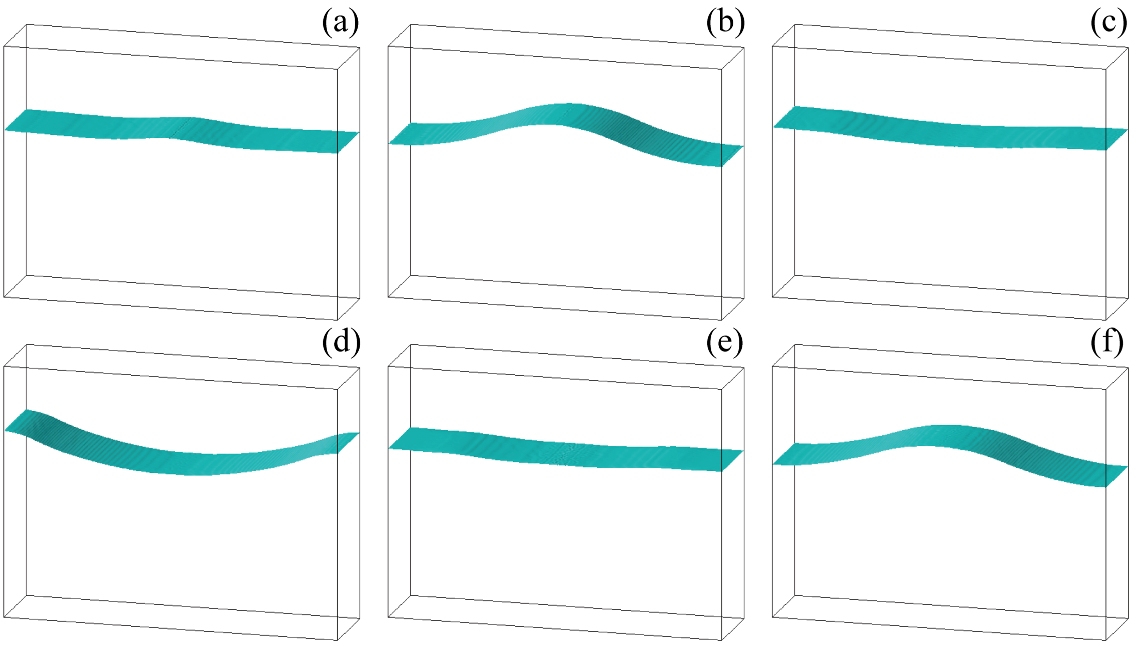


Figure 3.17: The calculated interface shapes with MRT-LBM (Ma = 0.01), (a) $t = 0.3$s, (b) $t = 0.6$s, (c) $t = 0.9$s, (d) $t = 1.1$s, (e) $t = 1.4$s, (f) $t = 1.7$s.

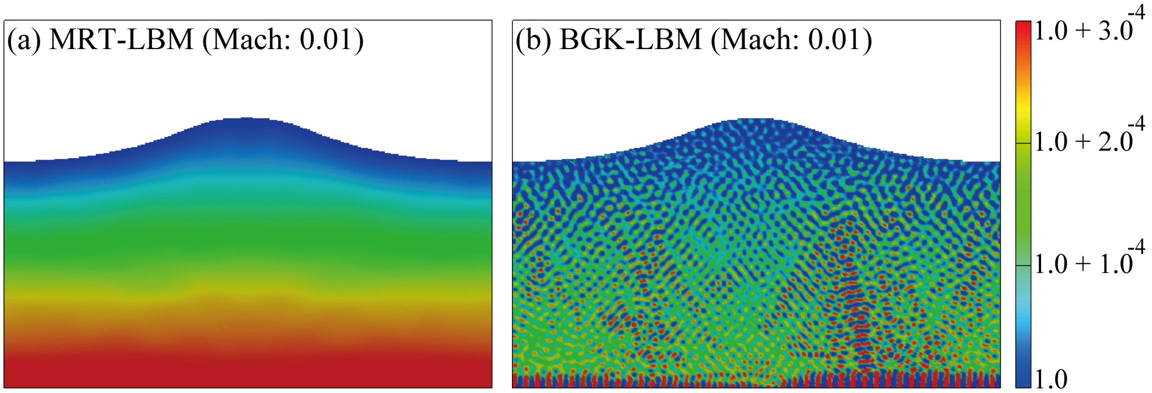


Figure 3.18: The comparison of the spacing density profiles at t = 0.5s, (a) MRT-LBM, (b) BGK-LBM.

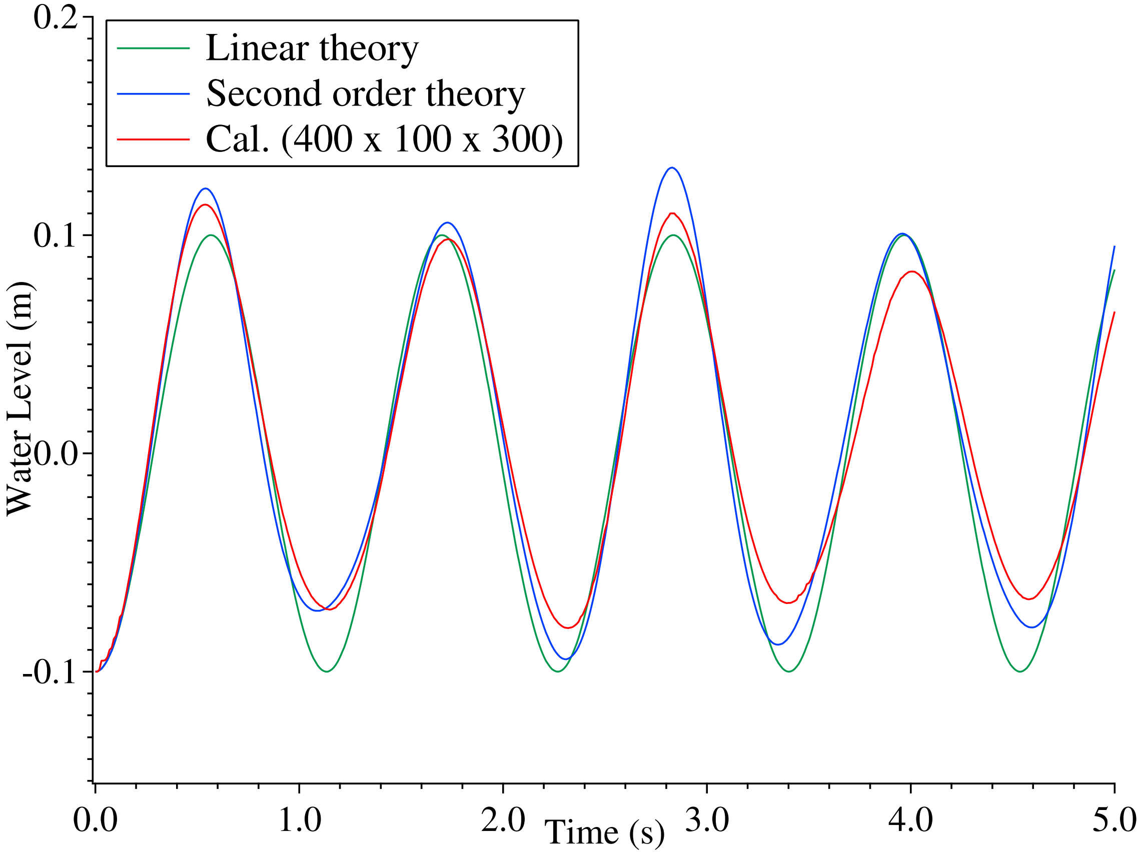


Figure 3.19: Timeseries of the water level at the centre of the numerical tank.

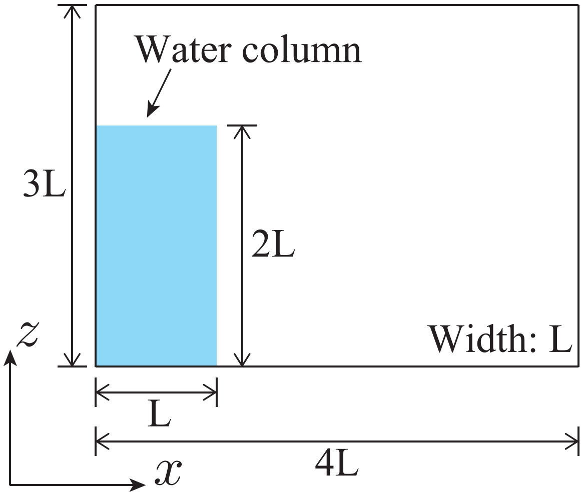


Figure 3.20: The calculation domain of the Koshizuka and Oka [KO96] dam breaking flows.

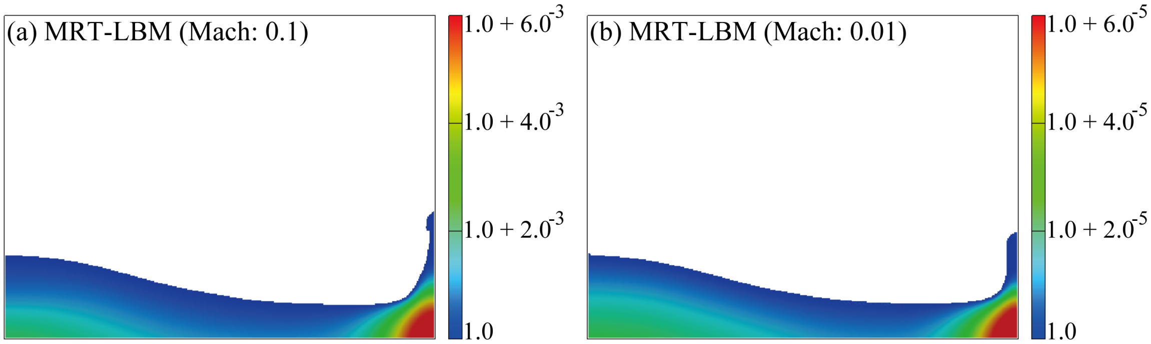


Figure 3.21: Fluid density profiles of the MRT-LBM due to the difference in Mach numbers at $t = 0.3$s.

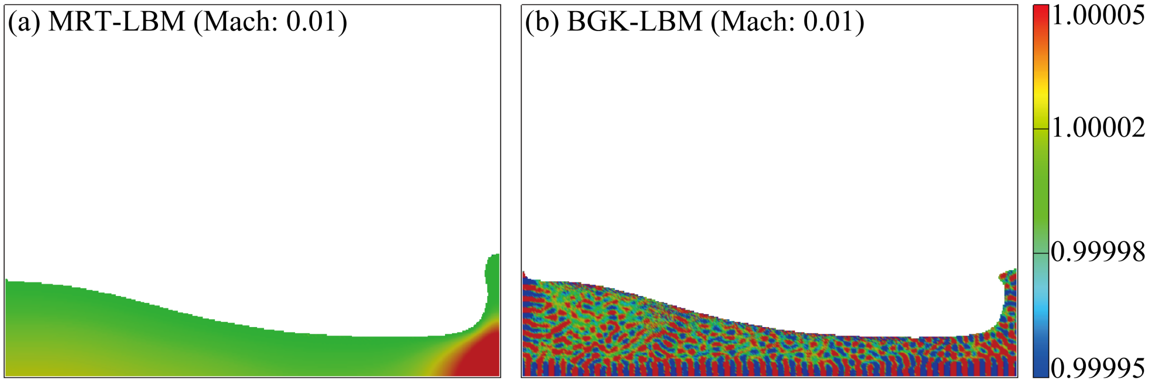


Figure 3.22: Fluid density profile of the MRT-LBM and BGK-LBM at at $t = 0.3$s.

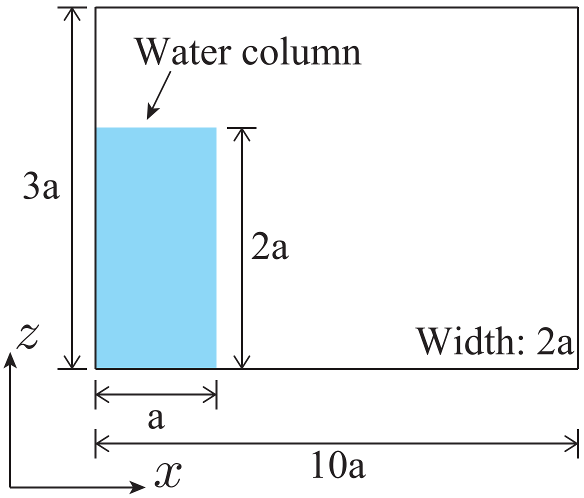


Figure 3.23: The calculation domain of the Martin and Moyce [MM52] dam breaking flows.

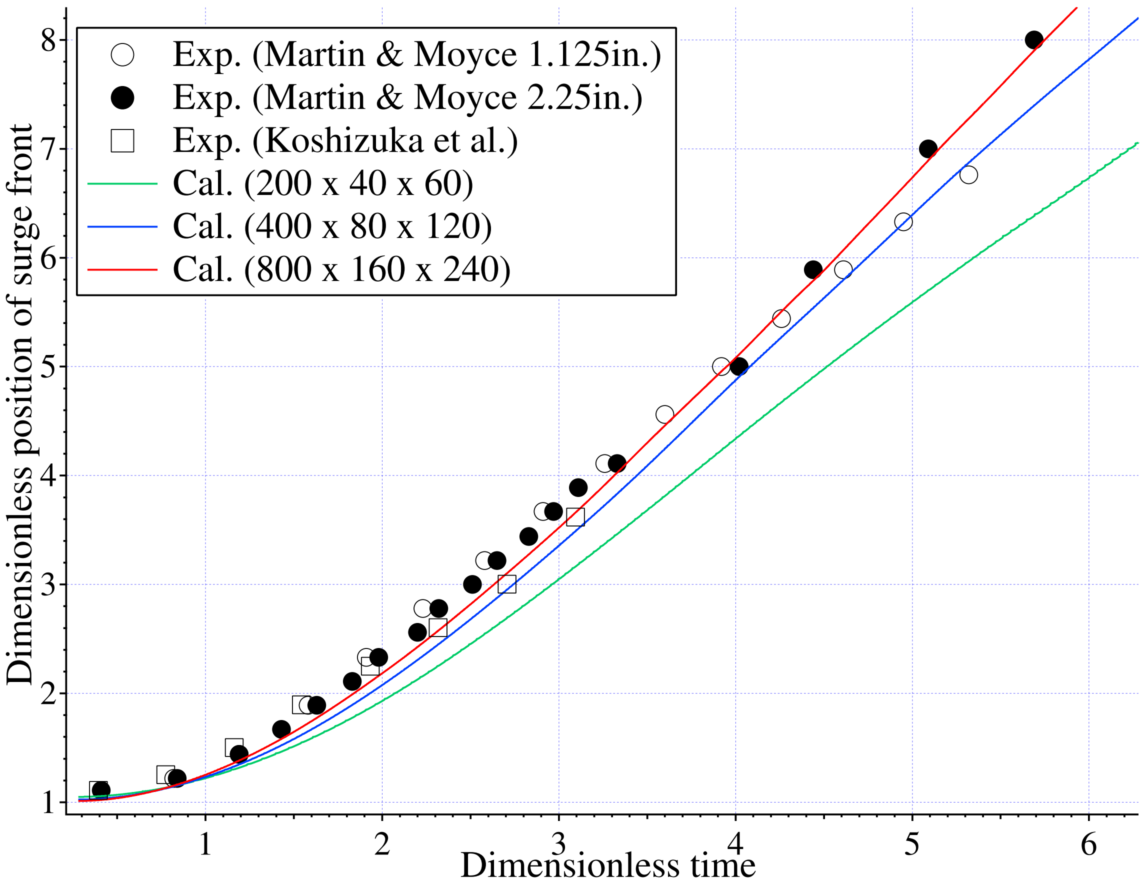


Figure 3.24: Timeseries of the dimensionless position of the surge front.

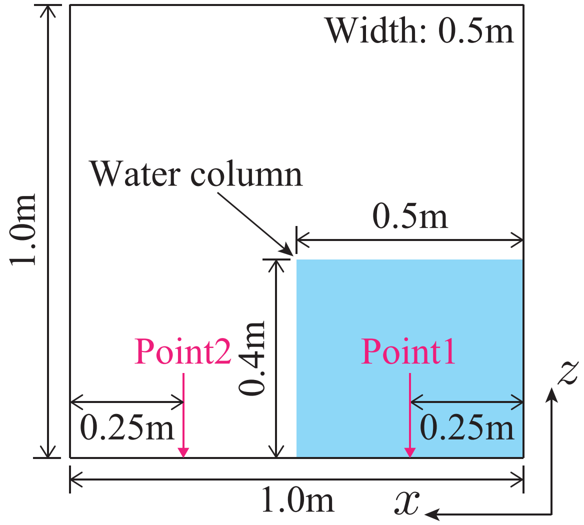


Figure 3.25: The calculation domain of the Araki and Koshimura [AK09] dam breaking flows.

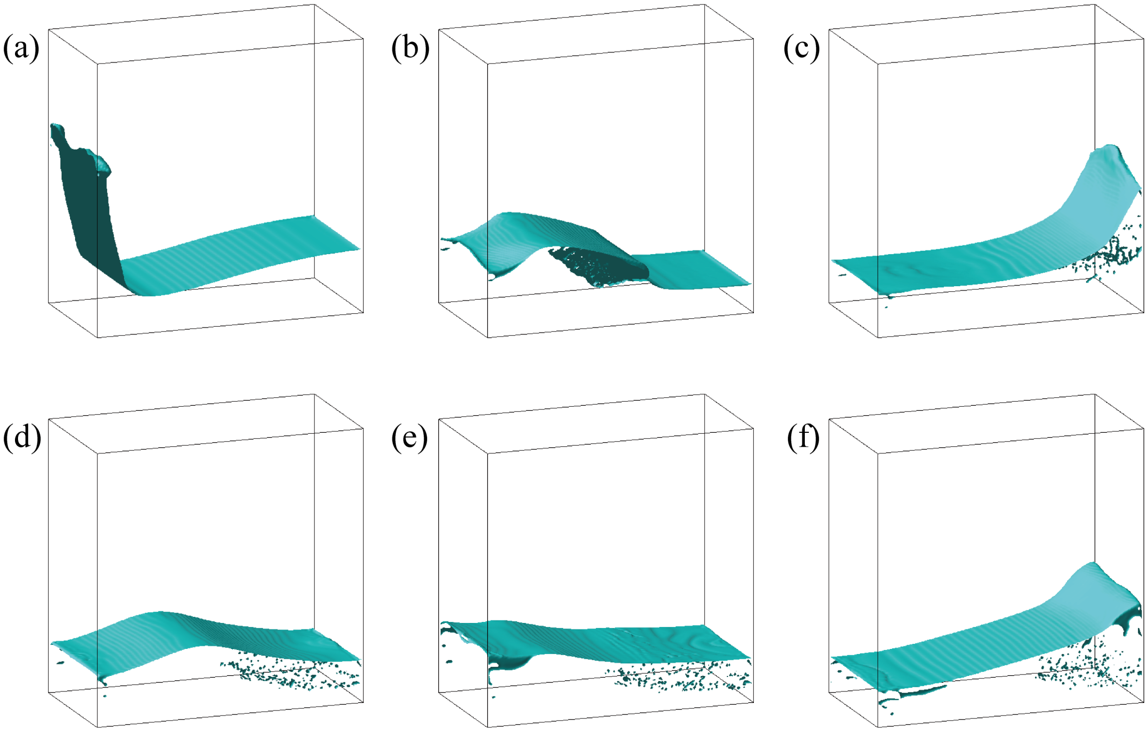


Figure 3.26: Free surface shapes with case2 simulation in three-dimensional view, (a) $t = 0.50$s, (b) $t = 1.0$s, (c) $t = 1.5$s, (d) $t = 2.0$s, (e) $t = 2.5$s, (f) $t = 3.0$s.

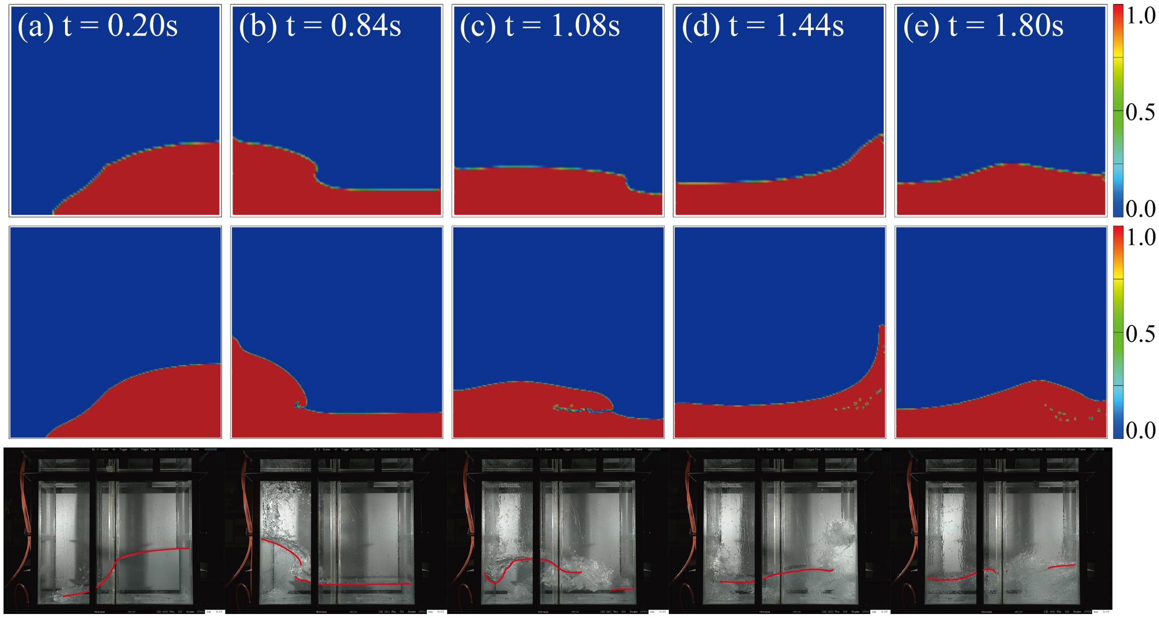


Figure 3.27: Free surface shapes in two-dimensional view, (top) Case1, (middle) Case2, (bottom) Experimental data [AK09].

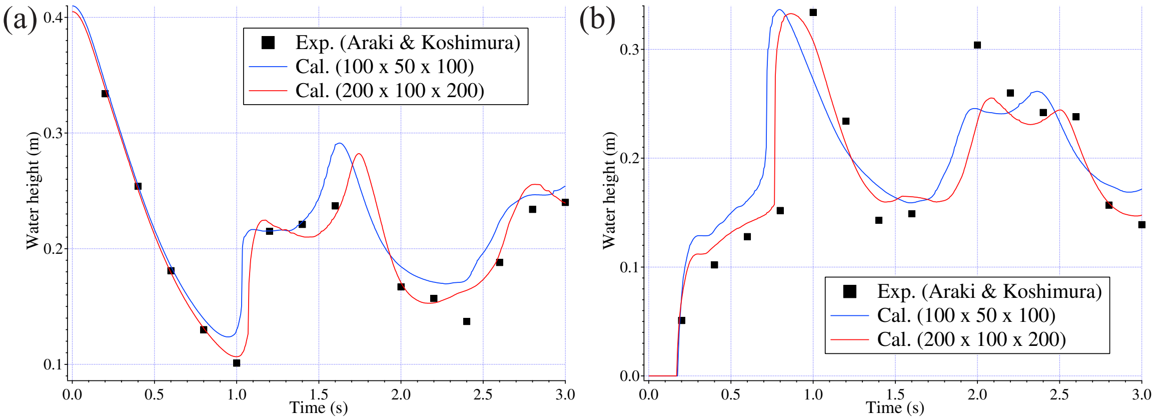


Figure 3.28: Timeseries water depth at the observed points (Figure 3.25), (a) Point1, (b) Point2.

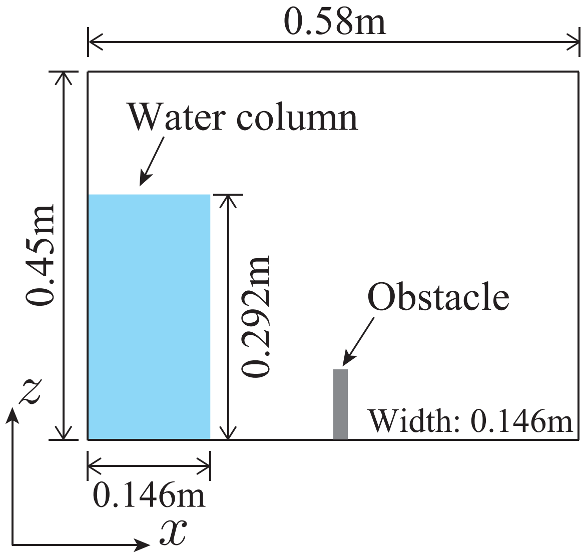


Figure 3.29: The calculation domain of the Kölke [Köl05] dam breaking flows.

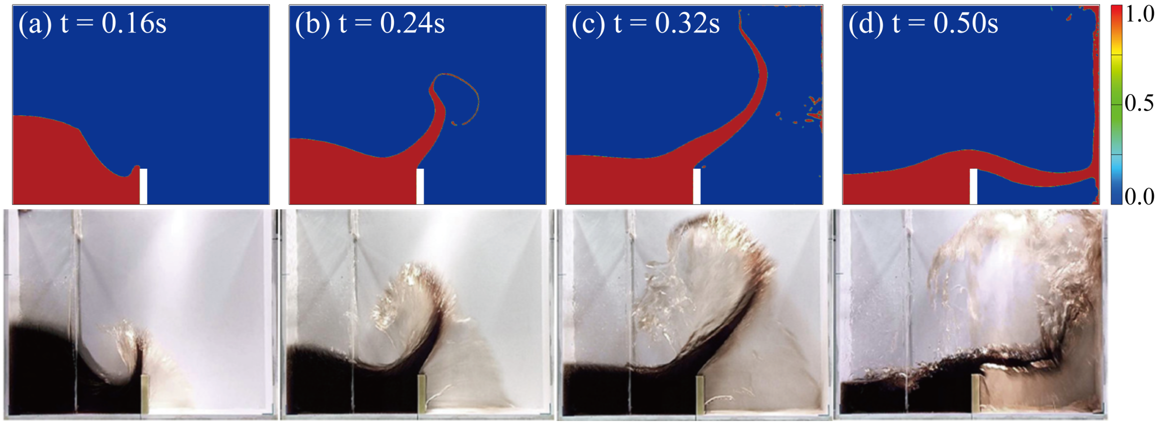


Figure 3.30: The comparison of the interface shapes in two-dimensional view, (top) our results, (bottom) experimental data [Köl05].

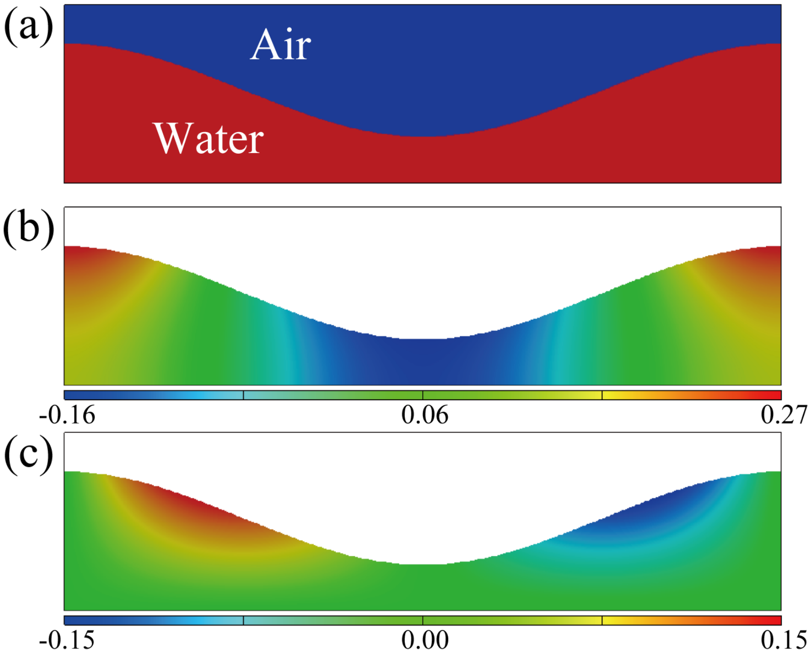


Figure 3.31: Initial velocity profiles for three-dimensional breaking waves benchmark [LVAC06], (a) initial phase profile, (b) velocity along x-axis, (c) velocity along y-axis.

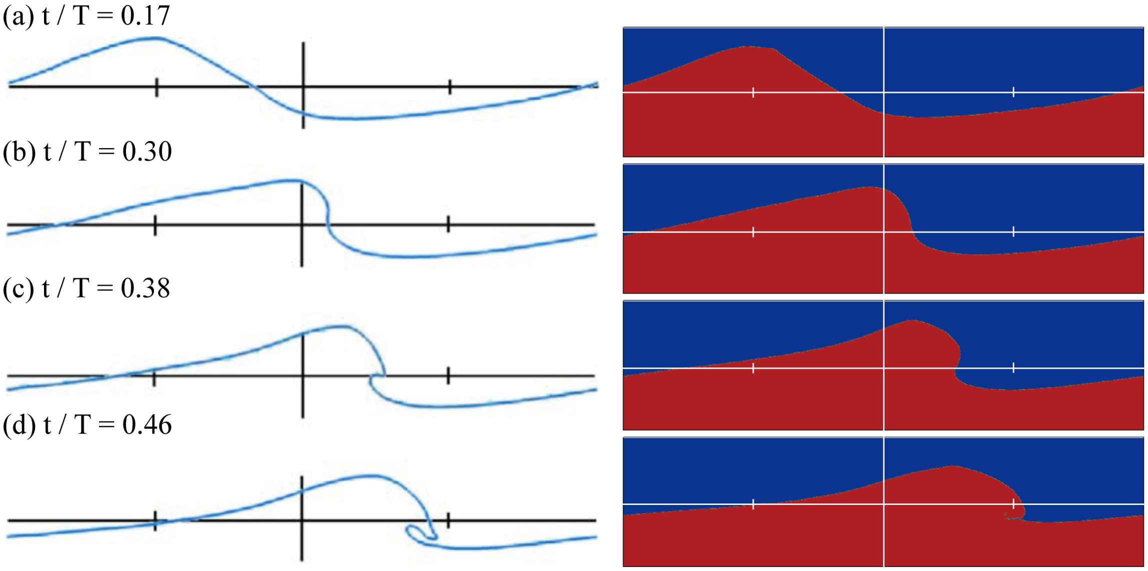


Figure 3.32: Time evolution of the three-dimensional breaking wave. (left) Lubin *et. al.*'s three-dimensional VOF results [LVAC06], (right) our results.

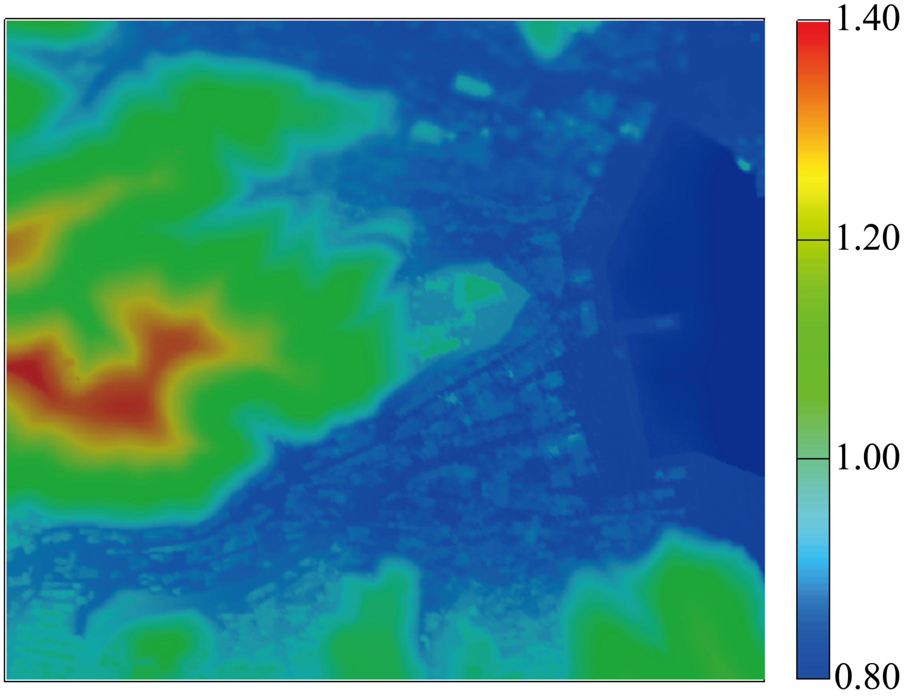


Figure 3.33: The bathymetry data of the three-dimensional Onagawa benchmark problem (from Disaster prevention research institute, Kyoto university).

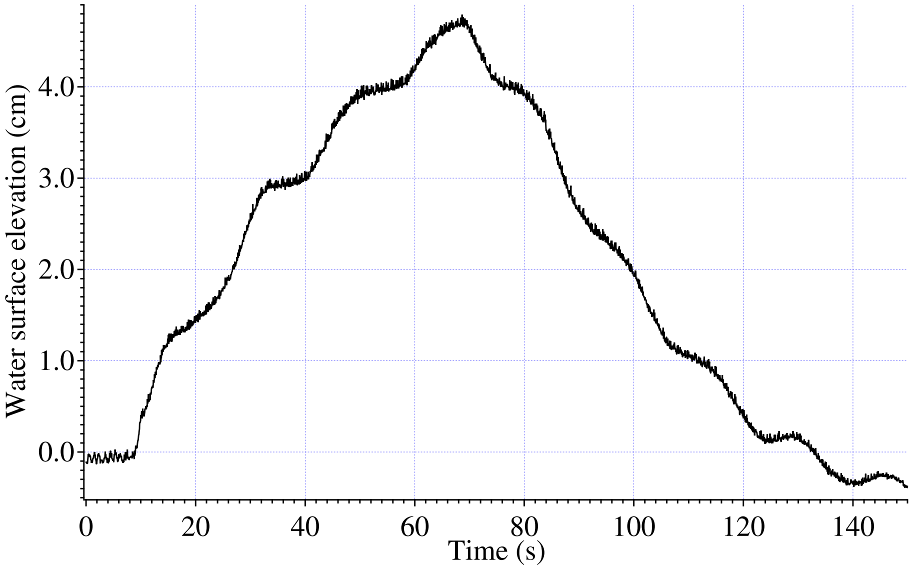


Figure 3.34: The incident wave from right-hand boundary in Figure 3.33.

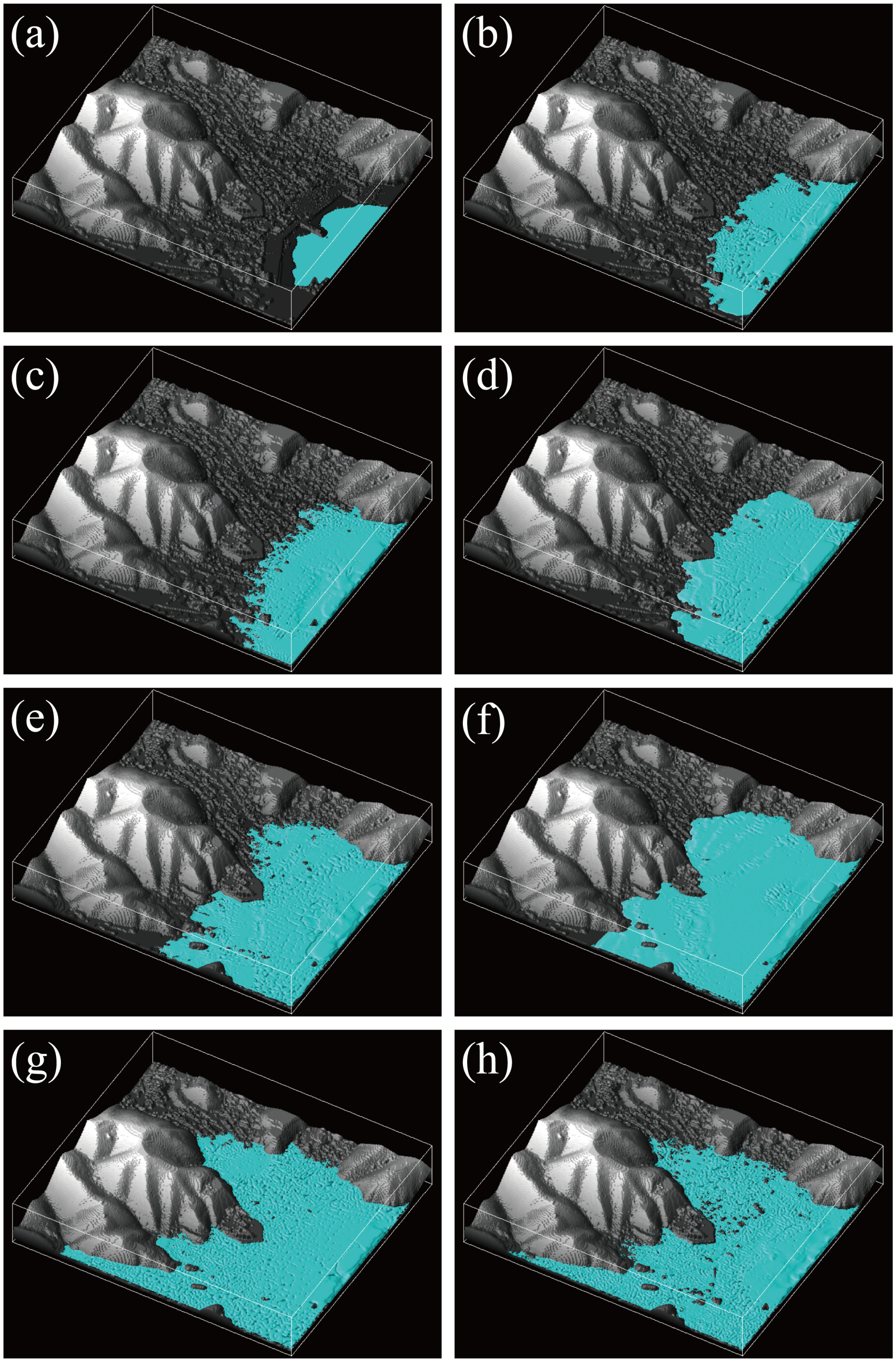


Figure 3.35: Free surface in three-dimensional view, (a) $t = 0.0$s, (b) $t = 30.0$s, (c) $t = 40.0$s, (d) $t = 50.0$s, (e) $t = 60.0$s, (f) $t = 70.0$s. (g) $t = 80.0$s, (h) $t = 90.0$s.

Table 3.1: Calculation parameters of the dam breaking flow [KO96].

Case No.	Ma	Resolution	$\Delta x\,(\mathrm{m})$	$\Delta t\,(\mathrm{s})$	τ
Case1	0.15	$400 \times 100 \times 300$	0.146×10^{-3}	5.28×10^{-5}	$0.5 + 7.5 \times 10^{-3}$
Case2	0.10	$400 \times 100 \times 300$	0.146×10^{-3}	3.52×10^{-5}	$0.5 + 5.0 \times 10^{-3}$
Case3	0.010	$400 \times 100 \times 300$	0.146×10^{-3}	3.52×10^{-6}	$0.5 + 5.0 \times 10^{-4}$

Table 3.2: Calculation parameters of the dam breaking flow [MM52].

Case No.	Resolution	$\Delta x\,(\mathrm{m})$	$\Delta t\,(\mathrm{s})$	τ
Case1	$200 \times 40 \times 60$	2.86×10^{-3}	7.14×10^{-6}	$0.5 + 2.6 \times 10^{-6}$
Case2	$400 \times 80 \times 120$	1.43×10^{-3}	3.57×10^{-6}	$0.5 + 5.3 \times 10^{-6}$
Case3	$800 \times 160 \times 240$	7.14×10^{-4}	1.79×10^{-6}	$0.5 + 1.1 \times 10^{-5}$

Table 3.3: Calculation parameters of dam breaking flow [AK09].

Case No.	Resolution	$\Delta x\,(\mathrm{m})$	$\Delta t\,(\mathrm{s})$	τ
Case1	$100 \times 50 \times 100$	1.0×10^{-2}	1.46×10^{-5}	$0.5 + 4.4 \times 10^{-7}$
Case2	$200 \times 100 \times 200$	5.0×10^{-3}	7.29×10^{-6}	$0.5 + 8.8 \times 10^{-7}$

References

[ABR$^+$14] Anderl, D., Bogner, S., Rauh, C., Rüde, U., and Delgado, A. Free surface lattice Boltzmann with enhanced bubble model. *Comput. Math. with Appl.*, 67(2):331–339, 2014.

[ADR$^+$13] Albadawi, A., Donoghue, D. B., Robinson, A. J., Murray, D. B., and Delauré, Y. M. C. Influence of surface tension implementation in Volume of Fluid and coupled Volume of Fluid with Level Set methods for bubble growth and detachment. *Int. J. Multiph. Flow*, 53:11–28, 2013.

[AK09] Araki, T. and Koshimura, S. Numerical modeling of free surface flow by the lattice Boltzmann method. *J. Jpn. Soc. Civ. Eng.*, B2-56(1):56–60 (in Japanese), 2009.

[AMSZ03] Aulisa, E., Manservisi, S., Scardovelli, R., and Zaleski, S. A geometrical area-preserving Volume-of-Fluid advection method. *J. Comput. Phys.*, 192(1):355–364, 2003.

[AMSZ07] Aulisa, E., Manservisi, S., Scardovelli, R., and Zaleski, S. Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry. *J. Comput. Phys.*, 225(2):2301–2319, 2007.

[AS64] Abramowitz, M. and Stegun, I. A. *Handbook of mathematical functions.* National Bureau of Standards, Washington, DC, 1964.

[DSG17] Dianat, M., Skarysz, M., and Garmory, A. A Coupled Level Set and Volume of Fluid method for automotive exterior water management applications. *Int. J. Multiph. Flow*, 91:19–38, 2017.

[DSS07] Ding, H., Spelt, P. D. M., and Shu, C. Diffuse interface model for incompressible two-phase flows with large density ratios. *J. Comput. Phys.*, 226(2):2078–2095, 2007.

[GSS89] Grilli, S. T., Skourup, J., and Svendsen, I. A. An efficient boundary element method for nonlinear water waves. *Engng. Anal. Bound. Elem.*, 6(2):97–107, 1989.

[HL97] He, X. and Luo, L. S. Lattice boltzmann model for the incompressible Navier-Stokes equation. *J. Stat. Phys.*, 88(3/4):927–944, 1997.

[HN81] Hirt, C. W. and Nichols, B. D. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39(1):201–225, 1981.

[Jac99] Jacqmin, D. Calculation of two-phase Navier-Stokes flows using phase-field modeling. *J. Comput. Phys.*, 155:96–127, 1999.

[JK11] Janßen, C. and Krafczyk, M. Free surface flow simulations on GPGPUs using the LBM. *Comput. Math. with Appl.*, 61(12):3549–3563, 2011.

[Kaw16] Kawano, A. A simple volume-of-fluid reconstruction method for three-dimensional two-phase flows. *Comput. Fluids*, 134-135:130–145, 2016.

[KFV+05] Kleefsman, K. M. T., Fekken, G., Veldman, A. E. P., Iwanowski, B., and Buchner, B. A Volume-of-Fluid based simulation method for wave impact problems. *J. Comput. Phys.*, 206(1):363–393, 2005.

[KK05] Kunugi, T. and Kino, C. DNS of falling film structure and heat transfer via MARS method. *Comput. Struct.*, 83(6-7):455–462, 2005.

[KO96] Koshizuka, S. and Oka, Y. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nucl. Sci. Eng.*, 123(3):421–434, 1996.

[Köl05] Kölke, A. *Modellierung und diskretisierung bewegter diskontinuitäten in randgekoppelten Mehrfeldsystemen.* PhD thesis, Technische Universität Braunschweig, 2005.

[KTH+05] Körner, C., Thies, M., Hofmann, T., Thürey, N., and Rüde, U. Lattice Boltzmann model for free surface flow for modeling foaming. *J. Stat. Phys.*, 121(1-2):179–196, 2005.

[LHC76] Longuet-Higgins, M. S. and Cokelet, E. D. The deformation of steep surface waves on water: a numerical method of computation. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 350, pages 1–26. The Royal Society, 1976.

[LVAC06] Lubin, P., Vincent, S., Abadie, S., and Caltagirone, J. P. Three-dimensional Large Eddy Simulation of air entrainment under plunging breaking waves. *Coast. Eng.*, 53(8):631–655, 2006.

[MBG96] Maier, R. S., Bernard, R. S., and Grunau, D. W. Boundary conditions for the lattice Boltzmann method. *Phys. Fluids*, 8(7):1788–1801, 1996.

[MM52] Martin, J. C. and Moyce, W. J. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philos. Trans. R. Soc. A Math. Phys. Eng.*, 244:312–324, 1952.

[ND13] Nishi, Y. and Doan, P. V. Hybrid boundary condition combined with data assimilation for simulations of free surface flows using lattice Boltzmann method. *Comput. Fluids*, 88:108–114, 2013.

[NH71] Nichols, B. D. and Hirt, C. W. Improved free surface conditions for numerical incompressible flow computations. *J. Comput. Phys.*, 8:434–448, 1971.

[NW76] Noh, W. F. and Woodward, P. SLIC (simple line interface calculation). In *Proceedings, Fifth International Conference on Fluid Dynamics*, volume 59, pages 330–340, Berlin, 1976. Springer.

[NY99] Nakamura, T. and Yabe, T. Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space. *Comput. Phys. Commun.*, 120(2):122–154, 1999.

[OF01] Osher, S. and Fedkiw, R. P. Level Set methods: an overview and some recent results. *J. Comput. Phys.*, 169(2):463–502, 2001.

[OI18] Onodera, N. and Idomura, Y. Acceleration of wind simulation using locally mesh-refined lattice boltzmann method on gpu-rich supercomputers. In *Supercomputing Frontiers*, pages 128–145, Cham, 2018. Springer International Publishing.

[OS88] Osher, S. and Sethian, J. A. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[PAB$^+$97] Puckett, E. G., Almgren, A. S., Bell, J. B., Marcus, D. L., and Rider, W. J. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *J. Comput. Phys.*, 130(2):269–282, 1997.

[PP04] Pilliod, J. E. and Puckett, E. G. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. Comput. Phys.*, 199(2):465–502, 2004.

[Pra17] Prasetyo, A. *Physical modeling and numerical analysis of tsunami inundation in a city scale.* PhD thesis, Kyoto university, 2017.

[PYMM18] Prasetyo, A., Yasuda, T., Miyashita, T. and Mori, N. Physical modeling and numerical analysis of tsunami inundation in a coastal city. *Front.*, submitted (2018/7).

[RK98] Rider, W. J. and Kothe, D. B. Reconstructing volume tracking. *J. Comput. Phys.*, 141:112–152, 1998.

[Rud97] Rudman, M. Volume-tracking methods for interfacial flow calculations. *Int. J. Numer. Methods Fluids*, 24:671–691, 1997.

[SCB98] Stansby, P. K., Chegini, A., and Barnes, T. C. D. The initial stages of dam-break flow. *J. Fluid Mech.*, 374:407–424, 1998.

[SFH89] Succi, S., Foti, E., and Higuera, F. Three-dimensional flows in complex geometries with the lattice Boltzmann method. *Europhys. Lett.*, 10(5):433–438, 1989.

[SFSO98] Sussman, M., Fatemi, E., Smereka, P., and Osher, S. An improved level set method for incompressible two-phase flows. *Comput. Fluids*, 27(5-6):663–680, 1998.

[SP00] Sussman, M. and Puckett, E. G. A coupled Level Set and Volume-of-Fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, 162(2):301–337, 2000.

[SS13] Shigeto, Y. and Sakai, M. Arbitrary-shaped wall boundary modeling based on signed distance functions for granular flow simulations. *Chem. Eng. J.*, 231:464–476, 2013.

[SS15] Sun, X. and Sakai, M. Three-dimensional simulation of gas-solid-liquid flows using the DEM-VOF method. *Chem. Eng. Sci.*, 134:531–548, 2015.

[SSO94] Sussman, M., Smereka, P., and Osher, S. A Level Set approach for computing solutions to incompressible two-phase flow, 1994.

[Sus03] Sussman, M. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *J. Comput. Phys.*, 187(1):110–136, 2003.

[SZ00] Scardovelli, R. and Zaleski, S. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *J. Comput. Phys.*, 164(1):228–237, 2000.

[TR09] Thürey, N. and Rüde, U. Stable free surface flows with the lattice Boltzmann method on adaptively coarsened grids. *Comput. Vis. Sci.*, 12(5):247–263, 2009.

[TY87] Takewaki, H. and Yabe, T. The cubic-interpolated pseudo particle (CIP) method: application to nonlinear and multi-dimensional hyperbolic equations. *J. Comput. Phys.*, 70(2):355–372, 1987.

[WT94] Wu, G. X. and Taylor, E. R. Finite element analysis of two-dimensional non-linear transient water waves. *Appl. Ocean Res.*, 16(6):363–372, 1994.

[YFLS06] Yang, X., Feng, J. J., Liu, C., and Shen, J. Numerical simulations of jet pinching-off and drop formation using an energetic variational phase-field method. *J. Comput. Phys.*, 218(1):417–428, 2006.

[YOD17] Yamamoto, T., Okano, Y., and Dost, S. Validation of the S-CLSVOF method with the density-scaled balanced continuum surface force model in multiphase systems coupled with thermocapillary flows. *Int. J. Numer. Methods Fluids*, 83(3):223–244, 2017.

[You82] Youngs, D. Time-dependent multi-material flow with large fluid distortion. *Numer. Methods Fluid Dyn.*, (October):273–285, 1982.

[You84] Youngs, D. L. An interface tracking method for a 3D Eulerian hydrodynamics code. *Tech. report. AWRE*, (Technical Report 44/92/35), 1984.

[YXU01] Yabe, T., Xiao, F., and Utsumi, T. The constrained interpolation profile method for multiphase analysis. *J. Comput. Phys.*, 169(2):556–593, 2001.

[ZAK15] Zabelok, S., Arslanbekov, R., and Kolobov, V. Adaptive kinetic-fluid solvers for heterogeneous computing architectures. *J. Comput. Phys.*, 303:455–469, 2015.

Chapter 4

Lattice Boltzmann method for shallow water flows

4.1 Introduction

The non-linear shallow water equations have been applied widely in river and coastal engineerings [MSH97, CPWC04, Zho04]. For example, it can be used to describe the tidal flows, tsunamis [Shu91], hydraulic jump, and open channel flows. Furthermore, coupled with the solute transport equation, the non-linear shallow water equations can be expanded to predict the solute transport such as distribution of pollution concentration and transport of suspended sediments [TJB$^+$01, SC06]. Although, non-hydrostatic fluid models such as Boussinesq models [Per67, MS92, WKGS95, GS96, SKH$^+$12] are increasingly used in research and damage assessment for major tsunami events using high-performance computing [PKAI11], the standard approach to simulate tsunami forecasting is still based on using models based on non-linear shallow water equations, which are a depth-integrated (hydrostatic) approximations of the Navier-Stokes equations [KM93]. The non-linear shallow water equations are accurate for a long wave propagation and run-up problem in an urban area, in which the horizontal length scale is much larger than the vertical length scale, such as for most tsunamis. If vertical variations have to be taken into account, these can be separated from the horizontal ones, resulting in a set of non-linear shallow water equations for a series of horizontal fluid layers [ZB11] (i.e., multilayer non-linear shallow water equations).

Although the lattice Boltzmann method is not widely known in the community working on free surface flow problems, several groups have already applied the lattice Boltzmann method to standard non-linear shallow water benchmark problems and test cases. For example, Frandsen [Fra08] presented a so-called two-dimensional 9-speed lattice Boltzmann method implementation for the simulation of wave run-up on a sloping beach. Thömmes *et. al.* [TSB07] applied a similar lattice

Boltzmann method to test cases including bed slope and friction terms. The main focus of their work was to demonstrate the ability of the method to cope with complex geometries and irregular bathymetry. This was illustrated by the simulation of the mean flow in the Strait of Gibraltar. These approaches were based on single relaxation time collision operators [BGK54, QDL92]. Tubbs [Tub10], however, applied an extended lattice Boltzmann method to a non-linear shallow water equations with and multiple-relaxation-time collision operators [LL00, LLS12]. Moreover, the author extended the standard lattice Boltzmann method to a multilayer approach that considers vertical, three-dimensional effects. The model is capable of simulating wind and density driven circulation over irregular bathymetry.

In this chapter, we describe conventional non-linear shallow water theory and its equations. Next we discuss the implementation of the lattice Boltzmann method for shallow water equations to tsunami modellings with moving shoreline algorithm and validate the proposed model in detail.

4.2 A non-linear shallow water equation

The governing equations for general incompressible flows are the three-dimensional continuity and Navier-Stokes equations which are derived from Newton's second law of motion and the mass conservation. The equations can be written in tensor form as:

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{4.1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial \left(u_i u_j\right)}{\partial x_j} = f_i - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j x_j} \tag{4.2}$$

where the subscripts i and j are space direction indices; f_i is the body force per unit mass acting on fluid in the i direction; and the Einstein summation convention is used.

The shallow water equations are derived from depth-integrating the Navier-Stokes equations with assumption implying that the vertical velocity of the fluid is small with hydrostatic pressure and constant velocities over the water depth [Zho04]. The shallow water equations are thus derived in the following section.

Firstly, the continuity equation for shallow water equations is derived by integrating Equation 4.1 over depth:

$$\int_{z_b}^{h+z_b} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) \mathrm{d}z = 0 \tag{4.3}$$

which results in:

$$\int_{z_b}^{h+z_b} \frac{\partial u}{\partial x} \mathrm{d}z + \int_{z_b}^{h+z_b} \frac{\partial v}{\partial y} \mathrm{d}z + w_s - w_b = 0 \tag{4.4}$$

where w_b and w_s are the vertical velocities at channel bottom and the free surface, respectively; and z_b is the bed elevation above a datum as shown in Figure 4.1. The first and second terms on the left hand side of Equation 4.4 can be written using the Leibnitz rule [SR66] as:

$$\int_{z_b}^{h+z_b} \frac{\partial u}{\partial x} \mathrm{d}z = \frac{\partial}{\partial x} \int_{z_b}^{h+z_b} u \mathrm{d}z - u_s \frac{\partial}{\partial x} (h + z_b) + u_b \frac{\partial z_b}{\partial x} \tag{4.5}$$

$$\int_{z_b}^{h+z_b} \frac{\partial v}{\partial y} \mathrm{d}z = \frac{\partial}{\partial y} \int_{z_b}^{h+z_b} v \mathrm{d}z - v_s \frac{\partial}{\partial y} (h + z_b) + u_b \frac{\partial z_b}{\partial y} \tag{4.6}$$

Substituting Equations 4.5 and 4.6 into Equation 4.4 leading to:

$$\frac{\partial}{\partial x} \int_{z_b}^{h+z_b} u \mathrm{d}z + \frac{\partial}{\partial y} \int_{z_b}^{h+z_b} v \mathrm{d}z + \left[w_s - u_s \frac{\partial}{\partial x} (h + z_b) - v_s \frac{\partial}{\partial y} (h + z_b) \right] - \left[w_b - u_b \frac{\partial z_b}{\partial x} - v_b \frac{\partial z_b}{\partial y} \right] = 0 \tag{4.7}$$

The kinematic conditions at the free surface and channel bottom are:

$$w_s = \frac{\partial}{\partial t} (h + z_b) + u_s \frac{\partial}{\partial x} (h + z_b) + v_s \frac{\partial}{\partial y} (h + z_b) \tag{4.8}$$

$$w_b = \frac{\partial z_b}{\partial t} + u_b \frac{\partial z_b}{\partial x} + v_b \frac{\partial z_b}{\partial y} \tag{4.9}$$

Substituting Equations 4.8 and 4.9 into Equation 4.7 results in:

$$\frac{\partial h}{\partial t} + \frac{\partial (h\bar{u})}{\partial x} + \frac{\partial (h\bar{v})}{\partial y} = 0 \tag{4.10}$$

which is just the continuity equation for the shallow water equations and $\bar{u}$ and $\bar{v}$ are depth-averaged velocity components and defined as:

$$\bar{u} = \frac{1}{h} \int_{z_b}^{h+z_b} u \mathrm{d}z \tag{4.11}$$

$$\bar{v} = \frac{1}{h} \int_{z_b}^{h+z_b} v \mathrm{d}z \tag{4.12}$$

Next, the momentum equation for shallow water flow will be derived. The Navier-Stokes equation along x-axis is integrated over water depth and the following expression can be obtained:

$$\int_{z_b}^{h+z_b} \left[\frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z} \right] \mathrm{d}z = \int_{z_b}^{h+z_b} f_c v \mathrm{d}z + \int_{z_b}^{h+z_b} \left[-\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \nabla^2 u \right] \mathrm{d}z \tag{4.13}$$

The Leibnitz rule [SR66] is used for the first three terms on the left-hand side leads to:

$$\int_{z_b}^{h+z_b} \frac{\partial u}{\partial t} \mathrm{d}z = \frac{\partial}{\partial t} \int_{z_b}^{h+z_b} u \mathrm{d}z - u_s \frac{\partial}{\partial t} (h + z_b) + u_b \frac{\partial z_b}{\partial t} \tag{4.14}$$

$$\int_{z_b}^{h+z_b} \frac{\partial (uu)}{\partial x} \mathrm{d}z = \frac{\partial}{\partial x} \int_{z_b}^{h+z_b} uu \mathrm{d}z - u_s u_s \frac{\partial}{\partial x} (h + z_b) + u_b u_b \frac{\partial z_b}{\partial x} \tag{4.15}$$

$$\int_{z_b}^{h+z_b} \frac{\partial (vu)}{\partial y} \mathrm{d}z = \frac{\partial}{\partial y} \int_{z_b}^{h+z_b} vu \mathrm{d}z - v_s u_s \frac{\partial}{\partial y} (h + z_b) + v_b u_b \frac{\partial z_b}{\partial y} \tag{4.16}$$

The last term on the left hand side of Equation 4.13 can be integrated, leading to:

$$\int_{z_b}^{h+z_b} \frac{\partial (wu)}{\partial z} \mathrm{d}z = w_s u_s - w_b u_b \tag{4.17}$$

combing it with the above equations and rewriting the results yields the following equation:

$$\begin{aligned}\int_{z_b}^{h+z_b} \left[\frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z}\right] \mathrm{d}z = \frac{\partial}{\partial t} \int_{z_b}^{h+z_b} u \mathrm{d}z + \frac{\partial}{\partial x} \int_{z_b}^{h+z_b} uu \mathrm{d}z + \frac{\partial}{\partial y} \int_{z_b}^{h+z_b} vu \mathrm{d}z \\ + u_s \left[w_s - \frac{\partial}{\partial t} (h + z_b) - u_s \frac{\partial}{\partial x} (h + z_b) - v_s \frac{\partial}{\partial y} (h + z_b) \right] - u_b \left(w_b - \frac{\partial z_b}{\partial t} - u_b \frac{\partial z_b}{\partial x} - v_b \frac{\partial z_b}{\partial y} \right)\end{aligned} \tag{4.18}$$

refer to the kinematic conditions. Combined with the Equations 4.11 and 4.12, the above equation can be rewritten as:

$$\int_{z_b}^{h+z_b} \left[\frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z}\right] \mathrm{d}z = \frac{\partial (\bar{u} h)}{\partial t} + \frac{\partial}{\partial x} \int_{z_b}^{h+z_b} uu \mathrm{d}z + \frac{\partial}{\partial y} \int_{z_b}^{h+z_b} vu \mathrm{d}z \tag{4.19}$$

with the second mean value theorem for integrals:

$$\int_a^b f(x) g(x) \mathrm{d}z = f(\zeta) \int_a^b g(x) \mathrm{d}z \tag{4.20}$$

the second term on the right-hand side of Equation 4.19 can be expressed as:

$$\int_{z_b}^{h+z_b} uu \mathrm{d}z = \breve{u}_1 \int_{z_b}^{h+z_b} u \mathrm{d}z = \breve{u}_1 h \bar{u} \tag{4.21}$$

and the last term on the right-hand side of Equation 4.19 can be expressed as:

$$\int_{z_b}^{h+z_b} vu \mathrm{d}z = \breve{u}_2 \int_{z_b}^{h+z_b} u \mathrm{d}z = \breve{u}_2 h \bar{v} \tag{4.22}$$

assuming $\breve{u}_1 = \theta_1 \bar{u}$ and $\breve{u}_2 = \theta_2 \bar{u}$ and substituting Equations 4.21 and 4.22 into Equation 4.19 leads to:

$$\int_{z_b}^{h+z_b} \left[\frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z}\right] \mathrm{d}z = \frac{\partial (\bar{u} h)}{\partial t} + \frac{\partial (\theta_1 h \bar{u} \bar{u})}{\partial x} + \frac{\partial (\theta_2 h \bar{v} \bar{u})}{\partial y} \tag{4.23}$$

where $theta_1$ and $theta_2$ are momentum correction factors and are determined by Equations 4.21

and 4.22 as:

$$\theta_1 = \frac{1}{h\bar{u}\bar{u}} \int_{z_b}^{h+z_b} uu\mathrm{d}z \tag{4.24}$$

$$\theta_2 = \frac{1}{h\bar{v}\bar{u}} \int_{z_b}^{h+z_b} vu\mathrm{d}z \tag{4.25}$$

Similarly, the following expression for the terms on the left hand side by the Navier-Stokes equation along y-axis can be obtained:

$$\int_{z_b}^{h+z_b} \left[\frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} + \frac{\partial (vv)}{\partial y} + \frac{\partial (wv)}{\partial z}\right]\mathrm{d}z = \frac{\partial (\bar{v}h)}{\partial t} + \frac{\partial (\theta_2 h\bar{u}\bar{v})}{\partial x} + \frac{\partial (\theta_3 h\bar{v}\bar{v})}{\partial y} \tag{4.26}$$

in which, an additional momentum correction factor θ_3 is defined by:

$$\theta_3 = \frac{1}{h\bar{v}\bar{v}} \int_{z_b}^{h+z_b} vv\mathrm{d}z \tag{4.27}$$

the first term on the right hand side of Equation 4.13 is integrated as:

$$\int_{z_b}^{h+z_b} f_c v\mathrm{d}z = f_c h\bar{v} \tag{4.28}$$

Since the vertical acceleration can be ignored in comparison with the horizontal effect in shallow water flows, the momentum along z-axis is reduced with $w \approx 0$ to:

$$\frac{\partial p}{\partial z} = -\rho g \tag{4.29}$$

which is integrated and the following expression can be obtained:

$$p = -\rho g z + C_0 \tag{4.30}$$

where C_0 is an integration constant. The pressure at the free surface is the atmospheric pressure p_a, for example: $p = p_a$ at $z = h + z_b$, in above equation, C_0 can be calculated by:

$$C_0 = \rho g \left(h + z_b\right) + p_a \tag{4.31}$$

substituting the above equation into Equation 4.30 results in:

$$p = \rho g \left(h + z_b - z\right) + p_a \tag{4.32}$$

p_a is almost constant in the corresponding area and often assumed to be zero. Because the difference in atmospheric pressure at water surface is usually small in most shallow water flows, the above equation reads:

$$p = \rho g \left(h + z_b - z \right) \tag{4.33}$$

the above equation is often referred to as the hydrostatic pressure approximation in shallow water flows and differentiating it to z gives:

$$\frac{\partial p}{\partial x} = \rho g \frac{\partial}{\partial x} \left(h + z_b \right) \tag{4.34}$$

because both of water depth h and the bed height z_b are functions of the horizontal coordinates x and y only, the following expression can be obtained:

$$\int_{z_b}^{h+z_b} \frac{1}{\rho} \frac{\partial p}{\partial x} \mathrm{d}z = \frac{h}{\rho} \frac{\partial p}{\partial x} \tag{4.35}$$

combing Equation 4.34 with Equation 4.35 can give:

$$\int_{z_b}^{h+z_b} \frac{1}{\rho} \frac{\partial p}{\partial x} \mathrm{d}z = gh \frac{\partial}{\partial x} \left(h + z_b \right) \tag{4.36}$$

the following approximations are given for the third and fourth terms on the right hand side of Equation 4.13:

$$\int_{z_b}^{h+z_b} \nu \frac{\partial^2 u}{\partial x^2} \mathrm{d}z \approx \nu \frac{\partial^2 \left(h\bar{u} \right)}{\partial x \partial x} \tag{4.37}$$

$$\int_{z_b}^{h+z_b} \nu \frac{\partial^2 u}{\partial y^2} \mathrm{d}z \approx \nu \frac{\partial^2 \left(h\bar{u} \right)}{\partial y \partial y} \tag{4.38}$$

the last term on the right hand side of Equation 4.13 is calculated as:

$$\int_{z_b}^{h+z_b} \nu \frac{\partial^2 u}{\partial z^2} \mathrm{d}z = \left(\nu \frac{\partial u}{\partial z} \right)_s - \left(\nu \frac{\partial u}{\partial z} \right)_b \tag{4.39}$$

the terms on the right hand side of Equation 4.39 can be approximated with the surface wind shear stress and the bed shear stress in the x-axis, respectively:

$$\left(\nu \frac{\partial u}{\partial z} \right)_s = \frac{\tau_{wx}}{\rho} \tag{4.40}$$

$$\left(\nu \frac{\partial u}{\partial z} \right)_b = \frac{\tau_{bx}}{\rho} \tag{4.41}$$

therefore, Equation 4.39 can be written as:

$$\int_{z_b}^{h+z_b} \nu \frac{\partial^2 u}{\partial z^2} \mathrm{d}z = \frac{\tau_{wx}}{\rho} - \frac{\tau_{bx}}{\rho} \tag{4.42}$$

combing the above equations leads to:

$$\frac{\partial\left(h\bar{u}\right)}{\partial t}+\frac{\partial\left(\theta_1 h\bar{u}\bar{u}\right)}{\partial x}+\frac{\partial\left(\theta_2 h\bar{v}\bar{u}\right)}{\partial y}=-g\frac{\partial}{\partial x}\left(\frac{h^2}{2}\right)+\nu\frac{\partial^2\left(h\bar{u}\right)}{\partial x\partial x}+\nu\frac{\partial^2\left(h\bar{u}\right)}{\partial y\partial y}-gh\frac{\partial z_b}{\partial x}+f_c h\bar{v}+\frac{\tau_{wx}}{\rho}-\frac{\tau_{bx}}{\rho} \tag{4.43}$$

the above equation is the momentum equation for shallow water flows in the x-axis. The momentum equation in the y-axis for shallow water flows can be derived similarly as:

$$\frac{\partial\left(h\bar{v}\right)}{\partial t}+\frac{\partial\left(\theta_1 h\bar{u}\bar{v}\right)}{\partial x}+\frac{\partial\left(\theta_3 h\bar{v}\bar{v}\right)}{\partial y}=-g\frac{\partial}{\partial y}\left(\frac{h^2}{2}\right)+\nu\frac{\partial^2\left(h\bar{v}\right)}{\partial x\partial x}+\nu\frac{\partial^2\left(h\bar{v}\right)}{\partial y\partial y}-gh\frac{\partial z_b}{\partial y}+f_c h\bar{u}+\frac{\tau_{wy}}{\rho}-\frac{\tau_{by}}{\rho} \tag{4.44}$$

theoretically, if the velocity profiles for u and v are known, the momentum correction factors θ_1, θ_2 and θ_3 can be calculated. It is, however, not easy to calculate these momentum correction factors, normally because there are no universal velocity distribution which are valid for all flows. On the other hand, $\theta_1 = 1$, $\theta_2 = 2$ and $\theta_3 = 1$ are used widely in numerical simulation for shallow water flows and these study show that assumption can give good results for most shallow water flows [KV73, MR78, BA97].

Therefore, when $\theta_1 = 1$, $\theta_2 = 2$ and $\theta_3 = 1$ are adopted, the above equations become:

$$\frac{\partial\left(h\bar{u}\right)}{\partial t}+\frac{\partial\left(h\bar{u}\bar{u}\right)}{\partial x}+\frac{\partial\left(h\bar{v}\bar{u}\right)}{\partial y}=-g\frac{\partial}{\partial x}\left(\frac{h^2}{2}\right)+\nu\frac{\partial^2\left(h\bar{u}\right)}{\partial x\partial x}+\nu\frac{\partial^2\left(h\bar{u}\right)}{\partial y\partial y}-gh\frac{\partial z_b}{\partial x}+f_c h\bar{v}+\frac{\tau_{wx}}{\rho}-\frac{\tau_{bx}}{\rho} \tag{4.45}$$

$$\frac{\partial\left(h\bar{v}\right)}{\partial t}+\frac{\partial\left(h\bar{u}\bar{v}\right)}{\partial x}+\frac{\partial\left(h\bar{v}\bar{v}\right)}{\partial y}=-g\frac{\partial}{\partial y}\left(\frac{h^2}{2}\right)+\nu\frac{\partial^2\left(h\bar{v}\right)}{\partial x\partial x}+\nu\frac{\partial^2\left(h\bar{v}\right)}{\partial y\partial y}-gh\frac{\partial z_b}{\partial y}+f_c h\bar{u}+\frac{\tau_{wy}}{\rho}-\frac{\tau_{by}}{\rho} \tag{4.46}$$

after the overbars are dropped for convenience, the continuity equation and the momentum equations can be expressed in a tensor form concisely as:

$$\frac{\partial h}{\partial t}+\frac{\partial\left(hu_j\right)}{\partial x_j}=0 \tag{4.47}$$

$$\frac{\partial\left(hu_i\right)}{\partial t}+\frac{\partial\left(hu_iu_j\right)}{\partial x_j}=-g\frac{\partial}{\partial x_i}\left(\frac{h^2}{2}\right)+\nu\frac{\partial^2\left(hu_i\right)}{\partial x_j\partial x_j}+F_i \tag{4.48}$$

in which, the force term F_i is defined as:

$$F_i=-gh\frac{\partial z_b}{\partial x_i}+\frac{\tau_{wi}}{\rho}-\frac{\tau_{bi}}{\rho}+E_i \tag{4.49}$$

where E_i is the Coriolis term. The Coriolis term and the wind shear stress can be ignored in two-dimensional tsunami modelling because of the spacial scale.

The bed shear stress τ_{bi} in the i direction can be calculated by the depth-averaged velocities:

$$\tau_{bi}=\rho C_b u_i\sqrt{u_ju_j} \tag{4.50}$$

where, C_b is the bed friction coefficient, estimated $C_b = g/C_z^2$, where C_z is the Chezy coefficient calculated by the following Manning equation:

$$C_b = h^{\frac{1}{6}}/n_b \tag{4.51}$$

4.3 A Subgrid-scale stress model

The governing equations with the large-eddy-simulation (LES) for turbulent flows can be derived by including a space-filtered quantity in the continuity equation and the momentum equation. The space-filtered governing equations can be expressed as:

$$\frac{\partial \tilde{u}_j}{\partial x_j} = 0 \tag{4.52}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial \left(\tilde{u}_i \tilde{u}_j\right)}{\partial x_j} = f_i - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 \tilde{u}_i}{\partial x_j x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{4.53}$$

in which $\tilde{u}_i$ is the space-filtered velocity component in the i direction and is defined by:

$$\tilde{u}_i\left(x, y, z, t\right) = \int\int\int_{\Delta x \Delta y \Delta z} u\left(x, y, z, t\right) G\left(x, y, z, x', y', z'\right) \mathrm{d}x' \mathrm{d}y' \mathrm{d}z' \tag{4.54}$$

where G is a spatial filter function. τ_{ij} is the subgrid-scale stress (SGS) that reflects the effect of the unresolved scales with the resolved scales and determined by:

$$\tau_{ij} = \widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j \tag{4.55}$$

with the Bussinesq assumption for turbulent stresses, the subgrid-scale stress can be expressed using a SGS eddy viscosity ν_e as:

$$\tau_{ij} = -\nu_e \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) \tag{4.56}$$

substituting the above equation into the momentum equation gives:

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial \left(\tilde{u}_i \tilde{u}_j\right)}{\partial x_j} = f_i - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \left(\nu + \nu_e\right) \frac{\partial^2 \tilde{u}_i}{\partial x_j \partial x_j} \tag{4.57}$$

when the standard Smagorinsky SGS model [Sma63] is adopted and the eddy viscosity ν_e can be expressed by:

$$\nu_e = \left(C_s l_s\right)^2 \sqrt{S_{ij} S_{ij}} \tag{4.58}$$

The continuity and momentum equations are the modified continuity and the Navier-Stokes equations used as the LES for turbulent flows. The finer the grid size, the less the unresolved scale eddies.

Similarly, the shallow water equations including the SGS model [Zho04] can be derived as:

$$\frac{\partial h}{\partial t} + \frac{\partial \left(h\tilde{u}_j\right)}{\partial x_j} = 0 \tag{4.59}$$

$$\frac{\partial \left(h\tilde{u}_i\right)}{\partial t} + \frac{\partial \left(h\tilde{u}_i\tilde{u}_j\right)}{\partial x_j} = -g\frac{\partial}{\partial x_i}\left(\frac{h^2}{2}\right) + \left(\nu + \nu_e\right)\frac{\partial^2 \left(h\tilde{u}_i\right)}{\partial x_j \partial x_j} + F_i \tag{4.60}$$

where $\tilde{u}_i$ is the depth-averaged space-filtered velocity component, τ_{ij} is the depth-averaged subgrid-scale stress with eddy viscosity and is expressed by:

$$\tau_{ij} = -\nu_e\left(\frac{\partial \left(h\tilde{u}_i\right)}{\partial x_j} + \frac{\partial \left(h\tilde{u}_j\right)}{\partial x_i}\right) \tag{4.61}$$

the eddy viscosity ν_e takes the same form as Equation 4.58, but the S_{ij} is represented by:

$$S_{ij} = \frac{1}{2h}\left(\frac{\partial \left(h\tilde{u}_i\right)}{\partial x_j} + \frac{\partial \left(h\tilde{u}_j\right)}{\partial x_i}\right) \tag{4.62}$$

4.4 The lattice Boltzmann method for a non-linear shallow water equation

4.4.1 Derivation of the lattice Boltzmann equation

The lattice Boltzmann equation is not only evolved from the lattice gas automata, but can also be derived from the continuum Boltzmann equation [HL97, Abe97] as shown in the following.

The Boltzmann equation with the BGK collision operator reads [BGK54, QDL92]:

$$\frac{\partial f}{\partial t} + \boldsymbol{e} \cdot \nabla f = -\frac{1}{\lambda}\left(f - f^{eq}\right) \tag{4.63}$$

in which, $f = f\left(\boldsymbol{x}, \boldsymbol{e}, t\right)$ is the single-particle distribution in continuum phase space, $\boldsymbol{e}$ is the particle velocity, λ is a relaxation time, ∇ is the gradient operator given by:

$$\nabla = \boldsymbol{i}\frac{\partial}{\partial x} + \boldsymbol{j}\frac{\partial}{\partial y} \tag{4.64}$$

and f^{eq} is the Maxwell-Boltzmann equilibrium distribution function expressed as:

$$f^{eq} = \frac{\rho}{\left(2\pi/3\right)^{D/2}} \exp\left[-\frac{3}{2}\left(\boldsymbol{e} - \boldsymbol{V}\right)^2\right] \tag{4.65}$$

where D is the spatial dimension, $\boldsymbol{e}$ is particle velocity and $\boldsymbol{V}$ is fluid velocity; $\boldsymbol{e}$ and $\boldsymbol{V}$ are normalised by $\sqrt{3RT_c}$ (R is the ideal gas constant and T_c is the temperature), which leads to a sound speed of

$U_s = 1/\sqrt{3}$ [CD98]. The fluid density and velocity are computed as follows:

$$\rho = \int f \mathrm{d}\boldsymbol{e} \tag{4.66}$$

$$\rho V = \int \boldsymbol{e} f \mathrm{d}\boldsymbol{e} \tag{4.67}$$

If the fluid velocity $\boldsymbol{V}$ is relative small compared with the sound speed, the equilibrium distribution function can be expanded up to the second-order accuracy [Koe91] as follows:

$$f^{eq} = \frac{\rho}{(2\pi/3)^{D/2}} \exp\left(-\frac{3}{2}e^2\right)\left[1 + 3\left(e \cdot V\right) + \frac{9}{2}\left(e \cdot V\right)^2 - \frac{3}{2}V \cdot V\right] \tag{4.68}$$

For the purpose of developing a discrete model, a limited number of particle velocities are adopted $e_\alpha \left(\alpha = 1, \ldots, K\right)$, and the distribution function including these velocities can be changed to:

$$f_\alpha^{eq}\left(x, t\right) = f^{eq}\left(x, e_\alpha, t\right) \tag{4.69}$$

which satisfies Equation 4.63:

$$\frac{\partial f_\alpha}{\partial t} + e_\alpha \cdot \nabla f_\alpha = -\frac{1}{\lambda}\left(f_\alpha - f_\alpha^{eq}\right) \tag{4.70}$$

In the limited discrete space and time, the left hand side of Equation 4.70 is the Lagrangian time derivative and can be discretized as:

$$\frac{\partial f_\alpha}{\partial t} + e_\alpha \cdot \nabla f_\alpha = \frac{f_\alpha\left(x, t + \Delta t\right) - f_\alpha\left(x, t\right)}{\Delta t} + e_{\alpha x}\frac{f_\alpha\left(x + \Delta x, t + \Delta t\right) - f_\alpha\left(x, t + \Delta t\right)}{\Delta x} \tag{4.71}$$

$$\frac{f_\alpha\left(x, t + \Delta t\right) - f_\alpha\left(x, t\right)}{\Delta t} + e_{\alpha x}\frac{f_\alpha\left(x + \Delta x, t + \Delta t\right) - f_\alpha\left(x, t + \Delta t\right)}{\Delta x} = -\frac{1}{\tau}\left(f_\alpha - f_\alpha^{eq}\right) \tag{4.72}$$

in which $e_{\alpha x}$ can be defined by $e_{\alpha x} = \Delta x/\Delta t$. Combining the above equation with Equation 4.70 produces the standard lattice Boltzmann equation:

$$f_\alpha\left(x + e_\alpha \Delta t, t + \Delta t\right) - f_\alpha\left(x, t\right) = -\frac{1}{\tau}\left(f_\alpha - f_\alpha^{eq}\right) \tag{4.73}$$

where $\tau = \lambda/\Delta t$. In fact, τ should be a single dimensionless relaxation time.

4.5 The lattice Boltzmann equation

The governing equation which is generally valid for fluid flows including the shallow water flows [Zho04] in the lattice Boltzmann method is as follows:

$$f_\alpha\left(x + e_\alpha \Delta t, t + \Delta t\right) = f_\alpha\left(x, t\right) + \Omega_\alpha\left[f_\alpha\left(x, t\right)\right] + \frac{\Delta t}{N_\alpha e^2} e_{\alpha i} F_i\left(x, t\right) \tag{4.74}$$

where N_α is a constant and is determined by the lattice pattern as:

$$N_\alpha = \frac{1}{e^2} \sum_\alpha e_{\alpha i} e_{\alpha i} \tag{4.75}$$

where Ω_α is the collision operator which represents the rate of change of f_α during collision. Theoretically, Ω_α is a complex matrix and is determined by the microscopic dynamics. An idea to linearize the collision operator is given firstly by Higuera and Jimenez [HJ89]. Based on this idea, Ω_α can be expanded about its equilibrium value [NCGB95] as follows:

$$\Omega_\alpha\left(f\right) = \Omega_\alpha\left(f^{eq}\right) + \frac{\partial \Omega_\alpha\left(f^{eq}\right)}{\partial f_\beta}\left(f_\beta - f_\beta^{eq}\right) + O\left[\left(f_\beta - f_\beta^{eq}\right)^2\right] \tag{4.76}$$

the solution process of the lattice Boltzmann equation is characterised by $f_\beta \rightarrow f_\beta^{eq}$, indicating $\Omega\left(f^{eq}\right) \approx 0$. Furthermore, if the higher-order terms in Equation 4.76 are neglected, a linearized collision operator can be obtained:

$$\Omega_\alpha\left(f\right) = \Omega_\alpha\left(f^{eq}\right) + \frac{\partial \Omega_\alpha\left(f^{eq}\right)}{\partial f_\beta}\left(f_\beta - f_\beta^{eq}\right) \tag{4.77}$$

the Bhatnagar-Gross-Krook (BGK) scheme simplifies the lattice Boltzmann equation greatly and makes the LBM used widely in various sectors. If the local particle distribution is assumed to be relaxed to an equilibrium state at a single rate τ [CCMM91].

$$\frac{\partial \Omega_\alpha\left(f^{eq}\right)}{\partial f_\beta} = -\frac{1}{\tau} \sigma_{\alpha\beta} \tag{4.78}$$

in which $\delta_{\alpha\beta}$ is the Kronecker delta function as:

$$\delta_{\alpha\beta} = \begin{cases} 0 & (\alpha \neq \beta) \\ 1 & (\alpha = \beta) \end{cases} \tag{4.79}$$

the collision term can be rearranged as:

$$\Omega_\alpha\left(f\right) = -\frac{1}{\tau} \delta_{\alpha\beta}\left(f_\beta - f_\beta^{eq}\right) \tag{4.80}$$

leading to the lattice BGK collision operator:

$$\Omega_{\alpha}(f) = -\frac{1}{\tau}\left(f_{\beta} - f_{\beta}^{eq}\right) \tag{4.81}$$

in which τ in named as the single relaxation time. Because the BGK simplifies the lattice Boltzmann equation extremely and increases efficiency; it is widely used in the lattice Boltzmann model. Combining the governing equation and the BGK model, the following lattice Boltzmann equation can be obtained:

$$f_{\alpha}(x + e_{\alpha}\Delta t, t + \Delta t) - f_{\alpha}(x,t) = -\frac{1}{\tau}\left(f_{\alpha} - f_{\alpha}^{eq}\right) + \frac{\Delta t}{N_{\alpha}e^{2}} e_{\alpha i} F_{i}(x,t) \tag{4.82}$$

the above equation becomes the most popular form of the lattice Boltzmann equation used today.

4.6 Lattice pattern

As in traditional methods, a lattice pattern is needed to represent the grid points and discrete computational domain. Furthermore, the lattice pattern has been used to determine particles' motions in the LBM, in which a microscopic model for molecular dynamics has been defined. Besides, the constant N_{α} is decided by the lattice pattern.

Generally, there are two kinds of lattice patterns; square lattice and hexagonal lattice for 2D cases. The square lattice can have 9-speed and hexagonal lattice can have 7-speed model according to the number of particle speed at lattice node. However, not all of these models can recover the correct flow equations and that requires sufficient lattice symmetry [FHP86]. Studies show that both of 9-speed square lattice and 7-speed hexagonal lattice have such property and can give satisfactory performance in numerical simulations. Therefore, these two kinds of lattice patterns have been used widely in the LBM.

However, the studies indicate that the 9-speed square lattice usually can produce more accurate result than that from the hexagonal lattice [Sko93]. Furthermore, the use of the square lattice leads to an easy way to implement different boundary conditions [Zho01]. Consequently, the 9-speed square lattice model (D2Q9) shown in Figure 4.2 is adopted in this thesis.

In the two-dimensional 9-speed square lattice model, each particle moves one lattice unit at its velocity along one of the eight links indicated with 1-8 and 0 indicates the rest particle with zero speed. The velocity vector of particles is given by:

$$e_{\alpha} = \begin{cases} (0,0) & (\alpha = 0) \\ e\left[\cos\frac{(\alpha-1)\pi}{4}, \sin\frac{(\alpha-1)\pi}{4}\right] & (\alpha = 1,3,5,7) \\ \sqrt{2}e\left[\cos\frac{(\alpha-1)\pi}{4}, \sin\frac{(\alpha-1)\pi}{4}\right] & (\alpha = 2,4,6,8) \end{cases} \tag{4.83}$$

it is not difficult to demonstrate that D2Q9 has the following features:

$$\sum_{\alpha} e_{\alpha i} = \sum_{\alpha} e_{\alpha i} e_{\alpha j} e_{\alpha k} = 0 \tag{4.84}$$

$$\sum_{\alpha} e_{\alpha i} e_{\alpha j} = 6e^2 \delta_{ij} \tag{4.85}$$

$$\sum_{\alpha} e_{\alpha i} e_{\alpha j} e_{\alpha k} e_{\alpha l} = 4e^4 \left(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\right) - 6e^4 \Delta_{ijkl} \tag{4.86}$$

where:

$$\Delta_{ijkl} = \begin{cases} 1 & (i = j = k = l) \\ 0 & (\text{otherwise}) \end{cases} \tag{4.87}$$

substituting the Equation 4.83 into the Equation 4.75 and the following equation can be obtained:

$$N_\alpha = \frac{1}{e^2} \sum_{\alpha} e_{\alpha x} e_{\alpha x} = \frac{1}{e^2} \sum_{\alpha} e_{\alpha y} e_{\alpha y} = 6 \tag{4.88}$$

combing the above equation with the governing equation, the following equation can be obtained:

$$f_\alpha \left(x + e_\alpha \Delta t, t + \Delta t\right) = f_\alpha \left(x, t\right) + \Omega_\alpha \left[f_\alpha \left(x, t\right)\right] + \frac{\Delta t}{6e^2} e_{\alpha i} F_i \left(x, t\right) \tag{4.89}$$

This equation is the most common form of a lattice Boltzmann model with D2Q9.

4.6.1 The equilibrium distribution function

To determine a suitable equilibrium distribution function has an important role in the lattice Boltzmann method because it decides what flow equations could be solved by the lattice Boltzmann equation. In order to apply the equation to solve the two-dimensional shallow water equations, a suitable equilibrium function f_α^{eq} has to be derived.

According to the primitive lattice gas automata, an equilibrium function is based on the Maxwell-Boltzmann state which is often expanded using a Taylor expansion in macroscopic velocity to its second order [HL97, CD98]. The Navier-Stokes equations can be recovered by using such equilibrium function in the lattice Boltzmann equation [CD98] as shown the above section. The shallow water equation, however, cannot be recovered with this kind of approach. On the other hand, an alternative method is to assume that an equilibrium function can be expressed as a power series in macroscopic velocity [RZ97] which has been used successfully in [CCM92, Gua00] and show its accuracy and suitability [Zho04].

The equilibrium function can be expressed as:

$$f_\alpha^{eq} = A_\alpha + B_\alpha e_{\alpha i} u_i + C_\alpha e_{\alpha i} e_{\alpha j} u_i u_j + D_\alpha u_i u_i \tag{4.90}$$

because the equilibrium function has the same symmetry as the lattice model, there are:

$$\begin{aligned} A_1 = A_3 = A_5 = A_7 = \bar{A} \\ A_2 = A_4 = A_6 = A_8 = \tilde{A} \end{aligned} \tag{4.91}$$

and the similar expressions for B_α, C_α and D_α are used. Therefore, Equation 4.90 can be rewritten as:

$$f_\alpha^{eq} = \begin{cases} A_0 + D_0 u_i u_i & (\alpha = 0) \\ \bar{A} + \bar{B} e_{\alpha i} u_i + \bar{C} e_{\alpha i} e_{\alpha j} u_i u_j + \bar{D} u_i u_i & (\alpha = 1, 3, 5, 7) \\ \tilde{A} + \tilde{B} e_{\alpha i} u_i + \tilde{C} e_{\alpha i} e_{\alpha j} u_i u_j + \tilde{D} u_i u_i & (\alpha = 2, 4, 6, 8) \end{cases} \tag{4.92}$$

the coefficients can be determined by the constraints on the equilibrium distribution function, such as mass and momentum conservations. In the shallow water equations, the constraints are the following three conditions:

$$\sum_\alpha f_\alpha^{eq}\left(\boldsymbol{x}, t\right) = h\left(\boldsymbol{x}, t\right) \tag{4.93}$$

$$\sum_\alpha e_{\alpha i} f_\alpha^{eq}\left(\boldsymbol{x}, t\right) = h\left(\boldsymbol{x}, t\right) u_i\left(\boldsymbol{x}, t\right) \tag{4.94}$$

$$\sum_\alpha e_{\alpha i} e_{\alpha j} f_\alpha^{eq}\left(\boldsymbol{x}, t\right) = \frac{1}{2} g h^2\left(\boldsymbol{x}, t\right) \delta_{ij} + h\left(\boldsymbol{x}, t\right) u_i\left(\boldsymbol{x}, t\right) u_j\left(\boldsymbol{x}, t\right) \tag{4.95}$$

The equilibrium distribution function Equation 4.90 can be obtained under the above constraints.

When Equation 4.92 is substituted into Equation 4.93, the following equation can be obtained:

$$\begin{aligned} A_0 + D_0 u_i u_i + 4\bar{A} + \sum_{\alpha=1,3,5,7} \bar{B} e_{\alpha i} u_i + \sum_{\alpha=1,3,5,7} \bar{C} e_{\alpha i} e_{\alpha j} u_i u_j + 4\bar{D} u_i u_i \\ + 4\tilde{A} + \sum_{\alpha=2,4,6,8} \tilde{B} e_{\alpha i} u_i + \sum_{\alpha=2,4,6,8} \tilde{C} e_{\alpha i} e_{\alpha j} u_i u_j + 4\tilde{D} u_i u_i = h \end{aligned} \tag{4.96}$$

combing the lattice model and equating the coefficients of h and $u_i u_i$, respectively, the following equations can be obtained:

$$A_0 + 4\bar{A} + 4\tilde{A} = h \tag{4.97}$$

$$D_0 + 2e^2\bar{C} + 4e^2\tilde{C} + 4\bar{D} + 4\tilde{D} = 0 \tag{4.98}$$

Similarly, substituting Equation 4.92 into Equation 4.94 results in:

$$\begin{aligned} A_0 e_{\alpha i} + D_0 e_{\alpha i} u_i u_i + \sum_{\alpha=1,3,5,7} \big(\bar{A} e_{\alpha i} + \bar{B} e_{\alpha i} e_{\alpha j} u_j + \bar{C} e_{\alpha i} e_{\alpha j} e_{\alpha k} u_i u_k + \bar{D} e_{\alpha i} u_i u_i \big) \\ + \sum_{\alpha=2,4,6,8} \big(\tilde{A} e_{\alpha i} + \bar{B} e_{\alpha i} e_{\alpha j} u_j + \tilde{C} e_{\alpha i} e_{\alpha j} e_{\alpha k} u_i u_k + \tilde{D} e_{\alpha i} u_i u_i \big) = h u_i \end{aligned} \tag{4.99}$$

rearranged the above equation yields:

$$2e^2 \bar{B} + 4e^2 \tilde{B} = h \tag{4.100}$$

Inserting Equation 4.92 to Equation 4.95 leads to:

$$\begin{aligned} A_0 e_{\alpha i} + D_0 e_{\alpha i} u_i u_i + \sum_{\alpha=1,3,5,7} \big(\bar{A} e_{\alpha i} e_{\alpha j} + \bar{B} e_{\alpha i} e_{\alpha j} e_{\alpha k} u_k + \bar{C} e_{\alpha i} e_{\alpha j} e_{\alpha k} e_{\alpha l} u_k u_l + \bar{D} e_{\alpha i} e_{\alpha j} u_k u_k \big) \\ + \sum_{\alpha=2,4,6,8} \big(\tilde{A} e_{\alpha i} e_{\alpha j} + \bar{B} e_{\alpha i} e_{\alpha j} e_{\alpha k} u_k + \tilde{C} e_{\alpha i} e_{\alpha j} e_{\alpha k} e_{\alpha l} u_k u_l + \tilde{D} e_{\alpha i} e_{\alpha j} u_k u_k \big) = \frac{1}{2} g h^2 \delta_{ij} + h u_i u_j \end{aligned} \tag{4.101}$$

combing the lattice model, the above equation can be expressed by:

$$\begin{aligned} 2\bar{A} e^2 \delta_{ij} + 2\bar{C} e^4 u_i u_i + 2\bar{D} e^2 u_i u_i + 4\tilde{A} e^2 \delta_{ij} \\ + 8\tilde{C} e^4 u_i u_j + 4\tilde{C} e^4 u_i u_i + 4\bar{D} e^2 u_i u_i = \frac{1}{2} g h^2 \delta_{ij} + h u_i u_j \end{aligned} \tag{4.102}$$

based on the above equation, the following four equations can be obtained:

$$2e^2 \bar{A} + 4e^2 \tilde{A} = \frac{1}{2} g h^2 \tag{4.103}$$

$$8e^4 \tilde{C} = h \tag{4.104}$$

$$2e^4 \bar{C} = h \tag{4.105}$$

$$2e^2 \bar{D} + 4e^2 \tilde{D} + 4e^4 + \tilde{C} = 0 \tag{4.106}$$

substituted the Equation 4.104 into the Equation 4.105, the following equation can be obtained:

$$\bar{C} = 4\tilde{C} \tag{4.107}$$

because of the symmetry of the lattice model, the following relations can be assumed:

$$\bar{A} = 4\tilde{A} \tag{4.108}$$

$$\bar{B} = 4\tilde{B} \tag{4.109}$$

$$\bar{D} = 4\tilde{D} \tag{4.110}$$

combing the above equations and relations lead to:

$$A_0 = h - \frac{5gh^2}{6e^2}, \;\; D_0 = -\frac{2h}{3e^2} \tag{4.111}$$

$$\bar{A} = \frac{gh^2}{6e^2}, \;\; \bar{B} = \frac{h}{3e^2}, \;\; \bar{C} = \frac{h}{2e^4}, \;\; \bar{D} = -\frac{h}{6e^2} \tag{4.112}$$

$$\tilde{A} = \frac{gh^2}{24e^2}, \;\; \tilde{B} = \frac{h}{12e^2}, \;\; \tilde{C} = \frac{h}{8e^4}, \;\; \tilde{D} = -\frac{h}{24e^2} \tag{4.113}$$

Therefore, the equilibrium function can be expressed by:

$$f_\alpha^{eq} = \begin{cases} h - \frac{5gh^2}{6e^2} - \frac{2h}{3e^2}u_i u_i & (\alpha = 0) \\ \frac{gh^2}{6e^2} + \frac{h}{3e^2}e_{\alpha i}u_i + \frac{h}{2e^4}e_{\alpha i}e_{\alpha j}u_i u_j - \frac{h}{6e^2}u_i u_i & (\alpha = 1,3,5,7) \\ \frac{gh^2}{24e^2} + \frac{h}{12e^2}e_{\alpha i}u_i + \frac{h}{8e^4}e_{\alpha i}e_{\alpha j}u_i u_j - \frac{h}{24e^2}u_i u_i & (\alpha = 2,4,6,8) \end{cases} \tag{4.114}$$

with two-dimensional shallow water flows.

4.7 Macroscopic properties

The above sections have shown the lattice Boltzmann model for shallow water equations proposed by [Zho02]. In order to recover the shallow water equation, the link between microdynamic variables and macroscopic the physical quantities (such as the water depth h and velocity u_i) will be established in this section. The macroscopic properties of the lattice Boltzmann equation has been examined by [Zho04].

The sum of the 0-th moment of the distribution function with the lattice Boltzmann equation is shown by:

$$\sum_\alpha \left[f_\alpha\left(x + e_\alpha \Delta t, t + \Delta t\right) - f_\alpha\left(x, t\right)\right] = -\frac{1}{\tau}\sum_\alpha \left(f_\alpha - f_\alpha^{eq}\right) + \frac{\Delta t}{6e^2}\sum_\alpha e_{\alpha i}F_i \tag{4.115}$$

It is easy to demonstrate $\sum_\alpha e_{\alpha i}F_i = 0$, and the above equation can be simplified as:

$$\sum_\alpha \left[f_\alpha\left(x + e_\alpha \Delta t, t + \Delta t\right) - f_\alpha\left(x, t\right)\right] = -\frac{1}{\tau}\sum_\alpha \left(f_\alpha - f_\alpha^{eq}\right) \tag{4.116}$$

the mass and momentum, are the corresponding summations of the microdynamic mass and momentum which are conserved in the lattice Boltzmann method, should also meet the requirement of mass conservation. The continuity equation with microdynamic variables can be expressed by:

$$\sum_{\alpha} f_{\alpha}\left(x+e_{\alpha}\Delta t, t+\Delta t\right) \equiv \sum_{\alpha} f_{\alpha}\left(x, t\right) \tag{4.117}$$

substituting the above equation into Equation 4.116 results in:

$$\sum_{\alpha} f_{\alpha}\left(x, t\right) = \sum_{\alpha} f_{\alpha}^{eq}\left(x, t\right) \tag{4.118}$$

combining Equation 4.93 with the above expression, the water depth can be obtained as:

$$h\left(x, t\right) = \sum_{\alpha} f_{\alpha}\left(x, t\right) \tag{4.119}$$

Next, the velocity will be defined. The sum of the first moment of distribution function with the lattice Boltzmann equation is taken:

$$\sum_{\alpha} e_{\alpha i}\big[f_{\alpha}\left(x+e_{\alpha}\Delta t, t+\Delta t\right) - f_{\alpha}\left(x, t\right)\big] = -\frac{1}{\tau}\sum_{\alpha} e_{\alpha i}\left(f_{\alpha} - f_{\alpha}^{eq}\right) + \frac{\Delta t}{6e^2}\sum_{\alpha} e_{\alpha i} e_{\alpha j} F_i \tag{4.120}$$

combing with the feature of D2Q9, the above equation can be expressed:

$$\sum_{\alpha} e_{\alpha i}\big[f_{\alpha}\left(x+e_{\alpha}\Delta t, t+\Delta t\right) - f_{\alpha}\left(x, t\right)\big] = F_i\Delta t - \frac{1}{\tau}\sum_{\alpha} e_{\alpha i}\left(f_{\alpha} - f_{\alpha}^{eq}\right) \tag{4.121}$$

According to the Newton's second law, the momentum equation with microdynamic variables requires:

$$\sum_{\alpha} e_{\alpha i}\big[f_{\alpha}\left(x+e_{\alpha}\Delta t, t+\Delta t\right) - f_{\alpha}\left(x, t\right)\big] \equiv F_i\Delta t \tag{4.122}$$

substituting the above equation into Equation 4.121 leads to:

$$\sum_{\alpha} e_{\alpha i} f_{\alpha}\left(x, t\right) = \sum_{\alpha} e_{\alpha i} f_{\alpha}^{eq}\left(x, t\right) \tag{4.123}$$

combing Equation 4.94 with above equation, the velocity u_i can be defined as:

$$u_i\left(x, t\right) = \frac{1}{h\left(x, t\right)}\sum_{\alpha} e_{\alpha i} f_{\alpha}\left(x, t\right) \tag{4.124}$$

4.8 Recovery of the shallow water equations

For the purpose of demonstrating that the depth and velocities obtained from Equations 4.119 and 4.124 are the solution to the shallow water equations [Zho04], the lattice Boltzmann equation will recover the macroscopic shallow water equations, in which the Chapman-Enskog expansion and

Taylor expansion are used.

When Δt is assumed to be small and is equal to ϵ:

$$\Delta t = \epsilon \tag{4.125}$$

the lattice Boltzmann equation can be expressed by:

$$f_\alpha \left(x + e_\alpha \epsilon, t + \epsilon\right) - f_\alpha \left(x, t\right) = -\frac{1}{\tau}\left(f_\alpha - f_\alpha^{eq}\right) + \frac{\epsilon}{6e^2} e_{\alpha i} F_i \left(x, t\right) \tag{4.126}$$

A Taylor expansion is applied to the first term on the left-hand side of the above equation in time and space around point (x, t) and results in:

$$\epsilon \left(\frac{\partial}{\partial t} + e_{\alpha j}\frac{\partial}{\partial x_j}\right) f_\alpha + \frac{1}{2}\epsilon^2 \left(\frac{\partial}{\partial t} + e_{\alpha j}\frac{\partial}{\partial x_j}\right)^2 f_\alpha + O\left(\epsilon^2\right) = -\frac{1}{\tau}\left(f_\alpha - f_\alpha^{(0)}\right) + \frac{\epsilon}{6e^2} e_{\alpha j} F_j \tag{4.127}$$

in which $f_\alpha^{(0)} = f_\alpha^{(eq)}$ and f_α can be expanded around $f_\alpha^{(0)}$:

$$f_\alpha = f^{(0)} + \epsilon f^{(1)} + \epsilon^2 f^{(2)} + O\left(\epsilon^2\right) \tag{4.128}$$

Inserting Equation 4.128 into the governing equation and ignoring the high order term (higher than second order), the equations to order ϵ and ϵ^2 can be obtained, respectively:

$$\left(\frac{\partial}{\partial t} + e_{\alpha j}\frac{\partial}{\partial x_j}\right) f_\alpha^{(0)} = -\frac{1}{\tau} f_\alpha^{(1)} + \frac{1}{6e^2} e_{\alpha j} F_j \tag{4.129}$$

$$\left(\frac{\partial}{\partial t} + e_{\alpha j}\frac{\partial}{\partial x_j}\right) f_\alpha^{(1)} + \frac{1}{2}\left(\frac{\partial}{\partial t} + e_{\alpha j}\frac{\partial}{\partial x_j}\right)^2 f_\alpha^{(0)} = -\frac{1}{\tau} f_\alpha^{(2)} \tag{4.130}$$

substituting Equation 4.129 into Equation 4.130 and rearranging lead to:

$$\left(1 - \frac{1}{2\tau}\right)\left(\frac{\partial}{\partial t} + e_{\alpha j}\frac{\partial}{\partial x_j}\right) f_\alpha^{(1)} = -\frac{1}{\tau} f_\alpha^{(2)} - \frac{1}{2}\left(\frac{\partial}{\partial t} + e_{\alpha j}\frac{\partial}{\partial x_j}\right)\left(\frac{1}{6e^2} e_{\alpha k} F_k\right) \tag{4.131}$$

from $\sum(4.129) + \epsilon \times (4.131)$ about α, the following equation can be derived:

$$\frac{\partial}{\partial t}\left(\sum_\alpha f_\alpha^{(0)}\right) + \frac{\partial}{\partial x_j}\left(\sum_\alpha e_{\alpha j} f_\alpha^{(0)}\right) = -\epsilon \frac{1}{12e^2}\frac{\partial}{\partial x_j}\left(\sum_\alpha e_{\alpha j} e_{\alpha k} F_k\right) \tag{4.132}$$

when the force term with the first-order accuracy is applied, Combining Equations 4.83 and 4.114 with above equation leads to:

$$\frac{\partial h}{\partial t} + \frac{\partial\left(h u_j\right)}{\partial x_j} = 0 \tag{4.133}$$

which is just the continuity equation for the macroscopic shallow water equations.

Taking $\sum e_{\alpha i}\left[(4.129)+\epsilon\times(4.131)\right]$ about α can result in:

$$\begin{aligned}\frac{\partial}{\partial t}\left(\sum_{\alpha}e_{\alpha i}f_{\alpha}^{(0)}\right)+\frac{\partial}{\partial x_j}\left(\sum_{\alpha}e_{\alpha i}e_{\alpha j}f_{\alpha}^{(0)}\right)+\epsilon\left(1-\frac{1}{2\tau}\right)\frac{\partial}{\partial x_j}\left(\sum_{\alpha}e_{\alpha i}e_{\alpha j}f_{\alpha}^{(1)}\right)\\ =F_j\delta_{ij}-\epsilon\frac{1}{2}\sum_{\alpha}e_{\alpha i}\left(\frac{\partial}{\partial t}+e_{\alpha}\frac{\partial}{\partial x_j}\right)\left(\frac{1}{6e^2}e_{\alpha j}F_j\right)\end{aligned} \tag{4.134}$$

similarly, if the force term with the first-order accuracy is used, combining it with Equations 4.83 and 4.114, the above equation can be rearranged as:

$$\frac{\partial\left(hu_i\right)}{\partial t}+\frac{\partial\left(hu_iu_j\right)}{\partial x_j}=-g\frac{\partial}{\partial x_i}\left(\frac{h^2}{2}\right)-\frac{\partial}{\partial x_j}\Lambda_{ij}+F_i \tag{4.135}$$

in which:

$$\Lambda_{ij}=\frac{\epsilon}{2\tau}\left(2\tau-1\right)\sum_{\alpha}e_{\alpha i}e_{\alpha j}f_{\alpha}^{(1)} \tag{4.136}$$

combining Equations 4.129, 4.83 and 4.114 and making some algebra, the following expression can be obtained:

$$\Lambda_{ij}\approx-\nu\left[\frac{\partial\left(hu_i\right)}{\partial x_j}+\frac{\partial\left(hu_j\right)}{\partial x_i}\right] \tag{4.137}$$

substituting the above equation into the governing equation results in:

$$\frac{\partial\left(hu_i\right)}{\partial t}+\frac{\partial\left(hu_iu_j\right)}{\partial x_j}=-g\frac{\partial}{\partial x_i}\left(\frac{h^2}{2}\right)+\nu\frac{\partial^2\left(hu_i\right)}{\partial x_jx_j}+F_i \tag{4.138}$$

this equation is the momentum equation for the macroscopic shallow water equations; where the kinematic viscosity ν is defined by:

$$\nu=\frac{e^2\Delta t}{6}\left(2\tau-1\right) \tag{4.139}$$

4.9 Stability conditions

The lattice Boltzmann equation can be interpreted as a Lagrangian finite difference method [SC96]. Therefore, it is not surprising that it may suffer from numerical instability. Sterling and Chen [SC96] have carried out an analysis of the stability for the lattice Boltzmann method using perturbations method. In general, the stability conditions are not available for the method. Zhou [Zho04], however, has indicated the LBM for shallow water equations is stable if some basic requirements are met:

First is the fluid resistance. It indicates that the kinematic viscosity ν should be positive. The following expression can be obtained:

$$\nu=\frac{e^2\Delta t}{6}\left(2\tau-1\right)>0 \tag{4.140}$$

Thus, the relaxation time needs to meet:

$$\tau > \frac{1}{2} \tag{4.141}$$

Secondly, as indicated by Zhou [Zho04] and Liu *et. al.* [LZB10], the Courant number should be smaller than unity. It implies that the magnitude of the resultant macro velocity is smaller than the lattice speed:

$$\frac{u_j u_j}{e^2} < 1 \tag{4.142}$$

and the wave velocity also should be smaller than the lattice speed:

$$\frac{gh}{e^2} < 1 \tag{4.143}$$

up to now, the LBM for shallow water equations is limited to subcritical shallow water flows, it means:

$$\frac{u_j u_j}{gh} < 1 \tag{4.144}$$

meanwhile, the Froude number should smaller than unit:

$$F_r = \frac{\sqrt{u_j u_j}}{\sqrt{gh}} < 1 \tag{4.145}$$

It should be noted that the above first three stability conditions can be easily satisfied by adjusting the relaxation time τ, the lattice size Δx and time step Δt. It has been tested that the lattice Boltzmann method is stable normally, if these four stability conditions can be satisfied [Zho04].

As of today, Chopard *et. al.* [CPL13] proposed an asymmetric lattice pattern in the one-dimensional lattice Boltzmann method to apply the model in a supercritical flow. This approach, however, cannot be expanded in two-dimensional model because of its symmetry. We, thus, have to satisfy the condition 4.145 in whole simulating domain.

4.10 Improved force term and treatment of bed slope

In the above method for the force term, it includes the calculation of the first order derivative related to the bed slope. To improve the efficiency and remove the calculation of the derivatives. High order scheme [Zho11, ZL13] has introduced the bed level into the lattice Boltzmann equation, directly. In this scheme, the lattice Boltzmann equation can be rewritten as:

$$f_\alpha\left(x + e_\alpha \Delta t, t + \Delta t\right) - f_\alpha\left(x, t\right) = -\frac{1}{\tau}\left(f_\alpha - f_\alpha^{eq}\right) - W_\alpha \frac{g\bar{h}}{C_s^2}\left[z_b\left(x + e_\alpha \Delta t\right) - z_b\left(x\right)\right] + W_\alpha \frac{\Delta t}{C_s^2} e_{\alpha j} F_j \tag{4.146}$$

in which $\bar{h} = \left[h\left(x + e_\alpha \Delta t, t\right) + h\left(x, t\right)\right]/2$ and W_α are the weights defined as:

$$W_\alpha = \begin{cases} 1/9 & (\alpha = 1, 3, 5, 7) \\ 1/36 & (\alpha = 2, 4, 6, 8) \end{cases} \tag{4.147}$$

Equation 4.146 are adopted in this thesis.

4.11 Turbulence modelling

In order to simulate flows with relatively high Reynolds number, the LBM for shallow water equations is extended to the shallow water equations with turbulence modelling which is proposed by [Zho04] in this section. The LBM for shallow water equations with turbulence modelling includes the eddy viscosity term which is not present in the original governing equation. It can be noted that the only difference is the viscosity term. Because the kinematic viscosity ν is determined only by the relaxation time with constant time and space step, this means that a new relaxation time τ_t can be defined by:

$$\tau_t = \tau + \tau_e \tag{4.148}$$

which yields a total viscosity ν_t:

$$\nu_t = \nu + \nu_e \tag{4.149}$$

and the lattice Boltzmann equation can be rewritten as:

$$f_\alpha\left(x + e_\alpha \Delta t, t + \Delta t\right) - f_\alpha\left(x, t\right) = -\frac{1}{\tau_t}\left(f_\alpha - f_\alpha^{eq}\right) + \frac{\Delta t}{6e^2} e_{\alpha i} F_i \tag{4.150}$$

which can produce the solution to the shallow water equations. This is consistent with the idea of the lattice Boltzmann model with subgrid-scale stress designed by [HSCD96]. Therefore, the flow turbulence can be predicted easily by the standard lattice Boltzmann equation with the total relaxation time τ_t.

For the purpose of determining the total relaxation time τ_t, the strain-rate tensor S_{ij} needs to be calculated. As S_{ij} defined by Equation 4.62 involves calculation of derivatives, it is not suitable or efficient to use in practice. To keep consistent with the lattice gas dynamics, S_{ij} is expected to be expressed in terms of the distribution function. Using the Chapman-Enskog expansion, it can be seen that the strain-rate tensor S_{ij} is related to the non-equilibrium momentum flux tensor and S_{ij} can be calculated by:

$$S_{ij} = -\frac{3}{2e^2 h \tau_t \Delta t} \sum_\alpha e_{\alpha i} e_{\alpha j}\left(f_\alpha - f_\alpha^{eq}\right) \tag{4.151}$$

Assuming ν_t and τ_t satisfy the definition of the macroscopic viscosity ν in the lattice Boltzmann method, the following expression can be obtained:

$$\tau_t = \frac{1}{2} + \frac{3\nu_t}{e^2 \Delta t} \tag{4.152}$$

substituting Equation 4.148 and 4.149 into the above equation results in:

$$\tau_e + \tau = \frac{1}{2} + \frac{3\left(\nu_e + \nu\right)}{e^2 \Delta t} \tag{4.153}$$

substituting Equation 4.58 into the above equation provides:

$$\tau_e = \frac{3}{e^2 \Delta t} \left(C_s l_s\right)^2 \sqrt{S_{ij} S_{ij}} \tag{4.154}$$

combining Equation 4.151 with the above equation gives:

$$\tau_e = \frac{3}{e^2 \Delta t} \left(C_s l_s\right)^2 \frac{3}{2e^2 h \tau_t \Delta t} \sqrt{\Pi_{ij} \Pi_{ij}} \tag{4.155}$$

in which

$$\Pi_{ij} = \sum_{\alpha} e_{\alpha i} e_{\alpha j} \left(f_\alpha - f_\alpha^{eq}\right) \tag{4.156}$$

with Equation 4.148, if $l_s = \Delta x$ is adopted, Equation 4.155 can be rewritten as:

$$\tau_e = \frac{9}{2} \frac{C_s^2}{e^2 h \left(\tau_e + \tau\right)} \sqrt{\Pi_{ij} \Pi_{ij}} \tag{4.157}$$

solving the above equation gives:

$$\tau_e = \frac{-\tau + \sqrt{\tau^2 + 18 C_s^2 / \left(e^2 h\right) \sqrt{\Pi_{ij} \Pi_{ij}}}}{2} \tag{4.158}$$

the total relaxation time τ_t can be by:

$$\tau_t = \tau + \tau_e = \frac{\tau + \sqrt{\tau^2 + 18 C_s^2 / \left(e^2 h\right) \sqrt{\Pi_{ij} \Pi_{ij}}}}{2} \tag{4.159}$$

4.12 The multiple-relaxation-time collision model in the D2Q9 square lattice

Compared with the BGK scheme, the multiple-relaxation-time (MRT) is less used in the lattice Boltzmann method for shallow water flows. In order to improve the stability of the approach, the collision operator of MRT-LBM is incorporated into the lattice Boltzmann method for shallow water equations in this section. This model will be used in the latter chapter show its accuracy. Together with the new force term introduced in the above section, the shallow water equations are recovered correctly in this section.

The MRT-LBM can improve the stability of computation. It is incorporated into the lattice Boltzmann method for shallow water equations in this thesis and the shallow water equations will be recovered by the Chapman-Enskog expansion in the next section. When the D2Q9 model is adopted, the lattice Boltzmann equation with the MRT-LBM for shallow water equations are as follows:

$$f_\alpha\left(x+e_\alpha\Delta t,t+\Delta t\right)-f_\alpha\left(x,t\right)=-\left(\boldsymbol{T_r}^{-1}\boldsymbol{S}\right)_{\alpha i}\left(m_i-m_i^{eq}\right)+\Delta tF_\alpha \tag{4.160}$$

where $m=\boldsymbol{T_r}f$, and $\boldsymbol{S}$ is the relaxation matrix, $\boldsymbol{S}=\mathrm{diag}\left(s_0,s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8\right)$, $\boldsymbol{T_r}$ is the transform matrix defined in [LL00]:

$$\boldsymbol{T_r}=\begin{bmatrix}1&1&1&1&1&1&1&1&1\\-4&-1&2&-1&2&-1&2&-1&2\\4&-2&1&-2&1&-2&1&-2&1\\0&1&1&0&-1&-1&-1&0&1\\0&-2&1&0&-1&2&-1&0&1\\0&0&1&1&1&0&-1&-1&-1\\0&0&1&-2&1&0&-1&2&-1\\0&1&0&-1&0&1&0&-1&0\\0&0&1&0&-1&0&1&0&-1\end{bmatrix} \tag{4.161}$$

the relaxation parameters s_7 and s_8 is chosen according to fluid viscosity as same with single relaxation time τ, and the other parameters can be chosen freely during the range of $[0,2]$ for maximum stability. In practical application, these free parameters are little larger than 1 according to [LL00].

The equilibrium variables of moments m^{eq} are given as:

$$m^{eq}=\left(h,-4h+\frac{3gh^2}{e^2}+\frac{3h\left(u^2+v^2\right)}{e^2},4h-\frac{9gh^2}{2e^2}-\frac{3h\left(u^2+v^2\right)}{e^2},\frac{hu}{e},-\frac{hu}{e},\frac{hv}{e},-\frac{hv}{e},\frac{h\left(u^2-v^2\right)}{e^2},\frac{huv}{e^2}\right) \tag{4.162}$$

4.13 Recovery of the MRT-LBM for shallow water equations

The Chapman-Enskog expansion is used to recover the shallow water equations from the MRT-LBM model with improved force term. Assuming Δt is small and $\Delta t=\epsilon$, Equation 4.160 can be expressed as:

$$f_\alpha\left(x+e_\alpha\epsilon,t+\epsilon\right)-f_\alpha\left(x,t\right)=-\left(\boldsymbol{T_r}^{-1}\boldsymbol{S}\right)_{\alpha i}\left(m_i-m_i^{eq}\right)+\epsilon F_\alpha \tag{4.163}$$

taking a Taylor expansion to the first term on the left hand side of the above equation in time and space around point (x, t) leads to:

$$\epsilon \left(\partial_t + e_{\alpha j}\partial_j\right) f_\alpha + \frac{1}{2}\epsilon^2 \left(\partial_t + e_{\alpha j}\partial_j\right)^2 f_\alpha + O\left(\epsilon^2\right) = -\left(\boldsymbol{T_r}^{-1}\boldsymbol{S}\right)_{\alpha i}\left(m_i - m_i^{eq}\right) + \epsilon F_\alpha \tag{4.164}$$

according to the Chapman-Enskog expansion, f_α can be written in a series of ϵ:

$$f_\alpha = f_\alpha^{(0)} + \epsilon f_\alpha^{(1)} + \epsilon^2 f_\alpha^{(2)} + O\left(\epsilon^2\right) \tag{4.165}$$

which can be expressed in a vector form:

$$\boldsymbol{f} = \boldsymbol{f}^{(0)} + \epsilon \boldsymbol{f}^{(1)} + \epsilon^2 \boldsymbol{f}^{(2)} + O\left(\epsilon^2\right) \tag{4.166}$$

the above equation can be easily converted into an expression in moment space by being multiplied with $\boldsymbol{T}$:

$$\boldsymbol{m} = \boldsymbol{m}^{(0)} + \epsilon \boldsymbol{m}^{(1)} + \epsilon^2 \boldsymbol{m}^{(2)} + O\left(\epsilon^2\right) \tag{4.167}$$

when the centered scheme [Zho04] is used, the expression for F_α can be obtained:

$$F_\alpha = F_\alpha\left(x + \frac{1}{2}e_\alpha\epsilon, t + \frac{1}{2}\epsilon\right) \tag{4.168}$$

making the taylor expansion to the above yields:

$$F_\alpha\left(x + \frac{1}{2}e_\alpha\epsilon, t + \frac{1}{2}\epsilon\right) = F_\alpha\left(x, t\right) + \frac{1}{2}\epsilon\left(\partial_t + e_{\alpha j}\partial_j\right) F_\alpha\left(x, t\right) + O\left(\epsilon^2\right) \tag{4.169}$$

combining the above equations, the equations to orders ϵ^0, ϵ and ϵ^2 are given as:

$$m_i^{(0)} = m_i^{(eq)} \tag{4.170}$$

$$\left(\partial_t + e_{\alpha j}\partial_j\right) f_\alpha^{(0)} = -\left(\boldsymbol{T_r}^{-1}\boldsymbol{S}\right)_{\alpha i} m_i^{(1)} + F_\alpha \tag{4.171}$$

$$\left(\partial_t + e_{\alpha j}\partial_j\right) f_\alpha^{(1)} + \frac{1}{2}\left(\partial_t + e_{\alpha j}\partial_j\right)^2 f_\alpha^{(0)} = -\left(\boldsymbol{T_r}^{-1}\boldsymbol{S}\right) m_i^{(2)} + \frac{1}{2}\left(\partial_t + e_{\alpha j}\partial_j\right) F_\alpha \tag{4.172}$$

The above equations can be written with matrices and vectors:

$$\boldsymbol{m}^{(0)} = m^{(eq)} \tag{4.173}$$

$$\left(\partial_t \boldsymbol{I} + \boldsymbol{E}\right) \boldsymbol{T_r}^{-1}\boldsymbol{m}^{(0)} = -\boldsymbol{T_r}^{-1}\boldsymbol{S}\boldsymbol{m}^{(1)} + \boldsymbol{F} \tag{4.174}$$

$$\left(\partial_t \boldsymbol{I} + \boldsymbol{E}\right) \boldsymbol{T_r}^{-1}\boldsymbol{m}^{(1)} + \frac{1}{2}\boldsymbol{F_r}\left(\partial_t \boldsymbol{I} + \boldsymbol{E}\right)^2 \boldsymbol{T_r}^{-1}\boldsymbol{m}^{(0)} = -\boldsymbol{T_r}^{-1}\boldsymbol{S}\boldsymbol{m}^2 + \frac{1}{2}\left(\partial_t \boldsymbol{I} + \boldsymbol{E}\right)\boldsymbol{F} \tag{4.175}$$

where $\boldsymbol{I}$ is the identity matrix; $\boldsymbol{E}$ is a diagonal matrix as:

$$\boldsymbol{E} = \mathrm{diag}\left[e_{0j}\partial_j, e_{1j}\partial_j, e_{2j}\partial_j, e_{3j}\partial_j, e_{4j}\partial_j, e_{5j}\partial_j, e_{6j}\partial_j, e_{7j}\partial_j, e_{8j}\partial_j\right] \tag{4.176}$$

$$\boldsymbol{m}^{(1)} = \left[0, e^{(1)}, e^{2(1)}, 0, q_x^{(1)}, 0, q_y^{(1)}, p_{xx}^{(1)}, p_{xy}^{(1)},\right]^{\mathrm{T}} \tag{4.177}$$

$$\boldsymbol{m}^{(2)} = \left[0, e^{(2)}, e^{2(2)}, 0, q_x^{(2)}, 0, q_y^{(2)}, p_{xx}^{(2)}, p_{xy}^{(2)},\right]^{\mathrm{T}} \tag{4.178}$$

and

$$\boldsymbol{F} = \left[0, 3w_1\frac{e_1F_i}{e^2}, 3w_2\frac{e_2F_i}{e^2}, 3w_3\frac{e_3F_i}{e^2}, 3w_4\frac{e_4F_i}{e^2}, 3w_5\frac{e_5F_i}{e^2}, 3w_6\frac{e_6F_i}{e^2}, 3w_7\frac{e_7F_i}{e^2}, 3w_8\frac{e_8F_i}{e^2},\right]^{\mathrm{T}} \tag{4.179}$$

substituting Equation 4.174 into Equation 4.175 leads to:

$$\left(\partial_t\boldsymbol{I} + \boldsymbol{E}\right)\boldsymbol{T_r}^{-1}\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{S}\right)\boldsymbol{m}^{(1)} = -\boldsymbol{T_r}^{-1}\boldsymbol{S}\boldsymbol{m}^{(2)} \tag{4.180}$$

multiplying Equation 4.174 and the above equation by $\boldsymbol{T_r}$ gives:

$$\left(\partial_t\boldsymbol{I} + \boldsymbol{T_r}\boldsymbol{E}\boldsymbol{T_r}^{-1}\right)\boldsymbol{m}^{(0)} = -\boldsymbol{S}\boldsymbol{m}^{(1)} + \boldsymbol{T_r} \tag{4.181}$$

$$\left(\partial_t\boldsymbol{I} + \boldsymbol{T_r}\boldsymbol{E}\boldsymbol{T_r}^{-1}\right)\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{S}\right)\boldsymbol{m}^{(1)} = -\boldsymbol{S}\boldsymbol{m}^{(2)} \tag{4.182}$$

writing our Equation 4.181 for $s = 0, 1, 3, 5, 7$ and 8 yields:

$$\partial_t h + \partial_x(hu) + \partial_y(hv) = 0 \tag{4.183}$$

$$\partial_t\left(-4h + \frac{3gh^2}{e^2} + \frac{3h\left(u^2+v^2\right)}{e^2}\right) = -s_1e^{(1)} \tag{4.184}$$

$$\partial_t\left(\frac{hu}{e}\right) + \partial_x\left(\frac{gh^2}{2e} + \frac{hu^2}{e}\right) + \partial_y\left(\frac{huv}{e}\right) = \frac{F_x}{e} \tag{4.185}$$

$$\partial_t\left(\frac{hv}{e}\right) + \partial_y\left(\frac{gh^2}{2e} + \frac{hu^2}{e}\right) + \partial_x\left(\frac{huv}{e}\right) = \frac{F_y}{e} \tag{4.186}$$

$$\partial_t\left(\frac{h\left(u^2-v^2\right)}{e^2}\right) + \frac{2}{3}\partial_x(hu) - \frac{2}{3}\partial_y(hv) = -s_7p_{xx}^{(1)} \tag{4.187}$$

$$\partial_t\left(\frac{huv}{e}\right) + \frac{1}{3}\partial_x(hv) + \frac{1}{3}\partial_y(hu) = -s_8p_{xy}^{(1)} \tag{4.188}$$

writing out Equation 4.182 for the conserved moments, $s = 3$ and 5 results in the follow equations:

$$\partial_x\left[\frac{1}{6}\left(1 - \frac{s_1}{2}\right)e^{(1)} + \frac{1}{2}\left(1 - \frac{s_7}{2}\right)p_{xx}^{(1)}\right] + \partial_y\left[\left(1 - \frac{s_8}{2}\right)p_{xy}^{(1)}\right] = 0 \tag{4.189}$$

$$\partial_y \left[\frac{1}{6}\left(1 - \frac{s_1}{2}\right) e^{(1)} + \frac{1}{2}\left(1 - \frac{s_7}{2}\right) p_{xx}^{(1)} \right] + \partial_x \left[\left(1 - \frac{s_8}{2}\right) p_{xy}^{(1)} \right] = 0 \tag{4.190}$$

combining the 0-th order Equations 4.185 and 4.186 with the 1-st order equations Equations 4.189 and 4.190 respectively:

$$\begin{aligned} \partial_t (hu) + \partial_x \left(\frac{gh^2}{2} + hu^2 \right) + \partial_y (huv) = F_x - \epsilon \frac{e^2}{6} \left(1 - \frac{s_1}{2}\right) \partial_x \left(e^{(1)} \right) \\ + \epsilon \frac{e^2}{6} \left(1 - \frac{s_7}{2}\right) \partial_x \left(p_{xx}^{(1)} \right) - \epsilon e^2 \left(1 - \frac{s_8}{2}\right) \partial_y \left(p_{xy}^{(1)} \right) \end{aligned} \tag{4.191}$$

$$\begin{aligned} \partial_t (hv) + \partial_y \left(\frac{gh^2}{2} + hv^2 \right) + \partial_x (huv) = F_y - \epsilon \frac{e^2}{6} \left(1 - \frac{s_1}{2}\right) \partial_y \left(e^{(1)} \right) \\ + \epsilon \frac{e^2}{6} \left(1 - \frac{s_7}{2}\right) \partial_y \left(p_{xx}^{(1)} \right) - \epsilon e^2 \left(1 - \frac{s_8}{2}\right) \partial_x \left(p_{xy}^{(1)} \right) \end{aligned} \tag{4.192}$$

ignoring the higher order terms, the following expansion from Equations 4.184, 4.187 and 4.188 can be obtained:

$$e^{(1)} = 0 \tag{4.193}$$

$$p_{xx}^{(1)} = -\frac{2}{3s_7} \left[\partial_x (hu) - \partial_y (hv) \right] \tag{4.194}$$

$$p_{xy}^{(1)} = -\frac{2}{3s_8} \left[\partial_x (hv) - \partial_y (hu) \right] \tag{4.195}$$

substituting the above equations into Equations 4.191 gives:

$$\begin{aligned} \partial_t (hu) + \partial_x \left(\frac{gh^2}{2} + hu^2 \right) + \partial_y (huv) = F_x \\ + \epsilon \frac{e^2}{3} \left(\frac{1}{s_7} - \frac{1}{2} \right) \partial_x \left[\partial_x (hu) - \partial_y (hv) \right] + \epsilon \frac{e^2}{3} \left(\frac{1}{s_8} - \frac{1}{2} \right) \partial_y \left[\partial_x (hv) + \partial_y (hu) \right] \end{aligned} \tag{4.196}$$

when setting $s_7 = s_8 = 1/\tau$, in which τ is the single relaxation time, the above equation can be reduced to:

$$\partial_t (hu) + \partial_x \left(\frac{gh^2}{2} + hu^2 \right) + \partial_y (huv) = F_x + \nu \left[\partial_x^2 (hu) + \partial_y^2 (hu) \right] \tag{4.197}$$

$$\partial_t (hv) + \partial_y \left(\frac{gh^2}{2} + hv^2 \right) + \partial_x (huv) = F_y + \nu \left[\partial_x^2 (hv) + \partial_y^2 (hv) \right] \tag{4.198}$$

the above equations and Equation 4.183 are just the shallow water equations. It may be noted that when $s_0, \ldots, s_8 = 1/\tau$, the lattice Boltzmann method with a single relaxation time τ is recovered.

4.14 Boundary and initial conditions

4.14.1 No-slip boundary condition

One of the most attractive advantages of the lattice Boltzmann method is its simple treatment of boundary conditions such as the bounce-back scheme. It can be implemented easily for flows in arbitrary complex geometries. The bounce-back and similar schemes for different boundary conditions, it very simple and efficient for no-slip, semi-slip and slip boundary conditions and is used widely for various flow problems. For example, in the bounce-back scheme as shown in Figure 4.3, the part below AB is a wall and defined as solid; the part above is fluid. The unknown incoming distribution functions f_2, f_3 and f_4 from the solid are simply equal to f_6, f_7 and f_8, respectively. Consequently, the specific location of solid points is not required and the programming is easy, making it the most efficient treatment for a no-slip boundary condition for flows in complex geometry [Zho04, CD98, GNGB97].

4.14.2 Slip boundary condition

If the boundary is smooth with lattice friction, the slip boundary should be used. As Figure 4.3 shows, the unknown distribution functions f_2, f_3 and f_4:

$$f_2 = f_8, \quad f_3 = f_7, \quad f_4 = f_6 \tag{4.199}$$

It means that no momentum is changed in the direction normal to a wall and the velocity along the wall is kept.

4.14.3 Semi-slip boundary conditions

In practical flows, a large velocity gradient exists near the boundary for turbulent flows due to the effect of wall friction. It cannot be described by slip or no-slip boundary conditions, and hence the semi-slip boundary is described to deal with this case. In order to construct the semi-slip boundary condition, the wall shear stress should be included. The wall shear stress τ_{fi} can be expressed by [Zho04]:

$$\tau_{fi} = \rho C_f u_i \sqrt{u_j u_j} \tag{4.200}$$

in which, C_f is the friction factor at the wall and can be constant or determined by $C_f = g\frac{n_f^2}{h^{1/3}}$ with n_f is the Manning's coefficient at the wall. By adding the wall shear stress to the force term, the semi-slip boundary condition is developed together with slip boundary. It should be noted that the distribution function is still dealt with by the slip boundary condition.

4.14.4 Initial condition

Before computation, the initial for distribution function needs to be specified. Generally, there are two ways to specify the initial condition in the lattice Boltzmann method. One is to specify a random value between 0 and 1 for the distribution function. The other is to assume a flow field with water depth and velocity firstly, then compute the local equilibrium distribution function f_α^{eq} and use it as an initial condition for f_α. Normally, it is easier to specify a macroscopic quantity than a macroscopic. So, the second method is better in practical computation, which is used in the present models. It is obvious that there is no difference between results calculated with these two initial conditions for a steady flow problem.

4.15 Implementation for tsunami modellings

4.15.1 Moving shoreline algorithm

Solutions of the non-linear shallow water equations are only valid for a water depth h greater than 0. For a moving shoreline, however, there will be areas of tsunami wave runup-rundown where parts of the computational domain will successively be wetting and drying. The numerical treatment of such drying or wetting boundaries requires additional considerations. As of today, a lot of moving shoreline boundary condition are proposed for tsunami modellings. Hibberd and Peregrine [HP79] developed a computational model based on the conventional non-linear shallow water equations for wave run-up problems. Liu *et. al.* [LCB$^+$95] presented an experimental and numerical study of the interactions of solitary waves climbing up a circular island. Synolakis [Syn87] and Briggs *et. al.* [BSHG95] experimentally studied long-wave evolution and run-up on piecewise linear bathymetries. Li and Raichlen [LR02] investigated solitary wave run-up using the weighted essentially non-oscillatory (WENO) shock capturing scheme [LOC94]. Carrier *et. al.* [CG58]. used a semi-analytic solution technique to evaluate tsunami run-up and draw-down motions on a uniformly sloping beach based on non-linear shallow water wave theory.

In the lattice Boltzmann method for non-linear shallow water equations, a moving shoreline algorithm is used to artificially adjust the domain size in the wave runup-rundown region, depending on the rising or falling water level at the boundary nodes. In this simple step-slot algorithm, the water level in the neighbour of solid boundaries is checked via evaluating:

$$z_b\left(\boldsymbol{x}\right) + h\left(\boldsymbol{x}\right) > z_b\left(\boldsymbol{x} + \boldsymbol{e}_i \Delta t\right) + \delta, \quad \forall i \tag{4.201}$$

where z_b denotes the elevation of bottom grid cells, h is the total water depth, and δ is a proper threshold value avoiding a divide-by-zero. We set $\delta = 1.0 \times 10^{-5}$ in this thesis. If the water is overflowing a given cell, the neighbouring inactive solid cell is activated and converted to a wetting cell. The equilibrium distribution functions for an extrapolated water elevation and velocity are defined as:

$$f_i = f_i^{eq}\left(h_{\text{init}}, \boldsymbol{u}_{\text{init}}\right) \tag{4.202}$$

This approach leads to accurate results for tsunami modellings. For more complex bottom topographies, however, more advanced moving shoreline algorithm can be used. Balzano [Bal98] analysed and discussed ten different moving shoreline algorithms, mainly for implicit methods, and compared these for demanding run-up test cases. Lynett *et. al.* [LWL02] proposed used a linear extrapolation scheme, which is a pure mathematical manipulation to deal with the wetting-drying front. Frandsen [Fra08] incorporated both the linear extrapolation scheme and the thin water film into the one-dimensional lattice Boltzmann shallow water model to investigate the behaviour of free-surface waves. It appears from these references that the calculation of h_{init} and $\boldsymbol{u}_{\text{init}}$ is a challenging problem. In a very simple model, h_{init} is set by the linear extrapolation approach from the neighbouring cells as shown in Figure 4.4, and the initial velocities are set to zero (with $\boldsymbol{u}_{\text{init}} = 0$). This constant extrapolation of water levels works well for most test cases, but might lead to instabilities in some cases, due to inconsistent initialisation of velocities. In such cases, a linear extrapolation might improve the run-up, but based on the stability conditions, Froude number stability limitations must also be considered. Specifically, the lattice Boltzmann method for the non-linear shallow water equations is only applicable to subcritical flows, so that an upper bound or similar technique has to be satisfied during run-up, to limit the Froude number value (i.e., such that $|\boldsymbol{u}_{\text{init}}| \leq \sqrt{gh_{\text{init}}}$) and avoid inconsistent node initialisations. It was observed that the present run-up model is very sensitive to such limitations, which may lead to instabilities, especially for low fluid viscosities ν. Specifically, an improper choice of h_{init} induces shock waves that can lead to instabilities. Similar effects have been reported in the Navier-Stokes type lattice Boltzmann simulations of fluid-structure interaction problems, where nodes have to be initialised with proper values for fluid pressure and velocity [GKT$^+$06]. So far, to tackle this problem, we used increased values for the fluid viscosity.

4.15.2 Multiple spacing grid coupled model

There are several reasons for applying the nested multiple spacing grid model for tsunami modellings. When the water depth varies within the computational domain, it might be desirable that different spacing grid size be employed in different subregions so that the frequency dispersion is adequately represented. On the other hand, we would like to obtain detailed information in the coastal region. Finer grids should be used only in certain specific regions. We will briefly describe the technique for two-way exchanging information between two subregions of different grid sizes. As shown in Figure 4.5, a smaller grid system is nested in a larger grid system with the ratio of 1:3.

To interpolate depth averaged velocity $\boldsymbol{u}$ and total water depth h in a small subregion, we used first order interpolation scheme in this thesis. On the other hand, we calculated the ensemble average of the macroscopic variables from 9 small grids in the large size grid for connecting a small subregion to large region. The distribution functions f_i in each region are calculated by the equilibrium distribution functions f_i^{eq} by the interpolated macroscopic variables.

4.15.3 The calculation procedure

The solution procedure for the lattice Boltzmann method for shallow water equations are extremely simple. It involves only explicit calculations and consists of the following step:

Algorithm 4 Calculation loop of two-dimensional shallow water flows

```
collision with the BGK-LBM or MRT-LBM
forcing bed slope and friction term
streaming
apply the boundary conditions for unknown distribution functions
update the total water depth $h$ and depth averaged velocity $\boldsymbol{u}$
```

4.16 Primitive benchmarks

4.16.1 Simple dam-breaking flows

A typical dam-break flow problem was considered to compare the specific model features between the MRT and BGK collision models when combined with the large eddy simulation model. Figure 4.6 shows the flow geometry and the initial water depth. The 1st-order mirror scheme, which represents slip boundary conditions, was used at the solid-wall boundary. The numerical results were examined in terms of the spatial distribution of the water depth and velocity compared with the analytical solution [Sto57] after 1 second.

The various calculation cases, with different grid sizes and parameters, are summarised in Table 4.1. The macroscopic fluid viscosity was set to $\nu = 1.0 \times 10^{-6}\mathrm{m}^2/\mathrm{s}$. This is a realistic value for water at 20°C. Then, the relaxation time was calculated using Equation 4.140. The time step was chosen to fulfill the requirement that the Courant number e should be much larger than the wave velocity $\sqrt{gh_{\max}}$. We chose a value of $e = 10\mathrm{m/s}$ for a wave velocity of 3.13m/s.

Figure 4.7 presents the water depth profiles for Case 1 to 3. The results of the MRT collision model are in good agreement with the analytical solution, whereas the results of the BGK model show numerical oscillations. There is a possibility of reducing this effect even when using the BGK model. It is, however, assumed that the solution quality will degrade because of the viscosity effect.

As seen from the above numerical experiments, the large eddy simulation model based on the MRT collision model achieves results that are essentially identical the analytical solution when we set 100 computation points per unit length in the dam-break flow problem.

4.16.2 Flows around a square cylinder

This benchmark test was performed to validate the reproducibility of the vortex separation behind a square cylinder in a high-Reynolds-number flow. The calculation geometry, depicted in Figure 4.8, was prepared in reference to Lui *et al.* [LLS12]. The origin point was placed at the center of the square cylinder. We chose the length of the square cylinder to be $L = 1\text{m}$ as the characteristic length of the flow field. The x-axis velocity was initialised as the characteristic velocity $U = 0.535\text{m/s}$, whereas the y-axis velocity was set to zero following Lyn *et al.* [LWL02]. The initial still water depth h_0 was set to 4m. The macroscopic fluid viscosity was set to $\nu = 2.5 \times 10^{-5}\text{m}^2/\text{s}$ to obtain a flow with a Reynolds number of 21400. In this scenario, the Reynolds number is defined as

$$\text{Re} = \frac{UL}{\nu} \tag{4.203}$$

The mirror scheme was used at the top and bottom wall boundaries. The left boundary was initialised with the equilibrium distribution function calculated from the still water depth h_0 and the characteristic velocity U. At the right-hand boundary, we applied a non-reflecting boundary condition, which is also called an impedance boundary condition [Sch14]. A semi-slip boundary condition [Zho04] was applied at the boundary of the square cylinder to conduct a correct turbulence simulation with a Manning roughness coefficient of 0.006. The bed slope was defined as a plane with a roughness coefficient of 0.009. Table 4.2 summarises the calculation parameters for this test case. Four simulation conditions with different grid sizes were chosen to confirm the dependence of the simulated vortex behaviour on the spatial resolution. The total simulation time was 10 minutes in all cases.

Figures 4.9 and 4.10 show the velocity profile for each direction after 10 minutes of iteration for each component. Regular vortex appear behind the cylinder in Case1 and 2. On the other hand, it is clear that the vortex occurrence and movement are fairly irregular in high resolution. Moreover, Figure 4.11 shows the time series of velocities captured at $(x, y) = (10\text{m}, 0\text{m})$. It is seen that the velocity profile indicates not a steady or periodic flow but rather a complex and aperiodic turbulent flow. These results clearly differ from those of the previous study [LLS12]. It is speculated that the difference is due to the grid size. In other words, the previous results are equivalent to the solution of a zero-equation turbulence model because of the coarse grid size. This suggests that the shallow water large eddy simulation model based on the MRT collision model requires sufficient calculation points to resolve the turbulence vortexes, as do traditional computational fluid dynamics.

Figure 4.12 presents the time-averaged velocity profile for each calculation case. The trend of the numerical results is generally consistent with the experimental data. There is, however, a large quantitative discrepancy between the numerical and experimental values. We consider that it is difficult to evaluate this problem in terms of the time-averaged velocity because of the high irregularity of the flow in this test case, as mentioned above; therefore, an additional benchmark problem was considered for direct analysis of the velocity profile, as described in the next section.

4.16.3 Plane-driven cavity flows

A two-dimensional square-plane-driven cavity flow was investigated. This benchmark test has its origin in the two-dimensional lid-driven cavity flow problem [GGS82, ECG05, Wah12]. Figure 4.13 shows the computation domain and the boundary conditions for the test case. The initial still water depth h_0 was set to 1.0m [LLS12]. U is the top-driven velocity, which is the characteristic velocity in this flow. L is the vertical and horizontal length of the cavity, which is also the characteristic length. These values were set to $U_{\max} = 1.0$m/s and $L = 1.0$m, respectively, for all Reynolds numbers. The critical Reynolds number for this problem is between 8000 and 8050, with an error of less than 1%, in analogy to the previous study of Bruneau *et al.* [BS06]. Validation for the plane-driven cavity flow problem using the LBM for shallow water theory has previously been performed for Reynolds numbers of up to 10000 [LLS12]. We extended our numerical analysis up to a Reynolds number of 30000, as presented below. The Reynolds number, defined as $\mathrm{Re} = UL/\nu$, was set to 10000, 15000, 20000 or 30000 to simulate a turbulent flow. The large eddy simulation model should be applied for these Reynolds numbers to consider sufficient turbulence of the flow. The numerical results were assessed in terms of the streamlines and velocity profiles after 30 minutes of computation time, as it was considered that the flow would not be in a steady state, such as a laminar flow. The calculation conditions and parameters are summarised in Table 4.3.

The bounce-back scheme was applied for the edge boundary condition. The distribution function at the moving lid boundary was determined by the equilibrium distribution functions calculated from the characteristic velocity U and the still water depth h_0. The y-axis velocity was constrained to always be equal to zero at the top boundary.

To quantitatively evaluate the model accuracy, Figure 4.14 present comparisons of the profiles of the dimensionless calculated velocity components u_x and u_y along a vertical and a horizontal line at $y/L = 1/2$ and $x/L = 1/2$, respectively, consistent with the previous study. The results of the MRT collision model are in good agreement with the previous study's numerical solution. It is also concluded that the bounce-back boundary condition in the MRT collision model reproduced the no-slip boundary condition with satisfactory accuracy, as seen from the fact that the accuracy does not decrease near the boundary. The reason is that the MRT collision rates can determine the accuracy of the boundary condition, in analogy to the conventional Navier-Stokes-type lattice Boltzmann equation with the MRT collision model; the rates s_5 and s_7 work especially well. Accordingly, it is considered that the MRT relaxation rates must also be validated in detail. We intend to focus on this issue in our future analysis.

The results of the MRT collision model better reproduce the reference data as the number of grid points increases, especially at the solid wall boundary. It can be concluded that the convergence behaviour of our turbulence model depends on the size of the computation grid. This is an important factor in determining the applicability of our model to other problems from an engineering point of view. Moreover, for the current problem, it is necessary to set at least 256 × 256 grid points to obtain results of comparable precision to those that have been previously reported. An adaptive mesh refinement technique will be necessary to execute more efficient simulations using the MRT collision model, for example, to refine the mesh near the solid wall boundary.

4.17 Verification with tsunami by the 2011 Great East Japan Earthquake

Finally, we simulated the tsunami by the 2011 Great East Japan Earthquake to verify our proposed tsunami simulation model. Figure 4.15 shows the computational domains. This tsunami simulation consists of five sub-regions by multiple spacing grid coupled model. The grid sizes include 405, 135, 45, 15, and 5m for sub-domains (a), (b), (c), (d) and (e) respectively. Sub-domain (e) is the inundation domain (Onagawa). Table 4.4 shows calculation parameters in this simulation. The relaxation times τ manually determined to avoid the numerical instability for each sub-region. This means that the values are the minimum number to enable the tsunami simulation stable.

The initial free surface elevation was generated by an analytical solution [Oka85], which is used to simulate ground deformation produced by, e.g., tectonic faults, such as earthquakes. For a given rectangular fault geometry, the free surface elevation can be computed. The equivalent roughness model [III13] was adopted to model the resistance of building structures during tsunami inundation. This approach uses the digital elevation model data and composite equivalent roughness coefficient that is given by Equation 4.204 according to land use and building conditions:

$$n_{\mathrm{erm}} = \sqrt{\frac{100-\theta}{100} n_0^2 + \frac{\theta}{100} \frac{C_D}{2gk} h^{4/3}} \tag{4.204}$$

where n_0 is the roughness coefficient according to MLIT [MLI09] ($n_0 = 0.025$ in urban areas: $n_0 = 0.030$ otherwise), θ is the relation of built-up area in the computational cell, C_D is the drag coefficient ($C_D = 0.5$ in this thesis), g is the gravitational acceleration, k is the horizontal scale of houses, and h is the total water depth.

The numerical results for the amplitude at ten different wave gauges [SFHN13] in sub-region1 are compared to numerical results by the conventional finite difference method (FDM) in Figure 4.16. All in all, the proposed model results reproduce the salient features of the tsunami-induced flow quite well. The time of the initial wave impact and the arrival of the reflected wave match the FDM results very well for all ten probes. Figure 4.17 also shows the amplitude at the gauge in Onagawa (Sub-region3). Both the proposed model and FDM is a very good agreement with the observed wave. A bidirectional coupling approach has successfully been implemented in the proposed model for the tsunami modellings.

The numerical results for the maximum inundation depth in Sub-region5 is shown in Figure 4.18. The result of proposed model reproduce the inundation depth calculated by the FDM well. The surge of wave front, however, is a little less than the FDM results. Attempts to fine-tune the LBM run-up model and/or further reduce the fluid viscosity (i.e., relaxation time τ), to obtain a higher elevation induced instabilities in the run-up region because of the its stability condition (Equation 4.145). This can potentially be improved by more sophisticated wetting-drying algorithms for the simulation of tsunami run-up.

4.18 Conclusion

In this chapter, we developed a two-dimensional tsunami simulation model based on the lattice Boltzmann method for the non-linear shallow water equations with the multiple-relaxation-time approach. Through the validation of the proposed model on traditional turbulence benchmark problems and realistic tsunami flows, the following remarkable results were obtained:

- A comparison between the MRT and BGK collision models revealed that the latter model well reproduces the analytical solution for a dam-breaking flow. We determined that it is necessary to set at least 100 lattice points per unit wavelength in this problem to reproduce the analytical solution using the proposed model.

- In benchmark tests based on the flow behind a square cylinder and a plane-driven cavity flow, we demonstrated that the convergence behavior of the proposed model depends on the size of the computation grid, as in the case of other computational fluid dynamics. Therefore, it can be used for other, more realistic flow problems in hydrodynamic engineering, such as coastal, lake, and river flows. In addition, it was shown that the results of the proposed model markedly depend on the size of the computation grid. If there are insufficient lattice points to resolve the vortexes, the model results may be equivalent to the solution of a zero-equation turbulence model because of the coarse grid size.

- We presented application of the proposed model with practical tsunami simulation by the 2011 Great East Japan Earthquake. The validation shows a good performance of the code and a good agreement of the numerical results by the conventional FDM approach with the reference data in most of the cases.

In addition, it is well known that the LBM requires many lattice points to solve high-Reynolds-number flows compared with conventional computational fluid dynamics. It is believed that adaptive mesh refinement will be needed to perform more efficient simulations using the LBM. In future work, several model aspects will be improved, particularly concerning wetting-drying models, higher order boundary conditions near the no-slip wall.

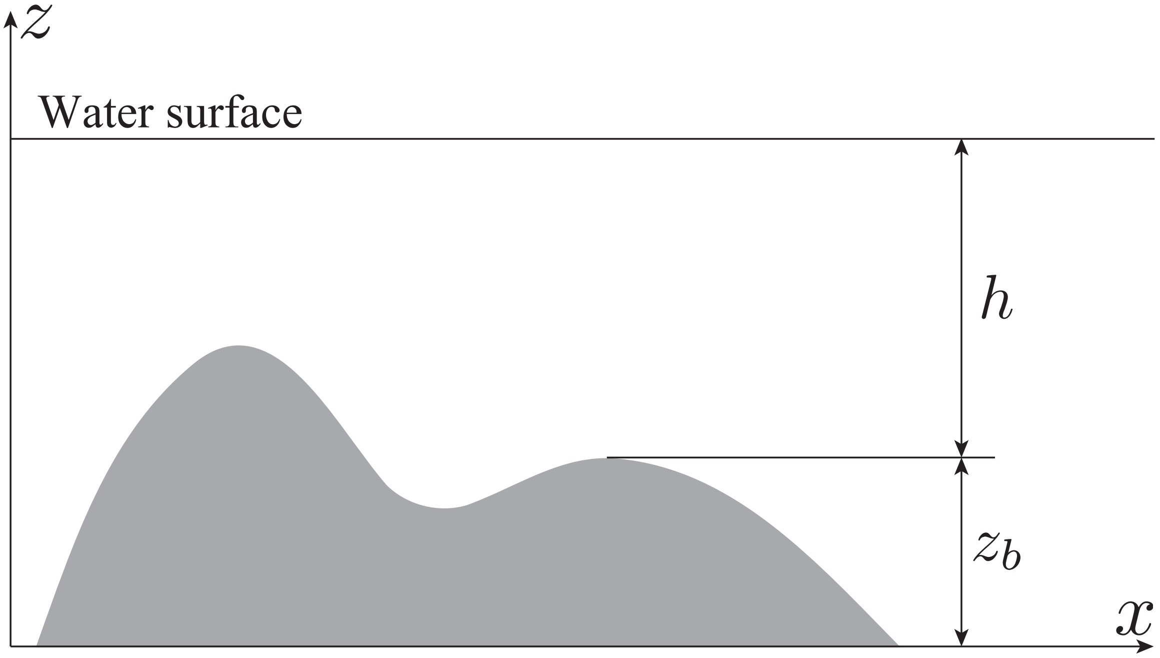


Figure 4.1: Shallow water flows sketch.

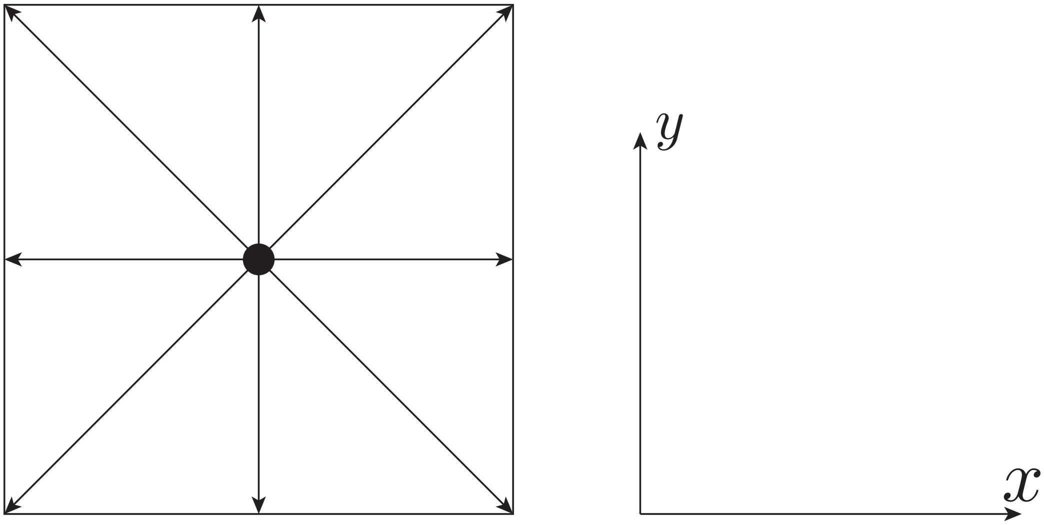


Figure 4.2: Two-dimensional 9-speed square lattice (D2Q9) in the horizontal plane.

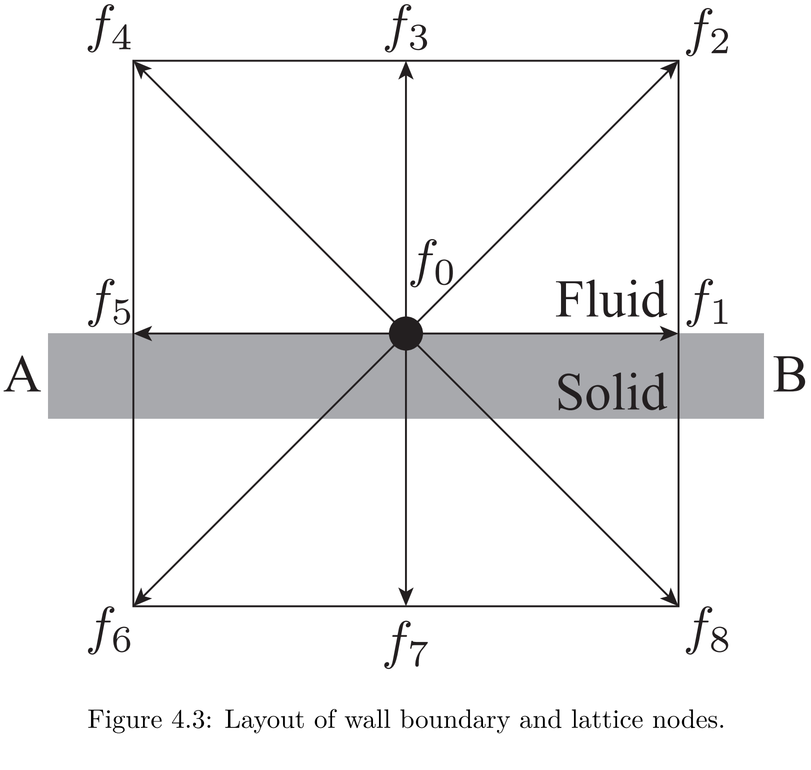


Figure 4.3: Layout of wall boundary and lattice nodes.

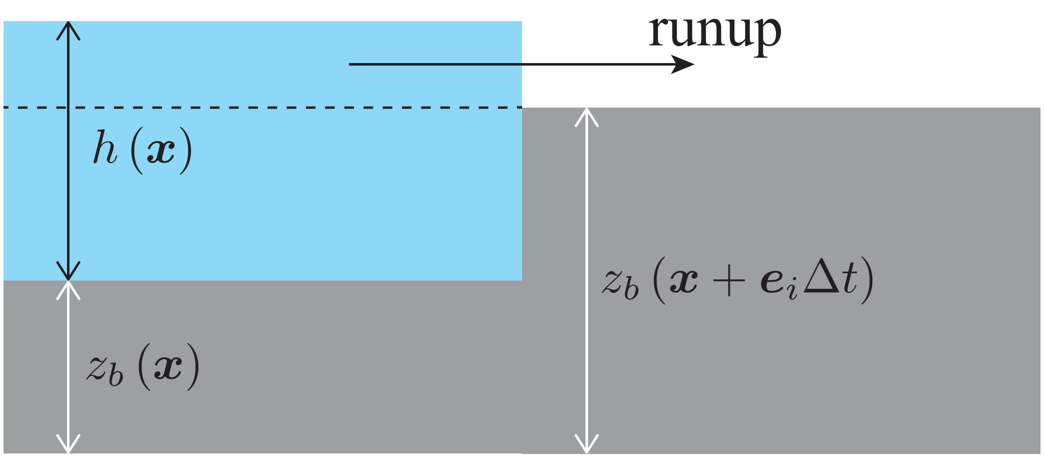

runup
$h\left(\boldsymbol{x}\right)$
$z_b\left(\boldsymbol{x}+\boldsymbol{e}_i\Delta t\right)$
$z_b\left(\boldsymbol{x}\right)$

Figure 4.4: Layout of moving shoreline algorithm in this thesis.

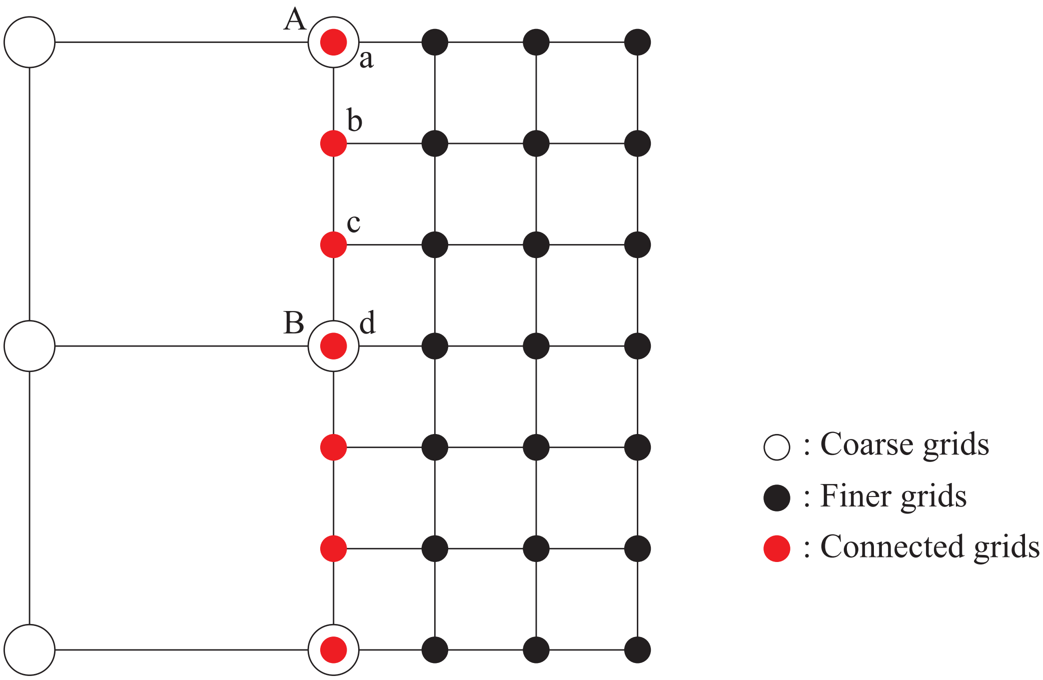


Figure 4.5: Layout of connecting algorithm for different spacing grid sizes.

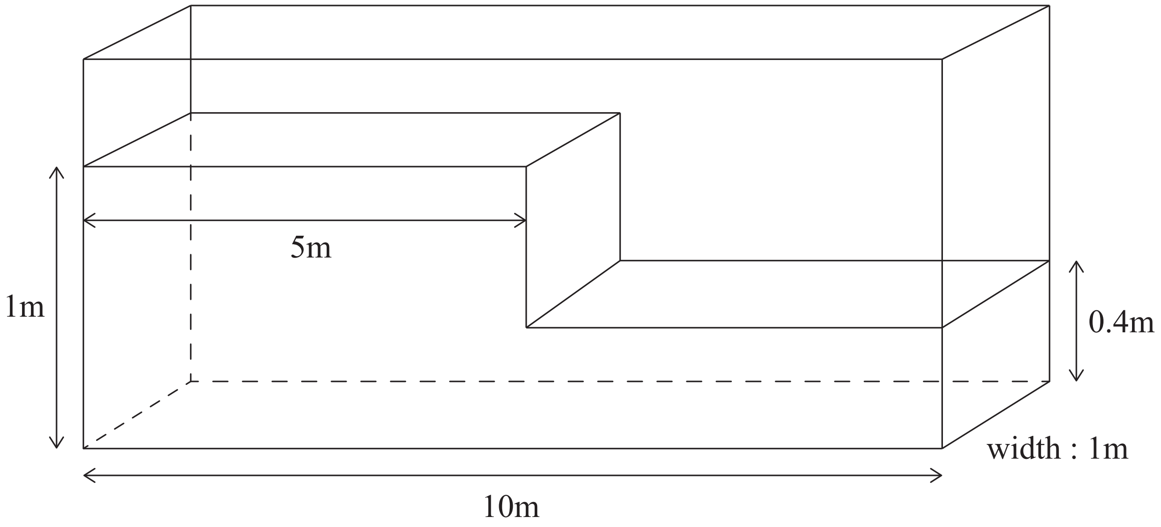


Figure 4.6: Geometry of the dam-breaking problem.

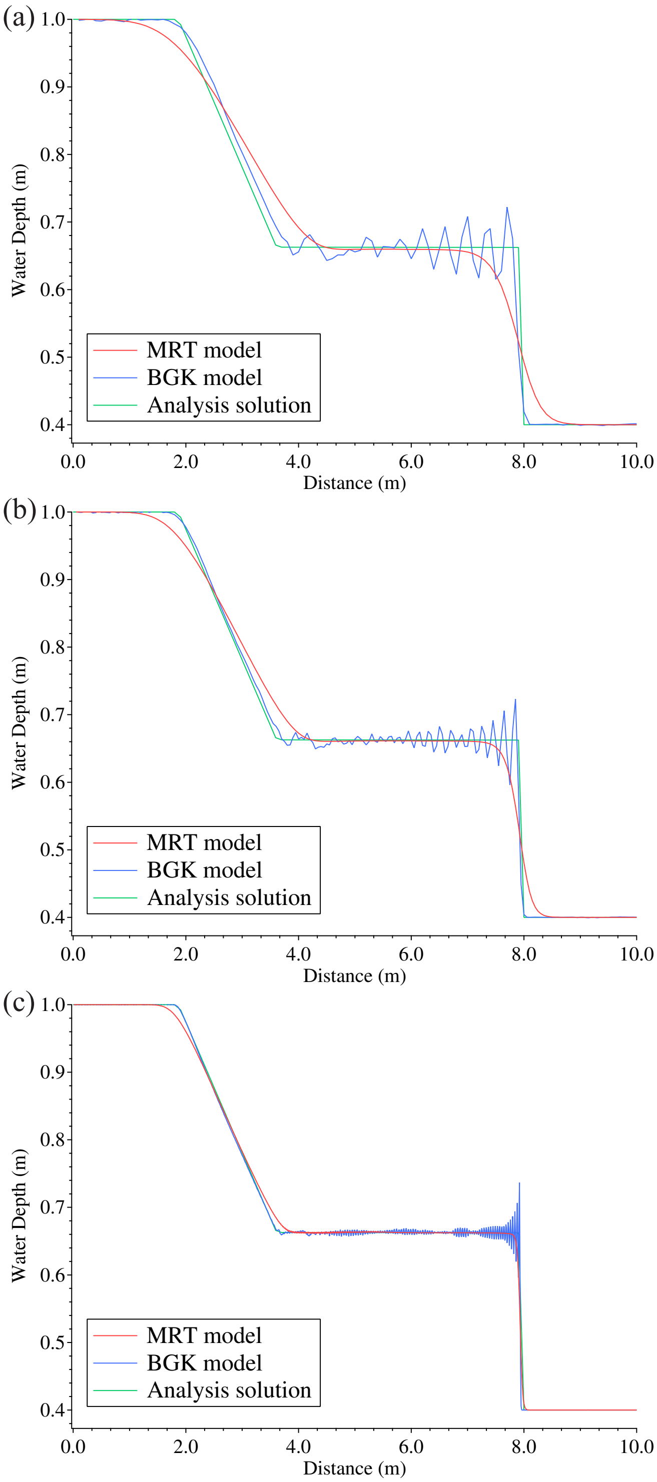


Figure 4.7: Comparison of the total water depth between the BGK and MRT models $t = 1.0$s: (a) Case1, (b) Case2, (c) Case3.

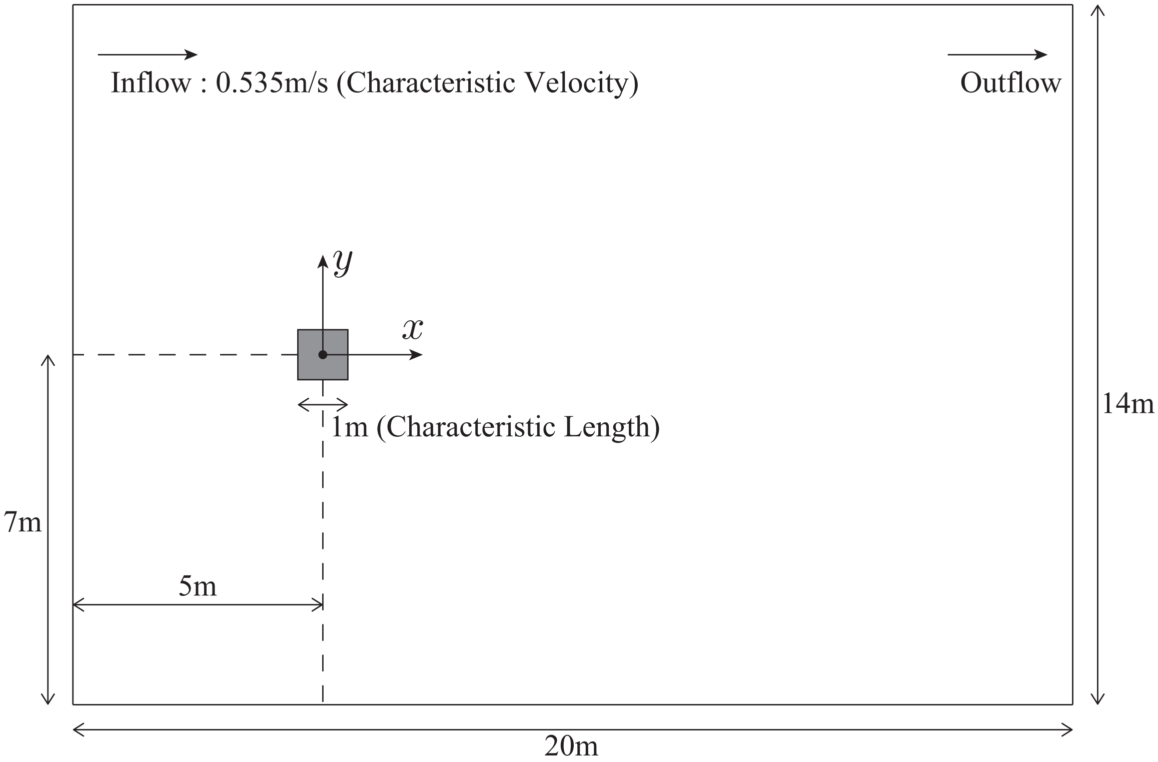

Inflow : 0.535m/s (Characteristic Velocity)
Outflow
y
x
1m (Characteristic Length)
14m
7m
5m
20m

Figure 4.8: Geometry of the square cylinder benchmark problem.

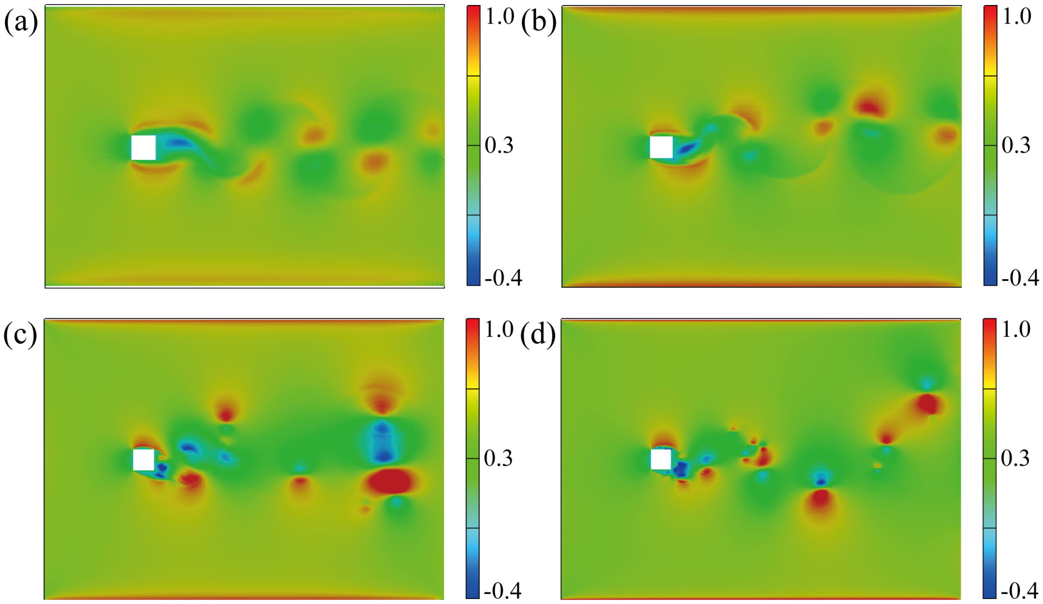


Figure 4.9: The x-axis velocity profiles after 10 minutes iterations: (a) Case1, (b) Case2, (c) Case3, (d) Case4.

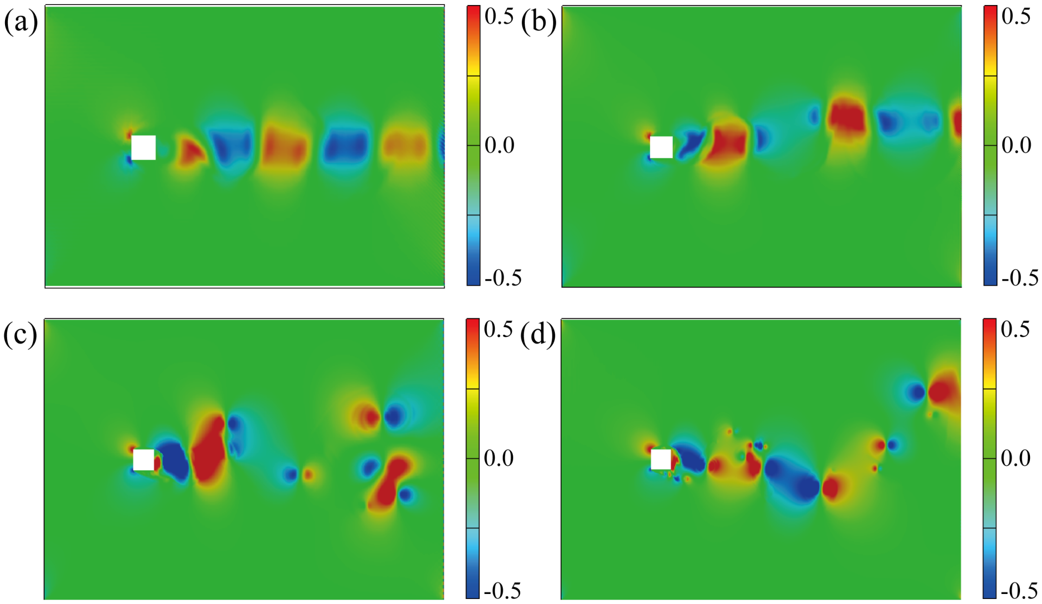


Figure 4.10: The y-axis velocity profiles after 10 minutes iterations: (a) Case1, (b) Case2, (c) Case3, (d) Case4.

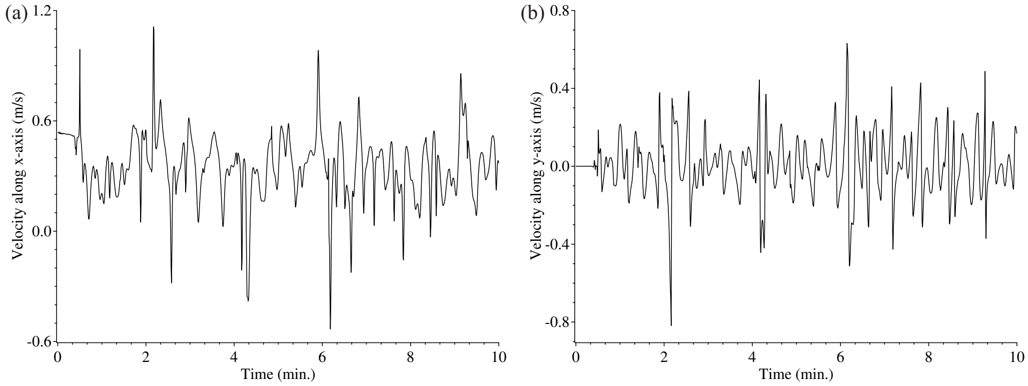


Figure 4.11: Timeseries of velocities along the x-axis at $(x, y) = (10.0\text{m}, 0.0\text{m})$, (a) velocity along x-axis, (b) velocity along y-axis in the calculation case 4.

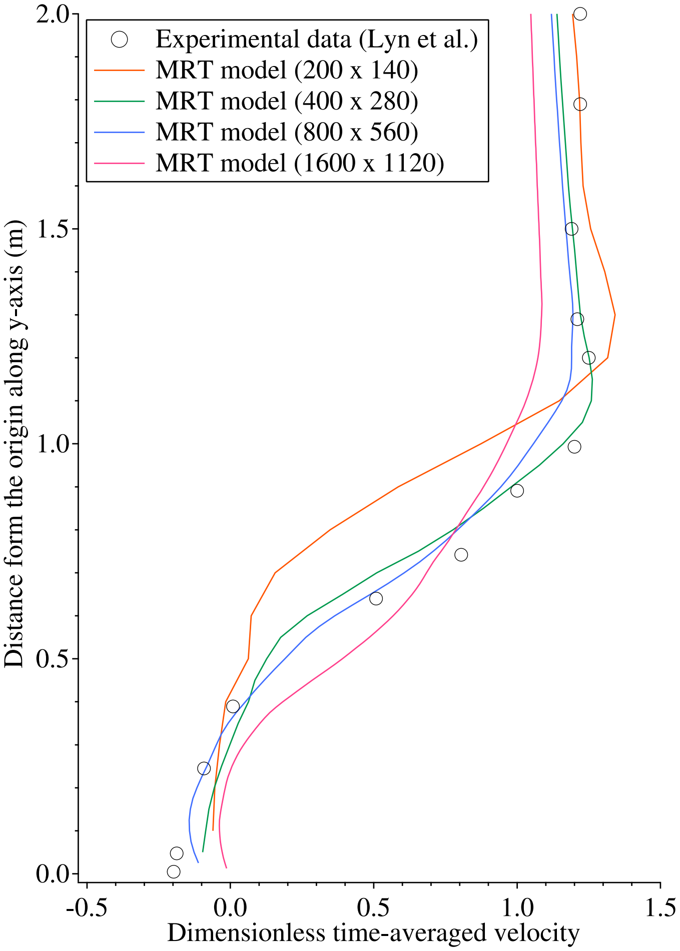


Figure 4.12: The time-averaged dimensionless velocity along the y-axis at $x = 1\text{m}$.

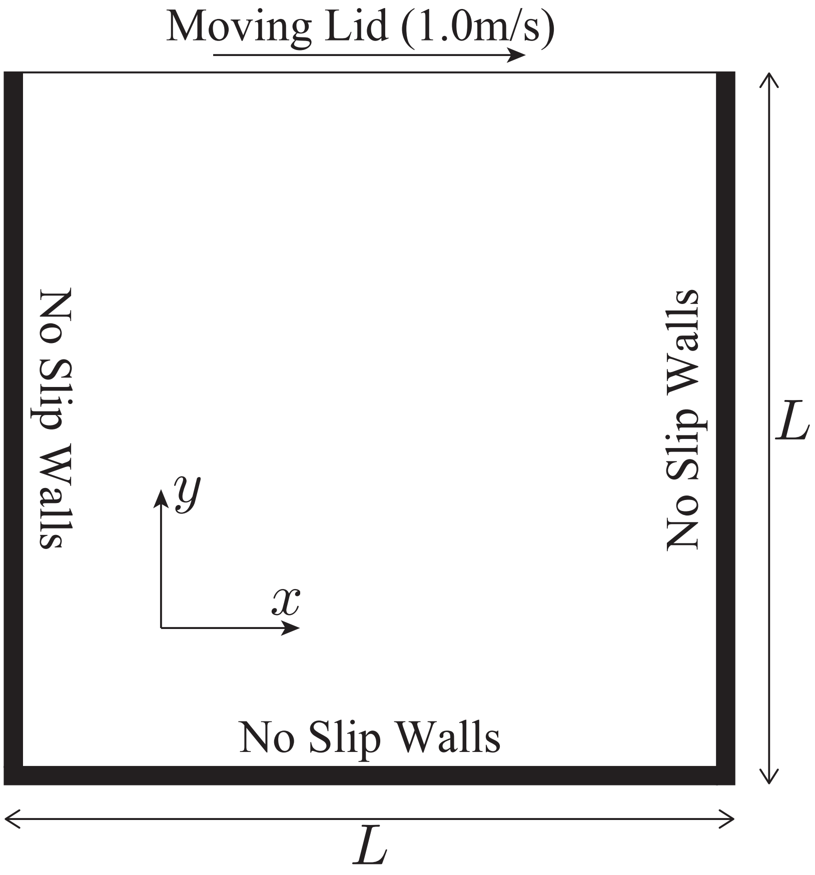


Figure 4.13: Geometry and boundary conditions of the plane-driven cavity flow.

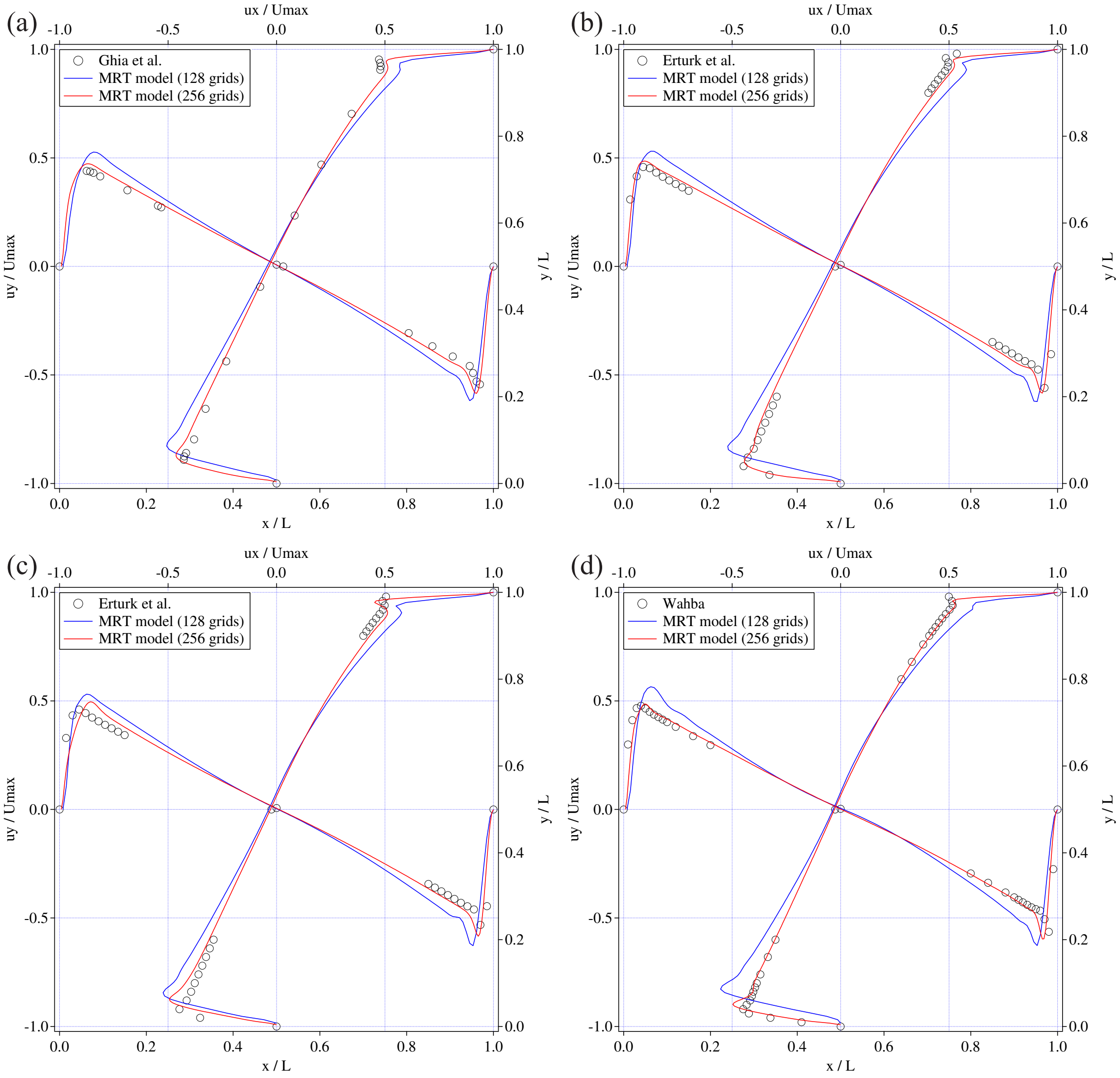


Figure 4.14: The velocity profiles along the centre line after 30 minutes iteration: (a) Re = 10000, (b) Re = 15000, (c) Re = 20000, (d) Re = 30000.

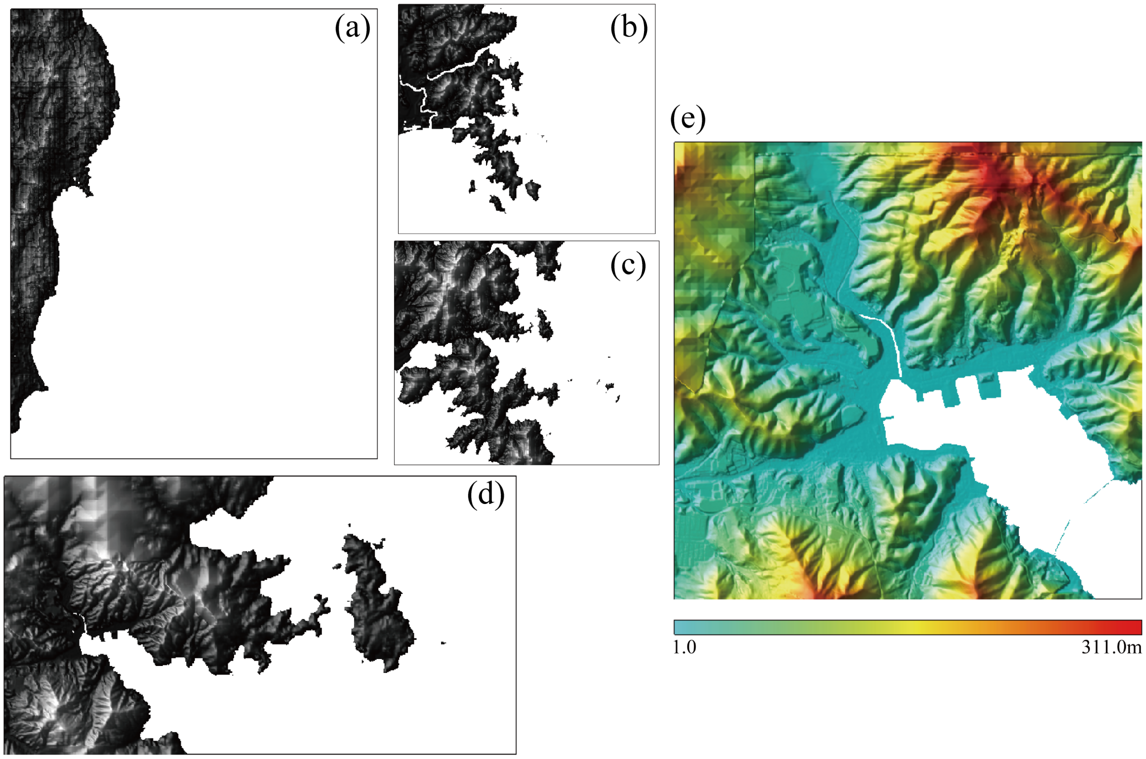


Figure 4.15: The calculation domains: (a) sub-region1, (b) sub-region2, (c) sub-region3, (d) sub-region4, (e) sub-region5. Sub-region5 is the inundation area (Onagawa).

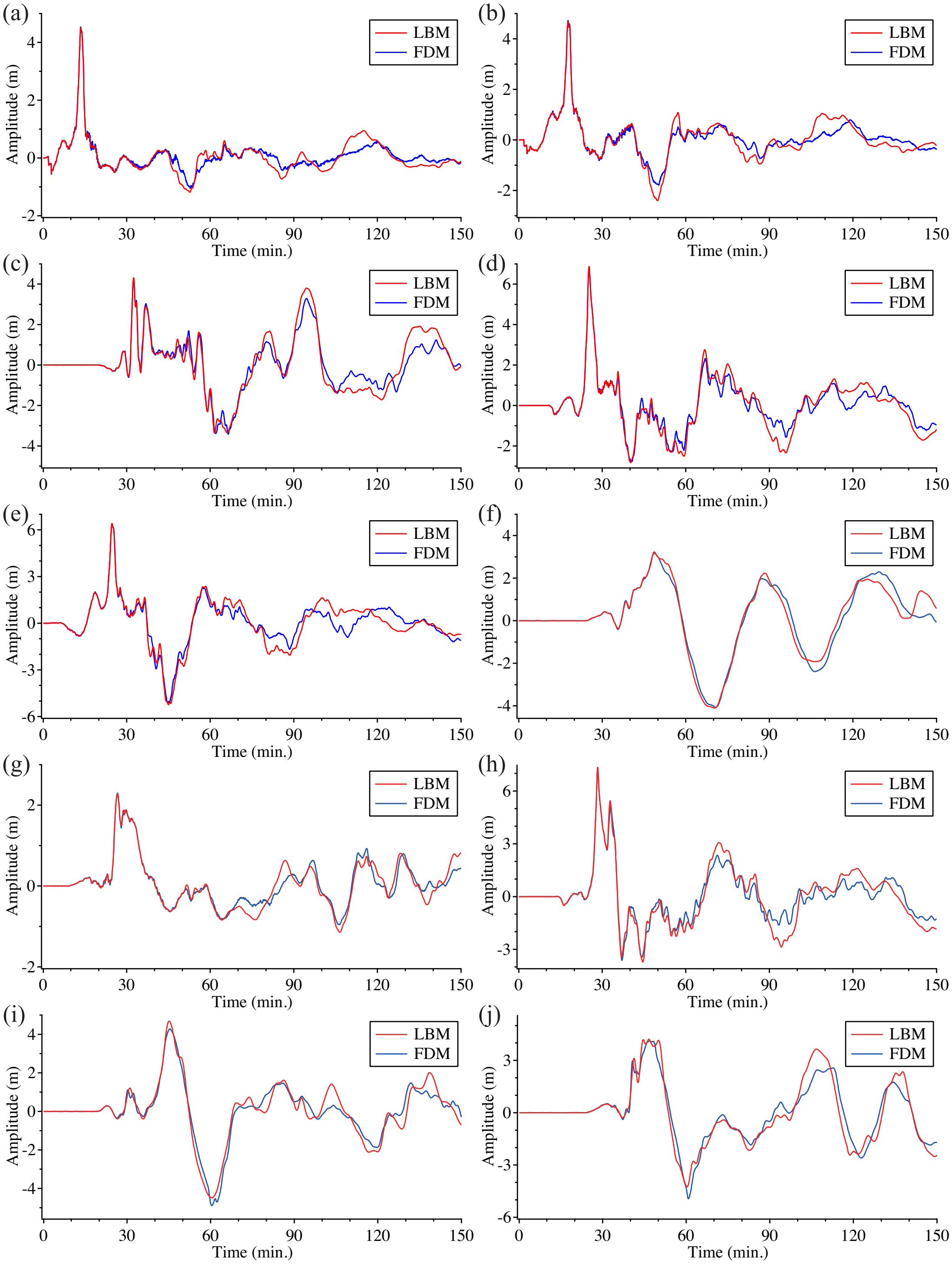


Figure 4.16: Comparisons of calculated tsunami waveforms between LBM and FDM in sub-region1: (a) TM-1, (b) TM-2, (c) Iwate N, (d) Iwate M, (e) Iwate S, (f) Miyagi M, (g) Mutsuogawara Port, (h) Kamaishi, (i) Sendai NewPort, (j) Choshi.

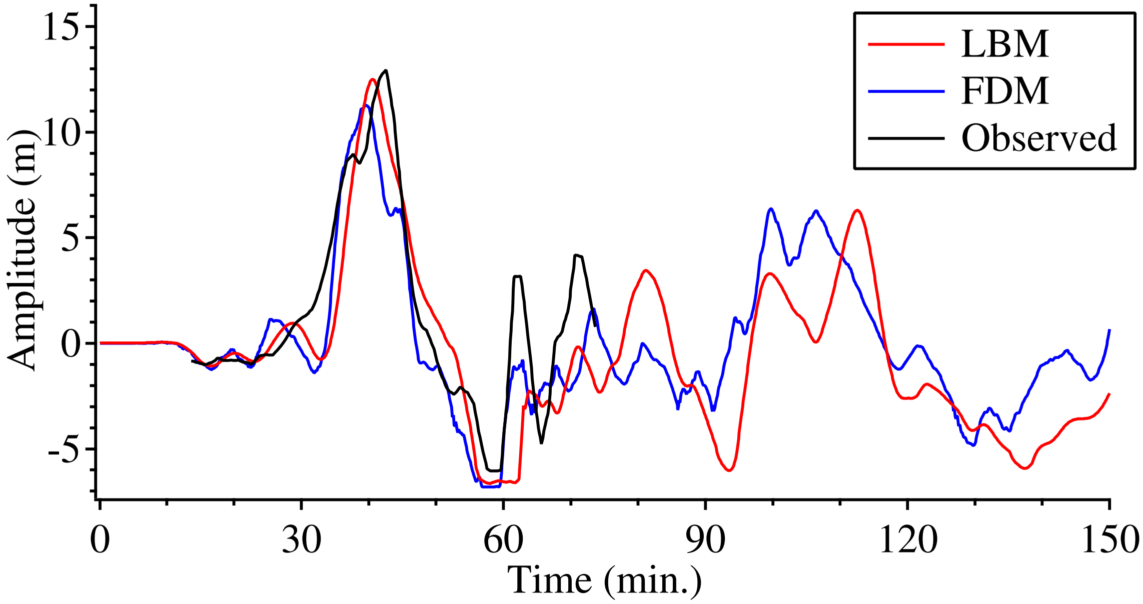


Figure 4.17: The observed and calculated tsunami waveforms in sub-region3 (Onagawa).

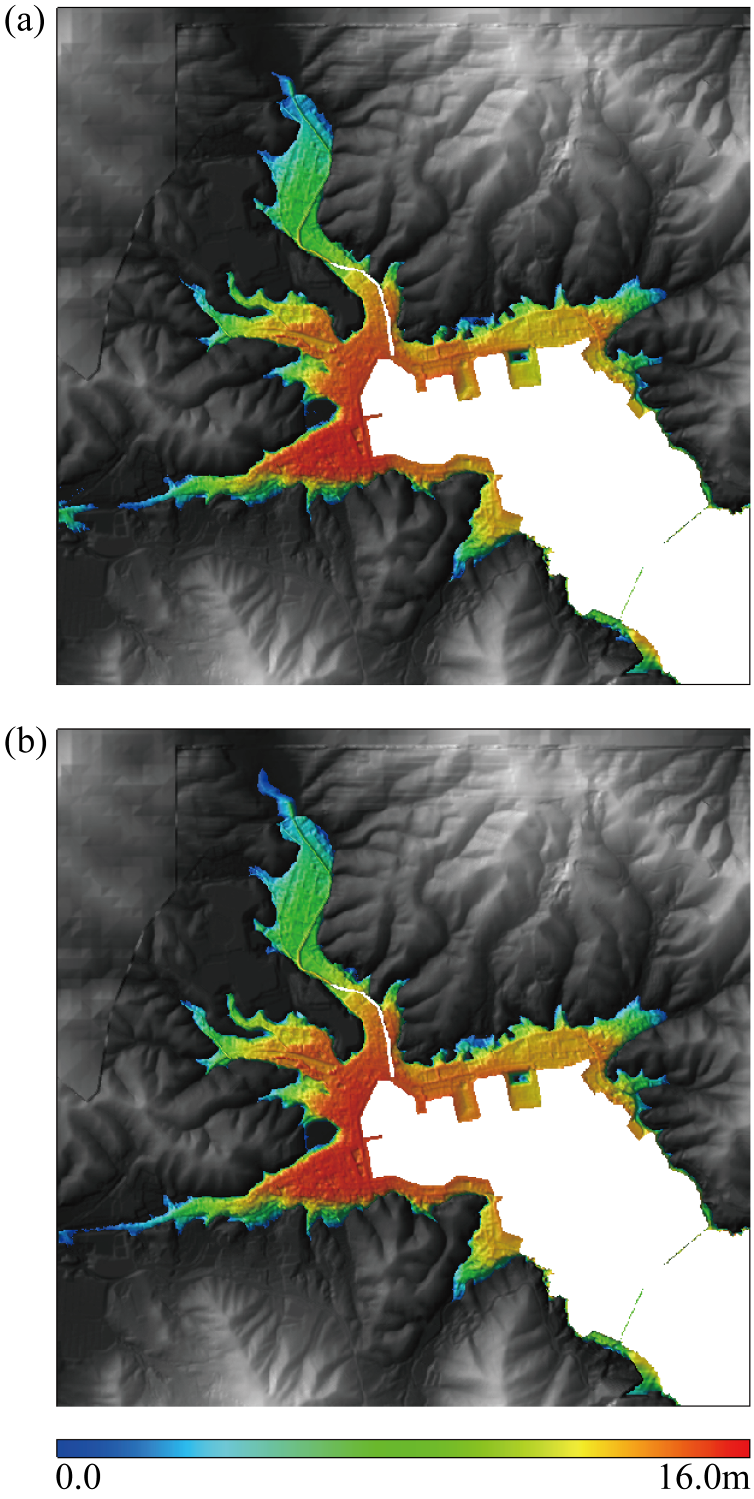


Figure 4.18: Maximum inundation depth in sub-region5: (a) LBM, (b) FDM.

Table 4.1: Parameters used in the dam-breaking problem.

Case No.	Resolution	Δx (m)	Δt (s)	τ
Case1	100×10	0.1	0.01	$0.5 + 3.0 \times 10^{-6}$
Case2	200×20	0.05	0.005	$0.5 + 6.0 \times 10^{-6}$
Case3	1000×100	0.01	0.001	$0.5 + 3.0 \times 10^{-5}$

Table 4.2: Parameters used in the square cylinder benchmark problem.

Case No.	Resolution	Δx (m)	Iterations	Δt (s)	τ
Case1	200×140	0.1	6.0×10^{5}	0.001	$0.5 + 7.5 \times 10^{-6}$
Case2	400×280	0.05	6.0×10^{5}	0.001	$0.5 + 3.0 \times 10^{-5}$
Case3	800×560	0.025	1.2×10^{6}	0.0005	$0.5 + 6.0 \times 10^{-5}$
Case4	1600×1120	0.0125	1.2×10^{6}	0.0005	$0.5 + 2.4 \times 10^{-4}$

Table 4.3: Parameters used in the plane-driven cavity problem.

Re	Resolution	Δx (m)	Iterations	Δt (s)	τ
10000	128×128	7.81×10^{-3}	3.6×10^{6}	5.0×10^{-4}	0.50246
	256×256	3.91×10^{-3}	3.6×10^{6}	5.0×10^{-4}	0.50983
15000	128×128	7.81×10^{-3}	3.6×10^{6}	5.0×10^{-4}	0.50164
	256×256	3.91×10^{-3}	3.6×10^{6}	5.0×10^{-4}	0.50655
20000	128×128	7.81×10^{-3}	3.6×10^{6}	5.0×10^{-4}	0.50123
	256×256	3.91×10^{-3}	3.6×10^{6}	5.0×10^{-4}	0.50492
30000	128×128	7.81×10^{-3}	3.6×10^{6}	5.0×10^{-4}	0.50082
	256×256	3.91×10^{-3}	3.6×10^{6}	5.0×10^{-4}	0.50327

Table 4.4: Parameters tsunami simulation in Onagawa.

Sub-region No.	Resolution	Δx (m)	Δt (s)	τ
Sub-region1	1330×1632	405.0	0.1	0.5001
Sub-region2	419×374	135.0	0.1	0.52
Sub-region3	558×472	45.0	0.1	0.515
Sub-region4	776×422	15.0	0.1	0.54
Sub-region5	680×665	5.0	0.1	0.55

References

[Abe97] Abe, T. Derivation of the lattice Boltzmann method by means of the discrete ordinate method for the Boltzmann equation. *J. Comput. Phys.*, 131(1):241–246, 1997.

[BA97] Borthwick, A. G. L. and Akponasa, G. A. Reservoir flow prediction by contravariant shallow water equations. *J. Hydraul. Eng.*, 123(5):432–439, 1997.

[Bal98] Balzano, A. Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models. *Coast. Eng.*, 34(34):83–107, 1998.

[BGK54] Bhatnagar, P. L., Gross, E. P., and Krook, M. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94(3):511–525, 1954.

[BS06] Bruneau, C. H. and Saad, M. The 2D lid-driven cavity problem revisited. *Comput. Fluids*, 35(3):326–348, 2006.

[BSHG95] Briggs, M. J., Synolakis, C. E., Harkins, G. S., and Green, D. R. Laboratory experiments of tsunami runup on a circular island. *Pure Appl. Geophys.*, 144(3-4):569–593, 1995.

[CCM92] Chen, H., Chen, S., and Matthaeus, W. H. Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. *Phys. Rev. A*, 45:R5339–R5342, 1992.

[CCMM91] Chen, S., Chen, H., Martnez, D., and Matthaeus, W. Lattice Boltzmann model for simulation of magnetohydrodynamics. *Phys. Rev. Lett.*, 67:3776–3779, 1991.

[CD98] Chen, S. and Doolen, G. D. Lattice Boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.*, 30:329–364, 1998.

[CG58] Carrier, G. F. and Greenspan, H. P. Water waves of finite amplitude on a sloping beach. *Journal Fluid Mech.*, 4(1):97–109, 1958.

[CPL13] Chopard, B., Pham, V.T., and Lefèvre, L. Asymmetric lattice Boltzmann model for shallow water flows. *Comput. Fluids*, 88:225–231, 2013.

[CPWC04] Cao, Z., Pender, G., Wallis, S., and Carling, P. Computational dam-break hydraulics over erodible sediment bed. *J. Hydraul. Eng.*, 130:689–703, 2004.

[ECG05] Erturk, E., Corke, T. C., and Gökçöl, C. Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. *Int. J. Numer. Methods Fluids*, 48(7):747–774, 2005.

[FHP86] Frisch, U., Hasslacher, B., and Pomeau, Y. Lattice-gas automata for the Navier-Stokes equation. *Phys. Rev. Lett.*, 56:1505–1508, 1986.

[Fra08] Frandsen, J. B. A simple LBE wave runup model. *Prog. Comput. Fluid Dyn. An Int. J.*, 8:222–232, 2008.

[GGS82] Ghia, U., Ghia, K. N., and Shin, C. T. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.*, 48(3):387–411, 1982.

[GKT$^+$06] Geller, S., Krafczyk, M., Tölke, J., Turek, S., and Hron, J. Benchmark computations based on lattice-Boltzmann, finite element and finite volume methods for laminar flows. *Comput. Fluids*, 35(8-9):888–897, 2006.

[GNGB97] Gallivan, M. A., Noble, D. R., Georgiadis, J. G., and Buckius, R. O. An evaluation of the bounce-back boundary condition for lattice boltzmann simulations. *Int. J. Numer. Methods Fluids*, 25(3):249–263, 1997.

[GS96] Grilli, S. T. and Subramanya, R. Numerical modeling of wave breaking induced by fixed or moving boundaries. *Comput. Mech.*, 17(6):374–391, 1996.

[Gua00] Guangwu, Y. A lattice Boltzmann equation for waves. *J. Comput. Phys.*, 161(1):61–69, 2000.

[HJ89] Higuera, F. J. and Jimenez, J. Boltzmann approach to lattice gas simulations. *Europhysics Lett.*, 9(7):663–668, 1989.

[HL97] He, X. and Luo, L. S. A priori derivation of the lattice Boltzmann equation. *Phys. Rev. E*, 55:R6333–R6336, 1997.

[HP79] Hibberd, S. and Peregrine, D. H. Surf and run-up on a beach: a uniform bore. *J. Fluid Mech.*, 95(2):323–345, 1979.

[HSCD96] Hou, S., Sterling, J., Chen, S., and Doolen, G. D. A lattice boltzmann subgrid model for high Reynolds number flows. *Fields Inst. Commun.*, 6:151–166, 1996.

[III13] Imai, K., Imamura, F., and Iwama, S. Advanced tsunami computation for urban regions. *J. Jap. Soc. Civ. Eng.*, 69(2):311–315 (in Japanese), 2013.

[KM93] Kowalik, Z. and Murty, T. S. *Numerical modeling of ocean dynamics.* World Scientific, 1993.

[Koe91] Koelman, J. M. V. A. A simple lattice Boltzmann scheme for Navier-Stokes fluid flow. *Europhysics Lett.*, 15(6):603–607, 1991.

[KV73] Kuipers, J. and Vreugdenhil, C. B. Calculation of two-dimensional horizontal flow. Technical report, Delft Hydraulic Laboratory, Delft, The Netherlands, 1973.

[LCB$^+$95] Liu, P. L. F., Cho, Y. S., Briggs, M. J., Kanoglu, U., and Synolakis, C. E. Runup of solitary waves on a circular Island. *J. Fluid Mech.*, 302:259–285, 1995.

[LL00] Lallemand, P. and Luo, L. S. Theory of the lattice boltzmann method: dispersion dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E*, 61(6):6546–6562, 2000.

[LLS12] Liu, H., Li, M., and Shu, A. Large eddy simulation of turbulent shallow water flows using multi-relaxation-time lattice Boltzmann model. *Int. J. Numer. Methods Fluids*, 70:1573–1589, 2012.

[LOC94] Liu, X. D., Osher, S., and Chan, T. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200–212, 1994.

[LR02] Li, Y. and Raichlen, F. Non-breaking and breaking solitary wave run-up. *J. Fluid Mech.*, 456:295–318, 2002.

[LWL02] Lynett, P. J., Wu, T. R., and Liu, P. L. F. Modeling wave runup with depth-integrated equations. *Coast. Eng.*, 46(2):89–107, 2002.

[LZB10] Liu, H., Zhou, J. G., and Burrows, R. Lattice Boltzmann simulations of the transient shallow water flows. *Adv. Water Resour.*, 33(4):387–396, 2010.

[MLI09] MLIT. National Land Use Classification (in Japanese), The Minister of Land, Infrastructure, Transport and Tourism (MLIT). 2009.

[MR78] Mcguirk, J. J. and Rodi, W. A depth-averaged mathematical model for the near field of side discharges into open-channel flow. *J. Fluid Mech.*, 86(4):761–781, 1978.

[MS92] Madsen, P. A. and Sørensen, O. R. A new form of Boussinesq equations with improved dispersion characteristics. Part 2. A slowly-varying bathymetry. *Coast. Eng.*, 18:183–204, 1992.

[MSH97] Meselhe, E. A., Sotiropoulos, F., and Holly, F. M. Numerical simulation of transcritical flow in open channels. *J. Hydraul. Eng.*, 123(9):774–783, 1997.

[NCGB95] Noble, D. R., Chen, S., Georgiadis, J. G., and Buckius, R. O. A consistent hydrodynamic boundary condition for the lattice Boltzmann method. *Phys. Fluids*, 7(1):203–209, 1995.

[Oka85] Okada, Y. Surface deformation due to shear and tensile faults in a half-space. *Bull. Seismol. Soc. Am.*, 75(4):1135–1154, 1985.

[Per67] Peregrine, D. H. Long waves on a beach. *J. Fluid Mech.*, 27(4):815–827, 1967.

[PKAI11] Pophet, N., Kaewbanjak, N., Asavanant, J., and Ioualalen, M. High grid resolution and parallelized tsunami simulation with fully nonlinear Boussinesq equations. *Comput. Fluids*, 40(1):258–268, 2011.

[QDL92] Qian, Y. H., D'Humières, D., and Lallemand, P. Lattice BGK models for Navier-Stokes Equation. *Europhys. Lett.*, 6(17):479–484, 1992.

[RZ97] Rothman, D. H. and Zaleski, S. *Lattice gas cellular automata.* London Cambridge University Press, 1997.

[SC96] Sterling, J. D. and Chen, S. Stability analysis of lattice Boltzmann methods. *J. Comput. Phys.*, 123(123):196–206, 1996.

[SC06] Simpson, G. and Castelltort, S. Coupled model of surface water flow, sediment transport and morphological evolution. *Comput. Geosci.*, 32(10):1600–1614, 2006.

[Sch14] Schlaffer, M. B. *Non-reflecting boundary conditions for the lattice Boltzmann method.* PhD thesis, Technische Universität München, Germany, 2014.

[SFHN13] Satake, K., Fujii, Y., Harada, T., and Namegaya, Y. Time and space distribution of coseismic slip of the 2011 Tohoku earthquake as inferred from tsunami waveform dataTime and space distribution of coseismic slip of the 2011 Tohoku earthquake. *Bull. Seismol. Soc. Am.*, 103(2B):1473–1492, 2013.

[Shu91] Shuto, N. Numerical simulation of tsunamis - Its present and near future. *Nat. Hazards*, 4(2-3):171–191, 1991.

[SKH$^+$12] Shi, F., Kirby, J. T., Harris, J. C., Geiman, J. D., and Grilli, S. T. A high-order adaptive time-stepping TVD solver for Boussinesq modeling of breaking waves and coastal inundation. *Ocean Model.*, 43-44:36–51, 2012.

[Sko93] Skordos, P. A. Initial and boundary conditions for the lattice Boltzmann method. *Phys. Rev. E*, 48:4823–4842, 1993.

[Sma63] Smagorinsky, J. General circulation experiments with the primitive equations. *Mon. Weather Rev.*, 91(3):99–164, 1963.

[SR66] Sokolnikoff, I. S. and Redheffer, R. M. *Mathematics of physics and modern engineering.* McGraw-Hill, 1966.

[Sto57] Stoker, J. J. *Water Waves.* Wiley, 1957.

[Syn87] Synolakis, C. E. The runup of solitary waves. *J. Fluid Mech.*, 185:523–545, 1987.

[TJB$^+$01] Teeter, A. M., Johnson, B. H., Berger, C., Stelling, G., Scheffner, N. W., Garcia, M. H., and Parchure, T. M. Hydrodynamic and sediment transport modeling with emphasis on shallow-water, vegetated areas (lakes, reservoirs, estuaries and lagoons). *Hydrobiologia*, 444(1):1–23, 2001.

[TSB07] Thömmes, G., Seaïd, M., and Banda, K. Lattice Boltzmann methods for shallow water flow applications. *Int. J. Numer. Methods Fluids*, 55:673–92, 2007.

[Tub10] Tubbs, K. *Lattice Boltzmann modeling for shallow water equations using high performance computing.* PhD thesis, Lousiana State University, 2010.

[Wah12] Wahba, E. M. Steady flow simulations inside a driven cavity up to Reynolds number 35,000. *Comput. Fluids*, 66(35):85–97, 2012.

[WKGS95] Wei, G., Kirby, J. T., Grilli, S. T., and Subramanya, R. A fully nonlinear Boussinesq model for surface waves. Part 1. Highly nonlinear unsteady waves. *J. Fluid Mech.*, 294:71–92, 1995.

[ZB11] Zhou, J. G. and Borthwick, A. G. L. Lattice Boltzmann method for variable density shallow water equations. *Comput. Fluids*, 49(1):146–149, 2011.

[Zho01] Zhou, J. G. An elastic-collision scheme for lattice Boltzmann methods. *Int. J. Mod. Phys. C*, 12(03):387–401, 2001.

[Zho02] Zhou, J. G. A lattice Boltzmann model for the shallow water equations. *Comput. Methods Appl. Mech. Eng.*, 191(32):3527–3539, 2002.

[Zho04] Zhou, J. G. *Lattice Boltzmann methods for shallw water flows.* Springer, 2004.

[Zho11] Zhou, J. G. Enhancement of the LABSWE for shallow water flows. *J. Comput. Phys.*, 230(2):394–401, 2011.

[ZL13] Zhou, J. G. and Liu, H. Determination of bed elevation in the enhanced lattice Boltzmann method for the shallow-water equations. *Phys. Rev. E*, 88(2):1–6, 2013.

Chapter 5

A 2D/3D hybrid coupling tsunami modelling

5.1 Introduction

This chapter will focus on scenes that have a much larger physical scale, e.g., a tsunami propagation from its generation concomitant with offshore earthquake to inundation in an urban area. As these phenomena are simulated by physical-based fluid modellings, several issues arise. The most significant problem is that the amount of computational work and the required resources grow when the resolution of the simulation is increased. The full simulation of the scales mentioned the above section with the volume-of-fluid (VOF) Navier-Stokes solver would hardly be possible even on a large supercomputer. In the following, a hybrid fluid simulation approach will be presented that computes the full fluid flow only in a bounded region of interest such as an unsteady flow around buildings, and uses a fast two-dimensional shallow water fluid simulation approach to compute the fluid surface around it.

As of today, a shallow water based two-dimensional (2D) and free surface three-dimensional (3D) coupling fluid simulation modelling is still a important research topic in coastal engineering [FMG02, YMM12, TT12, PY13] as an efficient tsunami modelling approach. Recently, approaches to couple two-dimensional fluid modellings based on shallow water equations or Boussinesq equations with three-dimensional particle fluid approaches such as Smoothed Particle Hydrodynamics [Luc77] or Moving Particle Semi-implicit methods [KO96] were proposed [SIS14]. A different coupling approach for grid-based modelling using OpenFOAM and shallow water equations was proposed [PISH14, KMS14]. These, however, do not connect the three-dimensional results to two-dimensional simulations. A two-way coupling model should be required in tsunami modelling in term of the return flow.

A different approach for detailed and precise two and three-dimensional hybrid fluid simulation model that will be developed by the lattice Boltzmann method to achieve more efficient tsunami simulation in this thesis. Several hybrid tsunami simulation techniques solving the conventional Navier-Stokes equations and shallow water equations has been carried out simulations for the tsunami of the 2011 off the Pacific coast of Tohoku Earthquake. These approaches, however, have only successfully applied only in limited computing environment as described the above because of the difficulty to solve the pressure Poisson equation, iteratively. More recently, Fukui *et. al.* [FKM10] presented hybrid tsunami simulation techniques by the lattice Boltzmann method based on Thürey. These works, however, were focused on scenes with a small deformation at the hybrid coupling boundary. The weak-compressibility of the three-dimensional free surface flow model by the lattice Boltzmann method has also to be considered to develop a hybrid simulation model. It cannot be ignored especially in tsunami modellings.

5.2 2D/3D hybrid tsunami simulation

An overview of the hybrid simulation approach is given in Figure 5.1. Both 2D and 3D calculation algorithms have been parametrised to solve the same fluid simulation problem. In the following it will be assumed that the shallow water simulation is performed in the x and y plane, and the gravitational force acts in the direction of the negative z axis. There is an inherent difference between the two simulation approaches that has to be overcome: the derivation of the shallow water simulation assumes a depth averaged velocity and has a coupling between fluid height and velocity. The three-dimensional simulation, on the other hand, can have a velocity varying along the z axis, and has boundary conditions that makes it independent of the initial height of the fluid surface h_{ini}. In order to be able to couple both simulations, the 2D/3D hybrid coupling algorithms explained in the following were developed.

5.3 Coupling 2D to 3D algorithm

The transfer from the 2D shallow water simulation to the 3D free surface simulation, is done by initialising the 3D domain cells to represent the 2D shallow water simulation height and velocity. For a 3D domain cell at (i, j) marked with orange colour in the upper part of Figure 5.2 cells are thus removed if $H(i, j) > h(i, j)$, otherwise new ones are added. Where H is the replaced fluid height in 3D domain and h is the height calculated by 2D simulation. To correctly initialise the new cells the velocity of the shallow water simulation is directly used as:

$$\boldsymbol{u}_{3\text{D},\alpha} = \boldsymbol{u}_{2\text{D},\alpha}, \quad \text{for} \quad \alpha = i, j \tag{5.1}$$

The k coordinate of the velocity is 0 because of depth averaged theory.

For the outer layer of the 3D simulation (square cells marked with yellow colour in the upper part in Figure 5.2) velocity boundary conditions with a fixed pressure are used. For the lattice Boltzmann method the pressure of a cell with height k in the domain is given by:

$$\rho_k = \rho_0 + 3\omega \left(h_{\text{in}} - k\right) \cdot g_l \tag{5.2}$$

where h_{in} total fluid depth in 3D domain calculated by 2D shallow water simulation and k is the calculation integer index. The pressure thus increase further down in the grid in 3D domain, with a gradient that depends on the relaxation time $\omega = 1/\tau$. This parameter determines the pressure gradient according to the dimensionless gravity acceleration caused by the weak-compressibility of the 3D free surface model based on the lattice Boltzmann approach. Equation 5.2, thus, satisfies pressure boundary condition at the gas-liquid interface cell (e.g., ρ_k becomes ρ_0 at the interface which means the fluid pressure $p = 0$ at the interface). The velocity can again be taken directly from the shallow water simulation as described above. Note that it is in this case not necessary to scale the shallow water simulation velocities, as the whole height of the 3D simulation is set. The distribution functions in 2D/3D coupling overlapping region can be calculated by the equilibrium distribution functions.

The fluid fraction values C in 3D domain has also to be initialised by 2D shallow water simulation water height as:

$$C = \begin{cases} 0 & \left(k\Delta x_{\text{3D}} > h + \Delta x_{\text{3D}}\right) \\ \left(h - k\Delta x_{\text{3D}}\right)/\Delta x_{\text{3D}} & \left(k\Delta x_{\text{3D}} \in S_k\right) \\ 1 & \left(k\Delta x_{\text{3D}} < h - \Delta x_{\text{3D}}\right) \end{cases} \tag{5.3}$$

where S_k is the existing range of the interface cells. It can be determined by the calculated 2D fluid height h and Δx_{3D} as $S_k = [h - \Delta x_{\text{3D}}, h + \Delta x_{\text{3D}}]$.

5.4 Coupling 3D to 2D algorithm

The height of the fluid at a position within the 3D simulation region is determined by searching for the first interface cell from bottom assuming the single-valued function of the fluid height based on the shallow water theory. Hence, the fluid height is computed for the first interface cell at (i, j, k_H) with:

$$H\left(i,j\right) = \left[\left(k_H - 1\right) + C\left(i,j,k_H\right)\right]\Delta x_{\text{3D}} \tag{5.4}$$

where C is the fluid fraction values of volume-of-fluid (VOF) approach. Thus, the height of the interface cells can be accurately considered by the value.

The velocity of 2D model is also given by the depth averaged velocity of 3D simulation result based on the shallow water theory as:

$$\boldsymbol{u}_{2\mathrm{D}}(i,j) = \frac{1}{k} \sum_{k\Delta x_{3\mathrm{D}} < H} \boldsymbol{u}_{\alpha,3\mathrm{D}}(i,j,k), \quad \text{for} \;\; \alpha = i,j \tag{5.5}$$

To transfer the information from the 3D simulation to the 2D shallow water simulation a cell at (i,j), shown as a circle marked with X in Figure 5.2, is initialised with the equilibrium distribution functions:

$$f_{2\mathrm{D},i}(i,j) = f^{eq}_{2\mathrm{D},i}\left(H(i,j), \boldsymbol{u}_{2\mathrm{D}}(i,j)\right) \tag{5.6}$$

the cells where height and velocity are set with the above equations represent the inner boundary for the 2D shallow water simulation. Further inwards the full 3D simulation is performed, thus the cells of the 2D shallow water simulation do not have to be updated in this region, as their values are never used. To ensure a transfer with as few disturbances as possible, a double layered transfer is used. Thus, a second type of boundary condition for the region of shallow water simulation cells directly outwards of the boundary cells (the circle marked with O in Figure 5.2) is applied. For the cells that are updated according to Equation 5.6 all distribution functions are reset each time step, while for the second boundary layer the existing distribution functions are only rescaled to match the calculated fluid height:

$$f^{*}_{2\mathrm{D},i} = f_{2\mathrm{D},i} \cdot \frac{H(i,j)}{h(i,j)} \tag{5.7}$$

The combination of these two boundary conditions ensures a correct transfer of both fluid surface height and velocity.

5.5 Calculation algorithm

The time step interval of the 3D free surface model $\Delta t_{3\mathrm{D}}$ generally smaller than the 2D shallow water simulation one $\Delta t_{2\mathrm{D}}$ due to the stability condition of each model. In this thesis, the simple linear interpolation technique is applied to determine the macroscopic variables between 2D and 3D model as shown in Figure 5.3.

The developed 2D/3D hybrid tsunami simulation, thus, carried out in the following chart.

Algorithm 5 Calculation loop 2D/3D hybrid tsunami simulation

```
execute the two-dimensional shallow water simulation (Algorithm 4)
connect variables from two dimension to three dimension as the boundary condition
execute the three-dimensional free surface simulation (Algorithm 1)
connect variables from three dimension to two dimension
```

5.6 Validation

5.6.1 A dam-breaking flow with obstacles

After the classical benchmark cases for both two and three-dimensional fluid model we proposed, the hybrid 2D/3D model is tested against the experimental dam-breaking flow with three obstacles by Fukui *et. al.* [FKM10]. The initial setting is shown in Figure 5.4. The 3D fluid simulation was implemented on an elective basis around the first obstacle against the wave gate to calculate three-dimensional fluid flow conditions in high accuracy. All the results of the 3D fluid simulation was returned to the two-dimensional shallow water flow model in wetting area in this benchmark. From the numerical point of view, this test case is demanding for two-dimensional shallow water model due to high Froude number immediately after gate opening. Here, we defined an upper limit parameter $\phi = 0.95$ for the velocity only for two-dimensional model. When, due to high fluid velocity, the stability condition cannot be hold, we compute a new suitable velocity magnitude as:

$$\boldsymbol{u}_{\text{new}} = \frac{\sqrt{\boldsymbol{u} \cdot \boldsymbol{u}}}{\sqrt{gh}} \cdot \phi \tag{5.8}$$

and replace the distribution functions of the cell with new ones calculated by its equilibrium distribution functions.

Table 5.1 shows the calculation parameters for each fluid model. The time step interval Δt of three-dimensional model generally much smaller than two-dimensional one, because the calculation stability conditions are different. This trend remarkably appears in the lattice Boltzmann fluid simulation compared with the conventional free surface model based on a direct solver for the Navier-Stokes equations due to the weak-compressibility of three-dimensional model. We should reduce the flow Mach number, which can be estimated by the maximum flow velocity magnitude to inhibit the growth of the compressibility. The costs of computing times and this parameter settings, however, are related to the trade-off relationship. The difference of the time step interval between 2D and 3D model should also be not large in term of the interpolation for values in time. We, thus, set $\text{Ma} = 0.1$ in the three-dimensional fluid model so as to have the same time stepping order with the two-dimensional fluid model in this benchmark.

Snapshots of the simulation results for selected time step $t = 1.34$s are given in Figure 5.5. This figure is just after the surge front hits against the first obstacle. The proposed hybrid 2D/3D coupling tsunami simulation model has been successfully implemented in the lattice Boltzmann method for the single-phase solver and is currently adapted to the shallow water model. There is no discontinuity connecting point and numerical oscillation in hybrid coupling region because the proposed hybrid model can consider the effect of the weak-compressibility in the coupling. The results of hybrid simulation, moreover, well reproduced three-dimensional flow around the obstacle compared with the only two-dimensional simulation, such as the hitting against the obstacle. On the other hand, the hybrid approach cannot calculate small splashes seen in the experimental data. Breaking waves ware observed just after the gate opening in the experiment. The two-dimensional model, however, cannot simulate the phenomena because it is based on the depth-averaged model. The three-dimensional computing area should be expanded to simulate such flows more accurately. Finally, the timeseries water level at the two observed points compared to the experimental data are shown in Figure 5.6. The hybrid simulation improves extremely overestimation of only 2D simulation around $t = 1.0$s. We restricted the 2D model velocity due to the stability condition. It can be considered that the total water depth, thus, grow up to satisfy the continuity condition locally. The three-dimensional model does not have to be limited the velocity. This is why the results of 2D simulation overestimates the experimental data. The numerical simulation for this hybrid model case took approximately 300 min, on the at most 8 cores with OpenMP. In combination with the high result accuracy, this emphasises the applicability of our proposed 2D/3D hybrid fluid simulation approach to an efficient tsunami simulation.

5.7 Conclusion

In this chapter, we developed a two and three-dimensional hybrid fluid model for large scale tsunami modellings, e.g., a tsunami propagation from its generation concomitant with offshore earthquake to inundation in an urban area. Through the validation of the proposed coupling approach on benchmark problems and realistic tsunami flows,

- The proposed coupling approach is formulated by considering the weak-compressibility of three-dimensional free surface model with the multiple-relaxation-time collision model at the time of 2D → 3D connecting.
- Both the 2D → 3D and 3D → 2D is free from the non-physical oscillation of the interface emerge with conventional Fukui *et. al.* [FKM10] model.
- We presented application of the proposed model with tsunami simulation benchmark problem. The validation shows a good performance of the code and reproduce three-dimensional flow phenomena efficiency and accurately such as flows around obstacles (e.g., buildings).

The coupling two and three-dimensional fluid simulation model we proposed in this thesis is based on the shallow water theory (i.e., two-dimensional fluid model). The coupling model, thus, should be applied in the region, in which the vertical acceleration can be ignored, because the vertical velocity cannot be connected by the algorithm. The accuracy depending on the three-dimensional area should deeply be validated with other benchmarks. This is an challenge for the future.

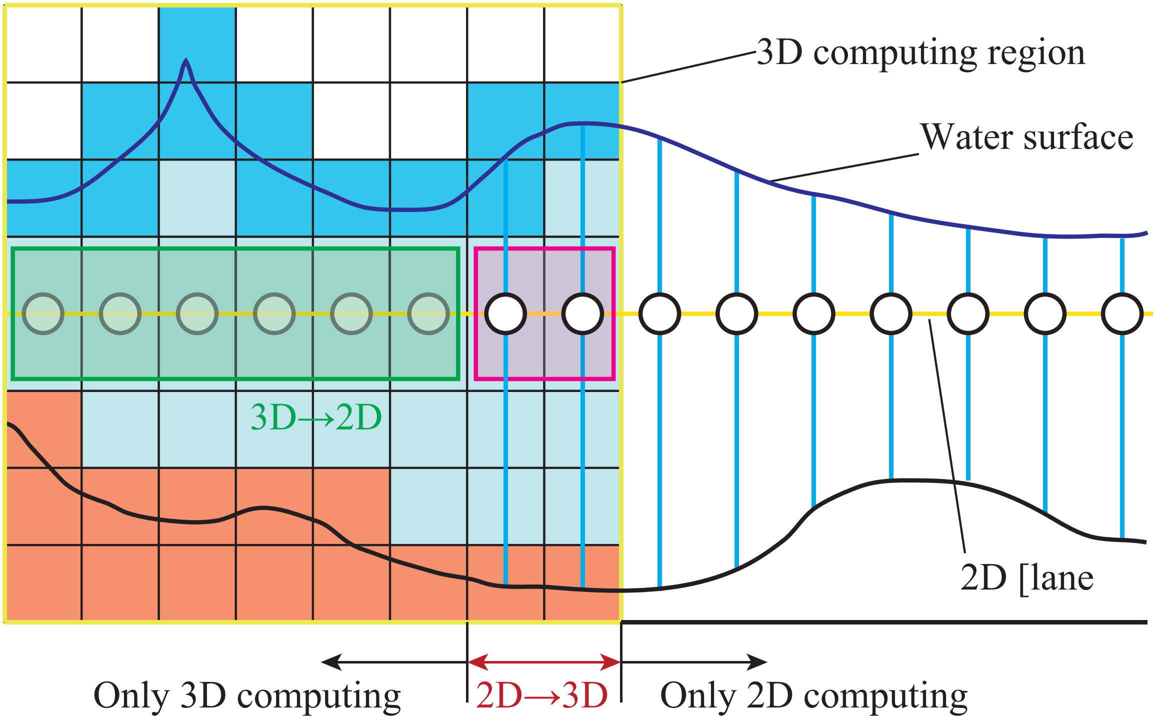


Figure 5.1: This schematic illustration gives an overview of the 2D/3D hybrid tsunami simulation method. The full three-dimensional fluid flow is solved in a given region of interest (illustrated by a 2D rectangle), and coupled to a two-dimensional shallow water simulation (shown as a one-dimensional line in the picture).

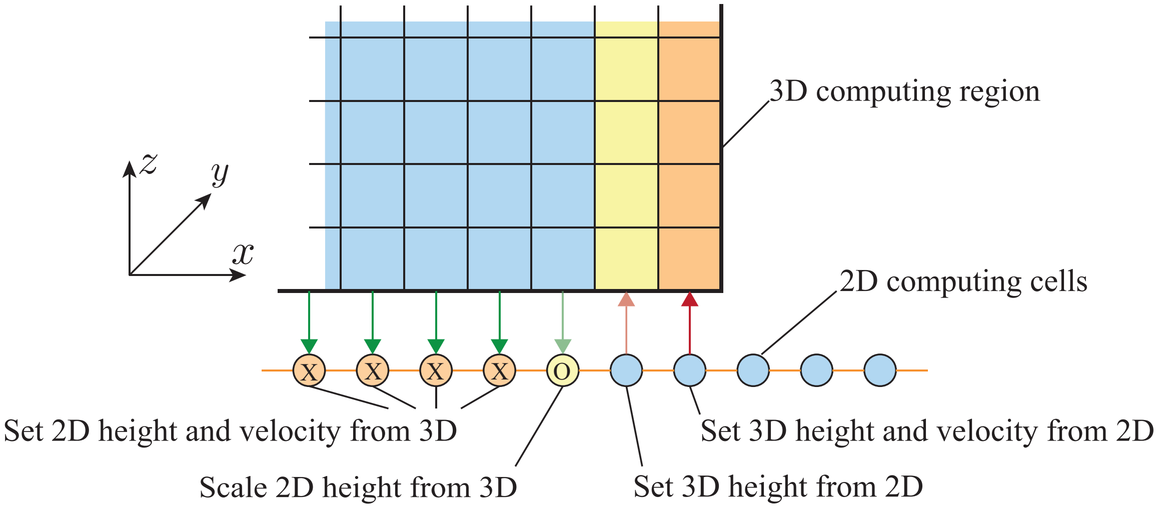


Figure 5.2: Detail of the double layer boundary conditions in the overlapping interface region.

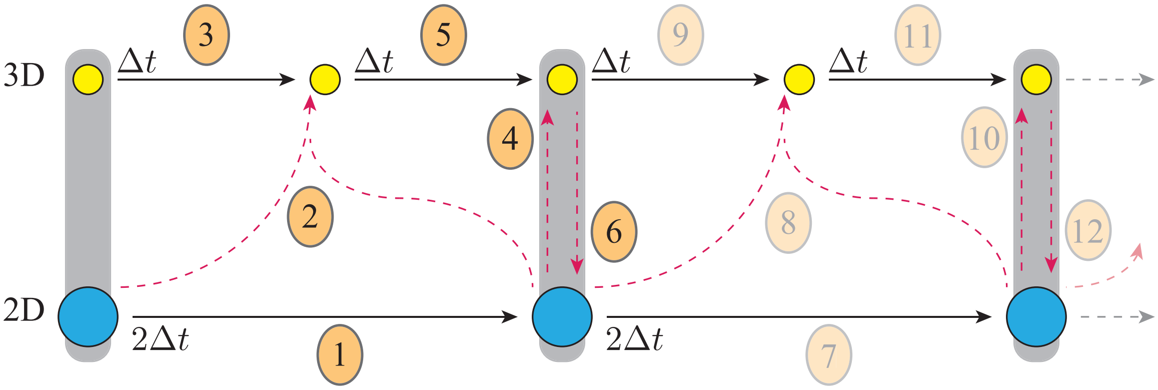


Figure 5.3: Here the effect of the different time step sizes for the 2D/3D hybrid fluid simulation is shown. The numbers indicate the order in which the steps are performed. Dashed arrows indicate interpolation, while straight arrows from one circle to another represent lattice Boltzmann steps with the indicated time step length.

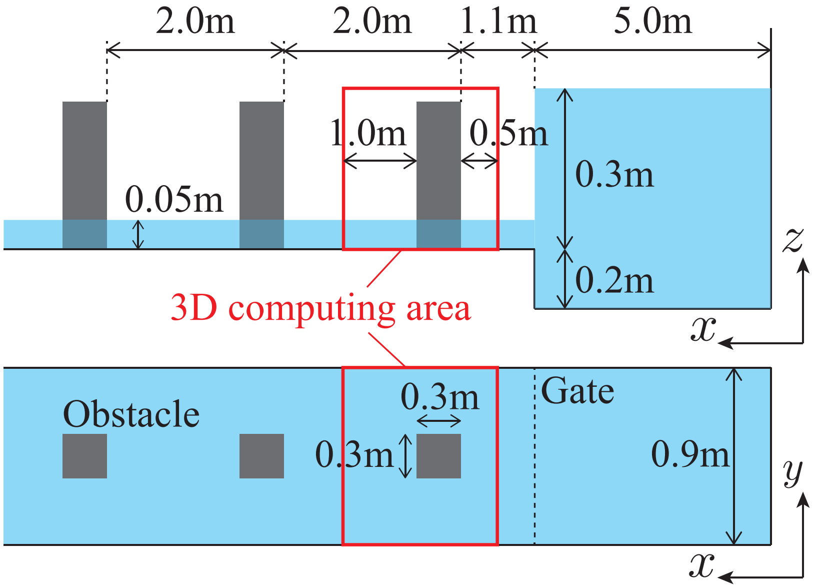


Figure 5.4: The calculation domain of the Fukui *et. al.* [FKM10] dam-breaking flows.

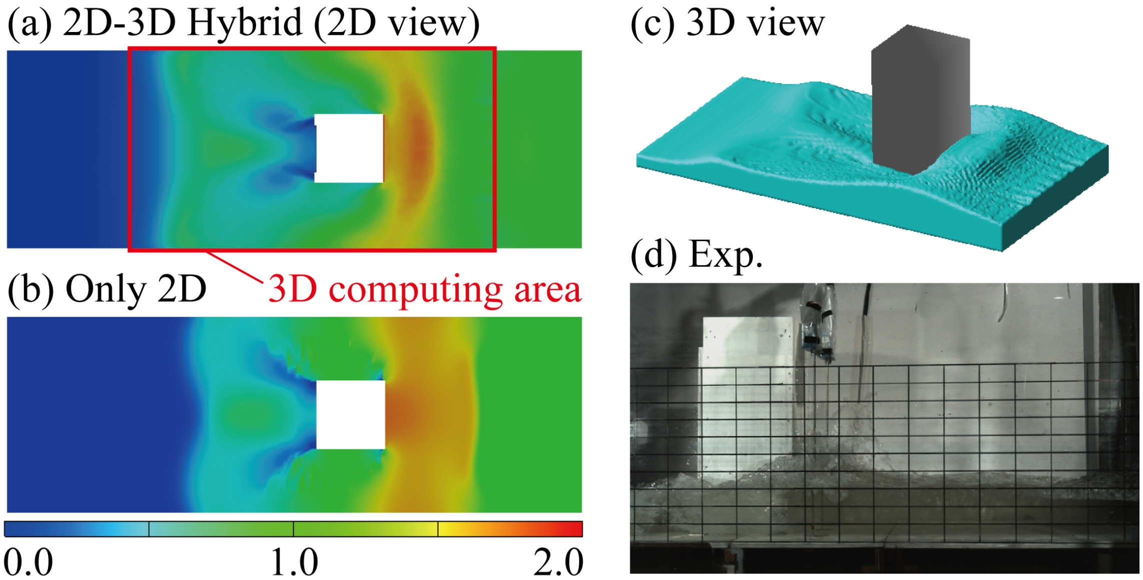


Figure 5.5: Free surface shapes at $t = 1.34$s: (a) two-dimensional view of hybrid simulation (b) two-dimensional view of only two-dimensional simulation, (c) three-dimensional view, (d) experimental data [FKM10].

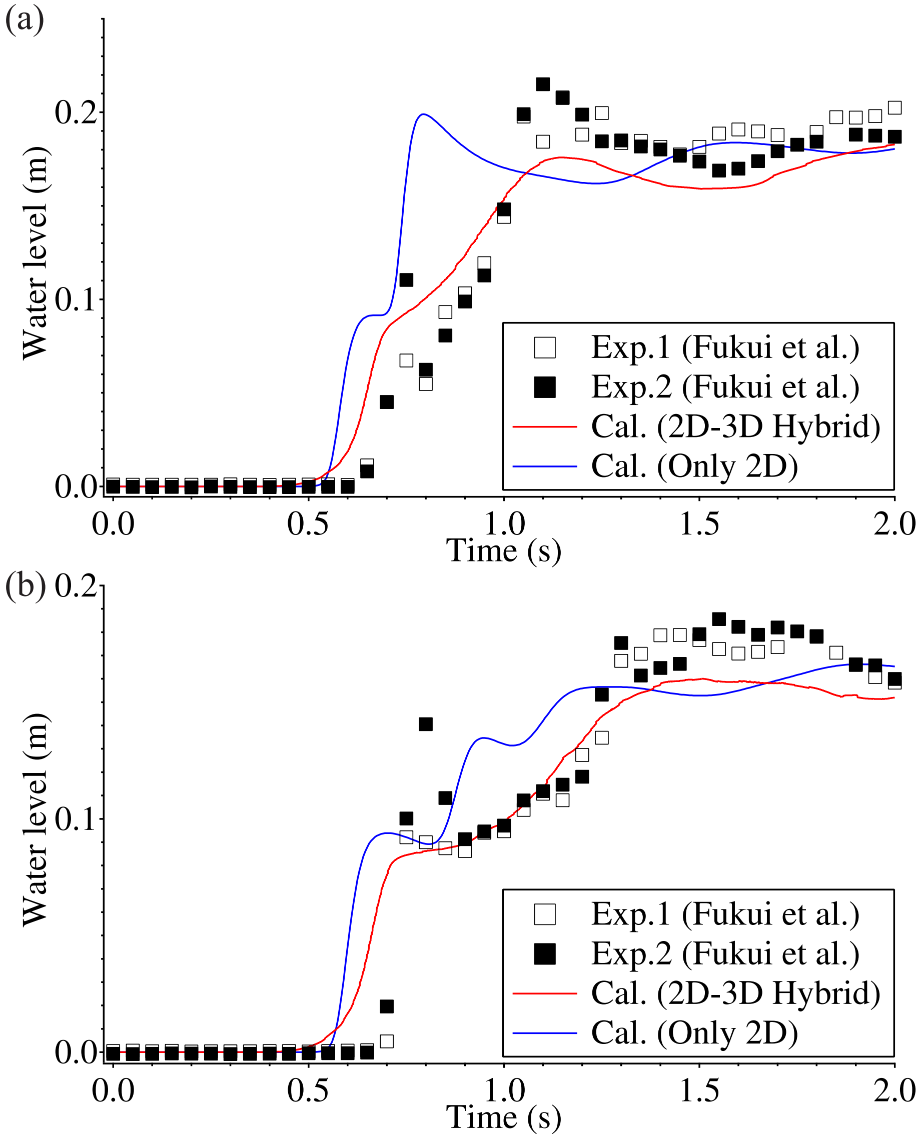


Figure 5.6: Timeseries water depth at: (a) $(x, y) = (6.0\text{m}, 0.1125\text{m})$, (b) $(x, y) = (6.0\text{m}, 0.3375\text{m})$.

Table 5.1: Parameters used in the dam-breaking benchmark [FKM10].

Model	Resolution	Ma	$\Delta x\,(\mathrm{m})$	$\Delta t\,(\mathrm{s})$
2D	1375×113	–	8.0×10^{-3}	4.0×10^{-4}
3D	$225 \times 115 \times 150$	0.10	8.0×10^{-3}	1.0×10^{-4}

References

[FKM10] Fukui, T., Koshimura, S., and Matsuyama, M. 2D-3D hybrid simulation of tsunami inundation flow by lattice Boltzmann method. *J. Jap. Soc. Civ. Eng.*, 66(1):61–65 (in Japanese), 2010.

[FMG02] Fujima, K., Masamura, K., and Goto, C. Development of the 2D/3D hybrid model for tsunami numerical simulation. *Coast. Eng. J.*, 44(4):373–397, 2002.

[KMS14] Kawasaki, K., Matsuura, S., and Sakatani, T. Horizontal two-dimensional and three-dimensional fluid tsunami analysis around Nagoya port due to Nankai trough earthquake. *J. Jap. Soc. Civ. Eng.*, 70(2):391–395 (in Japanese), 2014.

[KO96] Koshizuka, S. and Oka, Y. Moving-Particle Semi-implicit method for fragmentation of incompressible fluid. *Nucl. Sci. Eng.*, 123(3):421–434, 1996.

[Luc77] Lucy, L. B. A numerical approach to the testing of the fission hypothesis. *Nucl. Sci. Eng.*, 82(12):1013–1024, 1977.

[PISH14] Phuc, P. V., Imazu, Y., Sagawa, T., and Hasebe, M. Tsunami simulation using VOF method and falling mechanism of a building in Tohoku-Pacific coast earthquake. *J. Jap. Soc. Civ. Eng.*, 70(2):156–160 (in Japanese), 2014.

[PY13] Pringle, W. and Yoneyama, N. The application of a hybrid 2D/3D numerical tsunami inundation-propagation flow model to the 2011 off the pacific coast of Tohoku earthquake tsunami. *J. Jap. Soc. Civ. Eng.*, 69(2):306–310 (in Japanese), 2013.

[SIS14] Suwa, T., Imamura, F., and Sugawara, D. Development of a tsunami simulator integrating the Smoothed-Particle Hydrodynamics method and the nonlinear shallow water wave model. *J. Jap. Soc. Civ. Eng.*, 70(2):16–20 (in Japanese), 2014.

[TT12] Tomita, T. and Takahashi, K. Practical methods for precise computation of the 2011 off the pacific coast of Tohoku earthquake tsunami. *J. Jap. Soc. Civ. Eng.*, 68(2):191–195 (in Japanese), 2012.

[YMM12] Yoneyama, N., Mori, N., and Miwa, M. Numerical analysis of the 2011 off the pacific coast of Tohoku earthquake tsunami in Kamaishi bay. *J. Jap. Soc. Civ. Eng.*, 68(2):161–165 (in Japanese), 2012.

Chapter 6

Conclusion

The goal of this thesis was to develop a tsunami simulation model with high efficiency, stability and flexibility. The following section will summarise the contributions of this thesis that were presented in order to achieve this goal. Afterwards, drawbacks and future extensions will be discussed.

6.1 Summary

The efficient tsunami simulation model, as presented in this thesis, is given due to the use of the lattice Boltzmann method for the conventional Navier-Stokes equations and non-linear shallow water equations. The basic algorithm performs very well on modern computer architectures, and is suitable for shared and distributed memory machines using API for parallel programming. The single-phased piecewise linear interface calculation approach can be directly integrated into the lattice Boltzmann method, and likewise yields a high efficiency. It preserves mass, and allows a local treatment of the free surface boundary conditions. The stability is mainly guaranteed by the turbulence model. In its different forms the Navier-Stokes simulation, a simplified lattice Boltzmann method based on the non-linear shallow water theory was developed to achieve a consistent tsunami simulation from its occurrence to inland penetration. The coupling of the three-dimensional free surface simulation with the non-linear shallow water simulation allows the tsunami simulation of large open water surfaces that could hardly simulated due to memory and time requirements.

We presented several applications of the proposed model with practical relevance for various coastal and tsunami engineering applications. Wave propagation, wave impact were addressed for both two, three-dimensional and hybrid flow problems. The validations show a good performance of the code and a good agreement of the numerical results with the reference data in most of the cases.

6.2 Discussion and future work

One of the main strength of the tsunami simulation model developed in this thesis is the computational efficiency of its components due to the lattice Boltzmann method. The computations usually involve only the local neighbourhood of a cell. Therefore, global properties - such as the mass conservation or incompressibility - are guaranteed by the local treatment of the algorithm. Several aspects will discuss areas of future work for coastal engineering applications: the solid-liquid coupling simulation to investigate soil and fluid structure interaction and parallel computing with GPGPUs for more efficient simulation.

6.2.1 Parallel adaptive mesh refinement implementations

Although using the piecewise linear interface calculation approach in the lattice Boltzmann method rectifies the need to have very high resolutions, a great deal of computational time and memory is still wasted on regions far from the interface or gas-phase where a coarse lattice resolution may suffice the smooth variations in the fluid fraction as well as pressure and velocity. An adaptive mesh refinement (AMR) technique was not carried out in this thesis so as to refrain from the complications it brings to the parallel processing. Nevertheless, efficient GPGPU implementations for AMR algorithm could be considered as a future work. In particular, NVIDIA's Kepler generation GPGPUs provide a so-called dynamic parallelism framework which are aimed at facilitating adaptive computations through launching multiple levels of nested kernels, each resolving one lattice level in the refinement hierarchy. Conventional multi-block lattice Boltzmann frameworks using variable time steps for each level as in [GZZ11], or those based on a single time step finite difference formulations [FL14, FL15] could then be considered as candidates for such parallel implementations.

6.2.2 Multiple GPGPU implementations

Parallel computing implementations in this thesis were designed using MPI and OpenMP on CPUs. It is well known that the lattice Boltzmann method is originally suitable for GPUs which have manycore and is low clock frequency. Although the convergence trend in three-dimensional benchmark flows proved that the results are approaching those of direct numerical simulation for the Navier-Stokes equations, multiple GPGPU implementations will indeed facilitate reaching the same accuracy level via high resolution simulations. In case a more memory efficient an adaptive mesh refinement programming is also pursued, such massively parallel implementations also allow to treat large real scale problems of interest for industrial applications within reasonable times and costs.

6.2.3 Coupling with the discrete element method

Typical multiphysics problems occur in many different fields of civil and coastal engineering such as investigating the flow structures between water and soil. The dynamic flow of water and other viscous fluids through granular soil is commonly modeled by Darcy's law with time-independent permeability coefficients. Experimental studies show that this law is not valid for large nonlaminar fluid velocities which may develop under high hydraulic gradients. Furthermore, particle rearrangements leading to substantial variations in porosity are prevalent during soil liquefaction and other extreme flow driven phenomena, such as piping. These variations in porosity affect significantly the soil hydraulic conductivity and stiffness characteristics. Experimental investigations furnish only macroscopic phenomenological characterisation of these complex response mechanism. Microscopic numerical analyses provide effective alternative tools to assess the hydraulic and deformation characteristics or granular soil in a coupled fashion.

As of today, the coupling discrete element method and the lattice Boltzmann method has been proposed [RR18]. The free surface flow model proposed in this thesis is very suitable for coupling soil and fluid simulation Hence, the extensions to compute the moving boundary by the immersed boundary method with AMR [LS12, VPB14, JLZ15] is also important for an application for civil engineering.

References

[FL14] Fakhari, A. and Lee, T. Finite-difference lattice Boltzmann method with a block-structured adaptive-mesh-refinement technique. *Phys. Rev. E*, 89:033310, 2014.

[FL15] Fakhari, A. and Lee, T. Numerics of the lattice boltzmann method on nonuniform grids: Standard LBM and finite-difference LBM. *Comput. Fluids*, 107:205–213, 2015.

[GZZ11] Guo, Z., Zheng, C., and Zhao, T. A lattice BGK scheme with general propagation. *J. Sci. Comput.*, 16:569–585, 2011.

[JLZ15] Ji, H., Lien, F. S., and Zhang, F. A GPU-accelerated adaptive mesh refinement for immersed boundary methods. *Comput. Fluids*, 118:131–147, 2015.

[LS12] Li, Z. and Song, P. An adaptive mesh refinement strategy for Immersed Boundary/Interface methods. *Commun. Comput. Phys.*, 12(2):515–527, 2012.

[RR18] Rettinger, C. and Rüde, U. A coupled lattice Boltzmann method and discrete element method for discrete particle simulations of particulate flows. *Comput. Fluids*, 172:706–719, 2018.

[VPB14] Vanella, M., Posa, A., and Balaras, E. Adaptive mesh refinement for immersed boundary methods. *J. Fluids Eng.*, 136(4):040901:040901–9, 2014.