長岡技術科学大学
Nagaoka University of Technology

Application of macro and mesoscopic numerical models to hydraulic problems with solid substances

「固体を含む水理学的問題に対するマクロおよびメソスコピックな数値モデルの適用」

A dissertation
Submitted to the Graduate School of Engineering of the
Nagaoka University of Technology

By
Ayurzana Badarch (14701491)

In Partial Fulfillment for the Degree of
Doctor of Engineering

Approved by the Dissertation Committee:
Prof. Dr. Hosoyamada Tokuzo, Academic supervisor
Prof. Dr. Minjiao Lu, Co-examiner
Prof. Sc.D. Hidetsugu Kitatani, Co-examiner
Prof. Dr. Tsutomu Takahashi, Co-examiner
Assoc. Prof. Dr. Toshiro Kumakura, Co-examiner

Nagaoka University of Technology
Nagaoka, Japan
June, 2017

ABSTRACT

Sediment and ice, solid substances, can be found in rivers in cold regions. In some case, they are a problem together as considered often problems separately. As approaching to the ultimate purpose of modeling the interactions between sediment and ice, this thesis properly aimed to study sediment and ice problems separately in two different modeling scales, respectively. Numerical studies of sediments and ices have been done in order to understand their physics in fluid flows and to find whether the approach to accomplish the ultimate purpose found in different modeling scales. Herein the studies are organized in two parts.

The sediment problem in an estuary of the Ohkouzu diversion channel of the Shinano River studied by macroscopic models is discussed in Part 1. The macroscopic models include the Navier-Stokes equations simplified by the Boussinesq approximation for estuarine flows and the advection-diffusion equation and a novel sediment particle tracking method for the sediment transports. The particle tracking method is featured by a simple model accounting a flocculation effect for sediments and provides quantitative results involving sediment local and spatial characteristics in transport and interactions with estuarine hydrodynamics.

For the ice problem, as mesoscopic modeling, the lattice Boltzmann methods are proposed as a numerical framework in Part 2. Lattice Boltzmann models for free surface flows, heat transports with liquid-solid phase transitions and fluid-solid interactions are discussed and validated. The results of validation and testing applications for the proposed numerical framework indicated the potential of the method to be applied to real field problems. Applications for the open water forming mechanism in ice covered-outlet channel of a small hydropower plant results the detailed interpretation of the open water forming process in open channel flows.

Consequently, the modeling of the sediment and ice provides confidence for the ultimate purpose that can be modeled whether in macro and mesoscopic modeling. Particularly, the lattice Boltzmann method has advantages to bring a simple solution for the complex physics such like the sediment and ice.

Key words:

Sediment transport modeling, Sediment particle tracking method, Ice-free surface flow modeling, Thermal-free surface-immersed boundary-lattice Boltzmann method

QR codes for: Author’s profile and Ph.D. defense video

Abstract in Japanese and Mongolian

本論文では、土砂と氷を含んだ水理学における諸問題について焦点を当てた研究を取り扱っている。土砂と氷の水理学的問題は流れにおいて固体が浮遊しているという点において共通点を有している。これらの現象を数値モデルによって表現する手法はマクロスコピックおよびメソスコピックという観点に分けられる。

具体的には河口における土砂の輸送に対しては海水によるフロック過程を考慮した粒子追跡モデルをマクロスコピックモデル、氷の輸送に対しては固液二相の相変化と自由水面の変動に対応した格子ボルツマン法をメソスコピックモデルとしてそれぞれ開発し、複雑な流れ場に対応できるモデルとした。開発した数値計算モデルは信濃川河口やモンゴルにおける水力発電施設に適用され、計算結果と計測結果には良好な一致が見られた。

Энэхүү докторын судалгааны ажил нь шингэний урсгал дахь хагшаас болон мөсний үзэгдлийг тоон загварчлалын аргаар судалсан тухай өгүүлнэ. Шингэний урсгал дахь хагшаас болон мөс нь хатуу биет гэдэг ерөнхий ойлголтоор хоорондоо холбогдоно. Гэвч шингэний урсгалыг тайлбарлах цар хүрээнээс хамаарч эдгээр үзэгдлийг тус тусд нь макро болон мезо-орчны загваруудыг ашиглан судалсан юм.

Тухайлбал голын адаг дахь хагшаасны зөөгдөх хөдөлгөөнд зориулсан макро-орчны загваруудыг хагшаасны бөөгнөрөх үзэгдлийг илэрхийлэхүйц хагшаасны мөхлөгийг мөшгөн тодорхойлох нэн шинэ аргын хамт боловсруулсан юм. Харин чөлөөт гадаргуутай шингэний урсгалд байх мөсний үзэгдэлд зориулан шингэн ба хатуу төлөвийн шилжилтийг тооцох сүлжээний Больцманы арга дээр үндэслэсэн тооцон бодох нэгдэл бүхий аргачлалыг мезо-орчны загварчлал гэсэн нэрийн дор шинээр үйлдсэн юм. Дээрхи макро ба мезо-орчны загварчлалуудыг Шинано голын (Япон) адаг дахь хагшаасны хөдөлгөөн болон Тосонцэнгэл усан цахилгаан станцын (Монгол) ус зайлуулах сувгийн мөсний үзэгдэлд тус тус ашиглаж зохих үр дүнгүүдийг гарган авч шинжилсэн болно. Тоон загварчлалын үр дүнгүүд дээрх газруудад хийсэн ажиглалтын үр дүнтэй нийцэж байсан гэдгийг дурьдах нь зүйтэй юм.

DEDICATED

To my live Gods father Badarch Tseden,

mother Jigjidmaa Oyunchimeg,

My late Grandmother Lojoo Purewsuren for their great merits

And

To my eternal love Erdenebayar Tungalagtamir for her endless care.

ACKNOWLEDGEMENT

First of all, I would like to express my sincere gratitude to my supervisor Prof. Dr. Hosoyamada Tokuzo for his great support, guidance and encouragement during this Ph.D. course. I was so excited when he accepted me as a Ph.D. student and I feel always the same when we reach at particular achievements like run codes successfully, publishing papers, participating in conferences etc. and now completing my Ph.D. thesis. As Mongolian, I am deeply indebted to my supervisor in the rest of my life.

I would like to offer my special thanks to Prof. Dr. Baranchuluun Sh and Dugarmaa M for their warmth supports, to Prof. Dr. Chuluunhuyag S for his faith in me, to Dr. Baasandash Ch and Prof. Dr. Duinkherjaw Y for their valuable advices. My intellectual gratitude goes to Dr. Janchivdorj L, Mr. Ganzorig Sh, Batbayar Ts, Boldbaatar B, Dr. Ganbat D, Chinzorig S, Oyuntsatsral Ts and Mrs. Odsuren B for inspiring messages and helps. I have greatly benefited from my university friends Mr. Batzorig B, Myagmardorj G and other friends. I also want to thank my high school friends Mr. Baasantsogt J, Bayarmagnai M, Gantulga P, Amgaa A, Ms. Munguntuya Ts and others for their priceless friendships.

I have received generous supports from teachers of Mongolian University of Science and Technology, namely Prof. Davaajamts Sh, Mr. Bold P, Nasanbayar N, Boldbaatar N, Dr. Tuul Y, Mrs. Altantsetseg, Baasansuren S and Boldmaa L. I am appreciated to Prof. Dr. Scott Kenner, Mr. Nasanbayar N and their family for making our times wonderful in US and Germany.

I am deeply grateful to Mr. Ohtake Takeshi for his great tutor to get started new life and study in Nagaoka. He also had put his great contributions to my thesis. I also want to say thanks to Mr. Nil Illa Mestre for being a good lab-mate ever. And thanks go to all NUT-International students, including Mongolian students and Japanese students from the hydraulic engineering laboratory who shared their precious moment with me during their study in the NUT.

I thank Assoc. Prof. Dr. Inukai Naoyuki for his kind discussion and invitation for field trips. I appreciate all professors of the NUT who taught me courses and helped directly and indirectly. And discussions with Prof. Dr. Khenmedekh L, Asst. Prof. Dr. Fukumoto Y and Tomohiro D have been illuminating me to see things differently, thank you very much. I am thankful to Mrs Ikarashi Ch and other staffs of Kokusai-ka and Gakusei shein-ka, Mr. Yamamoto H from the hydraulic engineering lab, Mr. Sakura Y and Dr. Wada M from the division of Institutional Strategy, Ms. Kiryu E from Information processing center for their supports.

I am honored to be a Ph.D. student funded by the MEXT scholarship. I owe a very important debt to the people of Japan and Japanese government. Without the MEXT scholarship, I and many other international students could not complete studies in Japan.

And finally, thanks to my great family, brothers and sisters, relatives for all the support over the years. Especially, I thank my love Tungalagtamir E for her support and contribution to this thesis. She always provides me happy and lovely environment. Thank you all good people who know me and not mentioned here, I say “thanks” to you in person when we meet.

Өндөр төрийн минь сүлд ивээж
Өнөр түмний минь заяа түшив.

Ayurzana Badarch

Jun 2017, Japan

Abbreviations

1D	One dimensional
2D	Two dimensional
3D	Three dimensional
ACM	Artificial compressibility method
ADE	Advection-diffusion equation
BBGKY	Bogoliubov, Born and Green, Kirkwood and Yvon
BGK	Bhatnagar-Gross-Krook
CDE	Convection-diffusion equation
CPU	Central processing unit
CUDA	Compute Unified Device Architecture
D1Q3	1 dimension, 3 velocities
D2Q9	2 dimension, 9 velocities
D3Q15	3 dimension, 15 velocities
DEM	Digital elevation model
DF	Distribution function
DNS	Direct numerical simulation
EFG	Element-free Galerkin method
ELBM	Entropic LBM
EOS	Equation of State
ExLBM	Extended LBM
FDM	Finite difference method
FEM	Finite Element method
FHP	Frisch, Hasslacher and Pomeau
FPM	Finite point method
FVM	Finite volume method
GFLOPs	Giga floating point operations per second
GPGPU	General-Purpose computing on Graphics Processing Units
HPP	Hardy, Pomeau and Pazzis
IB	Immersed Boundary
LB	Lattice Boltzmann
LBE	Lattice Boltzmann equation
LBM	Lattice Boltzmann method
LES	Largy Eddy Simulation
LGA	Lattice Gas automata
MAC	Marker and Cell
MPS	Moving particle semi-implicict method
MRT	Multiple-relaxation time
NSE	Navier-Stokes equations
NUT	Nagaoka University of Technology
NZ	McNamara and Zanetti
PM	Projection method
PTM	the particle tracking method
rms	Root-mean-square
SHPP	small hydropower plant
SIMPLE	Semi-implicit method for pressure-linked equations
SMAC	Simplified MAC
SOLA	Solution Algorithm
SPH	Smoothed particle hydrodynamics
SRT	Single Relaxation time
SS	Suspended solids
StLBM	Standard LBM
SWA	Shallow water approximation
TRT	Two-relaxation time
VoF	Volume of Fluid
T-FS-IB-LBM	Thermal-free surface-immersed boundary-LBM

DECLARATIONS AND EXTERNAL PUBLICATIONS

Herein I assure that this thesis has been composed solely by me under supervision of Prof. Dr. Hosoyamada Tokuzo and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by references, the works presented are entirely my own as the following papers co-authored by me during my Ph.D. course.

Peer-reviewed journal paper

Badarch Ayurzana, Tokuzo Hosoyamada: Application of the lattice Boltzmann method to liquid-solid phase change in free surface flow: an example of Mongolian small hydropower plant. *Journal of Japan Society of Civil Engineers, Ser. B1 (Hydraulic Engineering)*, Vol.73, No.4, pp: I_607-I_612, 2017. DOI:10.2208/jscejhe.73.I_607 (English paper)

Takeshi OHTAKE, Ayurzana BADARCH, Tokuzo HOSOYAMADA: A numerical study of sediment distribution discharged from a river considering enhanced settling due to flocculation caused by mixing process of river and sea water. *Journal of Japan Society of Civil Engineers, Ser. B2 (Coastal Engineering)*, Vol.72, No.2, pp: I_1309-I_1314, 2016. DOI:10.2208/kaigan.72.I_1309 (Japanese paper with English abstract)

B Ayurzana: Free surface flow through porous media by lattice Boltzmann method. *Proceedings of the Mongolian Academy of Sciences*, Vol.56, No.01 (217), pp: 42-54, 2016. DOI:10.5564/pmas.v56i01.674 (Mongolian paper with English abstract)

Takeshi OTAKE, Ayurzana BADARCH, Tokuzo HOSOYAMADA: Numerical study of settling and transport of sediment discharged from large river mouth: an application to the Okhouzu river mouth. *Journal of Japan Society of Civil Engineers, Ser. B1 (Hydraulic Engineering)*, Vol.72, No.4, pp: I_355-I_360, 2016. DOI:10.2208/jscejhe.72.I_355 (Japanese paper with English abstract)

Ayurzana Badarch, Khenmedekh Lochin and Hosoyamada Tokuzo: Parallel implementation of Entropic lattice Boltzmann method for flow past a circular cylinder at high Reynolds number, *Transactions on GIGAKU* , Vol.4, No.1, pp: 04006/1-8, 2017 (English paper)

Conference proceedings

Tokuzo Hosoyamada, Ayurzana Badarch, Takeshi Otake: *Sediment Discharge Modeling at the Estuary of Ohkouzu Diversion Channel*. Two-pHase modElling for Sediment dynamicS in geophysical flows-THESIS-2016, Research and Development Initiative, Chuo University, Tokyo, Japan; pp:129-132, 09/2016.

Ayurzana Badarch, Hosoyamada Tokuzo, Narantsogt Nasanbayar: *Hydraulics application of the Free-surface Lattice Boltzmann method*. Proceedings of 2016 11th International Forum on Strategic Technology (IFOST), IEEE, Novosibirsk, Russia; Vol.11, No.2, pp: 195-199, 06/2016. DOI:10.1109/IFOST.2016.7884225.

Ayurzana Badarch, Hosoyamada Tokuzo: *Phase change simulations of water near its density inversion point by Lattice Boltzmann method*. Proceedings of the 23th IAHR International Symposium on Ice, Ann Arbor, Michigan USA; pp: 01-08, 06/2016.

Table of Contents

1 INTRODUCTION

1.1 Research objectives and scopes

The main objective of this thesis is to model fluid flows with solid substances by using numerical approaches. In general, the fluid flows in hydraulics are considered. The word solid substance stands for a solid matter in a fluid flow. More precisely, the sediment and ice are the main interest of solid substances rather than the other solid object that founds in a flow or bounds a fluid. Sediments and ices are different but share common things such as both exist in a natural river in cold regions and are engineering problems (Ettema, 2008) in general. These particular interests are explained in motivation section. The general questions for the main objective are (1) how many possible approaches to model are exist, (2) how to model the problems and (3) can we provide a general model that valid for the purpose or for every flow with any solid substances. The question (1) leads the different descriptions of fluid flows. Furthermore, the different descriptions have many existing models that can be applied to the problem. The specific models should be selected and studied for the problem. The question (2) provides research works based on the answers from the question (1). And during the research work for the model development, the question (3) is evaluated and clarified.

There are three different description scales for fluid flows (Guo & Shu, 2013) described in Section 1.3. Theoretically, we have an opportunity to get involved with two approaches namely macro and mesoscopic modeling. With these approaches, we conducted studies investigating the existing numerical models and developing novel extensions to the models considering the specific objectives and motivations. Thus, this thesis focuses on the macro and mesoscopic numerical models of the sediment and ice in different flow regions. The scopes of this thesis are the finite difference method for sediment transport in an estuary and the lattice Boltzmann modeling of ice melting or water freezing in open channel flows. The latter one can be stated as a liquid-solid phase transition in free surface flow (Ayurzana & Hosoyamada, 2017).

As a macroscopic modeling for a fluid flow with solid substances, the Navier-Stokes equations (NSE) formulated in the Boussinesq approximation governs a fluid flow, whereas the traditional advection-diffusion equation (ADE) and a novel particle tracking method (PTM) models the sediment transport. The fluid governing equations are solved by the solution algorithm (SOLA) (Hirt, et al., 1975) with the finite difference approximations for sediment transport models. One particular highlight of the PTM is an introduction of a practical flocculation model in an estuary. From the mesoscopic models, the lattice Boltzmann methods (LBM) are emerging numerical tools (Chen & D.Doolen, 1998) in computational fluid dynamics (CFD). Herein lattice Boltzmann (LB) solutions for fluid flows, scalar transports, free surface flows and fluid-solid interactions are studied with the implementation techniques. Firstly, the existing models for these are combined with each other for the liquid-solid phase transition in free surface flows as a main contribution. Among with the several attractive new solutions, an immersed boundary method is applied to the phase transition as well as a freely moving body in free surface flows. I hope this thesis will share useful knowledge for readers who are interested in the above methods and problems.

1.2 Motivations

Fluids are always interacting with solids. Fluid flows are bounded by solids and transport solids with a fluid (G. K. Batchelor, 1967). However, the most of researches have devoted to understanding effects of the solid that bounds a fluid and this is termed as a fluid-structure interaction (Belytschko, 1980). In cold nature, a common fluid, the water, interacts with two major solid substances named sediment and ice (Ettema, 2002), (Prowse, 1993). Those are not boundaries, but substances transporting with water flows. Fluid flows with such substances are motivations for this study. For example, hydropower plants in cold region face problems caused by sediments and ices (Tuthill, 1998). During a flood event, a large amount of sediments transports through a river and traps by hydraulic structures at upstream side as sedimentation (Morris & Fan, 1998). The sedimentation reduces capacity of a reservoir and it causes the reduction of the power production. Meanwhile, the ice makes a huge problem during winter time (Gebre, et al., 2013). Ice and sediments can be a problem at the same time during the spring, since a large amount of broken ice as an ice jam prompts the sediments from the bottom and bank of a river (Ettema, 2002). However, in this thesis, we do not intend to model these problems together. Rather that we explore and develop the numerical models for these problems separately as research parts in this thesis. Generally, our implicit aim is remained to lay the foundation for the models to solve these problems simultaneously.

For the sediment problem, we devoted our study to the sediment transport in an estuary. Estuaries are unique places where sediment and fluid flows interact in complicated dynamics (細山田得三, et al., 2001). The density current reasons interesting transport phenomena for sediments while sea water activates the flocculation process for cohesive sediments (Einstein & Krone, 1961). These were the specific motivations to study and develop numerical models in Part 1. On the other hand, the LBM had been expecting many extensions from the researcher to every field of fluid dynamics (Guo & Shu, 2013). The LBM was also starting to be applied to hydraulic problems (Karpiński, et al., 2013). Free surface flows and phase change processes were rarely connected in the LBM. More elaborations for the phase change have to be done. Especially, the immersed boundary method has been shown potentials for fluid-structure interactions (Mittal & Iaccarino, 2005). These were the motivations for the numerical works reported in Part 2.

Besides, I had internal motivations to complete my Ph.D. course with these studies with respect to build my skills. My numerical career starts with my Ph.D. course and is aged 3 years now. Before starting my Ph.D. course in the NUT, I have many wishes to do numerical works related to fluid dynamics. But that time, I did not know how to get started and even did not know how numerical studies get done. There was nobody to mentor me in my environment. My motivations were too specific and a few, namely to learn numerical methods, coding the numerical models and to solve common problems in hydraulics. I found a seemingly impossible plan when I was a first-year Ph.D. student that if I have a time, I would study one of the above problems: the sediment or ice. Attractive features of the computational fluid dynamics with the other great factors surrounds brought me here with the accomplishment of a most percent of my wishes. With the motivations, the research questions in Section 1.1 are herein answered along with the outcomes of the numerical studies during this Ph.D. course.

1.3 Descriptions of fluid flow in different scales

Let us discuss about different descriptions of fluid flow in fluid dynamics. Consider a proper heavy ball falls into stagnant water from a certain position above the free surface of water in confined domain. Humans with rare eyes only can see the splash of water due to falling ball and then can see the free surface of water sloshing above the sinking ball into the water. Interestingly, humans with special interest, say explorer, can see spatial distribution of physical variables of splashing water, sloshing waves and trajectory of the ball with experimental tools. Even advanced one with special sight, say engineer or physicist, can predict the spatial distribution of physical variables without falling ball into water in reality with modern tools and can show variables with easily readable colorful images and graphs, as shown in **Figure 1** (a). In this case, some researchers take the assumption that the fluid is a continuum and it is regardless that the fluid is made of molecules and atoms. This is generally a **macroscopic scale** of fluid flow. In the different point, with knowledge of the science about the matter, one wants to explore the fluid flow at the molecular level to understand the molecular interaction between solid and liquid states, maybe. This can be said **microscopic scale** of fluid flow and illustration shown in **Figure 1** (c). At the molecular level, fluid molecules are traveling at a high speed and are doing multiple collisions with other molecules and bombarding the bounding solid molecules. If one groups molecules at a proper range as an ensemble of molecules and can observe the behavior of the group molecules. The proper range in length should exist between the macroscopic scale and microscopic scale in order to be said **mesoscopic scale**. As seen in **Figure 1** (b), the connection between micro- and macroscopic scale must be interpreted in the mesoscopic scale.

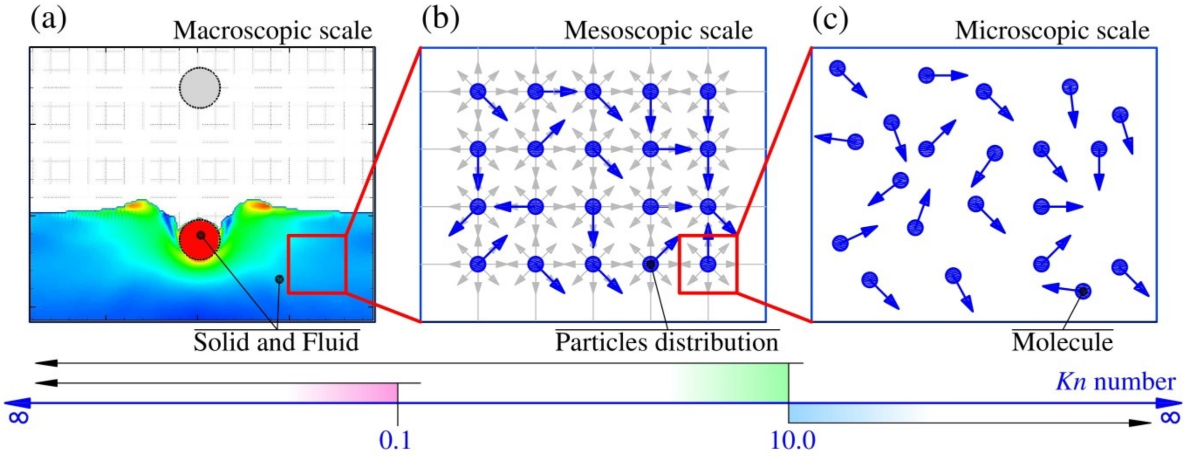


Figure 1. Illustration for different scales of fluid flows.

1.3.1 Macroscopic scale

In the macroscopic scale, physical variables of a continuous flow field such as density, velocity, pressure and temperature are considered. Fluids are considered continuum fields and fluid flows are described by continuum models. More precisely, a macroscopic continuum model is valid, if the Knudsen number is near or lower than one (Laurendeau, 2005). The Knudsen number (*Kn*) is defined as the ratio of a mean free path (λ) of molecules to a characteristic length (L) of a fluid flow. In the continuum model, the spatial distribution of physical variables with regards to others under the physical laws such as mass and momentum conservation, and expressed by a set of partial differential equations (Pope, 2001), respectively:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0\,, \tag{1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \tau\,, \tag{2}$$

where ρ is the density [kg m^{-3}], $\mathbf{u}$ is the velocity vector [m s^{-1}], p is the pressure term [pa] and τ is the deviatoric stress tensor including the viscosity. The continuity equation (Eq.(1)) and Navier-Stokes equations (Eq.(2)) contain ρ, $\mathbf{u}$ and p as unknown, dependent variables, and one further scalar equation is needed to make it possible the determination of fluid flows (G. K. Batchelor, 1967). This additional relation can be provided by the equation of state (EOS) for the fluid, which may be generally written as

$$f(p, \rho, T) = 0\,, \tag{3}$$

where T is the temperature. For many fluids, the viscosity μ significantly depends on the temperature. If the temperature difference in the fluid domain is small enough for the variation of μ, it is taken as uniform over the fluid. In that case, the flow system is called isothermal system. When the appropriate temperature difference exists in the flow domain it is necessary to regard μ as a function of position and the complexity of the deviatoric stress tensor is increased. In this case, the temperature in the system is not constant, the system is known as non-isothermal system and one needs to consider the energy conservation equation (Guo & Shu, 2013)

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho \mathbf{u} e) = -\nabla \cdot \mathbf{q} - p\nabla \cdot \mathbf{u} + \tau : \nabla \mathbf{u} \tag{4}$$

where e is the internal energy, q is the heat flux, which is usually related to the temperature gradient following the Fourier's law.

A solution of these differential equations for a fluid flow is difficult to be found by an analytic way due to many reasons such as nonlinearity and complex boundary of the flow domain, etc. To solve the equations numerically, the flow domain should be discretized into finite sets of elements as grids or meshes. With these elements, the above differential equations need to be approximated into linear algebraic equations using a numerical discretization method such as the finite difference (FDM) and finite volume method (FVM). The simple algebraic versions of the differential equations compute the time evolution of physical variables with the initial and boundary conditions in iterations. Part 1 of this thesis discusses the macroscopic solution for a fluid flow in case of sediment transportations in a river mouth.

1.3.2 Microscopic scale

In a smaller scale, the medium can be considered as space containing particles like the substance made of molecules. A particle in the space can be indexed by i to be distinguished with other particles in an account. The particle has a mass m_i, a velocity $\boldsymbol{\xi}_i$, at a position $\mathbf{x}_i$=(x_i, y_i, *z*$_i$) and a time moment *t*. The motion of that particle is described by the Newton's equation of motion

$$m_i\ddot{\mathbf{x}}_i = \mathbf{F}_i\,, \tag{5}$$

where $\mathbf{F}_i$ is the total force experienced by the particle. Another description of particle motion is the Hamiltonian mechanics about N-body system, where more generalized coordinates that are particle positions and momenta are considered.

As the nature of molecules motion, particles collide with other particles in a system and the collision results interaction forces between the particles, usually termed as interatomic potential. The total force term, including interaction force between the particles can be written as

$$\mathbf{F}_i = \sum_{j=1,j\neq i}^{N} -\phi(|\mathbf{r}_{i,j}|) + \mathbf{G}_i\,, \tag{6}$$

where ϕ is the interaction potential between colliding pairs *i* and *j* (Rapaport, 2003), $\mathbf{r}_{i,j}$ is the distance vector between two particles, *N* is a number of particles in the system and $\mathbf{G}_i$ can be an external force field involving the gravity. Solution of Eq.(5) gives only position and velocity of each particle at every time and procedure to solve particle motion is referred as molecular dynamic simulation. Molecular dynamic simulation is expensive in terms of time and memory of a computer, thus a simulation domain is often limited in a very small space. The governing equations are integrated using a numerical integration scheme with boundary conditions controlling pressure and temperature condition. Usually, during the integration, molecular dynamics simulation suffers with accumulation of errors. To reduce the cumulative error for the simulation, higher order numerical methods such as the predictor-corrector scheme, Verlet and Runge-Kutta schemes, are often employed. Numerical methods from the macroscopic scale are often applicable for the molecular dynamics.

It is difficult to determine macroscopic fluid flow characteristic with the information of particles defined by the molecular dynamic simulation in a direct way. In addition, physical parameters, such as viscosity and thermal diffusivity of fluid, are meaningless in microscale description and they are the result of the molecules interaction and transitions in a much bigger scale. However, it is possible to get approximated quantities for the macroscopic scale from the information on the microscopic scale by taking the ensemble average. For instance, temperature and pressure can be described by the ensemble averages of the kinetic energies of the particles and frequencies of the particle bombardments on the boundaries, respectively. Physical properties of fluid can be also measured according to the linear response theory (Guo & Shu, 2013). Thus, this scale of fluids is excluded from the study of interest.

1.3.3 Mesoscopic scale

This is well studied scale in terms of the description of gas flow. Generally, knowledge about a mesoscopic view of fluid flow fills the gap between the microscopic and macroscopic scales. Also, it has been discovered to explain fluid phenomena at a range where the theory of the both scales is invalid. The main assumption is that the behavior of the single particle in microscopic scale can't describe macroscopic phenomena of fluids; however, a characteristic of a group of particles might be important in the macroscopic scale. In the simplest way, the fluid density can be defined by the sum of the particle mass in a very small volume (Viggen, 2014) as

$$\rho(\mathbf{x},t) = \lim_{V\to 0}\left(\frac{1}{V}\sum_{x_i\in V}^{N} m_i\right). \tag{7}$$

In the selected volume, the average motion of particles can be expressed by the momentum as

$$\rho\mathbf{u}(\mathbf{x},t) = \lim_{V\to 0}\left(\frac{1}{V}\sum_{x_i\in V}^{N} m_i\,\boldsymbol{\xi}_i\right), \tag{8}$$

and temperature can be found by assuming the EOS of gas at a critical condition and the kinetic energy of particles as

$$T(\mathbf{x},t) = \frac{1}{3k}\lim_{V\to 0}\left(\frac{1}{V}\sum_{x_i\in V}^{N} m_i\,|\boldsymbol{\xi}_i|^2\right), \tag{9}$$

where k is the Boltzmann constant. Considering many particles in the system, calculation approaches to appreciate a macroscopic scale, but it leads complexity for the calculation. From the particle motion, gas pressure can be easily expressed by the kinetic energy of particles

$$p = \frac{2N}{3V}\left(\frac{mu_p^2}{2}\right) = \frac{2n}{3}KE\,, \tag{10}$$

where N is the number of molecules in unit volume, KE is a gas kinetic energy and u_p is the average velocity of molecules in the system. When introducing the gas pressure into the EOS, a relation of temperature and velocity can be found

$$T = \frac{2}{3}k\left(\frac{1}{2}mu_p^2\right), \tag{11}$$

where m is the molecule mass (particle mass). This relation is known as the kinetic temperature and is the background of Eq.(9). For the purpose of defining gas pressure and temperature, the velocity in Eq.(10) and Eq.(11) can be defined by Eq.(8). However, as so far mentioned, it is a difficult way to define the averaged velocity if the total number of particles is enormous. To see the correlation of the particle velocity, we can rewrite Eq.(11) as a function of temperature

$$u_p = \sqrt{\frac{3kT}{m}}\,. \tag{12}$$

From Eq.(12), it is obvious that average velocity of particles proportional to temperature. In other words, the particle velocity can be expressed in function of temperature and molecular/particle mass. It is also true that all particles can't move at similar velocity in the system. The velocity of a particle is delayed or sped up depending on its collision pattern with other particles. The collision is measureless and it happens with probability. Hence it is possible to define particle velocity based on the probability at a certain temperature. This was the idea of Maxwell and later Boltzmann to define particle velocity distribution function, ignoring the characteristic of each particle: a position and velocity of each particle is not important. The velocity probability of particles in the system can be described by the Maxwell speed distribution function as

$$f(c) = 4\pi \left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} c^2 e^{-\frac{mc^2}{2kt}} , \tag{13}$$

where c is the probable speed of the particle and $f(c)$ is the fraction of the molecules in the system at a range between velocity c and c+dc. With Eq.(13), we can define several characteristic velocity values of particles as shown in **Figure 2**, where speed distributions of dioxygen (O_2) is presented as an example.

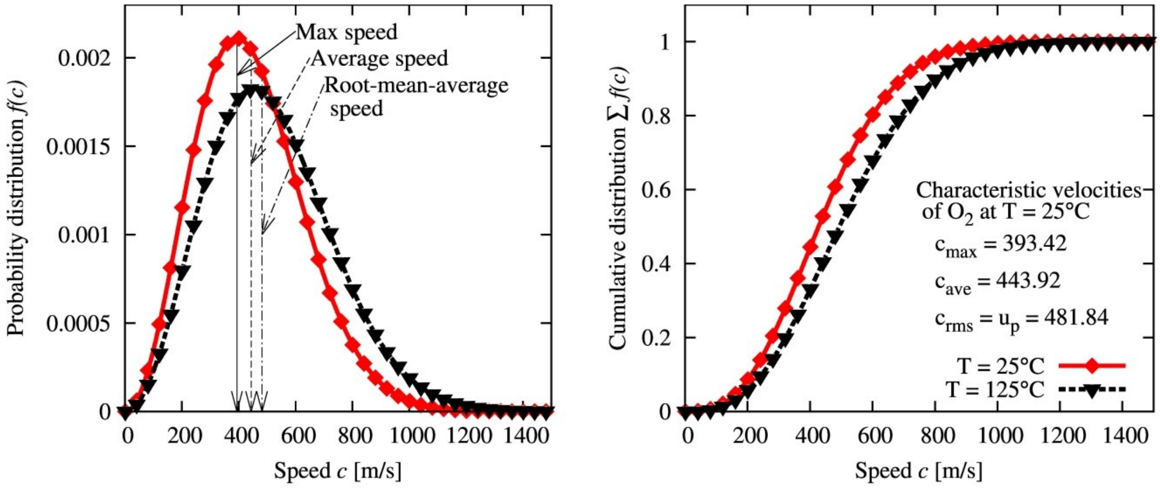


Figure 2. Probability and cumulative distribution function of dioxygen at two different temperatures.

The speed distribution function gives several characteristic speeds of particles in the system. For instance, the root-mean-average speed of the distribution function is the same as a velocity defined in Eq.(12). As temperature increases, speed distribution becomes smooth and the peak of the distribution moves in the right direction. Integration of the speed distribution function is a unit by its definition that any one particle can be found in the curve of the speed distribution function. One can derive the Boltzmann distribution function integrating the Maxwell speed distribution in velocity field. The Maxwell distribution function shows the distribution of particles by its kinetic energy, while the Boltzmann distribution function shows the distribution of particles by its potential energy. Equation (13) was detailed by Ludwig Boltzmann in 1872 to considering the total energy in the system and formulated as

$$f(E) = Ae^{-\frac{E}{kT}} . \tag{14}$$

The Maxwell and Boltzmann distributions are covered in Eq.(14) and it is known as the Maxwell-Boltzmann distribution function. This distribution is the cornerstone of the gas kinetic theory and thermodynamics. It is also used in the mesoscopic numerical model for fluid flows in part 2.

1.4 Outline of this thesis

This thesis is organized in two main parts to maintain the objectives of the thesis. Part 1 discuss about fluid flow and sediment transport simulations with macroscopic numerical models. Part 2 explores about the mesoscopic approach to flow simulations in particular hydraulic applications. The thesis outline is given in **Figure 3**. The thesis is written in simple and very detailed manners so that the new beginners for these models will get comprehensive understandings.

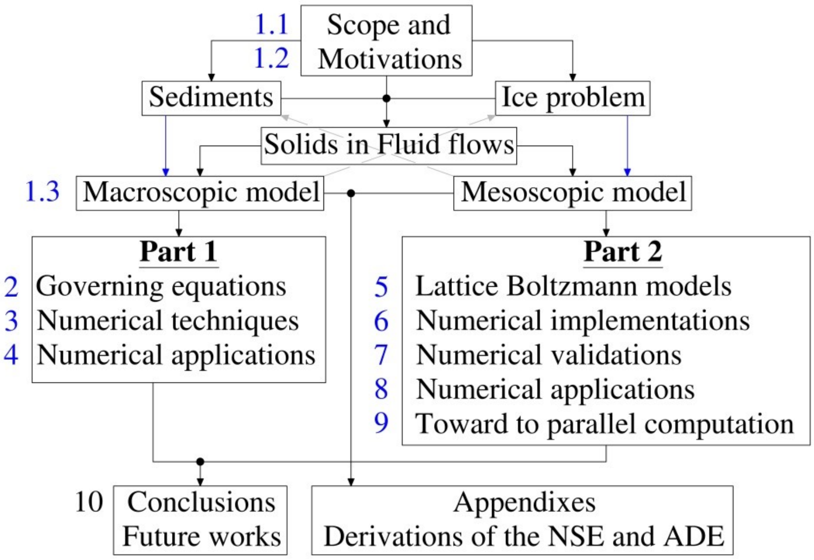


Figure 3. Dissertation outline. Light blue numbers at the front are Chapter numbers in the thesis. Black solid lines show direct flows, while the dotted line shows a potential of applications.

In part 1, the NSE with the approximation of Boussinesq and shallow water precedes the ADE and novel sediment PTM. For a particular application, the Smagorinsky sub-grid scale model is presented for the fluid governing equations. Then the numerical techniques for the models are discussed in great details. Before the application, a simple numerical validation, the lock-exchange problem, for the sediment transport numerical models are given. As a problem, the sediment transport in the estuary of the Ohkouzu diversion channel in the Japan Sea is investigated by the 2 and 3D numerical models.

In Part 2, the lattice Boltzmann (LB) models for fluid flows including free surfaces, scalar transports, phase changes and fluid-structure interactions. Then, the numerical implementations of these LB models are discussed in detail. After that, validations for fluid flows, liquid-solid phase changes and combination of above problems are given. The LBM is applied to the ice melting problems for the 2D open channel hydraulics in the latter. And finally an attempt to the parallel computation with the LBM is given in concise. The LB models base on the LBM with the BGK collision, including the Smagorinsky turbulent model, double distribution functions approach for scalar transport, enthalpy-based method for phase change, immersed boundary method, and the Entropic lattice Boltzmann method (ELBM).

Conclusions and future works summarize the studies in two parts and suggest possible studies in the future. QR codes through this dissertation direct to the animated results. Note that the symbol notations may coincide for some terms in two parts.

PART 1: MACROSCOPIC MODELING FOR SEDIMENT TRANSPORTS

includes the descriptions and results of research work using the macroscopic models for fluid flow with sediment transport.

2 GOVERNING EQUATIONS

The macroscopic fluid governing equations designed for particular applications are presented here.

2.1 Navier-Stokes equations (NSE)

As briefly discussed in Section 1.3.1, fluid flows qualified as $Kn \equiv \lambda/l \ll 1$ are considered as continuum fields (Pope, 2001). In the continuum field, common physical quantities averaged over a volume of size $V = l^{*3}$, where l^* is the intermediate length scale ($\lambda \ll l^* \ll l$), are the density, velocity, pressure and external forces. With the statistic description of molecular interactions and transports (Kerson, 1987), the viscosity, the physical quantities are related to each other in terms of space and time conserving the momentum. The relations of the physical quantities and properties are called the Navier-Stokes equations that govern the continuum fluid flow (Pope, 2001), (G. K. Batchelor, 1967), (Болд, et al., 2013) in macroscopic scale. For the incompressible fluid flow, the mass and momentum conservation equations in Eq.(1) and (2) can be rewritten as

$$\nabla \cdot u = 0 \qquad \text{The continuity equation}$$

$$\left.\begin{aligned}
\frac{\partial u_x}{\partial t} + u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y} + u_z\frac{\partial u_x}{\partial z} &= F_{mx} - \frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\nabla^2 u_x \\
\frac{\partial u_y}{\partial t} + u_x\frac{\partial u_y}{\partial x} + u_y\frac{\partial u_y}{\partial y} + u_z\frac{\partial u_y}{\partial z} &= F_{my} - \frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\nabla^2 u_y \\
\frac{\partial u_z}{\partial t} + u_x\frac{\partial u_z}{\partial x} + u_y\frac{\partial u_z}{\partial y} + u_z\frac{\partial u_z}{\partial z} &= F_{mz} - \frac{1}{\rho}\frac{\partial p}{\partial z} + \nu\nabla^2 u_z
\end{aligned}\right\} \quad \text{The Navier} - \text{Stokes equation,}$$

where **u** is the fluid velocity of coordinates of x, y, and z, F is the external forces, p is the pressure, ρ is the fluid density and ν is the viscosity of a fluid. Assuming the fluid is only accelerated by the gravitational field, the governing equations become as follows in summation convention

$$\frac{\partial u_i}{\partial x_i} = 0\,, \tag{15}$$

$$\frac{\partial u_i}{\partial t} + u_m \frac{\partial u_i}{\partial x_m} = -\mathrm{g}_i \delta_{i3} - \frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \nabla^2 u_i\,, \tag{16}$$

where g_i is the gravitational acceleration, i is the component index, m is the conventional notation and δ is the Kronecker delta function. Among the unknown variables in Eq.(15) and (16), the pressure can be satisfied the Poisson equation,

$$\nabla^2 p = -\rho \frac{du_m}{dx_n} \frac{\partial u_n}{\partial x_m}\,, \tag{17}$$

that is, in turn, a necessary and sufficient condition for the continuity equation (Pope, 2001). Also, the pressure for the incompressible flow is no longer connected to the density as the equation of state (EOS). However, the density variations due to physical and chemical additions in the fluid must be considered for the pressure, as well for the external force. For instance, either the salt dissolved in water or the sediment immersed in water or heat given to water (non-isothermal fluid flow) results the significant density and viscosity variations in system, which then leads to the onset of fluid flows.

The modeling water body on earth under the atmospheric pressure brings very good simplifications for the NSE. In the following we review two important approximations for the NSE to be convenient for numerical solution methods.

2.1.1 Boussinesq approximation

As a problem, an estuary is where the river and sea water meets. In consequence, the density current happens due to the density difference between river and seawater. A simple method to account the density difference for the governing equations is the Boussinesq approximation. In general discussion, one of the conditions the Boussinesq approximation to be effective (Tritton, 1988) is

$$\frac{\rho'}{\rho_o} \ll 1\,, \tag{18}$$

where ρ_o is the reference density and ρ' is the density variation. We can anticipate that the density of the fluid is

$$\rho = \rho_o + \rho'\,. \tag{19}$$

Multiplying Eq.(16) by ρ and then inserting Eq.(19) into ρ yields

$$(\rho_o + \rho')\left(\frac{\partial u_i}{\partial t} + u_m \frac{\partial u_i}{\partial x_m}\right) = -\mathrm{g}_i \delta_{i3} (\rho_o + \rho') - \frac{\partial p}{\partial x_i} + (\rho_o + \rho') \nu \nabla^2 u_i\,. \tag{20}$$

For flows satisfying the general condition (Eq.(18)) with certain conditions depending on the problems, Boussinesq in 1903 suggested that the density variations in the fluid can be neglected except in the gravity term (Kundu & Cohen, 2004). Applying this approximation to Eq.(20) and dividing the result by ρ_o gives

$$\frac{\partial u_i}{\partial t} + u_m \frac{\partial u_i}{\partial x_m} = -\mathrm{g}_i \delta_{i3} \left(1 + \frac{\rho'}{\rho_o}\right) - \frac{1}{\rho_o} \frac{\partial p}{\partial x_i} + \nu \nabla^2 u_i\,. \tag{21}$$

This is the conservation of momentum with the Boussinesq approximation. The Boussinesq approximation for the conservation of mass (Eq.(1)) gives the incompressible continuity equation given in Eq.(15). It should be noted that some physical or thermal properties such as the viscosity, diffusivity, and heat capacity are treated as constant in the Boussinesq approximation.

2.1.2 Shallow water approximation

Solving the pressure term in the NSE has been a difficult challenge. The pressure term was one of the motivations to develop different methods to solve the NSE. The pressure term also brings the simplification to the NSE without loss of generality. For this, the concept of the shallow water equations can be applied to the NSE to simply define the pressure term as a hydrostatic pressure. We do not derive the shallow water equations, but we use the concept of it for the further simplification to solve the NSE. The condition (Kundu & Cohen, 2004) to use the shallow water approximation (SWA) can be

$$\frac{H}{\lambda} \ll 1 \; \rightarrow \; \frac{H}{L} \ll 1 \,, \tag{22}$$

where H is the water depth (vertical length scale) and λ is the wavelength (L is the horizontal length scale). Under this condition, the effect of the vertical velocity component is negligible small in the conservation of mass and momentum. Thus, from the z component of the momentum equation, one can derive the following relations by eliminating the advection and viscous term:

$$0 = -\mathrm{g}_z\left(1 + \frac{\rho'}{\rho_o}\right) - \frac{1}{\rho_o}\frac{\partial p}{\partial z} \,. \tag{23}$$

This is the hydrostatic pressure equation derived under the SWA. A vertical integration from the bottom ($-h$) to the water surface (η) of Eq.(23) gives the hydrostatic pressure definition,

$$p = -g(\rho_o + \rho')(\eta - h) \,. \tag{24}$$

Along with the simple calculation of the pressure with Eq.(23), the time evolution of the velocities can be defined as

$$\begin{aligned}
&\text{For the x component:} && \frac{\partial u_x}{\partial t} + u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y} + u_z\frac{\partial u_x}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\nabla^2 u_x \\
&\text{For the y component:} && \frac{\partial u_y}{\partial t} + u_x\frac{\partial u_y}{\partial x} + u_y\frac{\partial u_y}{\partial y} + u_z\frac{\partial u_y}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\nabla^2 u_y \,. \\
&\text{For the z component:} && \frac{\partial u_z}{\partial z} = -\frac{\partial u_x}{\partial x} - \frac{\partial u_y}{\partial y}
\end{aligned} \tag{25}$$

2.1.3 Modeling the density variations

Using an example of density current in an estuary, we show how to model the effect of the density variations for the NSE. The density variation, i.e. concentration field, must be modeled by another differential equation involved with the velocity field obtained by the NSE. If the problem is temperature dependent, one should evaluate the temperature field with the additional differential equation: the heat equation (Tritton, 1988), so that the density variation can be calculated from the designated EOS. The differential equation of convection-diffusion is the best adequate equation for scalar type variations, such as temperature and concentration. The change of density induced by the salt concentration, which causes a density current flow in an ocean, can be written as

$$\frac{\partial \rho}{\partial t} + u_m \frac{\partial \rho}{\partial x_m} = A \nabla^2 \rho \, , \tag{26}$$

where ρ is the density of water, u_m is the velocity vector transporting the concentration and A is the diffusion coefficient expressing the rate of diffusion of concentration in a medium. In order to define the temporal evolution of density variations, we can only trace the density deviation instead of total density by substituting Eq.(19) into Eq.(26)

$$\frac{\partial \rho'}{\partial t} + u_m \frac{\partial \rho'}{\partial x_m} = D_s \nabla^2 \rho' \, . \tag{27}$$

In the common case, the scalar diffusion coefficient, D_s, can be considered as a turbulent viscosity coefficient resulted by the turbulent model.

2.2 Smagorinsky turbulent model

In nature and engineering field, most flows are turbulent and some characteristics of turbulence are randomness, nonlinearity, diffusivity, vorticity, and dissipative (Kundu & Cohen, 2004). Modeling of turbulence must present the above characteristic for the results. Very straight approach to model turbulent flow is the direct numerical simulation (DNS) (Orszag, 1970), which resolves turbulence based on the sufficient scale of discretization of the NSE without using any turbulent model. The DNS is expensive and inapplicable to large scale problems. To develop more effective models in terms of validity and computational expense, the idea to distinguish vorticities are developed. In turbulent flow, vorticities can be seen as large and small scale eddies. When solving the large scale eddies explicitly, because in the large scale problem, where the large eddies may be primary, one can model the effect of small scale eddies with a help of a model. The small scale eddies can be found in a scale range (an analogy to the sub-grid scale (Deardorff, 1970)) that is smaller than the characteristic filter length. To describe the large eddies in the NSE, the variables need to be filtered in space and time with filtering functions, which has dimensioned by the characteristic filter length. Like that, a simple model of the effect of the small scale eddies was introduced by Smagorinsky in 1963 closing the momentum equations for the filtered terms by an eddy viscosity term (Smagorinsky, 1963). The Smagorinsky turbulent model is a member of the Large Eddy Simulation (LES) modeling of turbulent flows. General conceptual ideas of the LES models are the filtering the variables, especially the velocity field, modeling the residual stress tensor for modeling the small scale eddies, and deriving the filtered NSE with the residual stress tensor (Pope, 2001). Here, we briefly review the Smagorinsky model for the NSE with the Boussinesq approximation.

According to the gas kinetic theory (Kerson, 1987), the diffusive properties of the gas molecules, roughly speaking the viscosity, is of the order of the product of root mean square (*rms*) speed of molecular motion and the mean free path defined as an average distance between successive collisions of molecules. By analogy with this hypothesis, using the eddy viscosity proposal with the turbulent stress of Boussinesq (Boussinesq, 1877) (Schmitt, 2007) , the order of eddy viscosity can be

$$\nu_e \sim u' l_m \ , \tag{28}$$

where u' is a typical scale of the fluctuating velocity or a residual (or sub-grid scale) component of the velocity and l_m is the mixing length defined as the cross-stream distance traveled by a fluid particle before it gives up its momentum and loses identity (Kundu & Cohen, 2004). More precisely, the eddy viscosity can be expressed by

$$\nu_e = l_m^2 \left| \frac{\partial \langle u \rangle}{\partial y} \right| , \tag{29}$$

where $\langle u \rangle$ is an average velocity. The velocity can be decomposed by the filtering function as

$$u(\mathbf{x}, t) = \bar{U}(\mathbf{x}, t) + u'(\mathbf{x}, t) \, , \tag{30}$$

where $\bar{U}$ is the filtered velocity. Having the same concept with Eq.(29), Smagorinsky in 1963 (Smagorinsky, 1963) proposed the eddy viscosity term

$$\nu_e \cong l_s^2 |\bar{S}| = (C_s \Delta)^2 \sqrt{2 \bar{S}_{ij} \bar{S}_{ij}} \ , \tag{31}$$

where l_s is the Smagorinsky length scale, $\bar{S}$ is the characteristic filtered rate of strain (Pope, 2001), C_s is the Smagorinsky constant and Δ is the filter width. With the Smagorinsky model, the NSE in Eq.(21) for the filtered velocity fields becomes

$$\frac{\partial \bar{U}_i}{\partial t} + \bar{U}_m \frac{\partial \bar{U}_i}{\partial x_m} = -\mathbf{g} \delta_{i3} \left(1 + \frac{\rho'}{\rho_o} \right) - \frac{1}{\rho_o} \frac{\partial p}{\partial x_i} + \nu_T \nabla^2 \bar{U}_i \, , \tag{32}$$

where $\nu_T = (\nu + \nu_e)$ is the turbulent viscosity, whereas the continuity equation becomes

$$\frac{\partial \bar{U}_i}{\partial x_i} = 0 \, . \tag{33}$$

2.2.1 Turbulence in stratified medium

Stratified medium, such as atmospheric and geophysical medium with the density variations, has an effect on the turbulent closures. If the density variations, induced by temperature or concentrations, are significant, the use of the Smagorinsky turbulent model may produce the unstable solution for the system. Stable and unstable stratified medium can be characterized by the Richardson number as

$$\begin{array}{ll} Ri = \dfrac{g}{\rho_o} \dfrac{\nabla \rho}{(\nabla u)^2} > 1 & \text{stable (buoyancy driven flow)} \\ Ri < 1 & \text{unstable (external force is dominant)} \end{array} \ . \tag{34}$$

It is claimed that if the flow is stable stratified flow, all turbulence are completely suppressed (Tennekes & Lumley, 1972). In other words, the buoyant destruction removes turbulence at a rate larger than the rate which it is produced by shear production (Kundu & Cohen, 2004).

Accounting the negative buoyancy effect, an anisotropic Smagorinsky model (Ikehata & Honji, 2000) can be defined for the stratified medium as

$$\nu_e = (C_s\Delta)^2\sqrt{2\bar{S}_{ij}\bar{S}_{ij}\left(1-\frac{Ri}{Pr_t}\right)}\,, \tag{35}$$

where Pr_t is the turbulent Prandtl number, which defines the relative efficiency of the vertical turbulent exchanges of momentum and heat. We can assume that $Pr_t \cong 1$. The recommended value for the Smagorinsky constant is $C_s = 0.1\sim0.2$, but it should be noted that the Smagorinsky constant is a function (Pope, 2001) relevant to a Reynolds number and filter width etc. The filter width can be computed as

$$\Delta = \left(\Delta_x\Delta_y\Delta_z\right)^{1/D}\ , \tag{36}$$

where $\Delta_x,\ \Delta_y,\ \Delta_z$ are considered as a grid spacing of the geometric discretization and D can be a number of dimensions. The Richardson number in Eq.(35) can be defined as

$$Ri = \frac{g}{\rho_o}\frac{\frac{\partial\rho'}{\partial z}}{2\bar{S}_{ij}\bar{S}_{ij}}\ . \tag{37}$$

Computationally, the Smagorinsky turbulent model can be applied for the determination of local turbulent viscosity based on the condition in Eq.(34).

2.3 Sediment transport modeling

Transporting the solid material by fluid flow is widely studied subject in physics. It is still leading subject for researchers and scientists to understand its undiscovered corners. As same as the other mechanical topics, the sediment transport was explained and modeled by using empirical description until the computer were available for computing. As the sediment transport in nature was complex, revealed understanding of it was widely applicable not only in engineering, but also in industrial processes. From the observation of not equal distribution of velocity over the depth of a river due to bed resistance to the knowledge of fluid-particle system, the countless number of people contributed their understanding into the sediment transport through several centuries in the past (Walter, 1984).

Speaking in hydraulic research, the phenomenon is well described by the phrase "the loose boundary hydraulics" (Bagnold, 1941), which identify a set of problems involving the interaction of fluid with the erodible material of its confined boundaries (Raudkivi, 1976). The early stage of the modeling was based on the sediment continuity equation accounting the entrainment and deposition rates in suspended and bed loads. Then, the modeling of the sediment transport adapted to use the advection-diffusion equation (ADE) with source and sink terms. Comprehensive review of the widely used sediment transport models are discussed in (Papanicolaou, et al., 2008). The importance of cohesive sediment transport and scale of consideration confines the modeling in continuous approach. In fact, however, the continuous approach of modeling was used to describe the particulate nature of the movement. As long as the continuum modeling is realized as inadequate for the non-cohesive sediment transport characteristic, reliable and comprehensive descriptions about the two-phase phenomenon of sediment and flow are developed (Drew, 1983). However, two-phase models diverse in continuous and discrete modeling, and the scale limitation of models have created the gap in the knowledge. To fill that and other gaps in the knowledge of sediment transport research, researchers have created the international organizations, such as the International sediment initiative (ISI) launched by UNESCO (Spreafico & Bruk, 2004), to support sustainable and systematic development of the sediment transport research.

In this thesis, we discuss two approaches for sediment transport: an ADE based model and a particle based model. Purpose to use the ADE system of sediment transport modeling is to validate the newly proposed particle based modeling (PTM). All presented results are partially reported in (Takeshi, 2016), (Takeshi, et al., 2016a) and (Takeshi, et al., 2016b).

2.3.1 Advection-diffusion equation

An amount of suspended sediment particles in fluid flow can be assumed as a concentration averaged over a control volume. Distributions of suspended sediments in fluid flows are commonly modeled by the differential equation of advection-diffusion in terms of conservation of mass. Although such a model does not account adequately for all influences, it has been found to explain satisfactorily many suspension problems (Walter, 1984). To determine a temporal evolution of spatial distributions of sediment concentration, the following ADE can be solved,

$$\frac{\partial c}{\partial t} + \mathbf{u}_i \frac{\partial c}{\partial x_i} + (w - w_s)\frac{\partial c}{\partial z} = D_c \nabla^2 c\,, i = 1{,}2 \tag{38}$$

where, c, $\mathbf{u}$, w, w_s and D_c are the sediment concentration, the horizontal and vertical velocity components, the settling velocity and the turbulent diffusion coefficient for the sediment concentration, respectively. For the diffusion coefficient, we assumed the turbulent diffusion coefficient, which is determined by a turbulent model described in Section 2.2. So that, the sediment concentration in Eq.(38) is seen as a filtered variable to be presented in turbulent flows.

The settling velocity w_s of sediment is an important rate for a vertical distribution of sediments. Proper values of the settling velocity depending on sediment mechanical properties are widely studied and recommended in many literatures (伊勢屋ふじこ, 1985), (鶴谷広一, et al., 1989). In the conventional studies, the Stokes velocity determined by the balance between ascending force (sum of the drag and buoyance forces) and gravity force is used for the settling velocity. However, the Stokes velocity is a final falling rate of particle in water. With a small time step in simulation, it is confirmed that the settling velocity takes the value of the Stokes velocity immediately in the ADE. Another way to define the settling velocity is to conduct experiments with the samples collected from the study area.

Since the ADE is solved by an Eulerian approach in the fixed grid, the local changes and historical properties for sediments does not determined in such an approach. To overcome the limits of a conventional approach with the ADE, the Lagrangian approach was proposed and solved for the sediment transport in the follow.

2.3.2 Sediment particle tracking approach

In order to capture the particulate nature of sediment transport, we purpose the sediment particle tracking method (PTM). In this method, an individual particle will be considered rather than a volume representation of sediments. This approach can be seen as a representative sediment particle tracking approach, in which the particles deputize the real sediment particles in a proper space. Conceptually, it is acceptable to use the representative particle for the model in order to represent the dynamics of the real sediment particles, since it is difficult to involve all of the sediment particles in the large scale sediment transport simulation. The representative sediment particle concept is depicted in **Figure 4** and the radius of that particle can expressed as,

$$r = \frac{\sum_{i=1}^{n} r_i}{n}, \tag{39}$$

where r_i is the radius of real sediment particles in selected region and n is the number of particles exist in the region. In computational practice, of course, we do not calculate the radius and position of the representative particles. The particles radius and position can be generated randomly by considering the problem scale and sediment concentration with respect to the grid sizes.

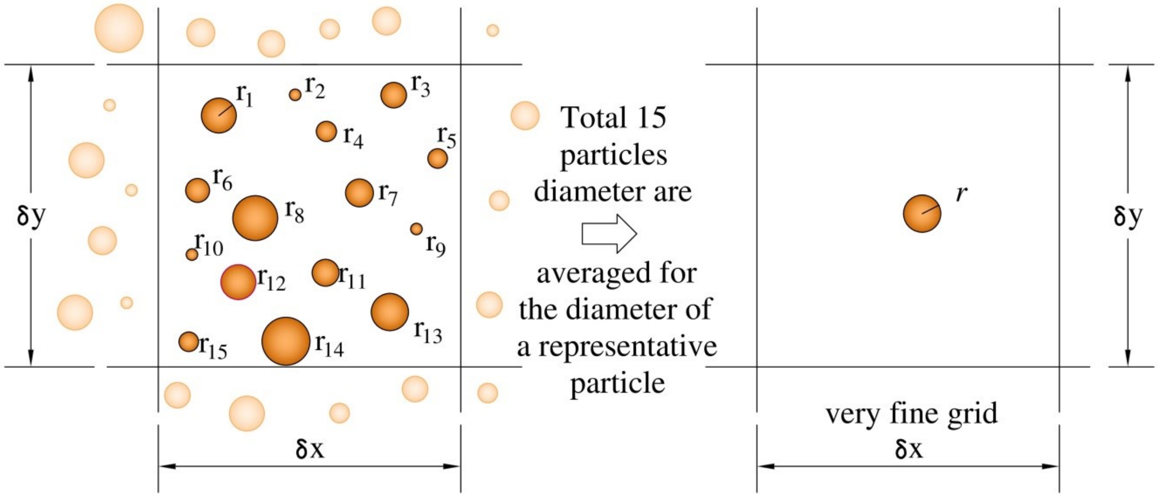

r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13
r14
r15
δy
δx
Total 15 particles diameter are averaged for the diameter of a representative particle
r
δy
very fine grid
δx

Figure 4. Example of defining representative sediment particles in very fine grid

In the PTM, time evolution of all particles will be continuously tracked in the space-time through the whole simulation. Advantages of considering individual particles provide some opportunities to handle specific behavior of sediment particles such as the flocculation process. The flocculation is a phenomenon that cohesive sediment particles aggregate with each other as attracting by their electromagnetic forces (Einstein & Krone, 1962). In consequence of the flocculation process, particle volume must be expanded by the circumstances of that process. For instance, the flocculation of sediment in the sea water may be thought as a function of time such as traveling time of the particles in the sea water. This process for individual particle traveling within sea water will be handled easily by storing the memory of the time history of particle accounted by the PTM. It also allows the particle to have a different traveling velocity at each time step in the simulation.

If a particle shape is simplified to be a sphere, the following equation of motion can be solved to capture particle position for every computational moment,

$$\frac{4}{3}\pi r^3 \rho_{sed} \frac{d\mathbf{u}_p}{dt} = \mathbf{F}_r \, , \tag{40}$$

where $\mathbf{u}_p$, r, ρ_{sed}, and $\mathbf{F}_r$ are the traveling velocity, the sediment particle radius, the density of a sediment particle, and the drag force acting on a particle. As mentioned, the sediment particle size and its spatial distribution in the simulation can be given by the generation of the random number according to the normal distribution and the uniform distribution function, respectively. The right hand side of Eq.(40) is a drag force and can be expressed as,

$$\mathbf{F}_r = \frac{1}{2} A \rho_w C_D \mathbf{u}_r |\mathbf{u}_r| \tag{41}$$

where ρ_w is the density of water, C_D is the drag coefficient of a sediment particle, A is the cross sectional area of a particle and $\mathbf{u}_r$ is the relative velocity between particles (see **Figure 5**) and a fluid flow. Since the drag coefficient is a function of the Reynolds number, we used an approximated function extracted from an experimental chart of dependency of C_D on the Reynolds number for an individual particle evaluation in Eq.(41). Consequently, depending on the instant situation of individual particle, whether laminar or turbulent regime, the drag coefficient can be estimated from such an approximated function. From the balance of forces acting on the single particle, it is presumable that sediment particles can take instantly its final sedimentation rate (Stokes velocity or settling velocity) from a stationary position when falling downward in still water. Therefore, it is reasonable to consider the Stokes velocity in Eq.(40) instead of the gravitational acceleration.

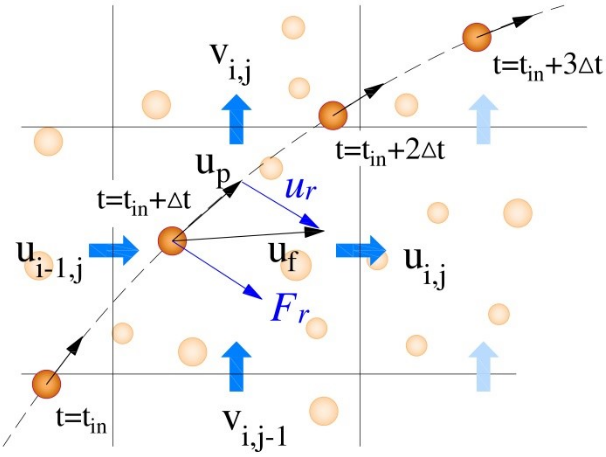


Figure 5. Schematic illustration of a particle moving through the computational cell in the particle tracking approach. A particle is traveling from initial time t_{in} to time instant $t_{in} + 3\Delta t$.

The relative velocity can be defined as

$$\mathbf{u}_r = \mathbf{u}_f - \mathbf{u}_p \, , \tag{42}$$

where the fluid velocity, $\mathbf{u}_f$, is obtained by weighting the velocities in a cell. The sediment particle velocity, $\mathbf{u}_p$, is updated by computing Eq.(40) in time advance.

2.3.3 Simple model for flocculation

Sediment transport and sedimentation are clearly dependent on a particle weight and cohesion. Cohesion and sediment particle size are solely related. Cohesion describes the tendency of fine sediment particles binding together under some circumstances. For instance, sediments in fresh water may have no repulsive force, because they are negatively charged, but this will no longer exist when sediments enter to salt water. After sediment particles are traveling in a certain time in salt water, the repulsive force does not exist between sediment particles and adjacent particles make flocs together. According to (Raudkivi, 1976), for the suspended particles of $D \leq 10\,\mu$ not only gravitational forces but also electrochemical forces become important. On the other point, a suspension of solid particles, where physicochemical forces are dominant because of chemical influence within the environment, made the collision with others and the results bring the flocs. In other words, the collisions of particles also make flocs. These processes of sediment particles stick to each other make a floc are called the result of the flocculation process (Walter, 1984). Studies of flocculation of suspended fine sediment are fronted by Kruyt (Kruyt, et al., 1952), Einstein (Einstein & Krone, 1961) and many others (Yanagi, 1989) generally in subject of estuary. An aggregation of neighboring particles as a floc accelerates the sedimentation rate of cohesive sediments (Walter, 1984). Many studies about the flocculation, these are rather descriptive or highly fundamental or even very empirical, have been carried out to understand the variation in the number of particles, or the changes in concentration (Winterwerp, 1998). However, rather complicated models are currently describing the result of flocculation, more significantly, the important variation in floc sizes and settling velocity with time. With the ADE system of suspended sediment transport, the relation between concentration and floc size are difficult to handle.

To understand the effect of flocculation for the sediment distribution, we propose a simple, practical flocculation model (Takeshi, et al., 2016a). Since the particle tracking approach or PTM is the method to track the representative particles, the number of particles in the system is no correlation with the flocculation. Simply, the flocculation process becomes only a matter for the changes of particle size in respect with time. Taking advantage of the PTM described above, the sediment particle size can be changed in the result of flocculation:

$$r = \alpha(t) r_{in}\,, \tag{43}$$

where α is the scaling rate in respect with time and r_{in} is the initial radii of the cohesive sediment particles. The scaling rate must be determined by the experimental studies and it can be expressed according to the sigmoid function of time as

$$\alpha = \alpha_o + \frac{1}{1 + e^{-t_p}} \text{ with } t_p = \frac{12}{t_{cr}} t_{touch} - 6\,, \tag{44}$$

where α_o is a base scaling, t_p is a time parameter, t_{touch} is a time for particle lapsed in flocculation condition and t_{cr} is a flocculation time. The base scaling can be assumed to be $\alpha_o = \alpha_{ex} - 1$, in which α_{ex} is experimentally defined scaling. In case of an estuary, the flocculation is caused by the salt concentration in seawater. Thus, the t_{touch} must be counted when the particle enters into seawater. The sigmoid function in the model just provides the smooth changes in particle size.

3 NUMERICAL TECHNIQUES

In solutions of fluid dynamic problems, the numbers of methods and algorithms have been used to solve the NSE. The methods differ by their concept and geometric configuration, or even designed conditions such as compressible or incompressible fluid flows. Widely used conventional methods are the finite-family methods, namely the finite difference method (FDM), finite element method (FEM) and finite volume method (FVM), based on an Eulerian grid discretization. Many of them share the same numerical algorithms in order to solve the NSE in preconditioned problems. In chronological orders, the marker and cell method (MAC) (Harlow & Welch, 1965), pseudo-compressibility method (or the artificial compressibility method - ACM) (Chorin, 1967), projection method (or velocity pressure simultaneous iteration method - PM) (Chorin, 1968), simplified MAC (SMAC) (Amsden & Harlow, 1970), semi-implicit method for pressure-linked equations (SIMPLE) (Patankar & Spalding, 1972) and highly simplified MAC (or the solution algorithm - SOLA) (Hirt & Cook, 1972) are the popular methods for the incompressible fluid flows and they are ancestor of the many other numerical algorithms (Chung, 2002). In this thesis, the SOLA method will be presented.

3.1 SOLA algorithm for the NSE

The marker-and-cell method (MAC), precursor of the solution algorithm (SOLA), uses an Eulerian finite difference formulation with pressure and velocity as the primary dependent variables and originally designed for problems involving free surfaces. As branching from the MAC, the SOLA method is simplified by eliminating some complicated refinements of the MAC algorithm, especially the marker particles (Hirt, et al., 1975) are removed from the scheme.

Geometric discretization of a problem is formed in a computational domain of the SOLA, which has an extra fictitious single layer of boundary cells surrounding the actual sized problem domain, as shown in **Figure 6** (a). A fictitious layer of boundary cells prevents unphysical oscillation that may emerge from the implementation of some boundary conditions. In other words, the boundary cells eliminate the influence of boundary condition for the problem domain. The computational domain consists of parallelepiped cells in the 3D space or rectangular cells in the 2D space having a grid spacing of $\Delta x, \Delta y,$ and Δz, respectively. Fluid variables are positioned on the cells as staggered fashion in order to avoid the checkerboard type error in pressure-velocity related formulation (Harlow & Welch, 1965). Precisely, we use u, v, w for the velocity components with their location indicated as i, j, k. The fluid velocities are located on the center of side or face of a cell in a 2D or 3D domain, respectively. Whereas, the scalar values of pressure, density or concentration are placed on the center of a cell in the 2D or 3D cell, as seen in **Figure 6** (b) and (c).

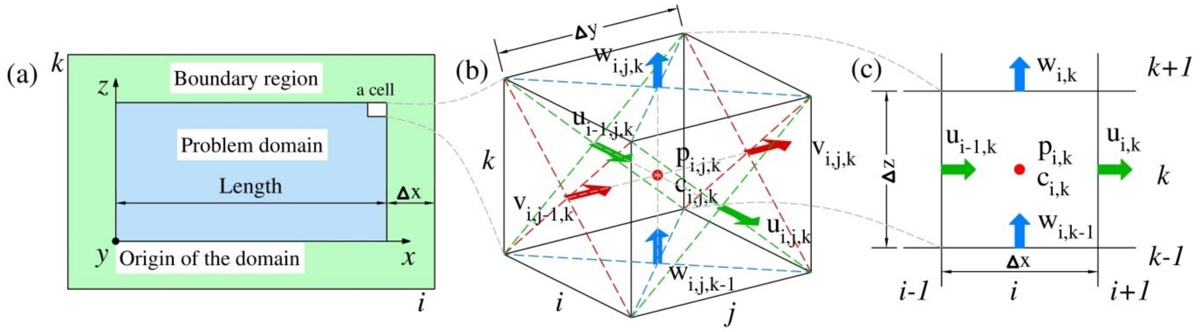


Figure 6. (a) Computational domain in the SOLA method, (b) a three dimensional cell and (c) a two dimensional cell possessing the fluid variables.

3.1.1 Explicit approximations for the governing equations

The simple Eulerian finite difference discretization of Eq.(32) in the x-direction can be written as

$$u_{i,j,k}^{n+1} = u_{i,j,k}^{n} + \Delta t\left(-\frac{1}{\rho_o}\frac{\partial p}{\partial x_i} - g_x\delta_{i3}\left(1+\frac{\rho'}{\rho_o}\right) - u_m\frac{\partial u_i}{\partial x_m} + \nu_T\nabla^2 u_i\right), \tag{45}$$

where the time dependent part of velocity is differed only. Assigning the notations for the differential terms in brackets on the right hand side of Eq.(45) becomes

$$u_{i,j,k}^{n+1} = u_{i,j,k}^{n} + \Delta t\left(P_x^n - R_x^n - F_{ux}^n - F_{uy}^n - F_{uz}^n + F_{visx}^n\right), \tag{46}$$

where P_x^n, R_x^n, F_{ux}^n, F_{uy}^n, F_{uz}^n and F_{visx}^n are the differential terms of the pressure, external forces (for the gravity in the x- and y-directional terms are zero), advection terms of velocity in the x-, y-, z-directions and viscosity term. Time advance is $t = (n+1)\Delta t$, in which n is the number of cycles in a computation. The staggered gird is very convenient for differencing the terms in Eq.(46), as we give them later. As like as the difference equation of the x-component of the NSE, the other component can be written as

$$\begin{matrix} v_{i,j,k}^{n+1} = v_{i,j,k}^{n} + \Delta t\left(P_y^n - R_y^n - F_{vx}^n - F_{vy}^n - F_{vz}^n + F_{visy}^n\right) & \text{for the y}-\text{direction} \\ w_{i,j,k}^{n+1} = w_{i,j,k}^{n} + \Delta t\left(P_z^n - R_z^n - F_{wx}^n - F_{wy}^n - F_{wz}^n + F_{visz}^n\right) & \text{for the z}-\text{direction} \end{matrix}. \tag{47}$$

If one wants to upgrade the first order of explicit systems in Eq.(46) and (47) by the second order explicit scheme, the Adams-Bashforth method may be employed as

$$\begin{aligned} u_{i,j,k}^{n+1} &= u_{i,j,k}^{n} + \Delta t(1.5T_x^n - 0.5T_x^{n-1}) \\ v_{i,j,k}^{n+1} &= v_{i,j,k}^{n} + \Delta t\left(1.5T_y^n - 0.5T_y^{n-1}\right) \\ w_{i,j,k}^{n+1} &= w_{i,j,k}^{n} + \Delta t(1.5T_z^n - 0.5T_z^{n-1}) \end{aligned}, \tag{48}$$

where $T^n = P^n - R^n - F_x^n - F_y^n - F_z^n + F_{vis}^n$ is the terms in the (n)th time step and T^{n-1} is the terms in the (*n-1*)th time step. Note that in the expression of T^n, we removed the notation of coordinates.

While, the continuity equation in Eq.(15) can get the discretization of

$$\frac{u_{i,j,k}^{n+1} - u_{i-1,j,k}^{n+1}}{\Delta x} + \frac{v_{i,j,k}^{n+1} - v_{i,j-1,k}^{n+1}}{\Delta x} + \frac{w_{i,j,k}^{n+1} - w_{i,j,k-1}^{n+1}}{\Delta z} = 0 \tag{49}$$

and the new velocities defined by the discretized NSE in Eq.(46) and (47) must satisfy the continuity equation in Eq.(49). However, in general, the new velocities computed from the NSE will not satisfy the continuity equation; moreover, the artificial incompressibility condition must be imposed for the scheme. Depending on the value of the divergence in Eq.(49), an appropriate adjusting for the pressure, which leads the correction for velocities, must be made. In other words, the divergence in a current cell is

$$D^m = \frac{u_{i,j,k}^{n+1} - u_{i-1,j,k}^{n+1}}{\Delta x} + \frac{v_{i,j,k}^{n+1} - v_{i,j-1,k}^{n+1}}{\Delta x} + \frac{w_{i,j,k}^{n+1} - w_{i,j,k-1}^{n+1}}{\Delta z}\,, \tag{50}$$

and the resultant pressure change can be defined as

$$\delta p^m = -D^m \frac{\partial D^m}{\partial p_{i,j,k}^{n,m}} = -D^m \beta = -D^m \frac{\omega}{2\Delta t\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right)}\,, \tag{51}$$

where m is the current iteration cycle, β is the constant containing the over relaxation parameter, $\omega(= 1.8)$, and the computational parameters such as a grid spacing and time step. The pressure and velocity corrections must be done iteratively, however, because when one cell is adjusted to its neighbors are affected. For the each cell of the computational domain, the most recent velocities, corrected with previous iteration or updated by new time step, are used to compute the divergence and the pressure change in current iteration n. During the iterations, the pressure change corrects the pressure in order to make D^m approaches to zero, as

$$p_{i,j,k}^{n,m+1} = p_{i,j,k}^{n,m} + \delta p^m\,, \tag{52}$$

and the cell relevant velocities become

$$\begin{aligned}
u_{i,j,k}^{n,m+1} &= u_{i,j,k}^{n,m} + \Delta t \frac{\delta p^m}{\Delta x}, & u_{i-1,j,k}^{n,m+1} &= u_{i-1,j,k}^{n,m} - \Delta t \frac{\delta p^m}{\Delta x} \\
v_{i,j,k}^{n,m+1} &= v_{i,j,k}^{n,m} + \Delta t \frac{\delta p^m}{\Delta x}, & v_{i,j-1,k}^{n,m+1} &= v_{i,j-1,k}^{n,m} - \Delta t \frac{\delta p^m}{\Delta x}\,. \\
w_{i,j,k}^{n,m+1} &= w_{i,j,k}^{n,m} + \Delta t \frac{\delta p^m}{\Delta x}, & w_{i,j,k-1}^{n,m+1} &= w_{i,j,k-1}^{n,m} - \Delta t \frac{\delta p^m}{\Delta x}
\end{aligned} \tag{53}$$

Convergence of the iteration is achieved when all cells have D^m values satisfying the inequality

$$\max D^m < \varepsilon\,, \tag{54}$$

where ε is a typical value order of 10^{-3} or smaller.

A constant time step in the numerical method, not only in the SOLA, leads the unstable solution for the computation. Once the grid spacing has been chosen to be a constant, the choice of the time step necessary for the stability is governed by a stability condition for the solving equations. Stability analysis of the numerical scheme is often carried out the von Neumann stability analysis (Crank & Phyllis, 1947) and the chosen time step should satisfy the von Neumann criteria in general (Vreugdenhil, 2012):

$$|\sigma| \le 1\,, \text{with } \sigma = u\frac{\Delta t}{\Delta x}\,, \tag{55}$$

where σ is the Courant number. Motivated from the Courant-Friedritchs-Lewy (CFL) condition in Eq.(55), general conditions of the time step for the stable solutions of the pressure-velocity iteration techniques (Bulgarelli, et al., 1984) are formulated as

$$\begin{cases} \Delta t \cdot \max\left(\dfrac{u}{\Delta x}, \dfrac{v}{\Delta y}, \dfrac{w}{\Delta z}\right) \leq \dfrac{1}{3} \\ \Delta t \leq \left[2\mu\left(\dfrac{1}{\Delta x^2} + \dfrac{1}{\Delta y^2} + \dfrac{1}{\Delta z^2}\right)\right]^{-1} \cdot \min(\rho) \end{cases} . \tag{56}$$

For the incompressible fluid flow, the criteria to choose a time step could be generalized from Eq.(56) as follow:

$$\begin{cases} \Delta t \leq \dfrac{1}{3 \cdot \max\left(\dfrac{u_{max}}{\Delta x}, \dfrac{v_{max}}{\Delta y}, \dfrac{w_{max}}{\Delta z}\right)} = dt1 \\ \Delta t \leq \dfrac{1}{\left[2\nu\left(\dfrac{1}{\Delta x^2} + \dfrac{1}{\Delta y^2} + \dfrac{1}{\Delta z^2}\right)\right]} = dt2 \end{cases} \rightarrow\rightarrow \quad \Delta t = a \cdot \min(dt1, dt2) \tag{57}$$

where a is a safety factor which insures the order of time step lower than the minimum grid spacing. Depending on the problem, the additional time step criteria can be embedded in the condition in Eq.(57).

3.1.2 Approximations for the terms in the governing equations

Now we give finite difference equations for the terms in Eq.(46) and (47) as follows. The pressure terms are

$$\begin{aligned} P_x^n &= -\frac{1}{\rho_o}\frac{(p_{i+1,j,k}^n - p_{i,j,k}^n)}{\Delta x} \\ P_y^n &= -\frac{1}{\rho_o}\frac{(p_{i,j+1,k}^n - p_{i,j,k}^n)}{\Delta y} \\ P_z^n &= -\frac{1}{\rho_o}\frac{(p_{i,j,k+1}^n - p_{i,j,k}^n)}{\Delta z} \end{aligned} \tag{58}$$

and the external force terms are

$$\begin{aligned} R_x^n &= a_x\left(1 + \frac{\rho'}{\rho_o}\right) \\ R_y^n &= a_y\left(1 + \frac{\rho'}{\rho_o}\right), \\ R_z^n &= a_z\left(1 + \frac{\rho'}{\rho_o}\right) \end{aligned} \tag{59}$$

where $\mathbf{a} = \{a_x, a_y, a_z\}$ is the force acceleration, for instance, in gravitational field, it is $\mathbf{a} = \{0,0,\mathrm{g}\}$.

The advection and viscous terms are relatively important terms in the SOLA method in terms of accuracy (Wilson, et al., 1988). They are expressing the advection of velocity flux and the acceleration of viscous forces in fluid flows. For example, F_{ux}^n is the advection flux of u in the x direction. These terms are evaluated using old time step values for velocities. For the regular grid, the advection flux terms can be approximated as:

For the x – component

$$F_{ux}^n = \frac{1}{4\Delta x}\left[\begin{array}{c}\left(u_{i,j,k}^n + u_{i+1,j,k}^n\right)\left(u_{i,j,k}^n + u_{i+1,j,k}^n\right) + \alpha\left|u_{i,j,k}^n + u_{i+1,j,k}^n\right|\left(u_{i,j,k}^n - u_{i+1,j,k}^n\right) - \\ \left(u_{i-1,j,k}^n + u_{i,j,k}^n\right)\left(u_{i-1,j,k}^n + u_{i,j,k}^n\right) - \alpha\left|u_{i-1,j,k}^n + u_{i,j,k}^n\right|\left(u_{i-1,j,k}^n - u_{i,j,k}^n\right)\end{array}\right] \tag{60}$$

$$F_{uy}^n = \frac{1}{4\Delta y}\left[\begin{array}{c}\left(v_{i,j,k}^n + v_{i+1,j,k}^n\right)\left(u_{i,j,k}^n + u_{i,j+1,k}^n\right) + \alpha\left|v_{i,j,k}^n + v_{i+1,j,k}^n\right|\left(u_{i,j,k}^n - u_{i,j+1,k}^n\right) - \\ \left(v_{i,j-1,k}^n + v_{i+1,j-1,k}^n\right)\left(u_{i,j-1,k}^n + u_{i,j,k}^n\right) - \alpha\left|v_{i,j-1,k}^n + v_{i+1,j-1,k}^n\right|\left(u_{i,j-1,k}^n - u_{i,j,k}^n\right)\end{array}\right] \tag{61}$$

$$F_{uz}^n = \frac{1}{4\Delta z}\left[\begin{array}{c}\left(w_{i,j,k}^n + w_{i+1,j,k}^n\right)\left(u_{i,j,k}^n + u_{i,j,k+1}^n\right) + \alpha\left|w_{i,j,k}^n + w_{i+1,j,k}^n\right|\left(u_{i,j,k}^n - u_{i,j,k+1}^n\right) - \\ \left(w_{i,j,k-1}^n + w_{i+1,j,k-1}^n\right)\left(u_{i,j,k-1}^n + u_{i,j,k}^n\right) - \alpha\left|w_{i,j,k-1}^n + w_{i+1,j,k-1}^n\right|\left(u_{i,j,k-1}^n - u_{i,j,k}^n\right)\end{array}\right] \tag{62}$$

For the y – component

$$F_{vx}^n = \frac{1}{4\Delta x}\left[\begin{array}{c}\left(u_{i,j,k}^n + u_{i,j+1,k}^n\right)\left(v_{i,j,k}^n + v_{i+1,j,k}^n\right) + \alpha\left|u_{i,j,k}^n + u_{i,j+1,k}^n\right|\left(v_{i,j,k}^n - v_{i+1,j,k}^n\right) - \\ \left(u_{i-1,j,k}^n + u_{i-1,j+1,k}^n\right)\left(v_{i-1,j,k}^n + v_{i,j,k}^n\right) - \alpha\left|u_{i-1,j,k}^n + u_{i-1,j+1,k}^n\right|\left(v_{i-1,j,k}^n - v_{i,j,k}^n\right)\end{array}\right] \tag{63}$$

$$F_{vy}^n = \frac{1}{4\Delta y}\left[\begin{array}{c}\left(v_{i,j,k}^n + v_{i,j+1,k}^n\right)\left(v_{i,j,k}^n + v_{i,j+1,k}^n\right) + \alpha\left|v_{i,j,k}^n + v_{i,j+1,k}^n\right|\left(v_{i,j,k}^n - v_{i,j+1,k}^n\right) - \\ \left(v_{i,j-1,k}^n + v_{i,j,k}^n\right)\left(v_{i,j-1,k}^n + v_{i,j,k}^n\right) - \alpha\left|v_{i,j-1,k}^n + v_{i,j,k}^n\right|\left(v_{i,j-1,k}^n - v_{i,j,k}^n\right)\end{array}\right] \tag{64}$$

$$F_{vz}^n = \frac{1}{4\Delta z}\left[\begin{array}{c}\left(w_{i,j,k}^n + w_{i,j+1,k}^n\right)\left(v_{i,j,k}^n + v_{i,j,k+1}^n\right) + \alpha\left|w_{i,j,k}^n + w_{i,j+1,k}^n\right|\left(v_{i,j,k}^n - v_{i,j,k+1}^n\right) - \\ \left(w_{i,j,k-1}^n + w_{i,j+1,k-1}^n\right)\left(v_{i,j,k-1}^n + v_{i,j,k}^n\right) - \alpha\left|w_{i,j,k-1}^n + w_{i,j+1,k-1}^n\right|\left(v_{i,j,k-1}^n - v_{i,j,k}^n\right)\end{array}\right] \tag{65}$$

For the z – component

$$F_{wx}^n = \frac{1}{4\Delta x}\left[\begin{array}{c}\left(u_{i,j,k}^n + u_{i,j,k+1}^n\right)\left(w_{i,j,k}^n + w_{i+1,j,k}^n\right) + \alpha\left|u_{i,j,k}^n + u_{i,j,k+1}^n\right|\left(w_{i,j,k}^n - w_{i+1,j,k}^n\right) - \\ \left(u_{i-1,j,k}^n + u_{i-1,j,k+1}^n\right)\left(w_{i-1,j,k}^n + w_{i,j,k}^n\right) - \alpha\left|u_{i-1,j,k}^n + u_{i-1,j,k+1}^n\right|\left(w_{i-1,j,k}^n - w_{i,j,k}^n\right)\end{array}\right] \tag{66}$$

$$F_{wy}^n = \frac{1}{4\Delta y}\left[\begin{array}{c}\left(v_{i,j,k}^n + v_{i,j,k+1}^n\right)\left(w_{i,j,k}^n + w_{i,j+1,k}^n\right) + \alpha\left|v_{i,j,k}^n + v_{i,j,k+1}^n\right|\left(w_{i,j,k}^n - w_{i,j+1,k}^n\right) - \\ \left(v_{i,j-1,k}^n + v_{i,j-1,k+1}^n\right)\left(w_{i,j-1,k}^n + w_{i,j,k}^n\right) - \alpha\left|v_{i,j-1,k}^n + u_{i,j-1,k+1}^n\right|\left(w_{i,j-1,k}^n - w_{i,j,k}^n\right)\end{array}\right] \tag{67}$$

$$F_{wz}^n = \frac{1}{4\Delta z}\left[\begin{array}{c}\left(w_{i,j,k}^n + w_{i,j,k+1}^n\right)\left(w_{i,j,k}^n + w_{i,j,k+1}^n\right) + \alpha\left|w_{i,j,k}^n + w_{i,j,k+1}^n\right|\left(w_{i,j,k}^n - w_{i,j,k+1}^n\right) - \\ \left(w_{i,j,k-1}^n + w_{i,j,k}^n\right)\left(w_{i,j,k-1}^n + w_{i,j,k}^n\right) - \alpha\left|w_{i,j,k-1}^n + w_{i,j,k}^n\right|\left(w_{i,j,k-1}^n - w_{i,j,k}^n\right)\end{array}\right] \tag{68}$$

The viscous terms for each component are

$$F_{visx}^n = \begin{array}{l}\frac{1}{\Delta x^2}\left[{\nu_T}_{i+1,j,k}^n\left(u_{i+1,j,k}^n - u_{i,j,k}^n\right) - {\nu_T}_{i,j,k}^n\left(u_{i,j,k}^n - u_{i-1,j,k}^n\right)\right] \\ + \frac{1}{\Delta y^2}\left[{\nu_T}_{i,j,k}^n\left(u_{i,j+1,k}^n - u_{i,j,k}^n\right) - {\nu_T}_{i,j-1,k}^n\left(u_{i,j,k}^n - u_{i,j-1,k}^n\right)\right], \\ + \frac{1}{\Delta z^2}\left[{\nu_T}_{i,j,k}^n\left(u_{i,j,k+1}^n - u_{i,j,k}^n\right) - {\nu_T}_{i,j,k-1}^n\left(u_{i,j,k}^n - u_{i,j,k-1}^n\right)\right]\end{array} \tag{69}$$

$$F_{visy}^n = \begin{array}{l}\frac{1}{\Delta x^2}\left[{\nu_T}_{i,j,k}^n\left(v_{i+1,j,k}^n - v_{i,j,k}^n\right) - {\nu_T}_{i-1,j,k}^n\left(v_{i,j,k}^n - v_{i-1,j,k}^n\right)\right] \\ + \frac{1}{\Delta y^2}\left[{\nu_T}_{i,j+1,k}^n\left(v_{i,j+1,k}^n - v_{i,j,k}^n\right) - {\nu_T}_{i,j,k}^n\left(v_{i,j,k}^n - v_{i,j-1,k}^n\right)\right] \text{ and} \\ + \frac{1}{\Delta z^2}\left[{\nu_T}_{i,j,k}^n\left(v_{i,j,k+1}^n - v_{i,j,k}^n\right) - {\nu_T}_{i,j,k-1}^n\left(v_{i,j,k}^n - v_{i,j,k-1}^n\right)\right]\end{array} \tag{70}$$

$$F_{visz}^n = \begin{array}{l}\frac{1}{\Delta x^2}\left[{\nu_T}_{i,j,k}^n\left(w_{i+1,j,k}^n - w_{i,j,k}^n\right) - {\nu_T}_{i-1,j,k}^n\left(w_{i,j,k}^n - w_{i-1,j,k}^n\right)\right] \\ + \frac{1}{\Delta y^2}\left[{\nu_T}_{i,j,k}^n\left(w_{i,j+1,k}^n - w_{i,j,k}^n\right) - {\nu_T}_{i,j-1,k}^n\left(w_{i,j,k}^n - w_{i,j-1,k}^n\right)\right]. \\ + \frac{1}{\Delta z^2}\left[{\nu_T}_{i,j,k+1}^n\left(w_{i,j,k+1}^n - w_{i,j,k}^n\right) - {\nu_T}_{i,j,k}^n\left(w_{i,j,k}^n - w_{i,j,k-1}^n\right)\right]\end{array} \tag{71}$$

For the SOLA method, initialization for the pressure can be arbitrary, i.e. hydrostatic pressure distribution can be assigned for open channel flows.

3.1.3 SOLA for the shallow water approximation

In Section 3.1.2, we saw the possibility to use the SWA for the numerical simulation with the NSE. In the SWA, the continuity equation is used for the calculation of the vertical component of velocity instead of the vertical component of the NSE. For a 3D computation, Eq.(25) can be solved. For a 2D computation, the momentum equation of the y direction is removed. Then the governing equations can be discretized as

$$\left.\begin{array}{ll}\text{For the } x \text{ component:} & \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + w\dfrac{\partial u}{\partial z} = -\dfrac{1}{\rho}\dfrac{\partial p}{\partial x} + \nu\nabla^2 u \\ \text{For the } z \text{ component:} & \dfrac{\partial w}{\partial z} = -\dfrac{\partial u}{\partial x}\end{array}\right. \rightarrow$$

$$\rightarrow \begin{array}{l} u_{i,k}^{n+1} = u_{i,k}^{n} + \Delta t\left(P_x^n - R_x^n - F_{ux}^n - F_{uz}^n + F_{visx}^n\right) \\ w_{i,k}^{n+1} = \dfrac{u_{i,k}^{n+1} - u_{i-1,k}^{n+1}}{\Delta x}\Delta z + w_{i,k-1}^{n} \end{array}, \tag{72}$$

where $P_x^n, R_x^n, F_{ux}^n, F_{uz}^n$ and F_{visx}^n are the same as the difference equations given in Eq.(58) – (71) after removing the variables with presence of the y – axis, i.e. variables with the index of j. For each time step with the SWA, the pressure is calculated by the finite difference approximation of Eq.(23)

$$p_{i,k}^n = p_{i,k+1}^n + (\rho_o + \rho')g_z\Delta z\,. \tag{73}$$

This hydrostatic pressure approximation also is valid for a 3D numerical simulation.

3.1.4 Finite difference scheme for the density variations

Since the Boussinesq approximation is considered for the problem, the convection-diffusion equation (CDE) for the density derivation must be solved with the NSE. The CDE in Eq.(27) can be approximated as

$$\rho'^{n+1}_{i,j,k} = \rho'^{n+1}_{i,j,k} + \Delta t\left[-f_{rx}^n - f_{ry}^n - f_{rz}^n + f_{rxx}^n + f_{ryy}^n + f_{rzz}^n\right], \tag{74}$$

where the convection/advection and diffusion terms exist in square bracket. The convection terms are expressed by using the upwind differences as follows:

$$f_{rx}^n = \begin{cases} u_c^n \dfrac{\rho'^n_{i+1,j,k} - \rho'^n_{i,j,k}}{\Delta x} & \text{if } u_c^n \le 0 \\ u_c^n \dfrac{\rho'^n_{i,j,k} - \rho'^n_{i-1,j,k}}{\Delta x} & \text{if } u_c^n \ge 0 \end{cases}, \tag{75}$$

$$f_{ry}^n = \begin{cases} v_c^n \dfrac{\rho'^n_{i,j+1,k} - \rho'^n_{i,j,k}}{\Delta y} & \text{if } v_c^n \le 0 \\ v_c^n \dfrac{\rho'^n_{i,j,k} - \rho'^n_{i,j-1,k}}{\Delta y} & \text{if } v_c^n \ge 0 \end{cases}, \tag{76}$$

$$f_{rz}^n = \begin{cases} w_c^n \dfrac{\rho'^n_{i,j,k+1} - \rho'^n_{i,j,k}}{\Delta z} & \text{if } w_c^n \leq 0 \\ w_c^n \dfrac{\rho'^n_{i,j,k} - \rho'^n_{i,j,k-1}}{\Delta z} & \text{if } w_c^n \geq 0 \end{cases} , \tag{77}$$

where u_c^n, v_c^n and w_c^n are the average velocities for a cell. The velocities are averaged as

$$u_c^n = \frac{u_{i,j,k}^n + u_{i-1,j,k}^n}{2} , \qquad v_c^n = \frac{v_{i,j,k}^n + v_{i,j-1,k}^n}{2} \text{ and } w_c^n = \frac{w_{i,j,k}^n + w_{i,j,k-1}^n}{2} , \tag{78}$$

with the velocities obtained from the NSE.

The diffusion terms in Eq.(74) are differenced by the central difference scheme as

$$f_{rxx}^n = \frac{D_{s_{i,j,k}}^n \left(\rho'^n_{i+1,j,k} - \rho'^n_{i,j,k}\right) - D_{s_{i-1,j,k}}^n (\rho'^n_{i,j,k} - \rho'^n_{i-1,j,k})}{\Delta x^2} , \tag{79}$$

$$f_{ryy}^n = \frac{D_{s_{i,j,k}}^n \left(\rho'^n_{i,j+1,k} - \rho'^n_{i,j,k}\right) - D_{s_{i,j-1,k}}^n (\rho'^n_{i,j,k} - \rho'^n_{i,j-1,k})}{\Delta y^2} , \tag{80}$$

$$f_{rzz}^n = \frac{D_{s_{i,j,k}}^n \left(\rho'^n_{i,j,k+1} - \rho'^n_{i,j,k}\right) - D_{s_{i,j,k-1}}^n (\rho'^n_{i,j,k} - \rho'^n_{i,j,k-1})}{\Delta z^2} , \tag{81}$$

where the diffusion coefficients, D_s, are written as a function of space-time, because the turbulent viscosity can be used instead the diffusion coefficient: $\nu_{T_{i,j,k}}^n = D_{s_{i,j,k}}^n$.

3.2 Implementation for sediment transport modeling

3.2.1 The finite difference advection-diffusion equation

The approximation of the ADE is the same as the previous CDE. The finite difference form of the ADE in Eq.(38) for the suspended sediment concentration can be written as

$$c_{i,j,k}^{n+1} = c_{i,j,k}^n + \Delta t\left[-f_{cx}^n - f_{cy}^n - f_{cz}^n + f_{cxx}^n + f_{cyy}^n + f_{czz}^n\right] . \tag{82}$$

Using the upwind difference scheme, the advection terms in Eq.(82) can be differenced as

$$f_{cx}^n = \begin{cases} u_c^n \dfrac{c_{i+1,j,k}^n - c_{i,j,k}^n}{\Delta x} & \text{if } u_c^n \leq 0 \\ u_c^n \dfrac{c_{i,j,k}^n - c_{i-1,j,k}^n}{\Delta x} & \text{if } u_c^n \geq 0 \end{cases} , \tag{83}$$

$$f_{cy}^n = \begin{cases} v_c^n \dfrac{c_{i,j+1,k}^n - c_{i,j,k}^n}{\Delta y} & \text{if } v_c^n \leq 0 \\ v_c^n \dfrac{c_{i,j,k}^n - c_{i,j-1,k}^n}{\Delta y} & \text{if } v_c^n \geq 0 \end{cases} , \tag{84}$$

$$f_{cz}^n = \begin{cases} (w_c^n - w_s) \dfrac{c_{i,j,k+1}^n - c_{i,j,k}^n}{\Delta z} & \text{if } w_c^n - w_s \leq 0 \\ (w_c^n - w_s) \dfrac{c_{i,j,k}^n - c_{i,j,k-1}^n}{\Delta z} & \text{if } w_c^n - w_s \geq 0 \end{cases} , \tag{85}$$

where the averaged velocities for a cell are determined with Eq.(78).

The diffusion terms in Eq.(82) are approximated by the central differencing scheme

$$f_{cxx}^{n} = \frac{D_{c_{i,j,k}}^{\ n}\left(c_{i+1,j,k}^{n} - c_{i,j,k}^{n}\right) - D_{c_{i-1,j,k}}^{\ n}\left(c_{i,j,k}^{n} - c_{i-1,j,k}^{n}\right)}{\Delta x^{2}} , \tag{86}$$

$$f_{cyy}^{n} = \frac{D_{c_{i,j,k}}^{\ n}\left(c_{i,j+1,k}^{n} - c_{i,j,k}^{n}\right) - D_{c_{i,j-1,k}}^{\ n}\left(c_{i,j,k}^{n} - c_{i,j-1,k}^{n}\right)}{\Delta y^{2}} , \tag{87}$$

$$f_{czz}^{n} = \frac{D_{c_{i,j,k}}^{\ n}\left(c_{i,j,k+1}^{n} - c_{i,j,k}^{n}\right) - D_{c_{i,j,k-1}}^{\ n}\left(c_{i,j,k}^{n} - c_{i,j,k-1}^{n}\right)}{\Delta z^{2}} , \tag{88}$$

in which the turbulent viscosity can be used instead the concentration diffusion coefficient. To improve the accuracy and stability for ADE (the same for the CDE), the Adams-Bashforth method applied to Eq.(48) can be employed.

3.2.2 The particle tracking method

The equation of motion for a sediment particle given in Eq.(40) can be rewritten as

$$\frac{d\mathbf{u}_{p_m}}{dt} = \frac{\mathbf{F}_{r_m}}{W_m} , \tag{89}$$

where m is a numbering of a representative sediment particle, simply a particle, in the system and W_m is the weight of the m^th^ particle:

$$W_m = \frac{4}{3}\pi r_m^2 \rho_{sed_m} . \tag{90}$$

The velocities of the m^th^ particle at $(n+1)\Delta t$ are computed with the following difference equations,

$$u_{p_m}^{\ n+1} = u_{p_m}^{\ n} + \frac{f_{px_m}^{\ \ n}}{W_m^n}\Delta t , \tag{91}$$

$$v_{p_m}^{\ n+1} = v_{p_m}^{\ n} + \frac{f_{py_m}^{\ \ n}}{W_m^n}\Delta t , \tag{92}$$

$$w_{p_m}^{\ n+1} = w_{p_m}^{\ n} + \frac{f_{pz_m}^{\ \ n}}{W_m^n}\Delta t , \tag{93}$$

where u_p, v_p and w_p are the particle velocity and W_m^n terms the temporal weights of particle, if the flocculation is present for the sediment particle. The forces for Eqs.(91-93) are the drag forces expressed as

$$f_{px_m}^{\ \ n} = \frac{1}{2}\rho_o C_D u_{r_m}^{\ n} U_{abs}^n A_m^n , \tag{94}$$

$$f_{py_m}^{\ \ n} = \frac{1}{2}\rho_o C_D v_{r_m}^{\ n} U_{abs}^n A_m^n , \tag{95}$$

$$f_{pz_m}^{\ \ n} = \frac{1}{2}\rho_o C_D w_{r_m}^{\ n} U_{abs}^n A_m^n , \tag{96}$$

where ρ_o is the water density, the same as the reference density in the NSE, C_D is the drag coefficient with respect to the Reynolds number of the m^th^ particle, ${u_r}_m^n$, ${v_r}_m^n$ and ${w_r}_m^n$ are the relative velocities, U_{abs}^n is the absolute relative velocity and A_m^n is the cross-sectional area of the m^th^ particle: $A_m^n = \pi r_m^2$. The relative velocities are defined by

$$\begin{aligned} {u_r}_m^n &= {u_f}_{i,j,k}^n - {u_p}_m^n \\ {v_r}_m^n &= {v_f}_{i,j,k}^n - {v_p}_m^n \ , \\ {w_r}_m^n &= {w_f}_{i,j,k}^n - {w_p}_m^n \end{aligned} \tag{97}$$

where $\mathbf{u}_f = (u_f, v_f, w_f)$ is the weighted velocity and the absolute value of the relative velocity is

$$U_{abs}^n = \sqrt{{u_r}_m^2 + {v_r}_m^2 + {w_r}_m^2} \, . \tag{98}$$

Since the particles are traveling through the computational cell with the Lagrangian coordinate of $\mathbf{X}_m$ (**Figure 5**), an appropriate fluid velocity cannot be fit the velocity defined on the side or face of the cell. Thus the weighted velocity in proper accuracy can be defined by the volume or areal weights for the relevant velocities of control volumes as

$$\mathbf{u}_f = \sum_{l=1}^{N} \mathrm{A}_l \mathbf{u}_{av}^n \ , \tag{99}$$

where l is the numbering of section or fragment divided by the position of the particle, as shown in **Figure 7**, N is the total number of section or fragment and A_l is the area of a section or the volume of a fragment in percent in the 2D or 3D cell, respectively.

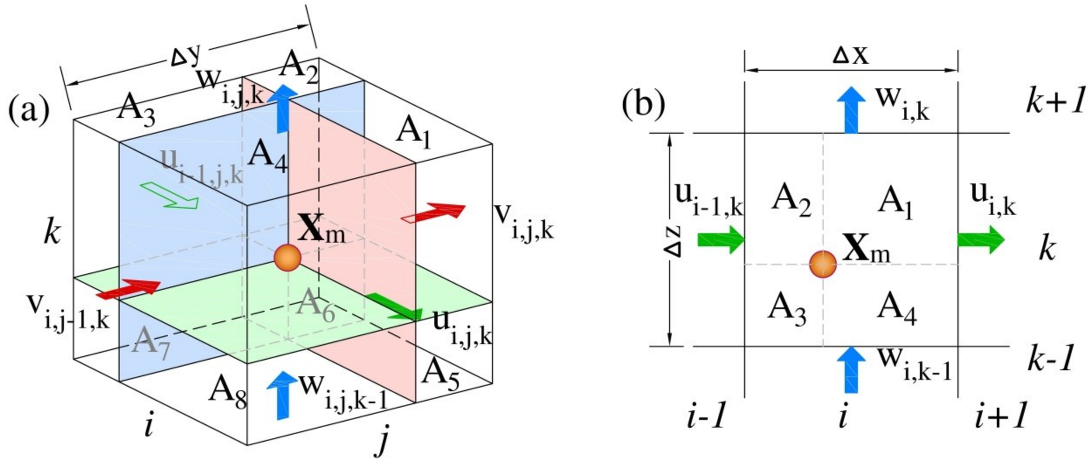


Figure 7. Fragments (a) and sections (b) of a cell divided by the position of particle in the cell. The fragments are separated by colored panels in the 3D cell, while the sections are divided by dashed lines in the 2D cell.

Special care must be given to average the velocity, $\mathbf{u}_{av}^n$, as Eq.(78) by their existing control volume. With the updated velocities for a new time step, the particle position is updated by the following equations

$$\mathbf{X}_m = \begin{cases} x_m^{n+1} = {u_p}_m^{n+1} \Delta t + x_m^n \\ y_m^{n+1} = {v_p}_m^{n+1} \Delta t + y_m^n \\ z_m^{n+1} = {w_p}_m^{n+1} \Delta t + z_m^n - {w_s}_p^{n+1} \Delta t \end{cases} , \tag{100}$$

where ${w_s}_p^{n+1}$ is the settling velocity of the particle and can be defined by the Stokes equation:

$$ {w_s}_p^{n+1} = \frac{2r_m^2(\rho_{sed} - \rho_o)g_y}{9\mu}, \tag{101} $$

where r_m is the radii of the sediment particle. For the ADE, it is difficult to use the settling velocity as flexible as that used in the PTM given in Eq.(101) and we just choose the settling velocity according to the references like (Flemming & Thum, 1978) and (Gibbs, 1974) or experiments in Section 4.2.5.

3.2.3 Boundary conditions

For fluid flow, there are an inlet, outlet, free-surface and solid surface boundaries. As shown in **Figure 6**, the computational domain has an extra layer of cells as a boundary region. Since one face or side (3D or 2D space) of a boundary cell belongs to the problem domain, the variables on that face/side will be defined by the computational procedure. The variables of another 5 faces or 3 sides and center will be defined by the boundary condition. For the bottom and the top, slip and non-slip boundary condition can be imposed to model smooth and rough surface of the reality, respectively. For instance, at the bottom, fluid flow interacting with the sea bed can be modeled as a non-slip surface. Either the solid surface is slip or non-slip, the velocity that is perpendicular to the surface must be zero to be impermeable. The velocity that is parallel to the solid surface can be

$$ u_{i,j-1,k}^n = \begin{cases} -u_{i,j,k}^n & \text{for a non} - \text{slip surface} \\ u_{i,j,k}^n & \text{for a slip surface} \end{cases}, \tag{102} $$

for the bottom surface.

At the inlet and outlet boundary, the discharge must be maintained the same. Usually, the prescribed inlet velocity will be given to the inlet boundary cells and the inlet discharge is estimated by integrating over inlet boundary cells. Then the outlet discharge is equated to the inlet discharge. The outlet boundary velocities are estimated from the outlet discharge.

For the scalar variables, for instance the density deviation and sediment concentration, the zero gradient boundary conditions can be assigned at the all boundaries including the inlet, outlet and solid surfaces.

4 NUMERICAL APPLICATIONS

In this section, we apply the models described in Section 2 and 3 to solve the sediment problem in a lab scale and in a practical scale. The lab scale application is devoted for the model validations.

4.1 Lock-exchange problem

Gravity currents, which result whenever fluid of heavy density flows horizontally into fluid of a lighter density, are frequent occurrences in both natural and engineering situation (Huppert & Simpson, 1980). There are several types of gravity currents observed in the both situations, e.g. turbidity current caused by suspended sediment and density current induced by the salt concentration (Benjamin, 1968). To validate the mathematical and numerical model designed for the density current flow, a lock-exchange problem is extensively solved and experimented as a benchmark problem. In a lock exchange experiment, fluids of different densities initially at rest are separated by a vertical gate in a tank. When the gate is removed, differences in the hydrostatic pressure cause the denser fluid to flow in one direction along the bottom boundary of the tank, while the lighter fluid flows in the opposite direction along the top boundary of the tank (Shin, et al., 2004).

We solved a 2D lock-exchange problem with the presence of the suspended sediment, which was originally reported in (Takeshi, et al., 2016a) and (Takeshi, 2016). The density current is caused by the salt concentration difference between the fresh and seawater. The main purpose of the laboratory scale lock-exchange problem was to validate the sediment PTM against the ADE of sediment concentration. The problem geometry is depicted in **Figure 8** where the left hand side is filled with fresh water having a density of 1.0 g cm^{-3}, while the right hand side is filled with seawater having the density of 1.01g cm^{-3}. In addition, the upper half of the domain of water has the suspended sediments: the concentration and sediment particles exist simultaneously. The density of sediment particles were 2.6 g cm^{-3} and settling velocity for the ADE system was given as 0.01 cm s^{-1}. The density current might be caused by the sediment was neglected in the simulations. Two cases of runs, without flocculation and with flocculation, were performed in order to understand the effects of flocculation on the sedimentation. The flocculation time criteria set as 5 s for simplicity and until that time the particle size becomes twice following the sigmoid function in Eq.(44). The total time of the simulations was 20 second. In both cases, a flow field was simulated by the SOLA method described in Section 3.1.

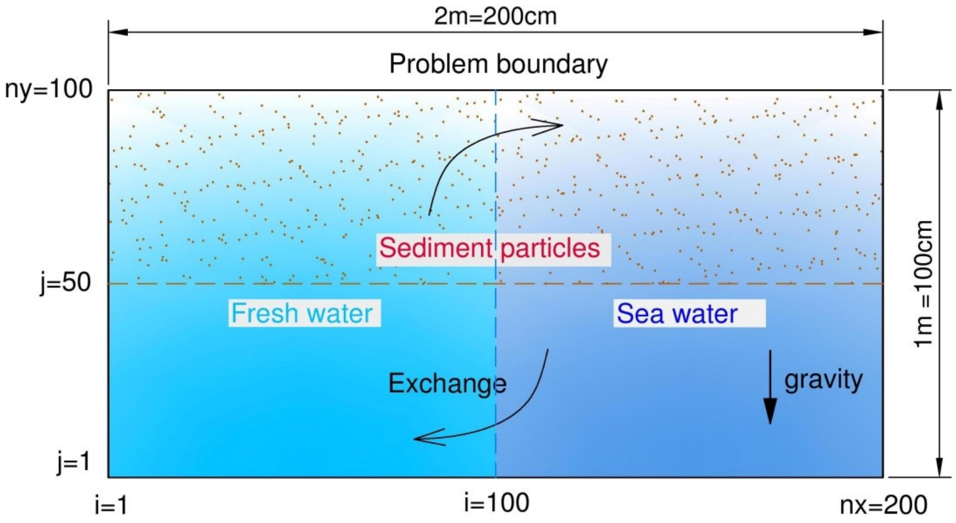


Figure 8. Specification of the lock-exchange problem with the hindered sediments in the upper part.

The results of the lock-exchange problem with the sediments – concentration field and particles – for two cases are shown in **Figure 9** and **Figure 10**. In the figures, red contour lines present the sediment concentration while blue contours show the salt concentration predicting the interface between fresh water and seawater. The brown points are the sediment particles. The figures show the situation of the exchange flow of fresh and seawater with the sediment transport at the selected times followed by the velocity magnitudes of the particles for each case. Most significant argument on the results shown in **Figure 9** is the agreement between the results of the PTM and the ADE. The PTM is properly validated against the ADE. We can observe that the development of the vortex of Kelvin-Helmholtz instability is appeared at 8 s and after due to the strong shear stress at the interface between seawater and fresh water. Because of the vorticity generated by shear stress in interface of fresh and sea water, the sediment particles and concentration field are also rounded up with the flow. Due to flocculation process, the particles traveled during 5 s in the sea water become larger and the sedimentation rate is increased. The growth of the sedimentation rate and the sizes of sediment are clearly shown in the second column of each case in **Figure 9** and **Figure 10**. Also, it is shown that the sedimentation occurred in the left bottom corner of the enclosure at 12 s and afterward in case of the flocculation is present, while the situation is different in the case without flocculation. In the case without flocculation, the particles follow the flow field dynamically and resuspension due to vertical flow is shown clearly (at 16 s left bottom corner in **Figure 9**). In the case with flocculation process, the resuspension of the sediment particle is lower than that of the other case. More precisely, no sediment ascending appears in the left top corner at 20 s in the case with flocculation, while the plenty of sediment particles are transported in the same area in the case without flocculation. Although it is obvious, the results of the simulations brought the concrete conclusion that the effect of flocculation process is important for the sedimentation of cohesive sediments.

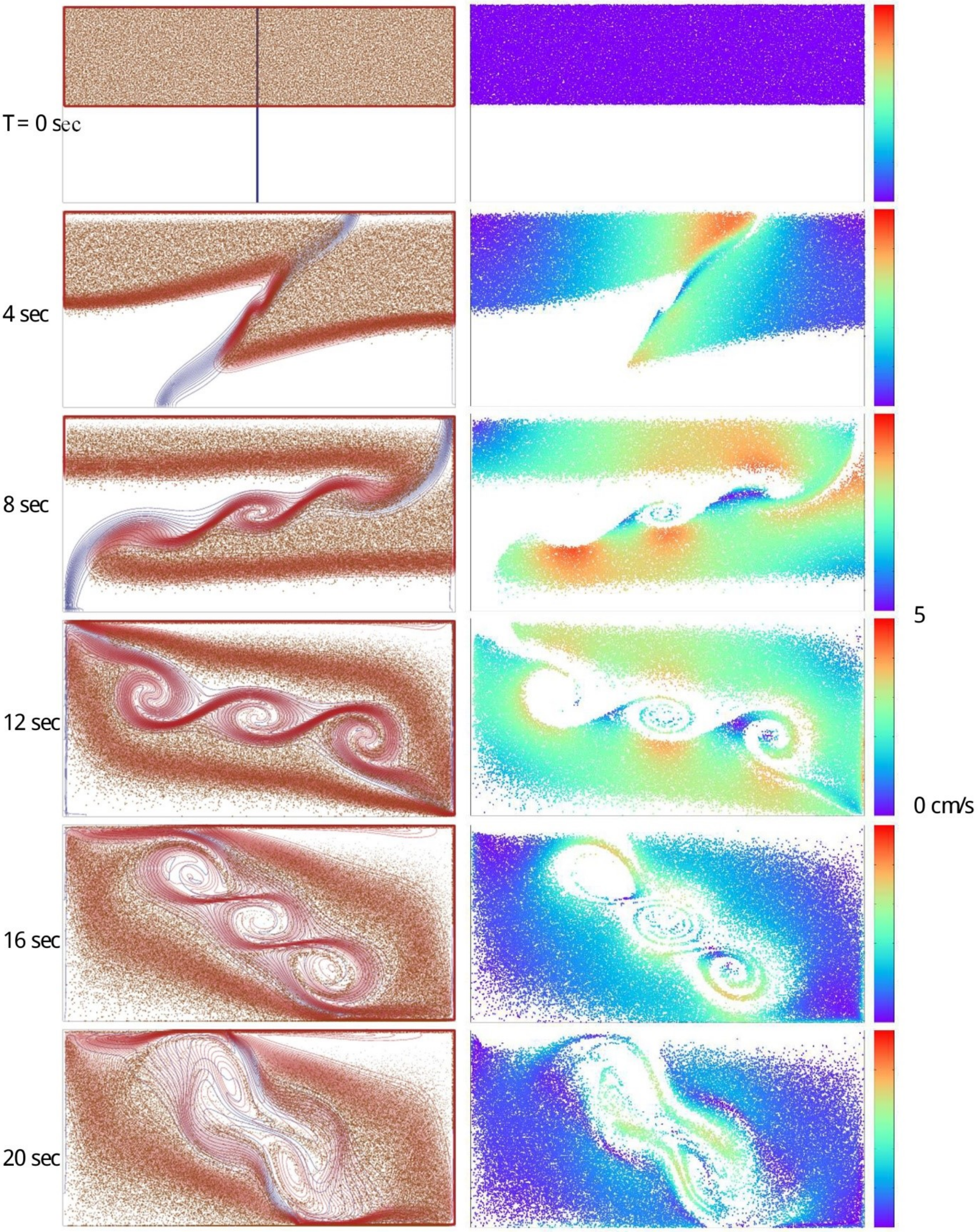


Figure 9. Time sequences snap shots of the lock-exchange without flocculation model. The first column of figures shows the concentration of salt and sediment as well sediment particles while the second column of figures only shows the absolute velocity of sediment particles.

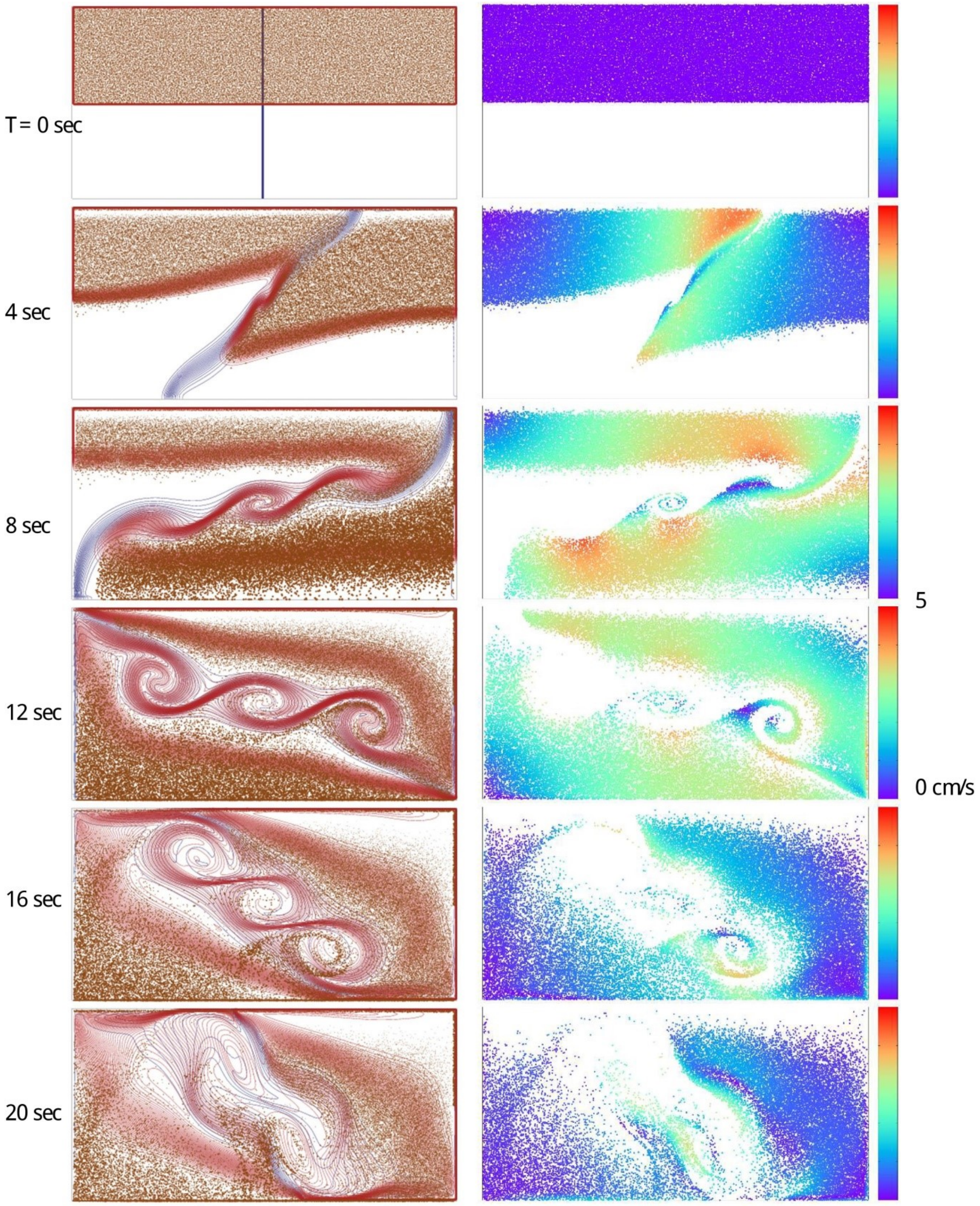


Figure 10. Time sequences snap shots of the lock-exchange with the flocculation model. The representations are the same as previous figure.

4.2 Sediment problem at an estuary of the Ohkouzu diversion channel

4.2.1 Introduction to the estuary of the Ohkouzu diversion channel.

The Ohkouzu diversion channel is a shortcut channel of the Shinano River to the Japan Sea preventing flood inundation disaster of the downstream area of the Shinano River. The Ohkouzu diversion channel was constructed 90 years ago and since then sediments have been discharged to an estuary of the channel. Origin of the estuary was formed by rocks. After the completion of the Ohkouzu diversion channel, a nearby coast has been gradually formed as sandy and has extended to the offshore about several hundred meters (see **Figure 11**). It is important to mention that a sediment contribution of the Shinano River to the Japan Sea has split into two estuaries; a mouth of the Ohkouzu diversion channel and a mouth of the Shinano River. An amount of sediment contribution in the estuary of the Ohkouzu diversion channel is higher than that of the Shinano river mouth, because of the flow regulation on a diversion structure at the head of the Ohkouzu diversion channel (Sane, et al., 2005). Sediment phenomena in estuaries of the Japan sea are affected by seasonal influences such as wind waves caused by monsoon in winter, floods due to snow melting in spring, large scale floods during rainy season and typhoon in summer. While the most of the coastlines near the estuaries of the Japan Sea has been eroding, the coastline of the estuary of the Ohkouzu diversion channel has been extended. However, the erosion is taking place in the river mouth and has been warning the possible damage to the control structures in the estuary.

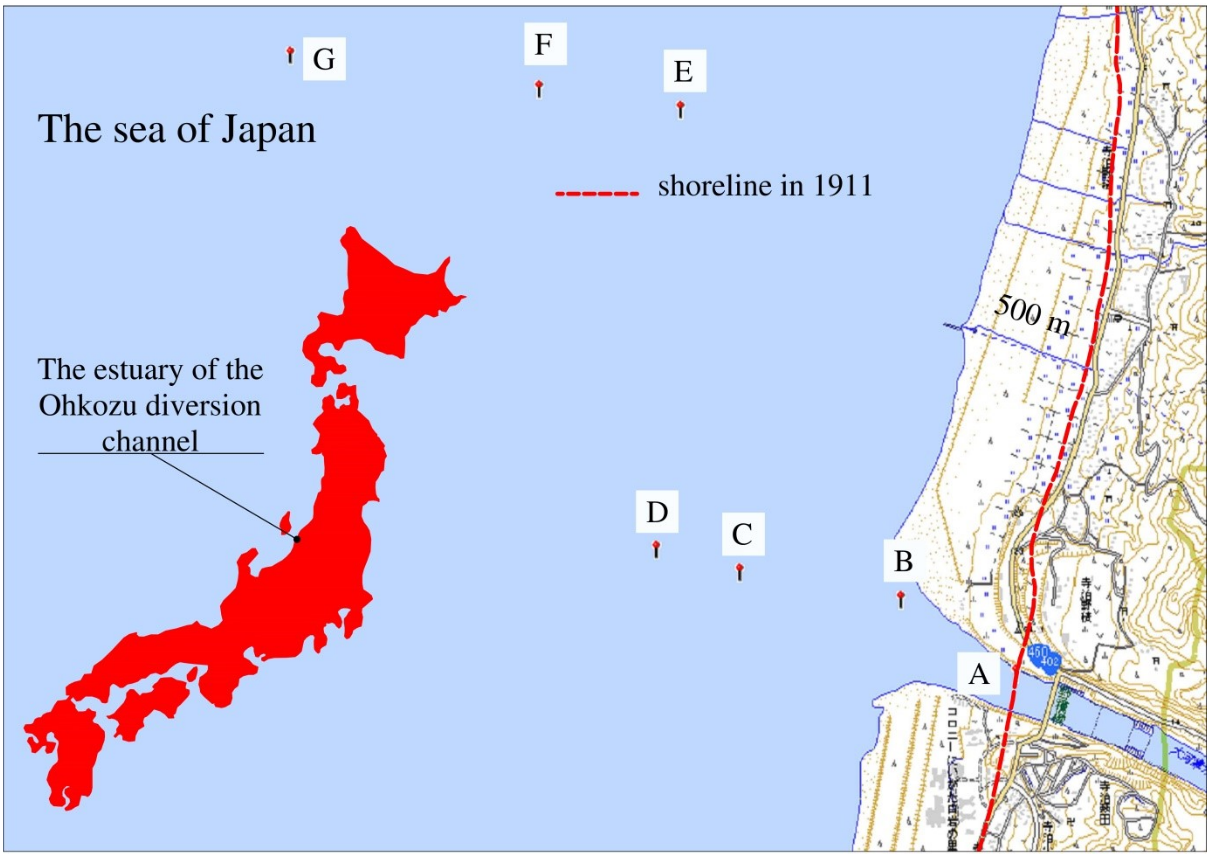


Figure 11. The estuary of Ohkouzu diversion channel, the Japan Sea. Points, A, B, C, D, E, F, and G are the observation sites where salinity and turbidity were measured.

4.2.2 Observations at the estuary

The estuary of the Ohkouzu diversion channel has experienced large floods and the most recent and substantial flood happened in July 2011. The maximum flow rate was more than 8310 m^3/s. During this flood event, large amount of sediment discharged to the Japan Sea by the estuary. In order to assess variation of the estuary terrain change after the flooding, the Shinano river authority conducted a narrow multi-beam bathymetric survey in the estuary, as shown in **Figure 12**. Also, researchers (Hideo, et al., 2013) have conducted a survey of the littoral sedimentary environment using core sampling of sedimentation layers and have concluded that the sediment supplement of the Ohkouzu diversion channel has played an important role in the formation of nearshore coastal area. In the survey, they (Hideo, et al., 2013) measured the vertical distribution of salinity and turbidity for several sites in onshore and offshore area, shown in **Figure 11**. An ADCP (Acoustic Doppler Current Profiler) flow measurement and vibrocore sampling of sedimentation layer were also conducted. In the report, they explained that the fine sediments transported by major flood late on July 2011 have created a mud layer in offshore sea bed, where samples were collected. **Figure 13** shows vertical distributions of salinity and turbidity on the site A to G, whose locations are shown in **Figure 11**. The configuration of the most shore sites, point A, was close to the river mouth site, point B, which does not have salinity and has huge account of turbidity. Amount of turbidity tends to be slightly increased through the river terrace according to the point A to B. However, turbidity and salinity of the point C had become different from the point A and B, because of the influence of sea water. On the point C, salinity actively increased up to 2 m below the sea surface and remained almost constant to down. In contrast, turbidity decreased rapidly until around 2 m below the sea surface and gradually decreased through the bottom of the sea. But in the offshore of the estuary, the point C and D, had a high turbidity in the vicinity of the sea bed, which might show that the sediment settlement process caused by the flocculation due to sea water in that area. From the points C to G, turbidity in near sea surface was still remained. This can be explained that the sediments may be re-suspended by the waves or density currents. When sediments discharged from the river mouth enter into the sea water, settling processes of flocculated sediments are described by turbidity decrease in the latter points. Near the free surface of the sea within 2 m depth, turbidity decrease with the increase of salinity shows the density current of the sea water. The fine sediments may be suspended in this region. This tendency of salinity and turbidity profile had been observed at the offshore measurement sites E, F and G. In the point F and G, a disappearance of turbidity showed the sediments eventually settled and deposited on the sea bed. Those are an interpretation of the sediment transport processes in the estuary of the Ohkouzu diversion channel given a basis of the observation. This observation need to be confirmed by the numerical simulation (Takeshi, et al., 2016a).

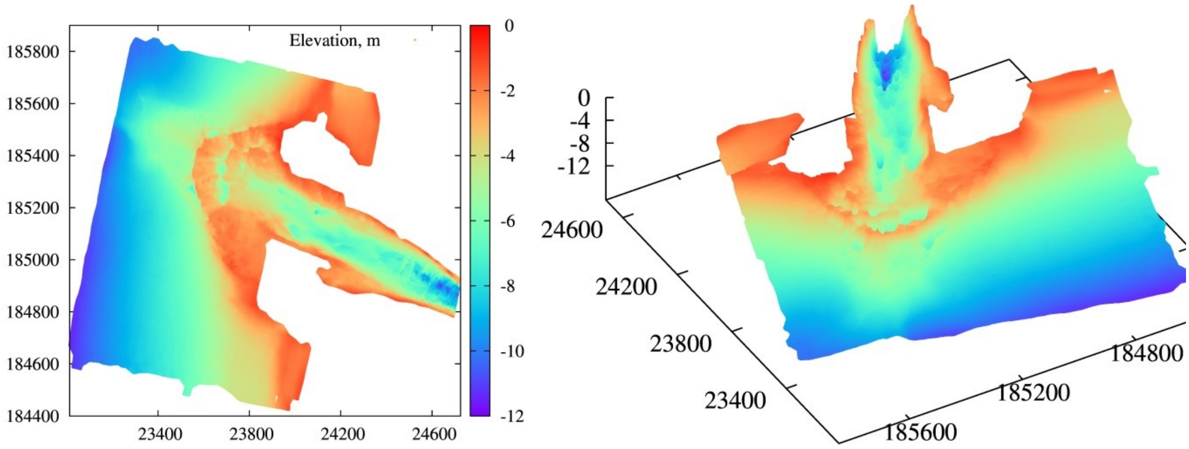


Figure 12. Two- and three-dimensional bathymetry data of the estuary of the Ohkouzu diversion channel.

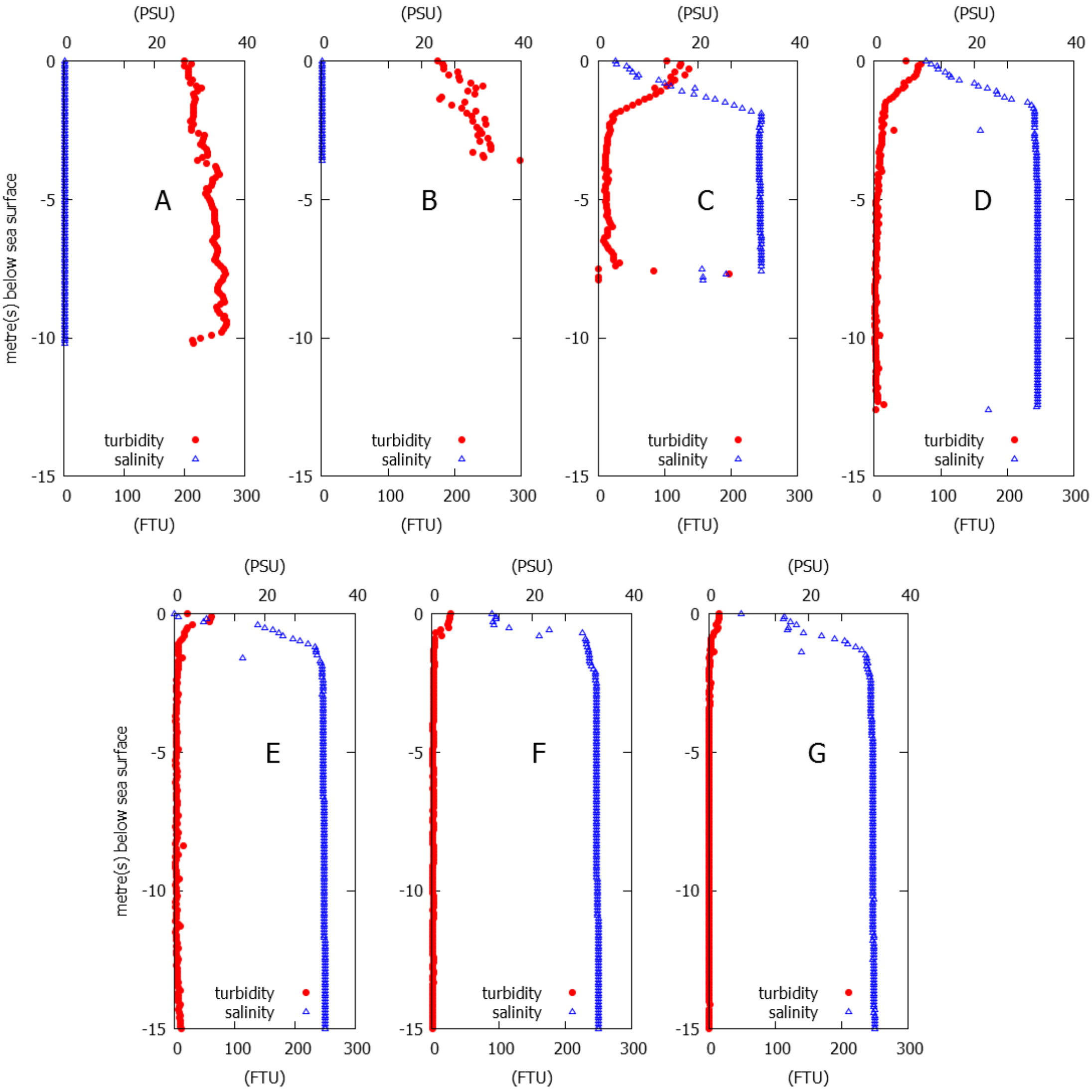


Figure 13. Measured salinity (PSU-Practical Salinity Unit) and turbidity (FTU-Formazin Turbidity Unit) at sampling sites in the estuary of the Ohkouzu diversion channel (Hideo, et al., 2013).

4.2.3 Problem statement

Sediments discharged from a river mouth during a flood are transported long distance away due to inertial flow force, even though cohesive sediments bind together when they enter sea water and increase their settling rate. Eventually, fine sediments settle down gradually on the sea bed as an unconsolidated layer. Unlike, coarse sediment particles settle down in sedimentation area, where they will contribute to the further development of a river terrace. Nearshore coastal area formation is performed by the transported sediments due to wind driven waves. Although, these general processes are the same for the estuary of the Ohkouzu diversion channel, more detailed understandings of the sediment transport at the estuary has been in demand.

Coarse sediments discharged from the river with fresh water are ascended by a salt wedge, before they enter into sea water. While, fine sediments are not only affected by density current, but also affected by the waves. Also, it is possible that sediments are re-suspended from weakly formed sand waves at the sea bottom by shear stress driven by the salt wedge. Successfully ascended sediment particles from bedload become fully suspended sediment particles until settling back in somewhere. All those known phenomena from the observation at the estuary need to be confirmed by the numerical simulations. In order to study the process, we have developed a numerical model with flocculation effects for sediment transport discharged from a river mouth using the models described in Chapter 2 and 3. The model also involves the density current, since it is stated that the density currents have significant influences for bedload in the vicinity of the salt wedge as well for the suspension of sediments. Density stratification and interaction between salt wedge and fresh water flow must be considered in an estuary sediment transport modeling (Tokuzo, et al., 2001). The sediment transport was modeled by the PTM and the ADE simultaneously. The salinity concentration of intrusion of density current was also treated by the CDE. It is worth to mention that the use of Lagrangian particle tracking approach for sediment transport provides many opportunities as well as handling the flocculation process in sea water.

The aim of this modeling was to describe the sediment transport process at the estuary of the Ohkouzu diversion channel reported by (Hideo, et al., 2013) to verify their observation and to get more a clear understanding. In order to fulfill the purpose, numerical modeling, including a density current and a flocculation of sediment was carried out in the 2D and 3D space.

4.2.4 Settling velocity and flocculation

The settling velocity for the sediment transport is the most important parameter in order to account the sedimentation. The settling velocity is a function of the sediment concentration. As increase of the concentration, the settling velocity of the sediment increase. When the concentration becomes about several thousands the interaction between the sediment particles decrease the settling rate of the sediment. This is considered as a hindered settling process. In a study of Tsurutani et al. (鶴谷広一, et al., 1989) corresponded with other studies, the maximum value of settling velocity of estuary sediments was observed around 0.258 cm/s, which gives the particle size of 55 μm when the same velocity applies to the Stokes velocity formula. Simplifying the Stokes equation yields the relation between the settling rate and the sediment diameter as

$$w_s = 8711d^2 \ , \tag{103}$$

where d is the sediment particle diameter in cm. With the minimum settling velocity observed in (鶴谷広一, et al., 1989), the minimum size of the sediment particle can be approximated around 10 μm. So the average particle size will be interpolated as 32 μm. With this average particle size, we can obtain settling velocity of 0.09 cm/s from Eq.(103).

We also carried out simple experiments settling sediments in fresh water and seawater. In a beaker of 500 cc, fresh and sea water with the suspended sediment of 4000 mg/L was filled and stirred for the same time. The seawater of 3 percent was sampled from the estuary. Also the sediments used in the experiments were prepared after collected from the bottom of the estuary. After the sediments are uniformly distributed in the two cases of water, the concentration of the turbidity (suspended solids - SS) was measured for each second with equally leveled optical turbidity meters for the beakers. The amount of the suspended solid against the experimental times is given in **Figure 14**. As seen in **Figure 14** (a), the decrease of the suspended solid in seawater is higher than that of fresh water. In other words, the sediment in seawater settles faster than the sediments in freshwater. This can be described by the flocculation process in seawater. An enlarged part of the SS against time shown in **Figure 14** (b) gives more detailed explanations for the settling rate and the flocculation. The SS increased for initial 20 s because of the presence of the velocity field by stirring. After the external velocity field is suppressed, the sediment particles are only experienced by the gravitational field and started to settle. The amount of the SS in two cases is almost the same until a point A, which is the starting point for the difference. Actually, the point A is the starting point of the sediment flocculation in seawater and is measured at 60 s. The amounts of settling SS in two different waters are bifurcated until a point B. From the point B, the rates of decreasing SS were the same in the two cases of water. This implicates that the flocculation of sediments in seawater is faded away and started to lose the effect on the sedimentation. From the point B to the end of the experiment, the decreases of the SS were the same and curves continued in parallel.

Using the data of the SS in two cases of water, the settling rates against time in these cases can be estimated from Eq.(38) by eliminating the advection term and the diffusion term in the x direction. The equation for the inverse problem for the settling rates can be written as

$$w_s \frac{\partial c}{\partial z} = \frac{\partial c}{\partial t} + D_{sed} \frac{\partial^2 c}{\partial z^2} \, , \tag{104}$$

where c takes the value of the SS. The settling rates obtained with Eq.(104) against the experimental times are plotted in **Figure 15**. The fluctuation of the SS yields the fluctuation for the settling rate. However, the trend lines for the estimation give the clearest view for the changes of the settling rates in time. A ratio of settling rates over whole time shows that the settling rate of sediment in seawater almost doubled the settling rate of sediment in freshwater. In the flocculation effected range, until the time of 400 s, the settling rate was accelerated 1.71 times in seawater. The flocculation results the bigger sediment particles stuck to each other. The increase of the particle size accelerates the settling rate. Thus, with these average settling rates no matter what is estimated in whole time or initial 400 s, the radius of an sediment particle is expressed as $r \cong 2r_{in}$ by using Eq.(103). This expression confidently proves the simple flocculation model given in Section 2.3.2 can be applied to the study. This experiment and the sediment size observation with a microscope gives the base scaling in the flocculation model as $\alpha_o = 1$. The critical time for the flocculation is found 300 s from the experiments. The average particle sizes are found as 26 ~ 37 μm, which agree with the average approximated from (鶴谷広一, et al., 1989).

Some of the complicated parts to model sediment transport with the traditional and the new particle approach have been severally clarified with the preceding discussions and experiments. Basically, the settling velocity discussed above is used for the simulation of the ADE for the sediment transport. Similarly, the particle sizes in the PTM are given in the sediment size range defined above. For each case of simulation, the settling velocity is the velocity for an average sized particle considered in the PTM.

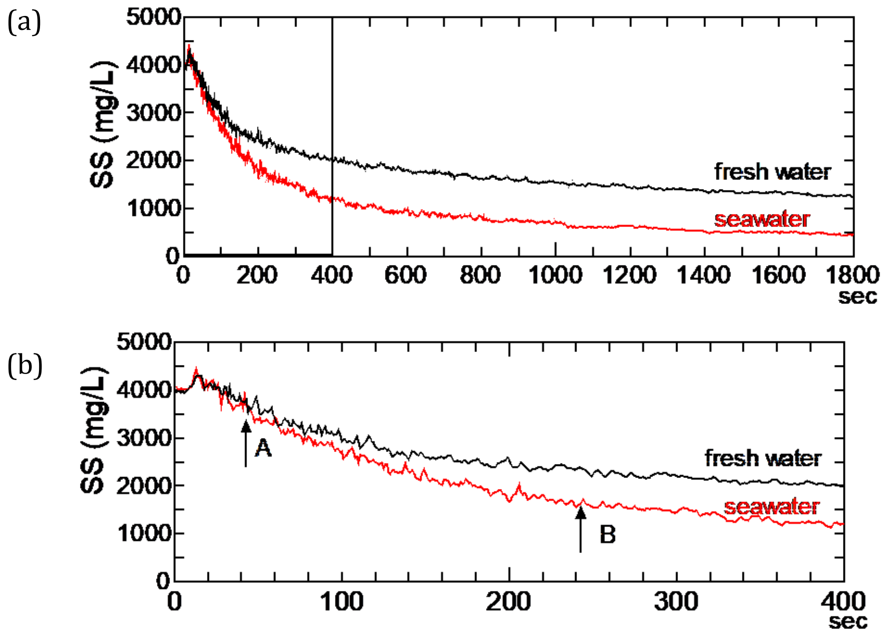


Figure 14. Sediment settling experiments: (a) time series of suspended solid in fresh and seawater in full experimental time and (b) time series of suspended solid in the enlarged area until 400 s.

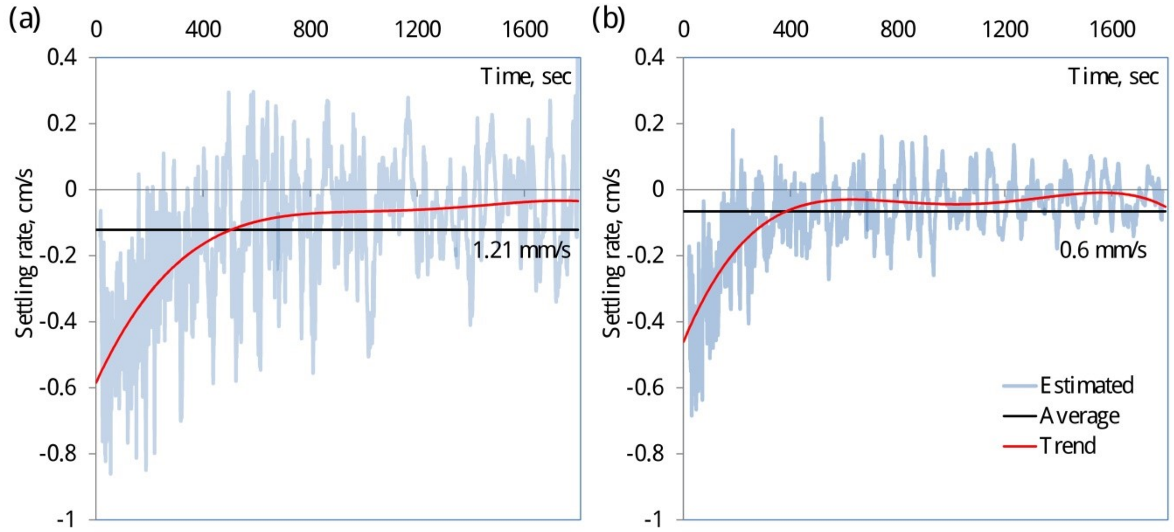


Figure 15. Settling rate estimation from the time series of suspended solids: settling rate in (a) seawater and (b) fresh water.

The sediment concentration at the inflow boundary in the ADE is assumed to be 1.0 and it will be distributed further due to river inflow. It is also possible to get higher concentration value than 1.0 at the bottom of the sea because of the sedimentation. In other words, the amount of sediment concentration $c = 1.0$ means that just for a reference value. Numerically, a total amount of sediment has no limitation for the large value.

4.3 2D numerical simulations for the sediment problem

4.3.1 Model setup

The results of the lock-exchange problem confirm that the novel particle tracking approach can be applied to the real field problem. The numerical scheme composed as **Figure 16** was applied to the two-dimensional numerical simulation of the estuary of the Ohkouzu diversion channel. The numerical scheme consists of two approaches for sediment transport and two options for the NSE. The particle tracking approach has coupled with the simple flocculation model. For the PTM, the sediment sizes are important parameters, which should be defined in the way that corresponds to the settling velocity of the ADE. In general, the settling velocity and particle sizes are defined by the experiments and are reconciled with [鶴谷広一, et al., 1989]. For the flocculation model, the flocculation time criteria, which is practically valid time for particles flocculating with each other gradually, for fine sediments traveling in seawater was given as parameter based on the experiments.

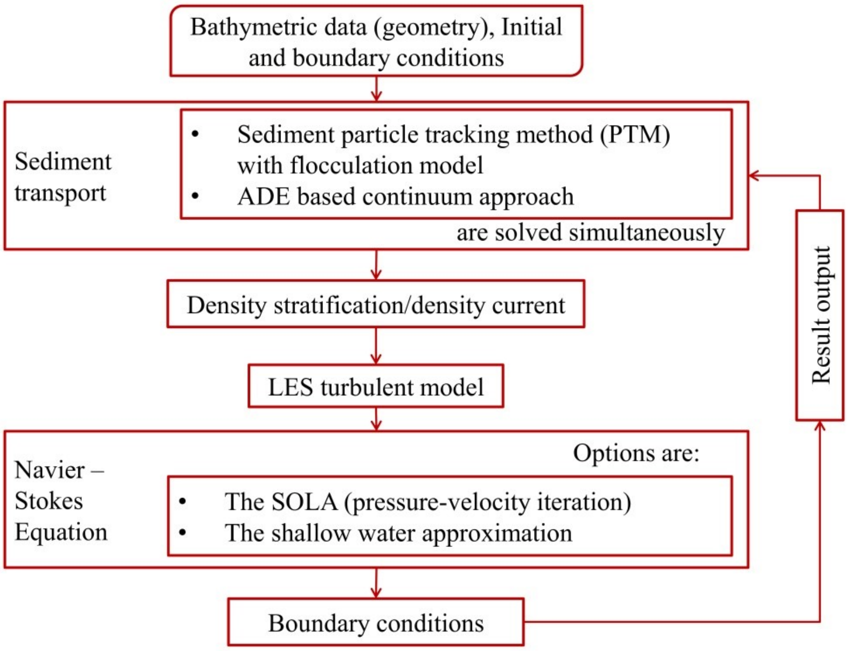

Bathymetric data (geometry), Initial and boundary conditions
Sediment transport
Sediment particle tracking method (PTM) with flocculation model
ADE based continuum approach
are solved simultaneously
Density stratification/density current
LES turbulent model
Navier – Stokes Equation
Options are:
The SOLA (pressure-velocity iteration)
The shallow water approximation
Boundary conditions
Result output

Figure 16. A numerical scheme for the sediment transport modeling in an estuary.

Two dimensional terrain data used in the computations was prepared from the digital bathymetry data of the Ohkouzu diversion channel of 2011 shown in **Figure 12**. A computational domain of the field scale simulation was formed with the rectangular grids in an area of 2.490 km in length and 29 m in height and the additional layer of the cells are generated for the boundary condition. The computational domain depicting the initial condition for sediment and salt stratification is given in **Figure 17**. For the ocean bed, the no-slip boundary condition was imposed. The upper surface of the domain where a free surface would be presented is modeled as the slip boundary. For inlet, the inlet velocity of 32 cm s^{-1} (10 percent of design flood discharge) was given in all cases. Whereas the outlet boundary condition is imposed in the condition that balances the discharge computed from the inlet velocity.

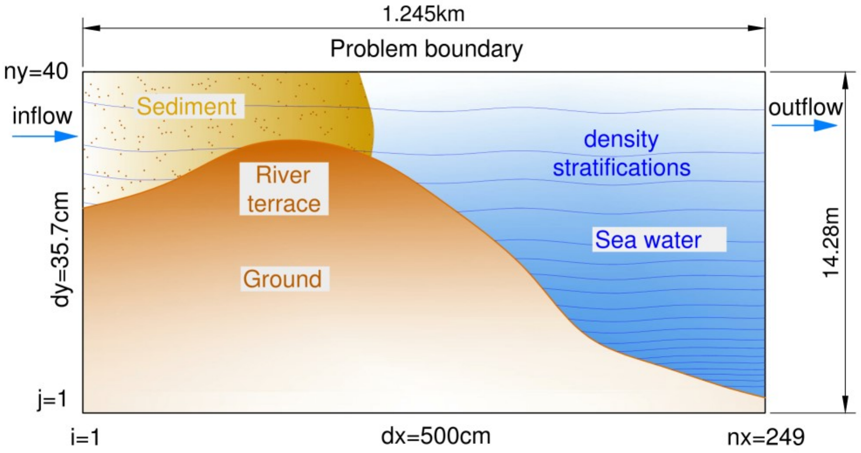


Figure 17. Schematic description of the computational domain for the sediment transport modeling in the estuary of the Ohkouzu diversion channel.

4.3.2 Numerical results

In the field scale simulation, the SWA was selected instead of the SOLA method for the NSE in order to exploit the computational efficiency and comfortability of geometry in the SWA. Meanwhile, the sediment transport in the estuary was solved by two methods explained in Section 2.3. The cases involved in this research are given in **Table 1** with the other important parameters in the simulations.

Table 1. The simulation cases and parameters

Cases	w_s	c	D_{max}	D_{min}	ρ_{sed}	N[1]	ρ'	Δx	Δy
1	0.12	1.0	54.4	4.0	2.6	100	0.02	500	35.7
2	0.09		32.2						
3	0.05		24.4						
4	0.02		12.2						
Units:	cm s^{-1}	-	µm	µm	g cm^{-3}	pieces	g cm^{-3}	cm	cm
Notes:	Parameters for the ADE		Parameters for the particle tracking method				Density current	Grid spacing	

Each case has experimental parameters including the particle sizes and settling velocity. The time step for the computations was adaptively computed by the stability condition described in Section 3.1.1. The computation was totally performed for 16 hours, 6 hours of which have spent only for creating the fully developed flow condition like a natural flow in the

[1] N is the number of representative sediment particles generated randomly in a single cell within given range of the diameter. Totally, 98200 particles were considered for the 2D simulations.

estuary and the rest are devoted to the simulation of the sediment transport and the density current. Times presented in the following explanation and figures are the count when the sediment and the density current simulation started in the simulation. Generally, sediment with fresh water is flowing from the left side inflow boundary (river mouth) toward to the right side outlet boundary (offshore), see **Figure 17**. The bathymetry profile shows a river terrace outside of the river mouth.

Based on the solution of the ADE, we measured the vertical distribution of salinity and sediment concentration at several points, since the simulation domain has included the observation sites B to D (see **Figure 11** and **Figure 13**) approximately. The vertical distribution of salinity and sediments are plotted in **Figure 18** for the approximated sites, which are close to the above observation points and are titled by distances measured from the inlet boundary. In **Figure 18**, the first graph, 50 m from the inlet, shows the guarantee of no salinity near the inlet boundary and the last graph at 950 m from the inlet shows the sediment and salinity distributions in offshore parts. Despite it, the other graphs are assumed to be equated to the corresponding the nearest observation points. Roughly speaking, the salinity is still not detected at 50 m and 250 m from the inlet. But the point at 750 m, the sediment concentration decreases with the salinity increases, which shows the same properties with the observation point C. The point at 850 m from the inlet has less sediment concentration near the sea surface than the bottom of the sea and the salinity distribution is kept its previous and next shapes of profiles. At the point at a 950 m from the inlet boundary, comparatively large sediment concentration takes place near to the sea surface and the sea bed as like as the observation site E. Practically, it is obvious that the sediment and salinity distributions by conventional method can be defined more accurately on the fine grid, however, it is difficult to evaluate how does the sediment particle size effects on the spatial distributions of sediments from the results obtained the conventional methods. Moreover, the combined effects of the flocculation and the density current on the sediment transport could be a challenge for the conventional method.

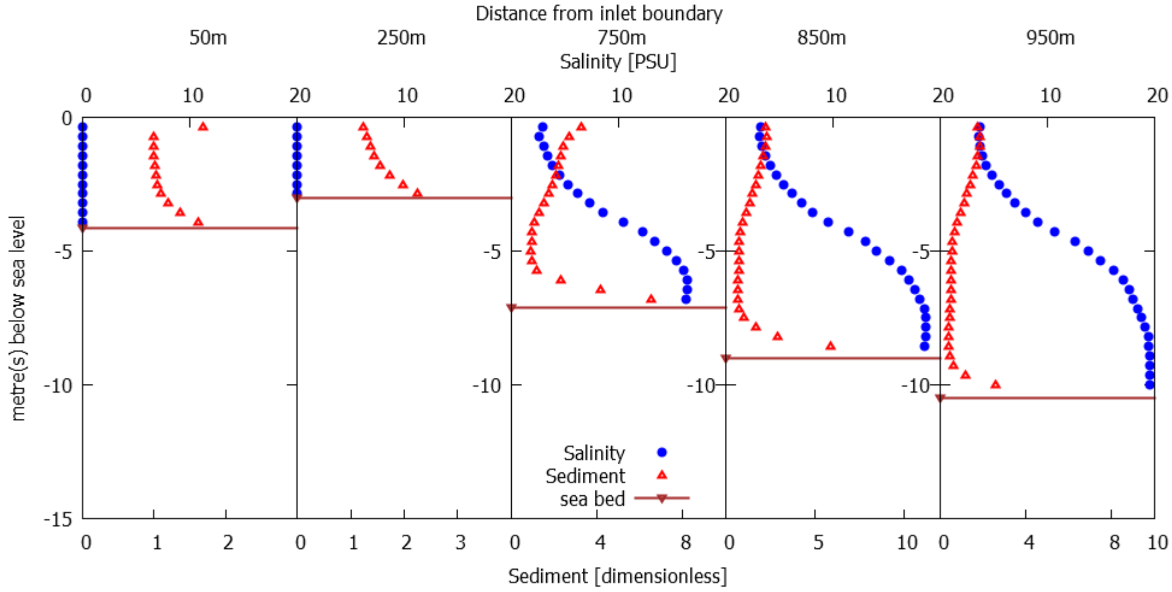


Figure 18. Simulated vertical distribution of salinity and sediments by the ADE. From those, the middle three plots are approximately nearest points to B, C, and D points in **Figure 13**, respectively. The profiles are measured at 2 hours after the simulation considered. A parameter used in this simulation: w_s=0.09 cm/s.

The numerical results of the sediment transport and density current field in the 2D domain are given in **Figure 19** and **Figure 20** for four cases changing the settling velocity and particles sizes. The blue and red contours show salinity and sediment concentration, while the particles are expressed by brown points. At the upstream side of the computational domain (see **Figure 17**), initially hundred particles are generated and distributed randomly in every single computational cell in order to represent the sediment particles. Only a one-time generation of the sediment particles for whole simulation time enables to figure out the sediment transport clearly in the domain rather than the generating particles in every time step of the computation. Lines of contour remark the spatial changes in salinity and sediment concentration in **Figure 19** and **Figure 20**. It is shown that the river water pushes the seawater toward the offshore direction, meanwhile the density current acts opposite direction to the flow. Subsequently, an interaction of freshwater and seawater makes the upward flux near an interface between them and creates the density current front backward of the river terrace. In front of the density current, the flow forms circulation due to strong shear stress between the river flow and the density current. When the settling velocity or sediment particles have a bigger value, the sediments are totally settled down around the river terrace and the steep side after the terrace until 5 hours, shown in **Figure 19** (a). Because the flocculation effect on the larger sediment particles intensively accelerates the settling process. The flow circulation and the front of a salt wedge cannot effect near bed sediment particle transport in case of large sediment particles, case 1. In **Figure 19** (b), case 2, the sediment particle size and settling velocities are smaller than case 1 in **Figure 19** (a) and the sediment particles near the front of the salt wedge are circulated with the flow and this is still appeared end of the simulation time. It shows that the settled or tumbling sediment particles near the front of the salt wedge or the interface between the fresh water and the seawater are re-suspended again and again. In **Figure 20**, the results with more fine sediment particles and lower settling velocities are shown. In case 3 with the most fine sediment particles, the sediments are transported straightly in the offshore direction and large amount of sediment stays suspended for a long time, as seen in **Figure 20** (d) during the middle of the simulation time. This reveals that the finer sediments such as fine silt and clay sediment can be transported for long distance and deposited in the deep ocean bed. This was also reported with the observation of sedimentation at the Ohkouzu diversion channel estuary in 2011 (Hideo, et al., 2013). Generally, the sediment particles are transported with the inertial flow force until a 1 hour and then the sediment particles greatly settle down due to the flocculation effect between 1 hour and 5 hours.

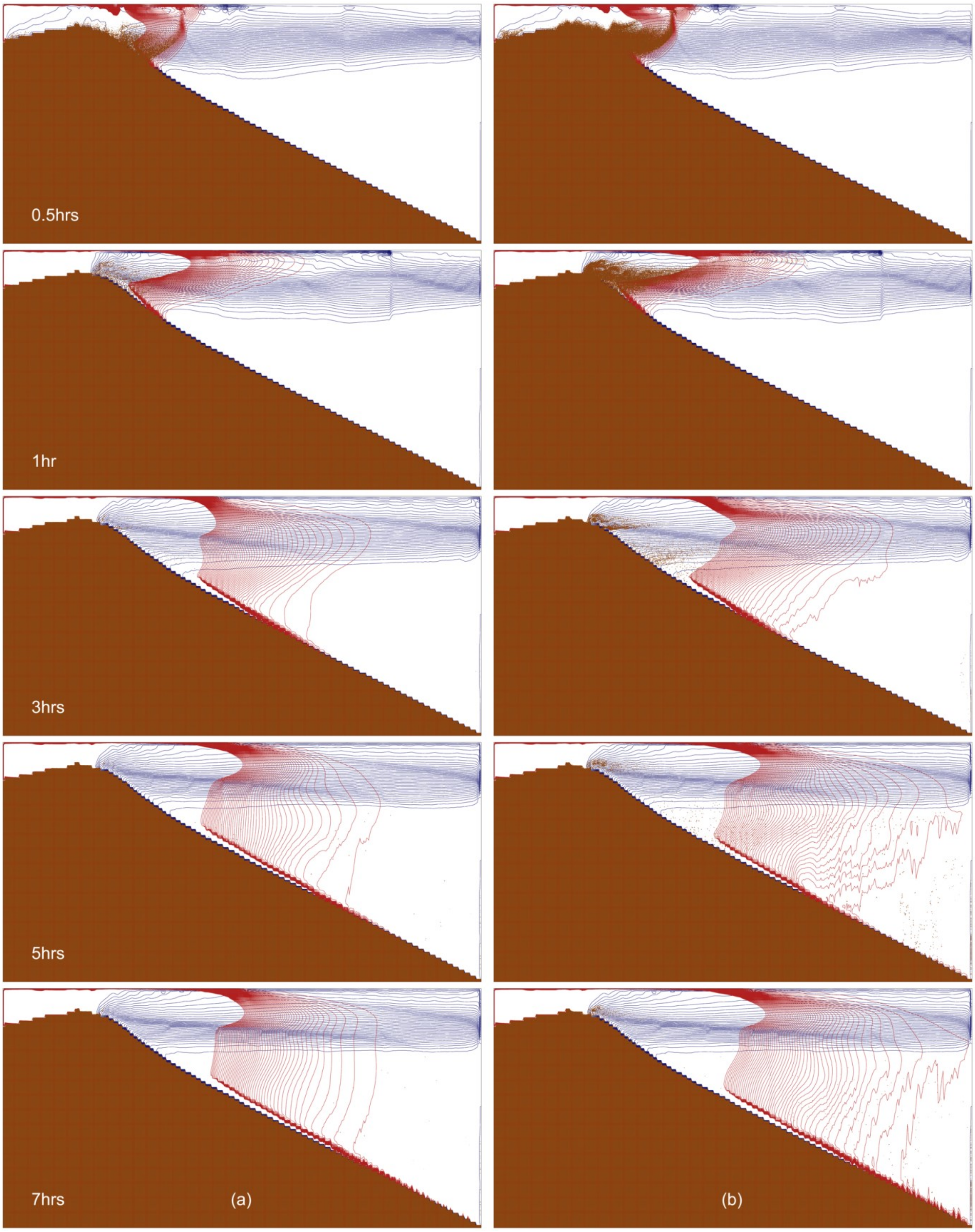


Figure 19. Time sequencing results of the sediment transport and density current at the estuary of the Ohkouzu diversion channel: (a) case 1 with the sediment settling velocity w_s=0.12 cm/s, sediment particle sizes D=54.4-4.0 μm, (b) case 2 with w_s=0.09 cm/s, D=32.2-4.0 μm.

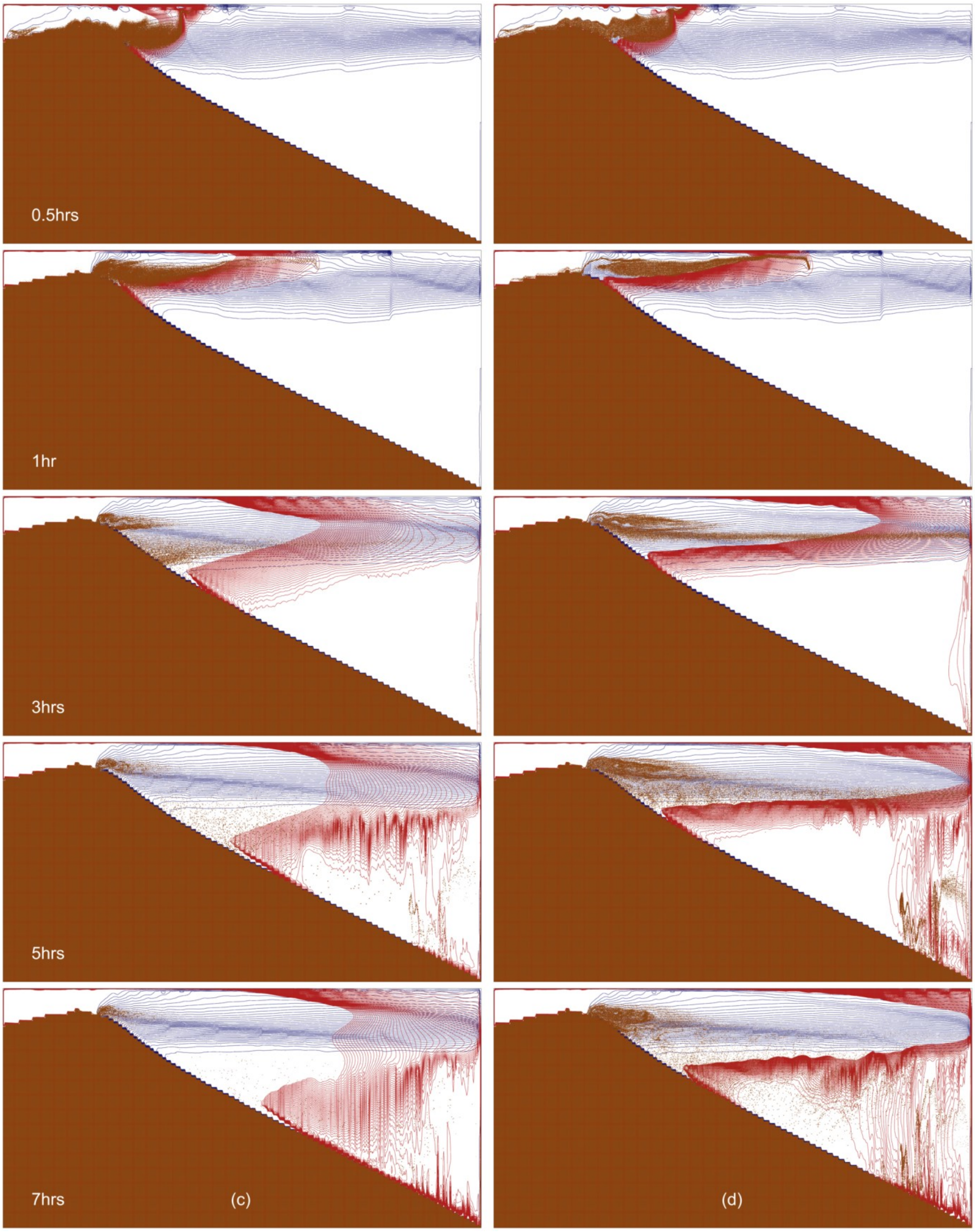


Figure 20. Time sequencing results for the sediment transport and density current at the estuary of Ohkouzu diversion channel: (c) case 3 with w_s=0.05 cm/s, D=24.4-4.0 μm and (d) case 4 with w_s=0.02 cm/s, D=12.0-4.0 μm.

The PTM has a drawback related to the treatment of the total amount of sediment deposition on the sea bed. Even if a new personal computer has a successful advancement of the performance, the total amount of sediment particles is beyond the limit of the computer memory and CPU processing speed. However, the distribution of probability of particles yields valuable information on the sediment distribution.

To investigate more about sediment distribution, we measured the vertical and horizontal distributions of sediment particle in four cases and shown in **Figure 21** and **Figure 22**, respectively. In case 1 with the coarsest sediments, the vertical distribution is not included in **Figure 21**, since the large number of sediment particles is settled down at around the river terrace. But it can be appeared with violet color in **Figure 22**. In case 2 with the particles of D=32.2-4 µm, the distribution of particle distance between 750 m to 950 m is lower than that of the succeeding cases (case 3 and 4) with the finer sediments as well as the lower settling rates. The sediment particles are settled hugely near 250 m and slightly at 850 m and 950 m, more slightly at 750 m and almost no sedimentation at 50 m at 5 hours after the simulation considered. The same tendency shown in case 3 with D=24.4-4.0 µm, but the distribution of the suspended sediment particle is increased during an 1 hour to 5 hours between distances 750 m to 950 m. Case of the most fine sediment particles, case 4, shows a huge amount of distribution stayed as suspended after 750 m, but it shows no sedimentation after 5 hours at the points of 750 m, 850 m and 950 m. In all cases, there are no suspended sediment particles transporting at 50 m and 250 m after the 1 hour, but a plentiful settled sediment particles. There are no settled sediment particles at 50 m and less settled particles at 250 m after 3 hours, which means that the sediments are re-suspended and transported in the offshore direction as seen clearly in **Figure 22**.

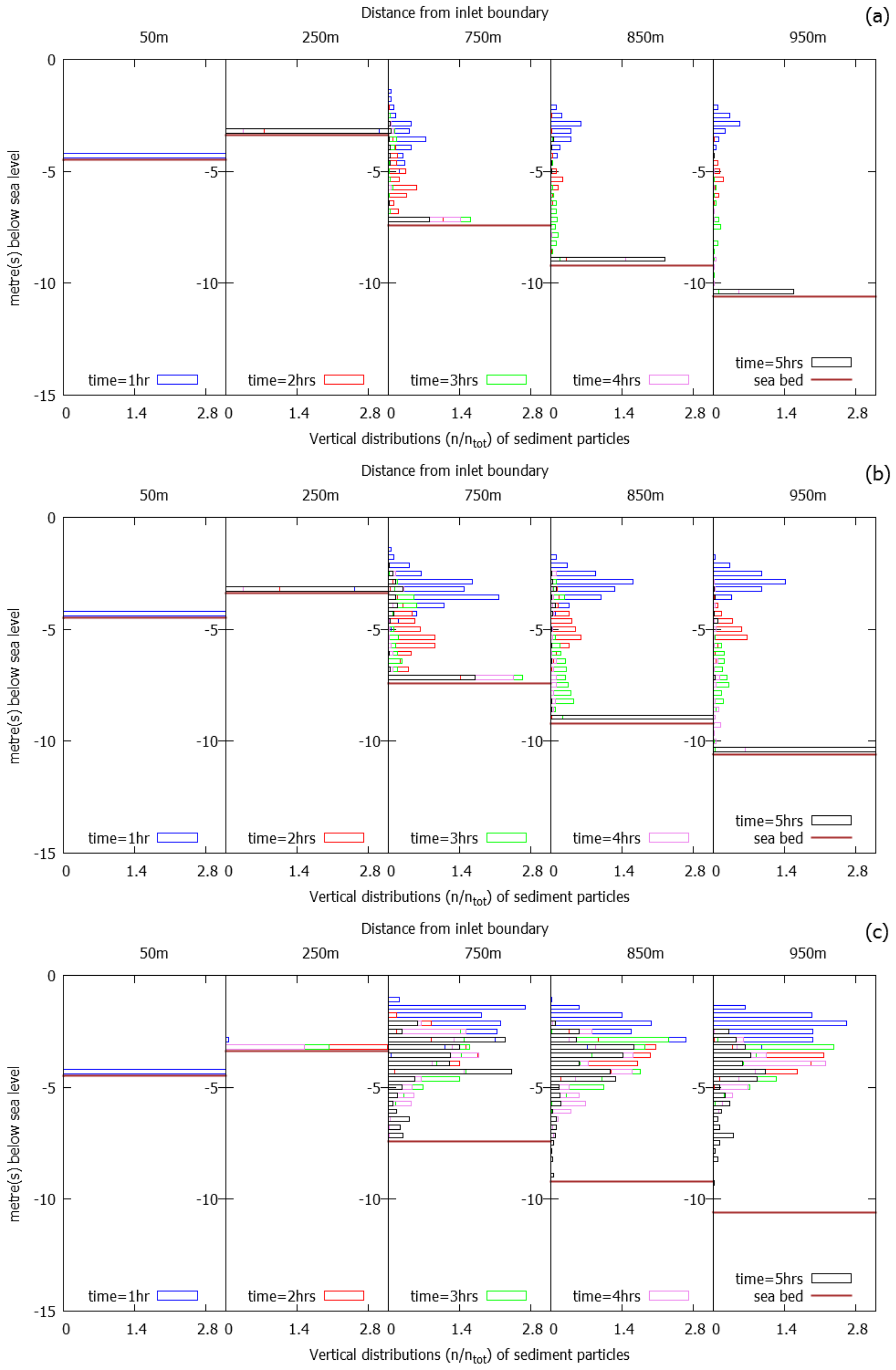


Figure 21. Vertical distributions of sediment particles at the specific times: (a) case 2 (b) case 3 (c) case 4. In single cell, hundred particles are randomly distributed and it can be assumed to be the amount of total particles n_{tot}.

With the measurement of more small time instants, the vertical distribution of sediment particle at arbitrary places can be seen more precisely; however the distribution of settling sediment particles on the bed and horizontal circumstances are not shown clearly. In order to show the horizontal distribution of sediment particles along the longitudinal section of the terrain, the particle distributions are measured at 3 hours, 5 hours and 7 hours in all simulation cases and are plotted in **Figure 22**. At 3 hours after the simulation considered, the sediment transport occupies a range from 150 m to the end of the domain (2490 m) in case 4 (a red histogram in **Figure 22** (a)), while the sediment transport only takes place between 150 m to 1000 m in case 1 (a violet histogram in **Figure 22** (a)). The distributions and settlements of the sediments in cases with D=32.2-4 μm and D=24.4-4.0 μm are situated in between (a blue and a green histogram in **Figure 22**, respectively). For the further simulation times, the beginning of the distribution range is addressed to the offshore direction up to around of the river terrace and the large number of coarse sediment particles settled down there, as seen in **Figure 22** (c). Whereas, the finer sediment particles are tended to settle down in the offshore part, as shown in **Figure 22** (b) and (c).

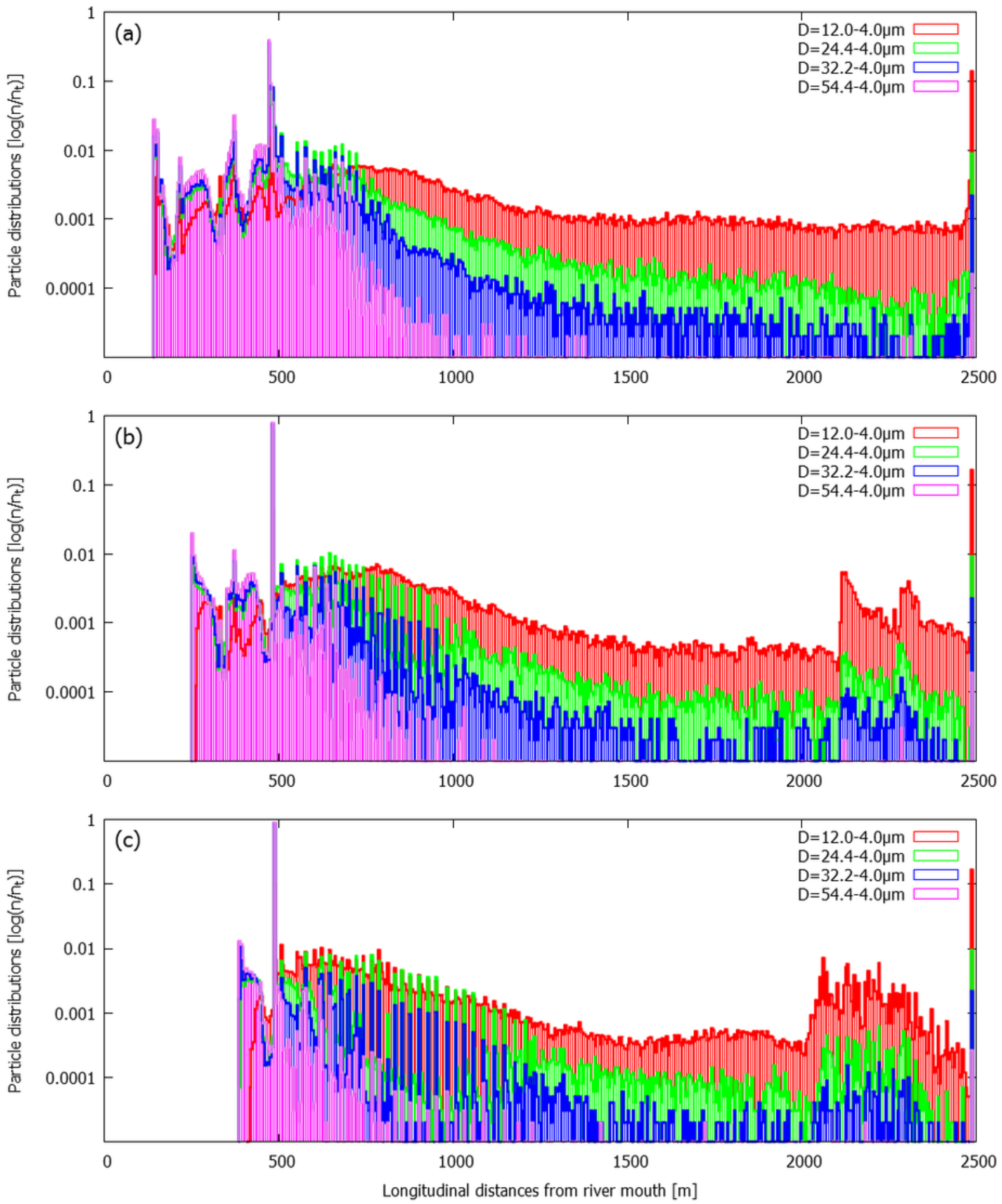


Figure 22. Longitudinal distributions of the sediment particles in the all simulation cases. The results were obtained at the time (a) t=3 hrs, (b) t=5 hrs, and (c) t=7 hrs. In those figures, the total amount of particles n_{tot} is the amount of all particles exists in the simulation domain.

4.4 3D numerical simulations for the sediment problem

4.4.1 Model setup

Less than a three dimensional modeling is sufficient excluding the specific directions of three dimensions that show no physical variations in the problem. An estuary, where a number of factors, including a uniqueness of the environment effects on the hydrodynamics, must be considered as a three-dimensional problem. The sediment transport in two-dimensional space, ignoring the third direction based on an assumption that may be the transports in shoreline direction would be less than that of the vertical direction. Thus the offshore direction of the sediment transport is discussed in the receding sections (the x-z plane model) ignoring the transport in the y directions, more clearly. We saw that the two-dimensional modeling can provide significant results on the sediment transport and spatial distributions in considering directions of x and z. But for the sediment problems relevant for the shoreline, one needs to consider the x-y plane for the modeling. Of course the number of substantial results will be found when one uses the three-dimensional model.

Here, we discuss the results of 3D numerical modeling of the sediment transport at the estuary of the Ohkouzu diversion channel, which were reported in (Takeshi, et al., 2016b) and extensively discussed in (Takeshi, 2016). For the simulations, we used the SOLA method to compute the flow field. The same cases given in **Table 1** were adapted in the 3D simulations, but some of the parameters are different because of the dimension and memory consumptions. A number of sediment particles generated in a cell at initial time has reduced to 10 particles and grids spacing are chosen as $\Delta x = 1000, \Delta y = 1000$ and $\Delta z = 50$ cm. The boundary conditions were the same as the condition applied in the 2D simulations. The bathymetry data for the 3D simulations, given in **Figure 23**, were interpolated to be a digital elevation model (DEM) from the raw point data shown in **Figure 12**. The discharge was constant for all simulations and was the same as the 2D simulations.

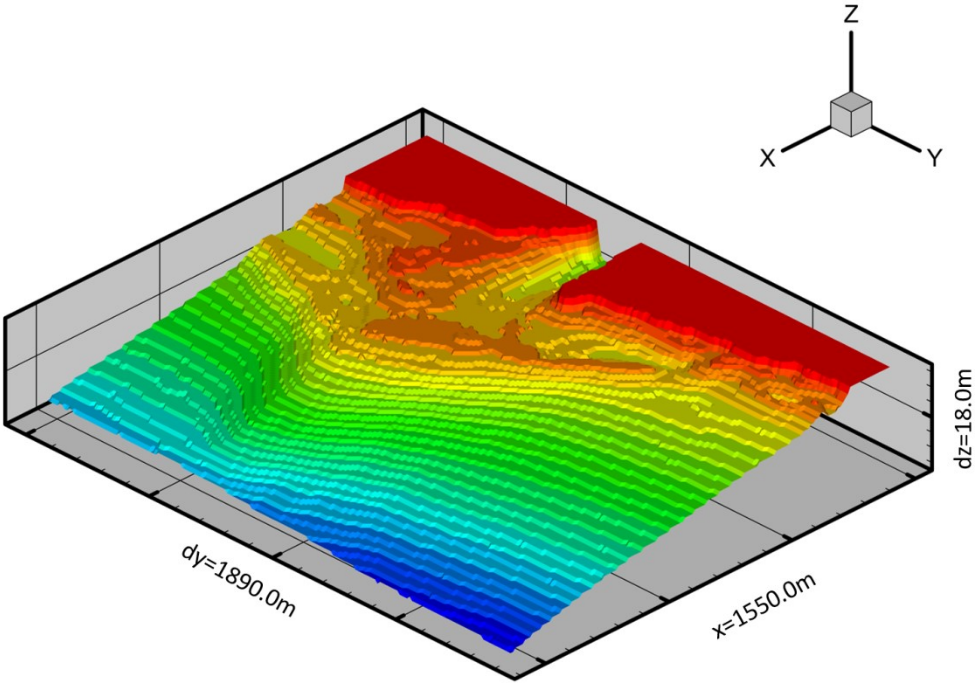


Figure 23. DEM model of the geometric data for the bathymetry of the estuary.

The computational time was 16 hours, 6 hours of which are devoted only for the calculation of the flow field. The rest of hours, total 10 hours were used to simulate the sediment transport and density currents. So that the times used for the results are counted since the subroutines of sediment and density current are active in the simulations. The flocculation model and its parameter were the same.

4.4.2 Numerical results

The reason choosing the SOLA method instead of the SWA is a three-dimensional behavior of the estuary flow. As applying the SWA for the NSE, the vertical velocity is roughly calculated and would be different from the velocity that could be calculated with the SOLA method. This difference might have an influence on the transporting mechanism in a spatial distribution of sediments. We run four cases with the same flow condition in order to study the influence of the settling velocity and sediment sizes. For simplicity, the axes in figures are named as shoreline direction for the y-direction, cross-shore direction in the x-direction and elevation for the z-direction. The flow field and salt concentration at the upper surface of the domain is given in **Figure 24** showing the absolute velocity and the intrusion of a salt wedge into the river mouth, respectively.

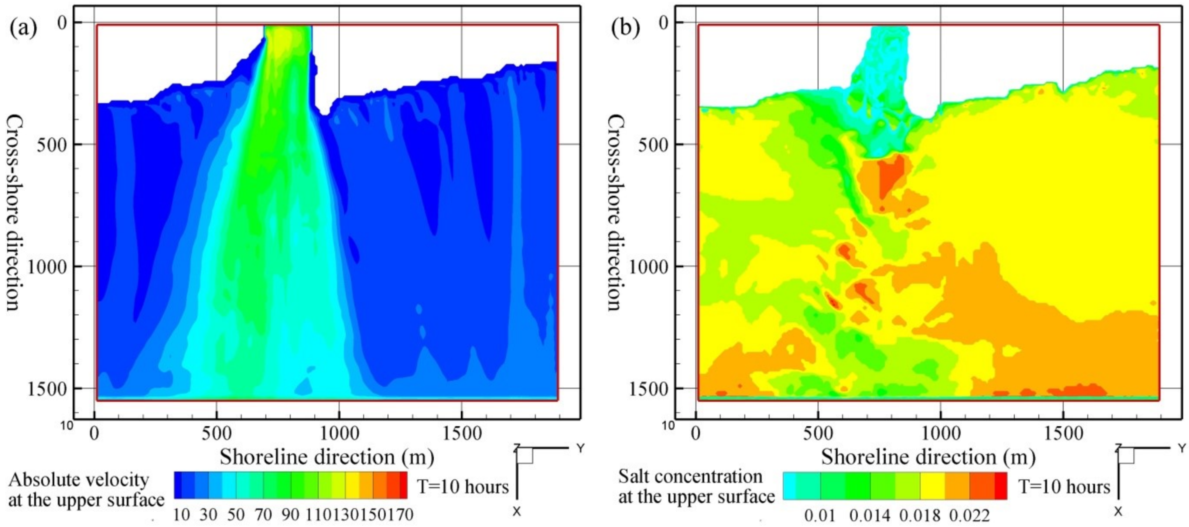


Figure 24. Plan views of the velocity magnitude and density deviation of the upper surface of domain at 10 hours after the simulation started in the estuary.

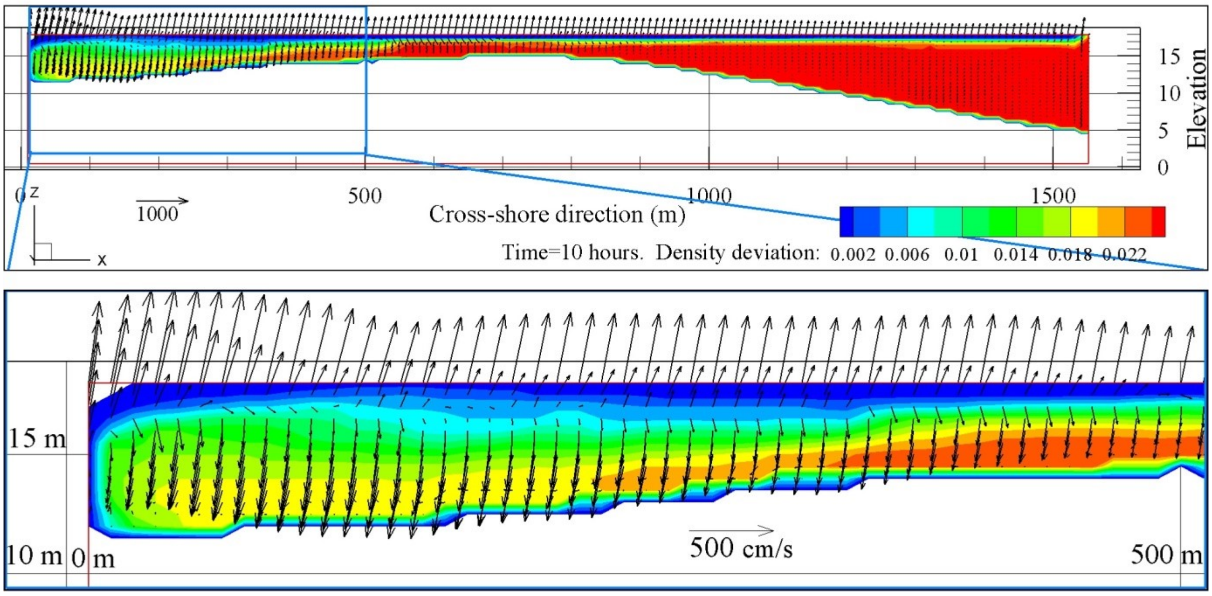


Figure 25. Slice views of the velocity vector and a salt wedge through the river channel at 10 hours after the simulation started in the estuary.

The given discharge, which is about 300 m^3 s^{-1} and the 10 percent of the design flood discharge, was moderate for the estuary. As that, the density current pushed river water up to the river mouth as the salt wedge, shown in **Figure 24** (b) and **Figure 25**. Similarly, the fewest amount of river water flows to the offshore direction as the exchange of the salt intrusion, as seen in **Figure 24** (a) and **Figure 25**. With the discharge more than 15 percent of the design flood discharge, the river flow could push the density current forward over the river terrace and other figures of the sediment transport resulted are discussed in (Takeshi, et al., 2016b).

The conventional solution for the sediment transport, the ADE, gives the general characteristic of the suspended sediment transport without the flocculation. Three-dimensional distributions of the suspended sediments and salt concentrations for four cases at 10 hours are given in **Figure 26**.

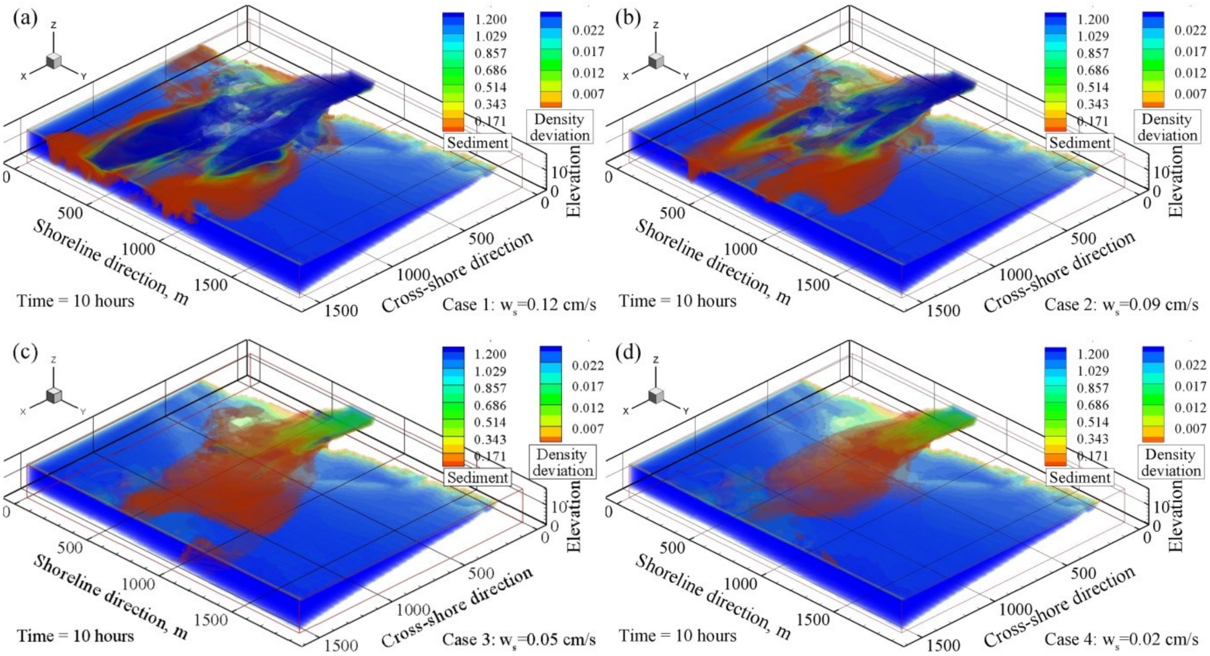


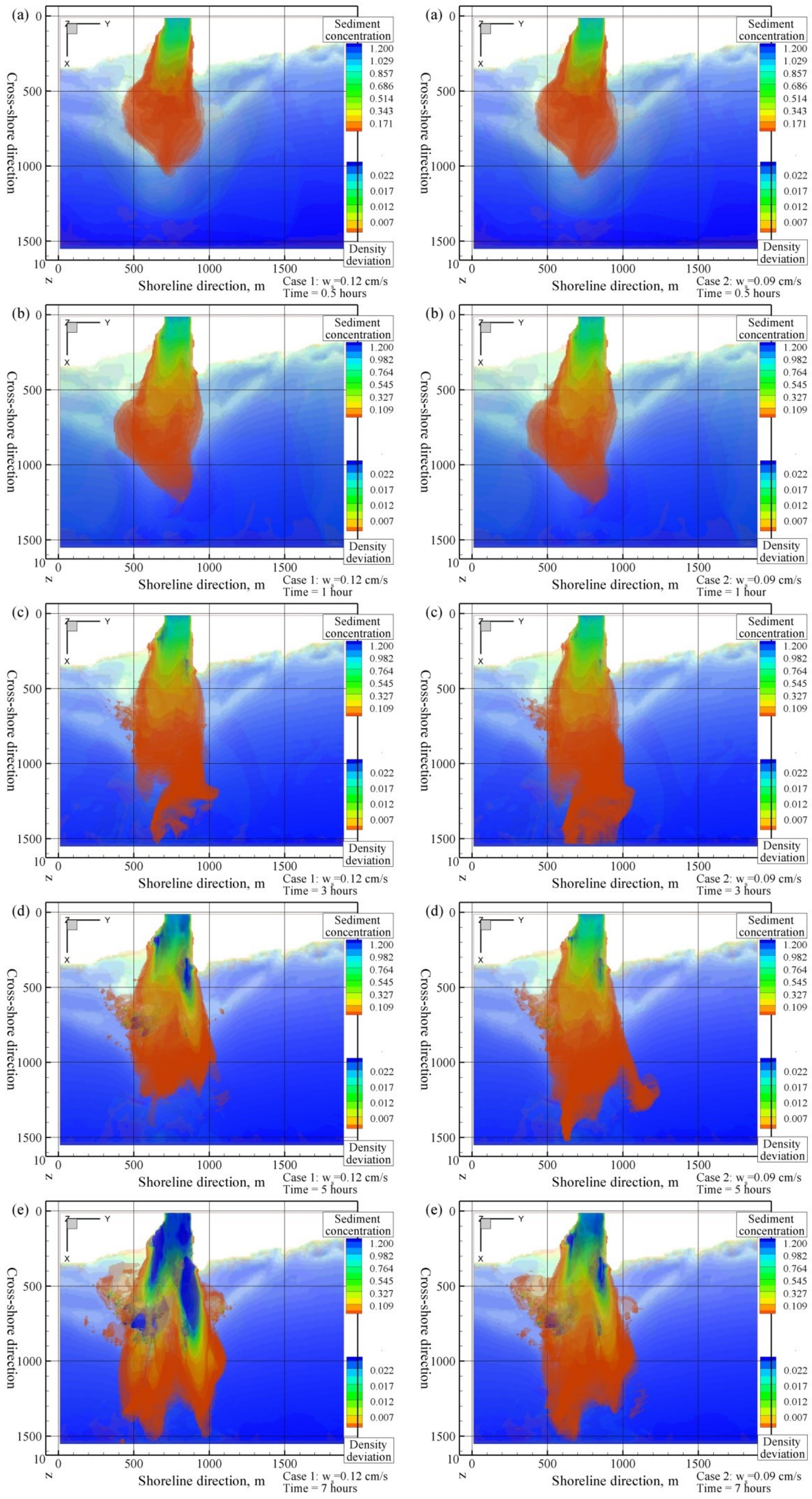
Figure 26. Suspended sediments of four cases changing the settling velocity transport to the offshore direction against the density current.

Sediments distributed in large area found in case 1 and in small area found in case 4. In other words, the results show that the settling velocities are key factors in the distribution of suspended sediments for each case when the transport happens in the same flow field. It is explained by the adjustments of the sediment advection by the settling velocity. It is not meant to be the sediment particles having higher value of settling velocity, which, in turn, gives the bigger particle size, can travel such a long distance.

In time development, the density current pushed the river flow back to the river terrace for an initial 30 minutes when the moderate discharge is given. Then the salt concentration in the estuary becomes steady state after that, as shown in **Figure 27** and **Figure 28**, and only the small interchanges took place. Whereas the time advances of the sediment transports were different for cases. At 0.5 hours after the transport considered, the distribution of the sediments was little different and case 4 shows the leading transport to the offshore direction. Like the salt concentration evolution, the sediment transports for an initial 30 minutes were active and the sediment front reaches 1000 m immediately, as seen in (a) of **Figure 27** and **Figure 28**. It is because that the sediment had transported with the inertial moments of the river flow. Soon after mixing sediments with the seawater, the advection rates of the sediment transport are reduced and slow but offshore directed transport took place for further hours, as seen in (b-e) of **Figure 27** and **Figure 28**. After 3 hours of transporting, the leading front of the sediment had been dispersed by the local flow field in case 1 and 2. Some of dispersion to the shoreline direction appeared on the cheek of the sediment mass, shown in **Figure 27** (c) and (d). As time advances, the sediment transport of case 1 lasted with the highest concentration at the upper part and with two bifurcations near the outlet part of the domain, as seen in **Figure 27** (e). Dispersed parts from the main body of the sediment mass in case 1 and 2 looked transporting to the shoreline in **Figure 27** (d) and (e). If the surface wave is present in the simulation, the sediment concentration at the upper part of the domain would definitely be transported into the shoreline.

In case 3 and 4, more interesting figures come up. Thanks to the less settling velocity, most of the sediment particles transported as suspended, so that the front of the sediment mass completely reached the outlet boundary after 3 hours, shown in **Figure 28** (c). The advection rate of the sediment reduces more slowly as the settling rate reduces. Like previous cases with higher settling velocity, the dispersion appeared at the leading front in **Figure 28** (d) and this dispersion transported to the shoreline more actively. Interestingly, the left cheek dispersion for all cases was constantly reduced as settling velocity reduces and completely disappeared in case 4. Additionally, the dispersion originated from the leading front was quantitatively increased to the right side of the sediment mass in case 4, as shown in **Figure 28** (e). Generally, the sediment concentrations higher than 0.5 in cases were stayed within the range of the 500 m from the river mouth, which is the inside of the river terrace.

Figure 27. Time sequences of the suspended sediment and salt concentration at the estuary. The first column is results of case 1 and the second is of case 2 (see next page).


(a)
Cross-shore direction
0
500
1000
1500
10
Z
X
Y
Sediment concentration
1.200
1.029
0.857
0.686
0.514
0.343
0.171
0.022
0.017
0.012
0.007
Density deviation
0
500
1000
1500
Shoreline direction, m
Case 1: w_s=0.12 cm/s
Time = 0.5 hours
(a)
Case 2: w_s=0.09 cm/s
Time = 0.5 hours
(b)
1.200
0.982
0.764
0.545
0.327
0.109
Case 1: w_s=0.12 cm/s
Time = 1 hour
(b)
Case 2: w_s=0.09 cm/s
Time = 1 hour
(c)
Case 1: w_s=0.12 cm/s
Time = 3 hours
(c)
Case 2: w_s=0.09 cm/s
Time = 3 hours
(d)
Case 1: w_s=0.12 cm/s
Time = 5 hours
(d)
Case 2: w_s=0.09 cm/s
Time = 5 hours
(e)
Case 1: w_s=0.12 cm/s
Time = 7 hours
(e)
Case 2: w_s=0.09 cm/s
Time = 7 hours

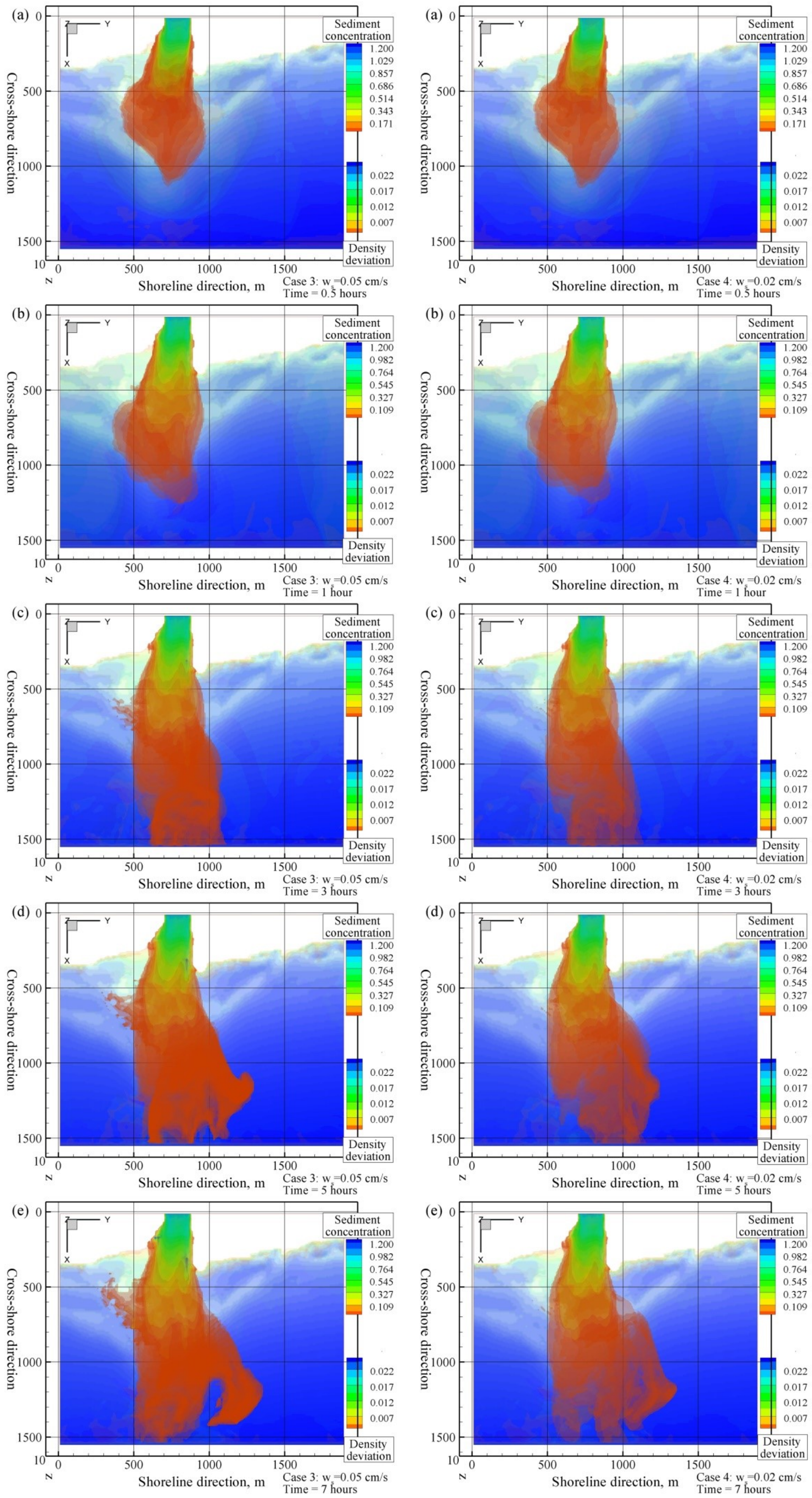

(a)
(b)
(c)
(d)
(e)
Cross-shore direction
0
500
1000
1500
Shoreline direction, m
0
500
1000
1500
Z
Y
X
10
Sediment concentration
1.200
1.029
0.857
0.686
0.514
0.343
0.171
0.982
0.764
0.545
0.327
0.109
0.022
0.017
0.012
0.007
Density deviation
Case 3: ws=0.05 cm/s
Case 4: ws=0.02 cm/s
Time = 0.5 hours
Time = 1 hour
Time = 3 hours
Time = 5 hours
Time = 7 hours

Figure 28. Time sequences of the suspended sediment and salt concentration at the estuary. The first column is results of case 3 and the second is of case 4 (see previous page).

Another figure that cannot be disclosed by the ADE of the sediment transport could be explained by the PTM. For the sediment transport modeling with the PTM, a total of 16,640 representative particles was tracked for all cases. The particle number for the simulation was limited by the memory of the running computer. In **Figure 29**, the three-dimensional distribution of sediment particles at 10 hours after the sediment transport considered is plotted for the four cases. We cannot see the large distribution area of the sediment particles having the higher settling velocity. It is not controversial for the results of the ADE, where the sediment distributed in the biggest area in case 1 which has a higher settling velocity or is representing the bigger particle size. If we discharge the sediment particles at every time step of the simulation, we could get the large area of sediment distribution with case 1 in the PTM. The bigger particles have a tendency to have a higher settling rate than the coarse particles. It makes sense that the bigger particles settled down immediately and hugely in the river mouth. Only the small number of coarse particles is transported in the offshore direction passing the river terrace in case 1 in **Figure 29** (a). The large area of distribution is seen in case 4 in **Figure 29** (d). The distribution of sediment particles is increased in case 1 to case 4, while the settling rate is increased, vice versa.

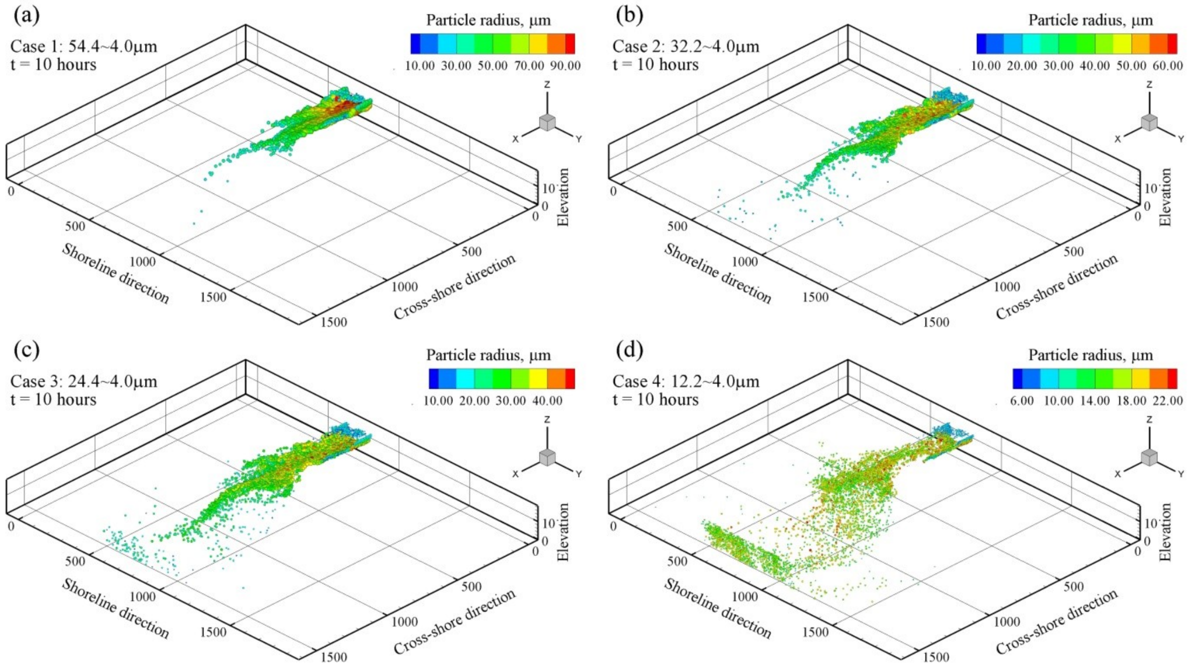


Figure 29. Spatial distribution of the sediment particles released from the river mouth at 10 hours for the four cases: (a) case 1, (b) case 2, (c) case 3 and (d) case 4.

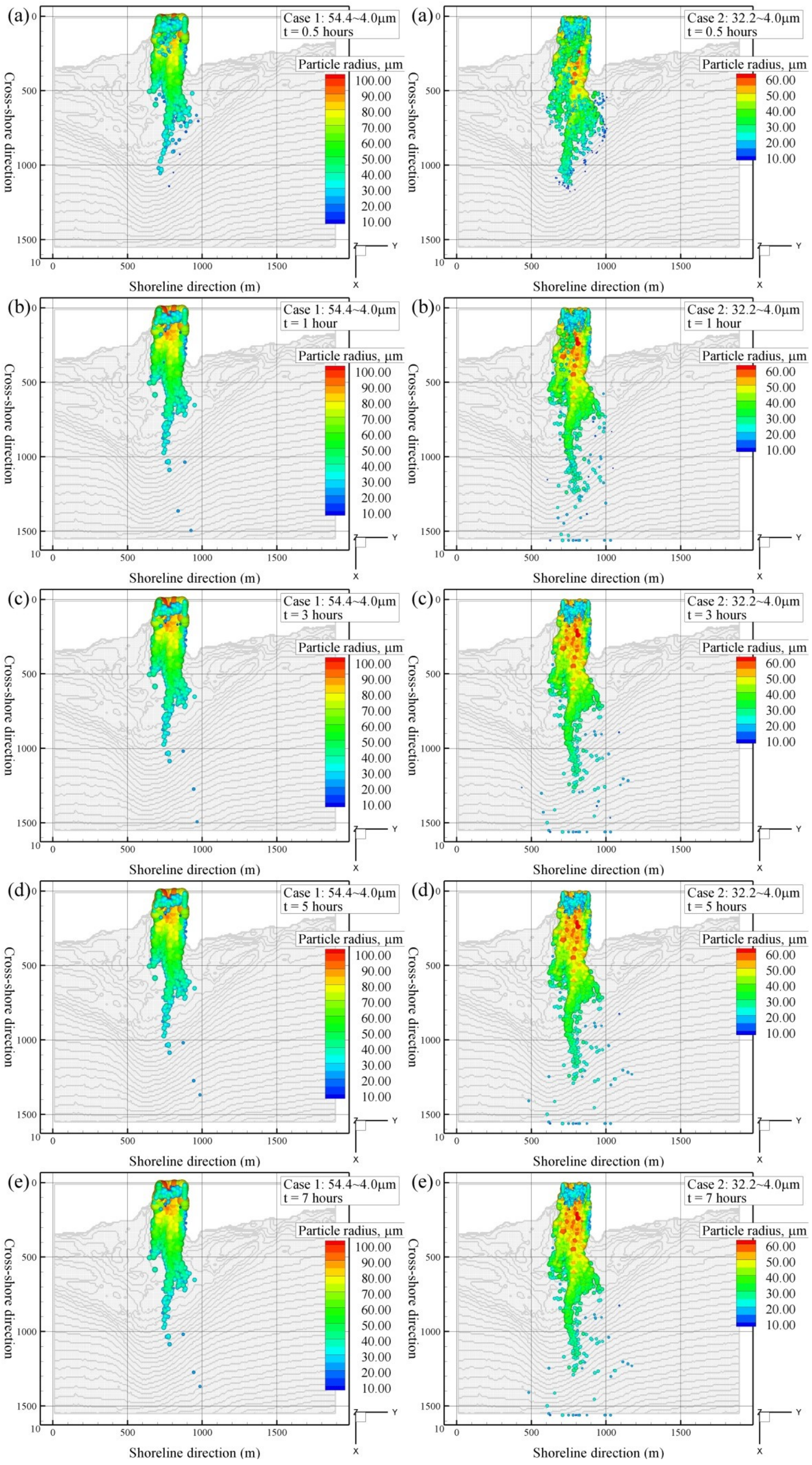
The large particles tend to settle at the estuary terrace, while the fine sediments are transported to the offshore directions in all cases and it can be seen in the time sequences of results given in **Figure 30** and **Figure 31**, where the color of the particle indicates the sizes of particles. The general tendency of the particle transport was the same as the results of ADE. For the first 30 min of the simulation, the sediment particles immediately reached at 1000 m with the inertia of the river flow and large particles settled around. Case 4 shows the most active transport to the offshore direction and it kept for all times, as seen in **Figure 31** (a-e). In **Figure 30**, almost all particles in case 1 are settled down within 1 hour and almost no particle motion observed after 3 hours except the couple of particles. These particles are the representative of the possible coarse particles transported in the offshore direction. In all cases, where a salt wedge invaded the river mouth and passed the river terrace, the sediment particles make flocs very quickly. This results the traveling distance short for the particles bigger than 30 μm. Like the seawater is important for the flocculation, the discharge is very dominant factor in the sediment transport. In that sense, the two types of the discharge with different cases of sediment particles were discussed in (Takeshi, et al., 2016b) and (Takeshi, 2016). In later cases, the front of the sediment particles reached the outflow boundary of the domain at 1 hour, which was 2 hours prior to the ADE results. This is one of the advances of the PTM over the ADE. The particle motion of independent from the others is likely in the PTM, but the concentration separated from the concentration mass might be recognized as error or instability. Unlike the ADE, the dispersion at the cheek of the sediment mass was not observed in case 1-3 in the PTM. However the sediment particles transporting to the shoreline were observed for all cases and the particle number was increased as the particle size decreased. In other words, the large amount of sediment particles, particularly in case 4, from the outlet part of the domain was transported to the shoreline at the left side of the sediment mass, as shown in **Figure 31** (e). Generally, the fine sediment particles lead the sediment mass in the transport. In the results of the PTM for sediment transport at the estuary of the Ohkouzu diversion channel, the sediment particles having sizes of less than 20 μm have a potential to be transported in the offshore direction over 1500 m from the river mouth. From the results, the right side of the shoreline is likely to receive more sediment particles than the left side. The horizontal distribution, more precisely the distribution in the shoreline direction (the y-direction), seems to be hugely dependent on the ocean surface dynamics as well as the interaction of the river flow and density current.

4.4.3 Advantages and disadventages of the new PTM

Based on the numerical experiments, I would like to stress some advantages and dis-advantages of the newly purposed PTM in a comparison with the ADE. The settling velocity is no longer an issue for the PTM because it depends on a sediment particle size and weight. Initialization in the PTM becomes more realistic than that of the ADE, which leads reliable results. The PTM can present a particulate nature of sediment transport and provides an error-free analysis of the sediment distribution. In general, the PTM is applicable for the cohesive and non-cohesive sediment transports. More importantly, the PTM brings simplicity for handling the individual characteristic of sediment particles like the flocculation and de-flocculation processes.

The main disadvantage is the computational cost, including memory and time consumptions. With respect to this, the increase of problem scale decreases the performance of the PTM. An inherent bottleneck of the PTM is the poor connectivity to the concentration based evaluation.

Figure 30. Distribution of the sediment particles released from the river mouth at once in case 1 and 2 (see next page).



(a)
Case 1: 54.4~4.0μm
t = 0.5 hours
(a)
Case 2: 32.2~4.0μm
t = 0.5 hours
(b)
Case 1: 54.4~4.0μm
t = 1 hour
(b)
Case 2: 32.2~4.0μm
t = 1 hour
(c)
Case 1: 54.4~4.0μm
t = 3 hours
(c)
Case 2: 32.2~4.0μm
t = 3 hours
(d)
Case 1: 54.4~4.0μm
t = 5 hours
(d)
Case 2: 32.2~4.0μm
t = 5 hours
(e)
Case 1: 54.4~4.0μm
t = 7 hours
(e)
Case 2: 32.2~4.0μm
t = 7 hours
Particle radius, μm
100.00
90.00
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
Particle radius, μm
60.00
50.00
40.00
30.00
20.00
10.00
Cross-shore direction
0
500
1000
1500
10
Shoreline direction (m)
0
500
1000
1500
Z
Y
X

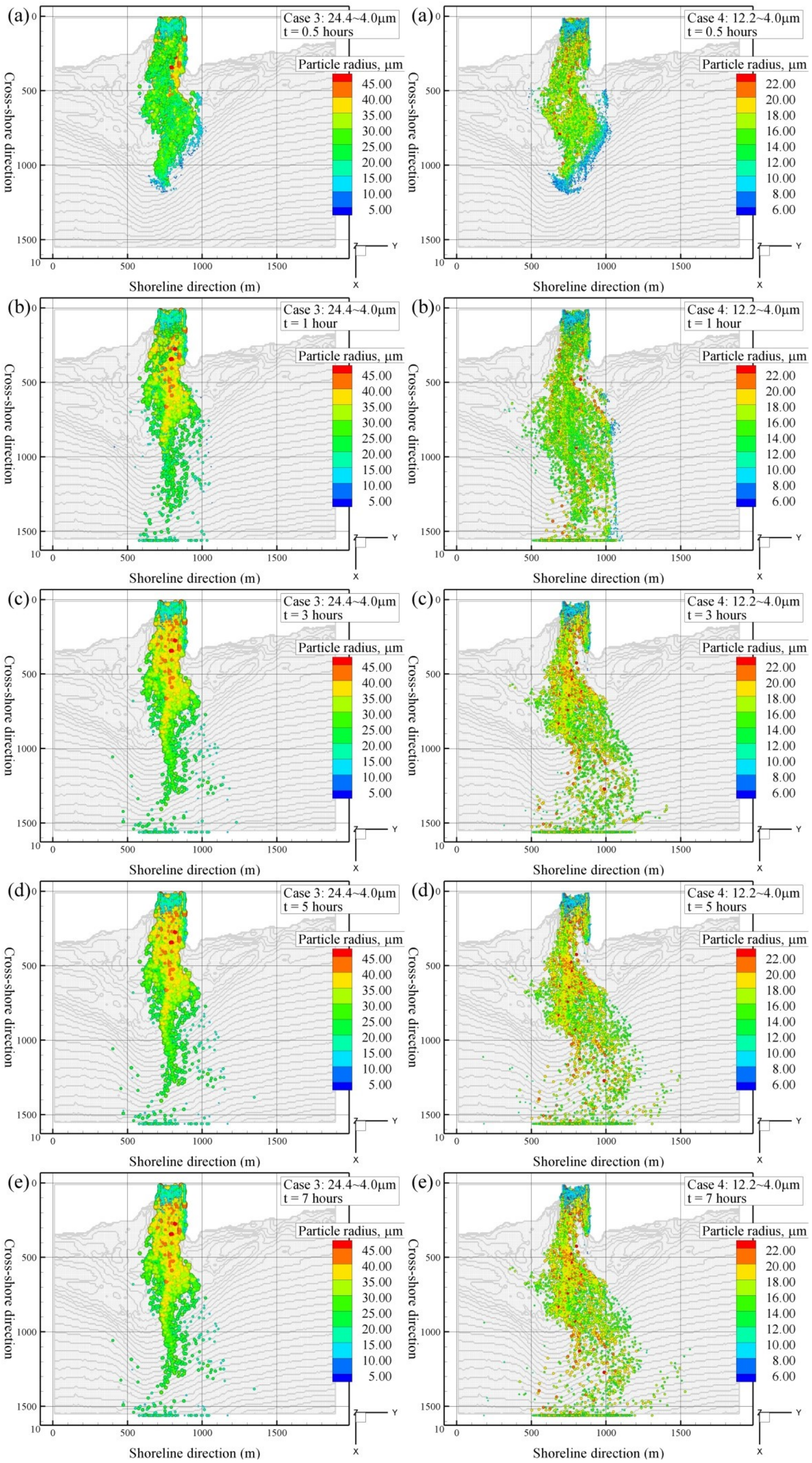

(a)
Case 3: 24.4~4.0μm
t = 0.5 hours
Case 4: 12.2~4.0μm
t = 0.5 hours
(b)
Case 3: 24.4~4.0μm
t = 1 hour
Case 4: 12.2~4.0μm
t = 1 hour
(c)
Case 3: 24.4~4.0μm
t = 3 hours
Case 4: 12.2~4.0μm
t = 3 hours
(d)
Case 3: 24.4~4.0μm
t = 5 hours
Case 4: 12.2~4.0μm
t = 5 hours
(e)
Case 3: 24.4~4.0μm
t = 7 hours
Case 4: 12.2~4.0μm
t = 7 hours
Particle radius, μm
45.00
40.00
35.00
30.00
25.00
20.00
15.00
10.00
5.00
22.00
20.00
18.00
16.00
14.00
12.00
10.00
8.00
6.00
Cross-shore direction
0
500
1000
1500
Shoreline direction (m)
0
500
1000
1500
Z
Y
X

Figure 31. Distribution of the sediment particles released from the river mouth at once in case 3 and 4 (see previous page).

SUMMARY FOR PART 1

In this part, we discussed the macroscopic modeling of the sediment transport problem in the estuary. The descriptions of models and methods were focused on the problem to be solved and the problem, sediment transports at the estuary of the Ohkouzu diversion channel, was studied by the 2D and 3D numerical models.

The numerical models are organized as the numerical framework. The fluid flow can be optionally solved by two different numerical techniques featured by the solution algorithm (SOLA), namely the pressure-velocity iteration and shallow water approximation, depending on requirements and problem condition. The two techniques implement the Navier-Stokes equation with the Boussinesq approximation. In addition, the Smagorinsky turbulent model was introduced to the fluid flow in a form for the stratified medium. The sediment transports were solved by another two models, but simultaneously. The first was the traditional solution: the advection-diffusion equation. The second was the sediment particle tracking method which is newly introduced in the external articles. The particle tracking method brought two important features that could be difficult or impossible in some aspect to be involved with the ADE. These are the local transport behavior and the flocculation process of the sediment particles. The solutions came with the simple flocculation model verified by the simple experiments of settling sediment particles in the fresh and seawater.

The numerical models for the sediment transport were validated with each other in the laboratory scale lock-exchange problem. Then the numerical framework was applied to the problem in the estuary of the Ohkouzu diversion channel. Four cases, changing the velocity and particle sizes, which originated from the experiment, were considered in the both 2D and 3D studies. The difference between 2D and 3D studies were not only the computational parameters, but also the solving technique for the flow governing equations. However, the two different dimensional studies provide the qualitative results which confirmed the observations made in the estuary of the Ohkouzu diversion channel. The comprehensive description of the sediment transport at the estuary was found in the 2D and 3D studies. The numerical framework with the results of the application to the estuary of the Ohkouzu diversion channel shows its potential to be applied for the sediment transport modeling in a large river estuary.

PART 2: MESOSCOPIC MODELING FOR ICE PROBLEMS

contains the description of lattice Boltzmann (LB) models, their implementation techniques and applications to problems.

5 LATTICE BOLTZMANN MODELS

The family of LB models is getting large. In this chapter, we will explore an origin of the LB models and will present developed models for fluid flows, scalar transports and other substantial problems in hydraulics.

5.1 Basic theory of the lattice Boltzmann method

In this section, we will briefly discuss the basic theory of the LBM and its originality.

5.1.1 Origin of the lattice Boltzmann model

The history of the LBM starts from the method in gas kinetics, so called lattice gas automata (LGA). Very first LGA was the HPP[2] model proposed by Hardy, Pomeau and Pazzis in 1973 (Hardy, et al., 1973). The HPP model describes a fluid flow using simple rules for motion of particles residing on the lattice nodes. Those simple rules conserve mass, moment and energy, and handle the steaming and collision of the particles on the lattice nodes. The main variable of the HPP model is the Boolean number, n_i, defining particle existence, $n_i = 1$, or absence, $n_i = 0$, on the lattice node. The motion of the particle containing the streaming and collision can be described by the following discrete kinetic equation (Guo & Shu, 2013),

$$n_i(x + c_i\delta t, t + \delta t) = n_i(x, t) + C_i\big(n(x, t)\big) \tag{105}$$

and the Boolean variable is used to define macroscopic variables,

$$\rho = \sum_i mn_i\,, \qquad \rho u = \sum_i mc_i n_i\,, \qquad \rho e = \sum_i \frac{m}{2}(c_i - u_i)^2 n_i\,, \tag{106}$$

where i is the lattice direction (see **Figure 32** (a)), c_i is the discrete velocity for each lattice direction and C_i is the collision term. The discrete velocity set in the HPP model is given by $c_i = s_l c_i$ with $c_1 = (1, 0)$, $c_2 = (0, 1)$, $c_3 = (-1, 0)$, and $c_4 = (0, -1)$, and $s_l = \delta x/\delta t$ is the lattice speed where δx is the lattice spacing and δt is the lattice time step, as seen in **Figure 32** (a). However, the HPP model satisfies the basic physical laws; consequently derived macroscopic variables do not manifest hydrodynamic continuum properties of the fluid flow. Reasons are

[2] HPP is the initials of the authors' names.

that the HPP model was devoted for microfluidic dynamics rather than macroscopic fluid flows and the square lattice used for a discretization has insufficient symmetric properties etc. Nevertheless, the HPP model had introduced very basic components of the modern LBM in its diligent period.

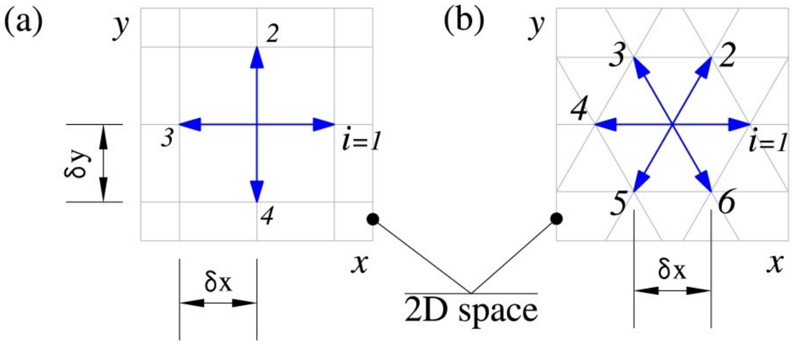


Figure 32. Two dimensional lattices for the Lattice Gas Automata: (a) Square lattice for HPP model, (b) Equilateral triangular lattice for FHP model

After over a decade, in 1986, Frisch group (Frisch, et al., 1986) and Wolfram (Wolfram, 1986) simultaneously revealed that the lattices used in the LGA must have symmetric properties. The model proposed by Frisch group has named the FHP model after the authors. The FHP model uses the equilateral triangular lattice (see **Figure 32** (b)) for the simulation and altered to have more collision rules than the previous HPP model. A big contribution of the FHP model is not limited by the symmetry properties of the lattice and the model also brings an equilibrium state for the collision term in Eq.(105). This revolution contribution was enabled the FHP model can recover the incompressible NSE in low Mach number. However, still in some case, the Galilean invariance was violated in the FHP model. Thanks to the equilibrium distribution in the collision term, several lattice terms and properties, such as the lattice speed of sound and the lattice EOS, are introduced to activate further development of the LGA. Still in the FHP model, the main variable is the Boolean number. Although the Boolean variable brings unconditional stability for the simulation, it leads whether a particle crowd in some place and a particle devastation in other place. This crowd and devastation are not described by the kinetic equation implemented on the lattice. The physically uneven distribution of the Boolean variable results a discordant macroscopic flow field and it was termed as a statistic noise. Further LGA models were successfully explored and applied to multiphase and porous media flows. However, these models were still suffering by the statistic noise caused by the Boolean variables, the violation of Galilean invariance and the velocity dependence on pressure. Those are said the key driver for the LGA to be altered into the LBM.

Historically, beginning of the LBM was initiated by Frisch group in 1987 improving the existing LGA based on the evaluation of the hydrodynamic properties and introducing the lattice Boltzmann equation (LBE) into the LGA (Frisch, et al., 1987). But they did not proclaim that their improved scheme was the new computational technique out of the LGA. Soon after, the LBE was formulated to solve fluid flow with the elimination of the statistic noise of the LGA as a new numerical method by McNamara and Zanetti in 1988 (McNamara & Gianluigi, 1988). The model uses the evolution of the mean values of one-particle distribution functions to eliminate a basic affliction of the LGA, the noise, instead of the Boolean variables on the lattice. The model named NZ model after the authors and Eq.(105) can be rewritten as the basic equation of the NZ model:

$$f_i(x + c_i\delta t, t + \delta t) = f_i(x, t) + \Omega_i\big(f(x, t)\big) \tag{107}$$

where Ω_i is the microscopic collision operator that was still the Boolean algebra expression that corresponds to the chosen set of collision rules (McNamara & Gianluigi, 1988) of the LGA. Further research works were mainly focusing on the simplification of the collision operator. However, the collision operator was still remained to be dependent from the LGA knowledge. Later, around in 1991, the collision operator was simplified as a linear term (Chen, et al., 1991), (Koelman, 1991) including the collision frequency and the equilibrium distribution function f_i^{eq}, which was the same as the Bhatnagar-Gross-Krook collision operator (Bhatnagar, et al., 1954) in the gas kinetic theory constructed back in 1953. Consequently, a simple LBE can correctly describe the hydrodynamics of a macroscopic flow and can derive the correct version of the incompressible NSE in isothermal condition at low Mach number. The simple collision operator is referred among the literature whether the Bhatnagar-Gross-Krook (BGK) operator or the single relaxation time operator (SRT). The equilibrium distribution function in the BGK operator is now fully independent from the LGA and made the LBE path straighten to the LBM. In isothermal flow, the equilibrium distribution function must satisfy the mass and moment conservation,

$$\rho = \sum_i f_i = \sum_i f_i^{eq} \text{ and } \rho\mathbf{u} = \sum_i \mathbf{c}_i f_i = \sum_i \mathbf{c}_i f_i^{eq}, \tag{108}$$

and the BGK collision operator is inserted into Eq.(105) as

$$f_i(\mathbf{x} + \mathbf{c}_i\delta t, t + \delta t) = f_i(\mathbf{x}, t) - \frac{f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)}{\tau_v}, \tag{109}$$

where τ_v is the relaxation time. Principally, the variations of approximation for the collision operator leads many versions of the LBM. In this thesis, we will mainly focus on the BGK collision operator exploiting its simplicity and capability.

5.1.2 The Boltzmann equation

We shall derive the Boltzmann transport equation in two ways to simply revisit underlying microscopic models and assumptions. Then we discuss about collision terms in the Boltzmann equation to lay the cornerstones of the LBM considered here.

The Boltzmann equation from the BBGKY hierarchy

Normally, let us consider an infinite volume V contains an N number of molecules. Please note that the molecule and particle are the interchangeable terms in this thesis. The function defining their state can be f^N and let it name the N particle distribution function. Thus, the distribution function f^N is the function in a phase space depending on a position ($\mathbf{x}^\alpha$) and velocity coordinates ($\dot{\mathbf{x}}^\alpha$) of a generic molecule α in volume V with α = 1, 2, …., N. A state of the N molecules can be defined as

$$f^N(\mathbf{x}^\alpha, \dot{\mathbf{x}}^\alpha)d\mathbf{x}^\alpha d\dot{\mathbf{x}}^\alpha\ ,$$

where $\alpha = 1, 2, \ldots, N$.

Above expression gives the probability to find N molecules in the position vector between $\mathbf{x}^\alpha$ and $\mathbf{x}^\alpha + d\mathbf{x}^\alpha$ and velocity vector between $\dot{\mathbf{x}}^\alpha$ and $\dot{\mathbf{x}}^\alpha + d\dot{\mathbf{x}}^\alpha$ at time t. The normalization of the distribution function over all time, t, gives the property

$$\int f^N\,(\mathbf{x}^\alpha, \dot{\mathbf{x}}^\alpha, t)d\mathbf{x}^\alpha d\dot{\mathbf{x}}^\alpha = 1\,. \tag{110}$$

The time evolution of the N particle distribution function follows the Liouville's equation (Guo & Shu, 2013) (Michael & Daneal T, 2006), i.e.,

$$\frac{\partial f^N}{\partial t} + \sum_{\alpha=1}^{N}\left(\dot{x}_i^\alpha \frac{\partial f^N}{\partial x_i^\alpha} + \ddot{x}_i^\alpha \frac{\partial f^N}{\partial \dot{x}_i^\alpha}\right) = 0 \tag{111}$$

Solving Eq.(111) is much more difficult than solving the equation of motion for N particles. Only a way to simplify Eq.(111) is to derive a time evolution equation for a lower particle distribution function. To reduce the complexity of the distribution function, we can derive another lower particle distribution function f^n from the distribution function f^N such that

$$f^n(\mathbf{x}^n, \dot{\mathbf{x}}^n, t)d\mathbf{x}^n, d\dot{\mathbf{x}}^n$$

$$= \left[\int f^N(\mathbf{x}^\alpha, \dot{\mathbf{x}}^\alpha, t)d\mathbf{x}^{n+1}, \ldots, d\mathbf{x}^N d\dot{\mathbf{x}}^{n+1}, \ldots, d\dot{\mathbf{x}}^N\right] d\mathbf{x}^n, \ldots, d\mathbf{x}^N d\dot{\mathbf{x}}^n, \ldots, d\dot{\mathbf{x}}^N \tag{112}$$

gives the probability to find n molecules in a certain interval of a position and velocity vector at time t. To derive the time evolution for the distribution function f^n (n<N), we can multiply Eq.(111) by $d\mathbf{x}^{n+1}, \ldots, d\mathbf{x}^N d\dot{\mathbf{x}}^{n+1}, \ldots, d\dot{\mathbf{x}}^N$ and it yields,

$$\int\left[\frac{\partial f^N}{\partial t} + \sum_{\alpha=1}^{N}\left(\dot{x}_i^\alpha \frac{\partial f^N}{\partial x_i^\alpha} + \ddot{x}_i^\alpha \frac{\partial f^N}{\partial \dot{x}_i^\alpha}\right)\right] d\mathbf{x}^{n+1}, \ldots, d\mathbf{x}^N d\dot{\mathbf{x}}^{n+1}, \ldots, d\dot{\mathbf{x}}^N = 0\,. \tag{113}$$

The first integral of Eq.(113) can be changed to the order of the time derivative of f^n according to Eq.(112):

$$\int \frac{\partial f^N}{\partial t} d\mathbf{x}^{n+1}, \ldots, d\mathbf{x}^N d\dot{\mathbf{x}}^{n+1}, \ldots, d\dot{\mathbf{x}}^N = \frac{\partial f^n}{\partial t}\,. \tag{114}$$

Moreover, the second and third terms in left hand side of Eq.(113) alters with some conditions and the subsequent form of Eq.(113) becomes

$$\frac{\partial f^n}{\partial t} + \sum_{\alpha=1}^{n}\left[\dot{x}_i^\alpha \frac{\partial f^n}{\partial x_i^\alpha} + \left(F_i^\alpha + \sum_{\beta}^{n} X_i^{\alpha\beta}\right)\frac{\partial f^n}{\partial \dot{x}_i^\alpha}\right]$$

$$= -\sum_{\alpha=1}^{n}(N-n)\frac{\partial}{\partial \dot{x}_i^\alpha}\int X_i^{\alpha,n+1} f^{n+1} dx^{n+1} d\dot{x}^{n+1}, \tag{115}$$

where F_i^α is the external force per unit of mass which acts on the molecule α being independent of the molecular velocities, $X_i^{\alpha\beta}$ is the force per unit of mass which acts on the molecule α due to its interaction with the molecule β and the forces can be composed as the acceleration of the α molecule as $\ddot{x}_i^{\alpha} = F_i^{\alpha} + \sum_{\beta=1}^{n} X_i^{\alpha\beta}$.

As we see, Eq.(115) is the time evolution equation for the distribution function f^n dependent on the distribution function f^{n+1}. Hence, Eq.(115) represents a hierarchy of coupled equations, which is known as BBGKY hierarchy, since it was deduced independently by the researchers Bogoliubov; Born and Green; Kirkwood and Yvon (Stewart, 2004). Still, we have complexity to solve Eq.(115) if *n* is enough big as *N*. The BBGKY hierarchy is equivalent to the original Liouville equation and lower order equation of the hierarchy can nearly represent its upper order equation of the hierarchy. With some assumptions, we can consider the first order equation from the BBGKY hierarchy and it reads

$$\frac{\partial f^1}{\partial t} + \dot{x}_i^1 \frac{\partial f^1}{\partial x_i^1} + F_i^1 \frac{\partial f^1}{\partial \dot{x}_i^1} = -(N-1)\frac{\partial}{\partial \dot{x}_i^1} \int X_i^{12} f^2 d\mathbf{x}^2 d\dot{\mathbf{x}}^2 , \qquad (116)$$

where f^2 is the two particle distribution function. As there are several ways to derive the Boltzmann equation from Eq.(116) in the literature, we shall follow Krikwood's method (Kirkwood, 1947) and shall consider following four hypotheses (Gilberto, 2010) to derive the Boltzmann equation.

Hypothesis 1: Only the interactions between two molecules are considered as a binary collision for a rarefied gas.

Hypothesis 2: The effect of the external force is small in comparison with the effect of the interaction force between two molecules.

Hypothesis 3: Initially, at *t*=0, molecules with uncorrelated velocities are unevenly distributed in space and are far from each other. In other words, a chaos assumption is valid for the state of molecules.

Hypothesis 4: The spatial gradient of the distribution function in comparison with the molecular size is small, so that the distribution function is assumed to be continuous.

The main procedure to derive the Boltzmann equation is to eliminate f^2 from Eq.(116) by expressing f^2 in function of f^1 with the hypotheses 1 to 4. Consequently, we can write the time evolution equation for the one particle distribution function independent from the two particle distribution functions as

$$\frac{\partial \overline{f^1}}{\partial t} + \dot{x}_i^1 \frac{\partial \overline{f^1}}{\partial x_i^1} + F_i^1 \frac{\partial \overline{f^1}}{\partial \ddot{x}_i^1} = \Omega\left(\overline{f^1}(\mathbf{x}, \dot{\mathbf{x}}, \tau^*, t)\right) , \qquad (117)$$

where the time average of the distribution function is taken as

$$\overline{f^1}(\mathbf{x}, \dot{\mathbf{x}}, t) = \frac{1}{\tau^*} \int_0^{\tau^*} f^1(x, x, t+s) ds \; . \qquad (118)$$

The time interval in Eq.(117) and (118) is chosen to be $\tau^c \ll \tau^* \ll \tau$, where τ^c is the mean collision time and τ is the mean free time for a mean free path. For simplicity, changing $\overline{f^1}$ notation with *f*, the Boltzmann transport equation can be written as

$$\frac{\partial f}{\partial t} + \dot{x}_i \frac{\partial f}{\partial x_i} + F_i \frac{\partial f}{\partial \dot{x}_i} = \Omega(f(\mathbf{x}, \dot{\mathbf{x}}, \tau^*, t)) \; . \qquad (119)$$

The Boltzmann equation from gas kinetics

Let us discuss another simple way to derive the Boltzmann transport equation. Before that, we will take some time to discuss about what is the meaning of the distribution function in the kinetic theory. We saw that the distribution function is a main variable in the gas kinetic theory with a mention of the Maxwell-Boltzmann distribution in Section 1.3.3 and the many (N or n) particle distribution function in the BBGKY hierarchy. Thus, the main aim of the kinetic theory is to define the distribution function in a given interaction of the molecules and to find the time evolution of the distribution function in the system with the purpose of the realization of thermodynamics. The distribution function gives the probability to find a molecule in a position between **x** and **x**+d**x** and a momentum between **p** and **p**+d**p** at time *t*. In other words, the distribution function gives the possible number of molecules in a given interval of the position and momenta at a time.

The distribution function is a continues function in phase space, μ, coordinated by a position (**x** = *x, y, z*) and a momentum (**p** = p_x, p_y, p_z). If the molecules in a system are indistinguishable, the velocity ($\dot{\mathbf{x}} = \mathbf{p}/m$) can be considered as a coordinate of the phase space. At any instant of time, the state of the entire system of N molecules can be presented by N points in μ space. The volume containing N points in μ space is $\mathrm{d}^3\mathbf{x}\mathrm{d}^3\mathbf{p}$ as an element and the total number of the points in infinitesimal volume is $f(\mathbf{x},\mathbf{p},t)\mathrm{d}^3\mathbf{x}\mathrm{d}^3\mathbf{p}$ by the definition. The μ space is obviously constructed by the number of elements and the density of point in each element does not vary rapidly from the element to a neighboring element, then the distribution function can be defined in entire μ space as

$$\sum f(\mathbf{x},\mathbf{p},t)\,\mathrm{d}^3\mathbf{x}\mathrm{d}^3\mathbf{p} \approx \int f(\mathbf{x},\mathbf{p},t)\mathrm{d}^3\mathbf{x}\mathrm{d}^3\mathbf{p}\,. \tag{120}$$

Let the volume element notation be $\mathrm{d}^3\mathbf{x}\mathrm{d}^3\mathbf{p} = \mathrm{d}\mu(\mathrm{t})$. The number of molecules in the volume element at time t is

$$n(t) = f(\mathbf{x},\mathbf{p},t)\mathrm{d}\mu(\mathrm{t})\,. \tag{121}$$

Assuming the hypothesis 3 for the system, the number of molecules in the volume element depends on time. Thus, after the instant of time Δt, the number of molecules in the volume element becomes

$$n(t+\Delta t) = f(\mathbf{x}+\Delta\mathbf{x},\mathbf{p}+\Delta\mathbf{p},t)\mathrm{d}\mu(\mathrm{t}+\Delta t)\,. \tag{122}$$

If the time instant Δt is small as the mean free time τ, the quantities $n(t)$ and $n(t+\Delta t)$ become equal to each other and the equality shows a free stream of the molecules. Generally, the time instant Δt is considered to be $\Delta t \gg \tau$ and a collision of the molecules needs to be considered in the number of molecules existing in the volume element. Further discussion should assume the hypothesis 1. An effective rate of a collision can be expressed by the two numbers of molecules at time t and $t+\Delta t$

$$\Delta n = n(t+\Delta t) - n(t)\,. \tag{123}$$

To show Eq.(105) from Eq.(123), we could change the notation of collision by C_i and rewrite Eq.(123)

$$n(t+\Delta t) = n(t) + C_i\,, \tag{124}$$

which is the basic kinetic equation used in the LGA. Substituting Eq.(121) and (122) to Eq.(123) gives

$$\Delta n = f(\mathbf{x}+\Delta\mathbf{x},\mathbf{p}+\Delta\mathbf{p},t)\mathrm{d}\mu(\mathrm{t}+\Delta t) - f(\mathbf{x},\mathbf{p},t)\mathrm{d}\mu(\mathrm{t})\,. \tag{125}$$

Note that evolution equation, Eq.(107), of the NZ model can also be seen from Eq.(125). If the change of the volume element is negligible, we can rewrite Eq.(125) as

$$\Delta n = [f(\mathbf{x}+\Delta\mathbf{x},\mathbf{p}+\Delta\mathbf{p},t) - f(\mathbf{x},\mathbf{p},t)]\mathrm{d}\mu(\mathrm{t})\,. \tag{126}$$

The changes of the position and momentum vectors of the molecules during the time interval Δt are defined by

$$\Delta\mathbf{x} = \dot{\mathbf{x}}\Delta t, \quad \Delta p = m\mathbf{F}\Delta t\,, \tag{127}$$

where **F** denotes a specific external force which acts on the molecules with the hypothesis 2. The expansion of $f(\mathbf{x}+\Delta\mathbf{x},\mathbf{p}+\Delta\mathbf{p},t)$ in around $(\mathbf{x},\mathbf{p},t)$ using the first order Taylor series (Баасандорж, et al., 1999) gives

$$f(\mathbf{x}+\Delta\mathbf{x},\mathbf{p}+\Delta\mathbf{p},t) = f(\mathbf{x},\mathbf{p},t) + \frac{\partial f}{\partial t}\Delta t + \frac{\partial f}{\partial x_i}\Delta\mathbf{x} + \frac{\partial f}{\partial p_i}\Delta\mathbf{p} + O[(\Delta t)^2] \tag{128}$$

Substituting Eq.(128) into Eq.(126) and dividing the both side by Δt yields

$$\frac{\Delta n}{\Delta t} = \left[\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x_i}\dot{x}_\iota + \frac{\partial f}{\partial p_i}mF_i + O[(\Delta t)^2]\right]\mathrm{d}\mu(\mathrm{t}) \tag{129}$$

and Eq.(129) can be rewritten in an organized way as,

$$\frac{\partial f}{\partial t} + \dot{x}_i\frac{\partial f}{\partial x_i} + F_i\frac{\partial f}{\partial \dot{x}_i} = \frac{\Delta n}{\Delta t\mathrm{d}\mu(\mathrm{t})} - O[(\Delta t)^2] \tag{130}$$

or

$$\frac{\partial f}{\partial t} + \dot{\mathbf{x}}\frac{\partial f}{\partial x_i} + \mathbf{F}\frac{\partial f}{\partial \dot{x}_i} = \Omega(f,f) \tag{131}$$

which is the Boltzmann equation, a nonlinear integro-differential equation for the one particle distribution function *f*. General form of the collision term in the Boltzmann equation is an integral equation (A.A.Mohamad, 2011).

1.1.3 The collision term

Now, we will briefly discuss about the collision term on the right side of Eq.(119) or (131). The collision term is only one complexity of solving the Boltzmann equation and contains all details about molecular interactions and the irreversibility of the kinetic description (Carlo, 1988). To approximate the integral form of the collision term, one needs to reduce the number of molecules of the interaction. Recalling the hypothesis 1, let us assume that only two elastic molecules undergo a collision during time interval τ. Asymptotic pre-collision velocities are denoted by v_1 and v_2 for the two colliding molecules, respectively. Let an origin of the coordinate be at the molecule 2 in order to be relative motion with the total momentum is zero. The molecule 1 is approaching to the center molecule 2, as shown in **Figure 33**.

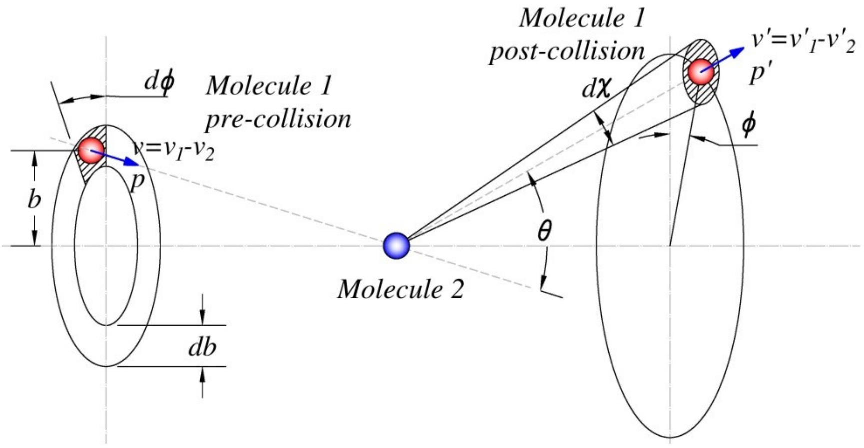


Figure 33. Sketch of a binary collision for the collision term and scattering of a molecule (Kerson, 1987).

The relative motion is characterized by the impact parameter b and by the azimuthal angle ϕ. Let their differential be db and $d\phi$. The pre-collision relative moment, p, of the molecule 1 become a rotated relative momentum, *p'*, after the collision with the molecule 2 without changing its magnitude. It is that the collision merely rotates the relative energy and momentum, the energy conservation, $|p| = |p'|$, is valid at the elastic collision (Kerson, 1987). The probability to find the molecule 1 with velocity *v₁* is $f_1 = f(\mathbf{x}_1, \mathbf{p}_1, t)$ while the probability to find the molecule 2 with velocity *v₂* is $f_2 = f(\mathbf{x}_2, \mathbf{p}_2, t)$ at the pre-collision state. If the probabilities at the post-collision become f_1' and f_2', the collision integral reads

$$\Omega(f, f) = \int (f_1' f_2' - f_1 f_2)|\boldsymbol{v}_1 - \boldsymbol{v}_2| b db d\phi d^3\mathbf{p}_2\ . \tag{132}$$

The collision integral, given in Eq.(132), is a general form of the collision term and the approximation or the model kinetic for the Boltzmann collision operator must hold some basic properties (Gilberto, 2010), (Kerson, 1987), (Ansumali, 2004) listed below.

a. Locality

The Boltzmann collision operator is local for the physical space and nonlocal for the momentum space. In other words, the molecules considered in the Boltzmann kinetic theory are points in physical space and the intermolecular forces in a limited range is considered for the interactions.

b. Summation invariants

Since the collision operator is reduced to the two-body elastic collision, the total mass, momentum and energy are conserved. This fact satisfies the following theorem.

Theorem 1: A continues function $\psi(\dot{\mathbf{x}})$ is a summation invariant if and only if

$$\psi(\dot{\mathbf{x}}) = A + \mathbf{B} \cdot \dot{\mathbf{x}} + D\dot{x}^2\ , \tag{133}$$

where A and D are two constant scalar functions (might be seen as density and energy) and B is a constant vector function (might be seen as momenta), all of them being independent of $\dot{\mathbf{x}}$.

The collision operator must satisfy the Theorem 1 in its form.

c. Zero point of the collision

Boltzmann describes that the all non-equilibrium system goes to an equilibrium state. In the equilibrium state, the collision integral becomes $\Omega(f,f) = 0$ and its solution gives

$$f_1 f_2 = f_1' f_2' \,, \tag{134}$$

which is an important balance in the Boltzmann kinetic theory. If this property is valid, a class of

$$f^0(\dot{\mathbf{x}}) = \exp(A + \boldsymbol{B} \cdot \dot{\boldsymbol{x}} + D\dot{x}^2) \tag{135}$$

function exist. In other words, a function satisfying the collision term in a collisionless state will be an equilibrium case of the distribution function *f*. In the class of Eq.(135), there is one physical rigorous solution, known as the Maxwell speed distribution function given in Eq.(13).

d. Boltzmann inequality

In any form of the distribution function *f*, Boltzmann stated that the property,

$$\int log\, f\ \Omega(f,f) d^3\mathbf{p} \leq 0\,, \tag{136}$$

holds. It is referred as a Boltzmann inequality and the equality sign applies if, and only if, *f* is given by Eq.(135) (Carlo, 1988). Hence, the collision term can describe the relaxation of the distribution function towards the local Maxwell distribution function. The Boltzmann inequality is also the implication of the Boltzmann ***H***-theorem, which is the property of the collision term, too.

The model kinetics, alternative collision operators for the Boltzmann collision term, not only satisfy above properties and but also retain the average qualitative and quality properties of the true collision operator. The widely used model for the kinetic theory and LBM is the Bhatnagar-Gross-Krook model (BGK) proposed by Bhatnagar's group (Bhatnagar, et al., 1954) and Welander (Welander, 1954), independently. The BKG model assumes that the average effect of the collision is to change the distribution function *f* by an amount proportional to the departure of *f* from an equilibrium distribution function f^{eq}. Thus, the BGK model is given by

$$\Omega_{BGK}(f,f) = w(f^0 - f)\,, \tag{137}$$

where w is a parameter of order of the collision frequency, which can be respected to $\dot{\mathbf{x}}$. With the BGK model, the Boltzmann transport equation (Eq.(119) or (131)) becomes

$$\frac{\partial f}{\partial t} + \dot{\mathbf{x}}\frac{\partial f}{\partial x_i} + \mathbf{F}\frac{\partial f}{\partial \dot{x}_i} = w(f^0 - f), \tag{138}$$

And it is the same as Eq.(109) without an external force term, where $w = 1/\tau_v$.

The BGK model is not a linear operator, because the equilibrium distribution function f^0 contains the moments of the distribution function *f*. So before the discussion about the equilibrium distribution function, let us consider the moments of the distribution function.

5.1.3 Macroscopic variables

The Boltzmann equation describes the time evolution of the distribution function, which is the main variable in the mesoscopic description of a fluid flow. The main question to be answered in this section is that how we can evaluate the macroscopic independent variables (density, velocity, energy, etc.) with the distribution function f. In other words, we shall discuss here the relationship between the Boltzmann equation and macroscopic fluid dynamics.

The distribution function, $f(\mathbf{x}, \mathbf{p}, t)$, is the function of phase space and the possibly molecule number in the system can be defined $N = \int f d^3\mathbf{x} d^3\mathbf{p}$ by its definition. If the molecules in a system have the same mass (a fluid is homogeneous), we can rewrite $f(\mathbf{x}, \mathbf{v}, t)$. Moreover, the distribution function behaves as the hypothesis 4, the number of molecules become

$$N = d^3\mathbf{x} \int f d^3\mathbf{v} \,. \tag{139}$$

Multiplying the both sides with m gives the total mass in the system:

$$mN = md^3\mathrm{x} \int f d^3\mathbf{v} \,. \tag{140}$$

From here, the density in a physical space can be defined as

$$\rho(\mathbf{x}, t) = \frac{mN}{d^3\mathbf{x}} = m \int f \, d^3\mathbf{v}$$

or, simply changing the space notation d^3 to d and including m into f:

$$\rho(\mathbf{x}, t) = \int f \, d\mathbf{v} \,. \tag{141}$$

This is called the zeroth order moment of the distribution function and is identical to the ensemble average in Eq.(7).

The macroscopic velocity can be given by the average of the molecular velocity

$$\mathbf{u} = \frac{\int \mathbf{v} f \, d\mathbf{v}}{\int f \, d\mathbf{v}} \,, \tag{142}$$

where the integral in the denominator is used to normalizing f to unity. According to Eq.(141), the velocity vector can also be written as

$$\rho\mathbf{u} = \int \mathbf{v} f \, d\mathbf{v} \,. \tag{143}$$

Even if a flow is at rest by the macroscopic observation, in other words **u**=0, the molecules have non zero velocity in the microscopic scale. This means there is the velocity deviation between the molecular velocity **c** and the macroscopic velocity **u**, which can be written as

$$\mathbf{C} = \mathbf{v} - \mathbf{u} \,. \tag{144}$$

The velocity **C** is called peculiar velocity (Carlo, 1988), (Gilberto, 2010) and is equal to molecular velocity **v** if the flow is macroscopically at rest. We have an important property of the peculiar velocity when we write its moment with the distribution function:

$$\int \mathbf{C} f \, d\mathbf{v} = \int \mathbf{v} f \, d\mathbf{v} - \mathbf{u} \int f \, d\mathbf{v} = \rho\mathbf{u} - \rho\mathbf{u} = 0 \,, \tag{145}$$

where Eq.(141) and (143) have been used.

Now it is interesting that how the other physical variables can be interpreted by the distribution function. To be simple, we can write the i-th component of velocity as

$$\rho u_i = \int v_i f \, d\mathbf{v} \, , \tag{146}$$

which can be said the momentum density in *i*-th direction. The *j*-th momentum flow in *i*-th direction can be written as

$$\int v_i (v_j f) \, d\mathbf{v} = \int v_i v_j f \, d\mathbf{v} \, , \tag{147}$$

where we use the general fact (Баасандорж, et al., 1999) and Eq.(147) shows that the momentum flow is described by a symmetric tensor of the second order. In order to find out the macroscopic implication of the term above, we have to change the molecular velocity by its peculiar part, $\mathbf{v} = \mathbf{u} + \mathbf{C}$. Equation (147) becomes

$$\begin{aligned} \int v_i v_j f \, d\mathbf{v} &= \int (u_i + C_i)(u_j + C_j) \, f d\mathbf{v} \\ = u_i u_j \int f \, d\mathbf{v}_{\lceil = u_i u_j \rho} &+ u_j \int C_i f \, d\mathbf{v}_{\lceil = 0} + u_i \int C_j f \, d\mathbf{v}_{\lceil = 0} + \int C_i C_j f \, d\mathbf{v} \\ &= u_i u_j \rho + \int C_i C_j f \, d\mathbf{v} \, . \end{aligned} \tag{148}$$

The resultant two terms describe the momentum flow decomposed into two parts, one of which can be recognized as the macroscopic momentum flow, while the second part can be related to a momentum flow driven by the peculiar velocity. Indeed, this can be addressed to the pressure tensor due to the peculiar velocity as

$$p_{ij} = \int C_i C_j f \, d\mathbf{v} \, . \tag{149}$$

The expression of Eq.(149) with the minus sign is known as the stress tensor.

The energy density, $\frac{1}{2} \int \mathbf{v}^2 f \, d\mathbf{v}$, can receive the similar analysis to Eq.(148) in order to identify the macroscopic appearance of it. The resultant term is

$$\frac{1}{2} \int \mathbf{v}^2 f \, d\mathbf{v} = \frac{1}{2} \rho \mathbf{u}^2 + \frac{1}{2} \int \mathbf{C}^2 f \, d\mathbf{v} \, , \tag{150}$$

where $\frac{1}{2} \rho \mathbf{u}^2$ is the macroscopic kinetic energy density and $\frac{1}{2} \int \mathbf{C}^2 d \, d\mathbf{v}$ can be ascribed to the internal energy of the fluid. If we note the internal energy as e, the second term in Eq.(150) gives the internal energy per unit volume as

$$\rho e = \frac{1}{2} \int \mathbf{C}^2 f \, d\mathbf{v} \, . \tag{151}$$

Introducing *i*=*j* into Eq.(149) gives the same term with the *i*-th component of internal energy. Hence we can write the relation between the pressure tensor and the internal energy as

$$p_{ii} = \int \mathbf{C}^2 f \, d\mathbf{v} = 2\rho e \, . \tag{152}$$

Moreover, the hydrostatic pressure of the fluid (gas) can be defined by the trace of the pressure tensor as

$$p(\mathbf{x},t)=\frac{1}{3}p_{ii}=\frac{1}{3}\int \mathbf{C}^2 f\,d\mathbf{v}\,. \tag{153}$$

The derivation of the internal energy leads us to the relation

$$p=\frac{2}{3}\rho e\,, \tag{154}$$

with Eqs.(151) and (152). This is the EOS of the gas and shows the possibility to define pressure in terms of the internal energy and density. Also, we can see that p/ρ is the constant when flow is isothermal. This statement will give us the EOS used in the LBM in the following sections. On the other hand, we have the EOS for the perfect gas (Boyle's law)

$$p=\rho RT\,, \tag{155}$$

where R is the constant and T is the absolute temperature. Equations (154) and (155) give

$$e=\frac{3}{2}RT\,, \tag{156}$$

or Eq.(153) and (155) gives

$$T(\mathbf{x},t)=\frac{p}{R\rho}=\frac{1}{3\rho R}\int \mathbf{C}^2 f\,d\mathbf{v}\,. \tag{157}$$

Let us investigate the energy flow; the total energy flow can be written by

$$\int c_i\left(\frac{1}{2}\mathbf{v}^2 f\right)d\mathbf{c}=\frac{1}{2}\int c_i\mathbf{v}^2 f\,d\mathbf{v}\,. \tag{158}$$

Introducing Eq.(144) into Eq.(158) and then applying Eq.(141), Eq.(145), Eq.(149), and Eq.(151) yields

$$\frac{1}{2}\int c_i\mathbf{v}^2 f\,d\mathbf{v}=u_i\left(\frac{1}{2}\rho\mathbf{u}^2-\rho e\right)-u_j p_{i,j}+\frac{1}{2}\int C_i\mathbf{C}^2 f\,d\mathbf{v}\,, \tag{159}$$

where the first term is the macroscopic energy flow due to the convection, the second term is the energy due to the work done by the pressure tensor per unit time and the third one is to be the additional energy called heat flux vector. The heat flux is given as

$$q_i(\mathbf{x},t)=\frac{1}{2}\int C_i\mathbf{C}^2 f\,d\mathbf{v}=\frac{1}{2}p_{jji}\,. \tag{160}$$

The common macroscopic variables such as Eqs.(141), (143), (151), (153) and (154) will be used for the LB modeling in further sections and the others will be mentioned if we need.

5.1.4 Equilibrium distribution function

If we observe the state of a fluid (gas) with an arbitrary distribution function at time without giving an external force to the system, we found that the state goes into an equilibrium situation, eventually. The distribution function belonging to the equilibrium state is called the equilibrium distribution function. In other words, the equilibrium distribution function of the gas will be the limiting form of the any distribution function, if time tends to infinity (Kerson, 1987). As said, let us assume that there is no external force. This assumption leads us to write the distribution function: $f(\mathbf{p},t)$, because the function is no longer dependent from physical space. The equilibrium distribution function, denoted by $f^0(\mathbf{p},t)$, can be found at $\frac{\partial f}{\partial t} = 0$. Putting this and the assumption into Eq.(131) yields

$$0 = \Omega(f,f)\,, \tag{161}$$

and $f^0(\mathbf{p},t)$ must satisfy the collision term. To solve Eq.(161), a sufficient condition is Eq.(134) and introducing $f^0(\mathbf{p},t)$ to Eq.(134) become

$$f^0(p_1)f^0(p_2) - f^0(p_1')f^0(p_2') = 0\,. \tag{162}$$

To show the sufficiency of Eq.(162), we write the Boltzmann functional

$$\boldsymbol{H}(t) = \int d^3\dot{\mathbf{x}} f(\mathbf{p},t) log\, f(\mathbf{p},t)\,, \tag{163}$$

and the distribution function in Eq.(163) is independent from physical space, but function of time, satisfying

$$\frac{\partial f(\mathbf{p},t)}{\partial t} = \Omega(f,f)\,. \tag{164}$$

If we differentiate the Boltzmann functional with time, we have

$$\frac{\partial \boldsymbol{H}(t)}{\partial t} = \int d^3\dot{\mathbf{x}} \frac{\partial f(\mathbf{p},t)}{\partial t}\big(1 + lnf(\mathbf{p},t)\big)\,. \tag{165}$$

To Eq.(165), substituting Eq.(164) and recalling $\Omega(f,f) = 0$, gives

$$\frac{\partial \boldsymbol{H}(t)}{\partial t} = 0\,. \tag{166}$$

This is the same as the condition in Eq.(162) and is a statement of finding the right form of the equilibrium distribution function. Equation (166) is also one case of the Boltzmann ***H***-theorem.

Theorem 2: If the distribution function satisfies the Boltzmann transport equation, then

$$\frac{dH(t)}{dt} \leq 0\,, \tag{167}$$

which is called the Boltzmann ***H***-theorem.

With Boltzmann ***H***-theorem, we saw that the equilibrium distribution function can be found from the condition in Eq.(162). Let us take logarithm of both sides of Eq.(162):

$$lnf^0(\mathbf{p}_1) + lnf^0(\mathbf{p}_2) = lnf^0(\mathbf{p}_1') + lnf^0(\mathbf{p}_2')\,. \tag{168}$$

Recalling the Theorem 1, the summation invariant is valid for function $\psi(\dot{\mathbf{x}})$ and the solution of natural logarithm of f^0 can have a form of the function $\psi(\dot{\mathbf{x}})$ as

$$lnf^0(\mathbf{p}) = \psi(\dot{\mathbf{x}})\,. \tag{169}$$

Hence, the general solution of Eq.(168) can be written as

$$lnf^0(\mathbf{p}) = \psi(\dot{\mathbf{x}}) + lnC$$

or

$$f^0(\mathbf{p}) = Ce^{\psi(\dot{\mathbf{x}})}\,, \tag{170}$$

where we would like to introduce a peculiar velocity $\mathbf{C}$ instead of a molecular velocity $(\mathbf{p} = m\mathbf{v} = \mathrm{m}\dot{\mathbf{x}})$, since the two velocities are equal in the case of an equilibrium state. The summation invariant $\psi(\dot{\mathbf{x}})$ as a polynomial of the $\dot{\mathbf{x}}$ with arbitrary scalar value in Eq.(133) can be rewritten in a simple form

$$\psi(\dot{\mathbf{x}}) = A + \mathbf{B}\cdot\dot{\mathbf{x}} + D\dot{x}^2 = -A(\dot{\mathbf{x}})^2 = -A(\mathbf{C})^2\,. \tag{171}$$

Mixing Eq.(170) and (171) gives

$$f^0(\mathbf{v}) = Ce^{-A(\mathbf{C})^2}\,, \tag{172}$$

where C and A are arbitrary constants and can be found by using the moments of the distribution function. Substituting Eq.(172) into Eq.(141), we can define the relation between density and the arbitrary constants as follow

$$\rho = \int f^0\,d\mathbf{v} = \int Ce^{-A(\mathbf{C})^2}\,d\mathbf{C} = C\left(\frac{\pi}{A}\right)^{\frac{3}{2}}, \tag{173}$$

where the Gaussian integral rule and three components of $\mathbf{C}$ have been considered. Since the arbitrary constants are stated as they are independent from velocity according to the Theorem 1, we do not find any relation from the analysis with Eq.(172) and (143). Now we proceed with Eq.(151) to see the connection of the constants with the internal energy:

$$e = \frac{1}{2}\int \mathbf{C}^2 f^0\,d\mathbf{v} = \frac{1}{2}\int \mathbf{C}^2 Ce^{-A(\mathbf{C})^2}\,d\mathbf{C} = \frac{3}{4A} \rightarrow A = \frac{3}{4e}\,. \tag{174}$$

We know A constant and substituting A to C in Eq.(173) gives

$$C = \rho\left(\frac{3}{4e\pi}\right)^{\frac{3}{2}}. \tag{175}$$

Gathering defined constants A and C and recalling the internal energy term with an absolute temperature in Eq.(156) into Eq.(172) yields

$$f^0(\mathbf{v}) = \rho\left(\frac{1}{2\pi RT}\right)^{\frac{3}{2}} e^{-\frac{(\mathbf{C})^2}{2RT}}$$

and then replacing the peculiar velocity $\mathbf{C}$ with its definition in Eq.(144) gives

$$f^0(\mathbf{v}) = \rho\left(\frac{1}{2\pi RT}\right)^{\frac{3}{2}} exp\left[-\frac{(\mathbf{v}-\mathbf{u})^2}{2RT}\right], \tag{176}$$

which is the Maxwell-Boltzmann distribution function, the probability of finding a molecule with velocity $\mathbf{c}$ in the gas at the equilibrium condition. To reduce the Maxwell speed distribution in Eq.(13), Eq.(176) must be multiplied by the factor $4\pi v^2$ (which is the surface area of a sphere in the velocity space) and replaced the density by the molecular mass.

To ensure mass and momentum conservation of the equilibrium distribution function, we can write following properties of the equilibrium distribution function with a help of the Gaussian integrals and they are:

$$\int f^0\,d\mathbf{v} = \int \rho\left(\frac{1}{2\pi RT}\right)^{\frac{3}{2}} exp\left[-\frac{(\mathbf{v}-\mathbf{u})^2}{2RT}\right] d\mathbf{v} = \rho\,, \tag{177}$$

and

$$\int \mathbf{v} f^0 \, d\mathbf{v} = 0 \, . \tag{178}$$

We will give the higher order moments of the equilibrium distribution function where they are used. Derivations of moments are the same. Equations (177) and (178) shows the mass and momentum conservation of the equilibrium distribution function.

So far we have successfully laid down the basic theory of the LBM discussing about the Boltzmann transport equation to its relation to the macroscopic scale. At some point of view, the LBM is the solving method for the Boltzmann equation. In the following sections, we will concentrate on the specific numerical techniques called the LB methods.

5.2 Lattice Boltzmann method for fluid flows

5.2.1 Discretized Boltzmann equation

We have derived the Boltzmann transport equation with the BGK collision operator. We have found that the equilibrium distribution function can be the Maxwell-Boltzmann distribution function. To solve such a differential Boltzmann equation in analytic way can be fulfilled by the moment method (Gilberto, 2010) and functional analysis (Carlo, 1988). We are interested to solve the Boltzmann equation by a numerical way, which requires the discretization for the Boltzmann equation in physical space. A distribution function in the Boltzmann equation is a function in phase space and one can replace the molecular velocity with a finite set of discrete velocities, which leads $f(\mathbf{x}, \mathbf{v}, t)$ to be associated with $f_i(\mathbf{x}, t)$. Hence, the Boltzmann equation in Eq.(138) becomes a discrete-velocity Boltzmann equation:

$$\frac{\partial f_i}{\partial t} + v_i \frac{\partial f_i}{\partial x_i} + F_i \frac{\partial f_i}{\partial v_i} = \frac{f_i^0 - f_i}{\tau_v} \, , \tag{179}$$

where $v_i(= \dot{x_\imath})$ is the finite set of discrete velocities and its set need to be defined by the way that the discrete-velocity Boltzmann equation can produce the Navier-Stokes equation correctly. To make it suitable for a general analysis, we can remove units of the variables from the discrete Boltzmann equation using the following parameters:

$$\hat{f}_i = \frac{f_i}{\rho_o}, c_i = \frac{v_i}{U}, \hat{t} = \frac{U}{L} t, \hat{x}_i = \frac{x_i}{L}, \hat{\tau}_v = \frac{\tau_v}{t_c}, \hat{F}_i = F_i \frac{U^2}{L} \, , \tag{180}$$

where ρ_o is the reference density of a fluid, U is the characteristic velocity, L is the characteristic length, t_c is the time between particle collisions. The equilibrium distribution function is also scaled as the distribution function with the reference density. The nondimensionalization with parameters in Eq.(180) for Eq.(179) gives the dimensionless discrete-velocity Boltzmann equation:

$$\frac{\partial \hat{f}_i}{\partial \hat{t}} + c_i \frac{\partial \hat{f}_i}{\partial \hat{x}_i} + \hat{F}_i \frac{\partial \hat{f}_i}{\partial c_i} = \frac{\hat{f}_i^0 - \hat{f}_i}{\hat{\tau}_v} \, , \tag{181}$$

The dimensionless discrete Boltzmann equation is not yet discrete in terms of space and time. To make the equation fully discrete, we need to discretize Eq.(181) in time and space as

$$\frac{\hat{f}_i(\mathbf{x},t+\delta\hat{t})-\hat{f}_i(\mathbf{x},t)}{\delta\hat{t}}+c_i\frac{\hat{f}_i(\mathbf{x}+\delta\hat{x},t+\delta\hat{t})-\hat{f}_i(\mathbf{x},t+\delta\hat{t})}{\delta\hat{x}}=\frac{\hat{f}_i^0-\hat{f}_i}{\hat{\tau}_v}-\hat{F}_i\frac{\partial\hat{f}_i}{\partial c_i}\,, \tag{182}$$

in which, introducing $c_i=\frac{\delta\hat{x}}{\delta\hat{t}}$ yields,

$$\hat{f}_i(x+\delta\hat{x},t+\delta\hat{t})-\hat{f}_i(x,t)=\frac{\delta\hat{t}}{\hat{\tau}_v}\left(\hat{f}_i^0-\hat{f}_i\right)-\delta\hat{t}\hat{F}_i\frac{\partial\hat{f}_i}{\partial c_i}\,. \tag{183}$$

The external force term in Eq.(183) is not calculated directly because the derivative of the distribution function depending on the velocity set is unknown. Assuming that the gradient of $\hat{f}$ can be approximated by the gradient of its equilibrium part (He, et al., 1998), the force term can be expressed as

$$\hat{F}_i\frac{\partial\hat{f}_i}{\partial c_i}\approx\hat{F}_i\frac{\partial\hat{f}_i^0}{\partial c_i}=\hat{F}_i\frac{\partial}{\partial c_i}\left(\rho\left(\frac{1}{2\pi RT}\right)^{\frac{3}{2}}exp\left[-\frac{(\hat{\mathbf{v}}-\hat{\mathbf{u}})^2}{2RT}\right]\right)=-\hat{F}_i\frac{\hat{\mathbf{v}}-\hat{\mathbf{u}}}{RT}\hat{f}_i^0\,, \tag{184}$$

where we have substituted Eq.(176). The discrete Boltzmann equation with the approximated external force term can be written as

$$\frac{\partial f_i}{\partial t}+v_i\frac{\partial f_i}{\partial x_i}=\frac{f_i^0-f_i}{\tau_v}+F_i\frac{\mathbf{v}-\mathbf{u}}{RT}f_i^0\,, \tag{185}$$

whereas the discretized Boltzmann equation can be given

$$\hat{f}_i(\mathbf{x}+\delta\hat{x},t+\delta\hat{t})-\hat{f}_i(\mathbf{x},t)=\frac{\delta\hat{t}}{\hat{\tau}_v}\left(\hat{f}_i^0-\hat{f}_i\right)+\delta\hat{t}\hat{F}_i\frac{\hat{\mathbf{v}}-\hat{\mathbf{u}}}{RT}\hat{f}_i^0\,. \tag{186}$$

Main terms in Eq.(186) are dimensionless and the results driven by this equation need to be scaled to a physical space. Scaling of results is discussed in Section 6.3. For simplicity, we can remove the tilde of variables denoting their dimensionless in Eq.(186). Hence, the dimensionless discretized Boltzmann equation for further discussions is given

$$f_i(\mathbf{x}+\delta x,t+\delta t)-f_i(\mathbf{x},t)=\frac{\delta t}{\tau_v}\left(f_i^0-f_i\right)+\delta tF_i\frac{\mathbf{v}-\mathbf{u}}{RT}f_i^0\,. \tag{187}$$

Note that the velocities in the force term are dimensional. Also the external force term and the equilibrium distribution functions are still continuous terms and should be discretized into the physical space from the phase space.

Approximated equilibrium distribution function

The Maxwell-Boltzmann equation in the discretized Boltzmann equation is non-linear and implicit still depends on molecular velocity. In order to approximate f_i^0 in a physical space with a finite set of velocities, we need to expand it by the Taylor series. Expanding Eq.(176) up to $O(\mathbf{C}^2)$ yields

$$f^0\approx f^{eq}=\rho\frac{1}{(2\pi RT)^{\frac{3}{2}}}exp\left(-\frac{\mathbf{v}^2-2\mathbf{uv}+\mathbf{u}^2}{2RT}\right)$$

$$= \rho \frac{1}{(2\pi RT)^{\frac{3}{2}}} e^{-\frac{\mathbf{v}^2}{2RT}} \left[e^{-\frac{2\mathbf{u}\mathbf{v}-\mathbf{u}^2}{2RT}} \right]$$

$$= \rho \frac{1}{(2\pi RT)^{\frac{3}{2}}} e^{-\frac{\mathbf{v}^2}{2RT}} \left[1 + \frac{\mathbf{u}\mathbf{v}}{RT} - \frac{\mathbf{u}^2}{2RT} + \frac{\mathbf{u}^2\mathbf{v}^2}{2(RT)^2} - \frac{\mathbf{u}^3\mathbf{v}}{2(RT)^2} + \frac{\mathbf{u}^4}{8(RT)^2} - O(\mathbf{C}^3) \right]$$

$$= \rho \frac{1}{(2\pi RT)^{\frac{3}{2}}} e^{-\frac{\mathbf{v}^2}{2RT}} \left[1 + \frac{\mathbf{u}\mathbf{v}}{RT} - \frac{\mathbf{u}^2}{2RT} + \frac{\mathbf{u}^2\mathbf{v}^2}{2(RT)^2} - O(\mathbf{C}^3) \right] \tag{188}$$

Before introducing a discrete velocity set in the equilibrium distribution function, let us write f^{eq} in the following form:

$$f^{eq} = \rho \frac{1}{(2\pi RT)^{\frac{3}{2}}} e^{-\frac{\mathbf{v}^2}{2RT}} \left[1 + \frac{\mathbf{u}\mathbf{v}}{RT} - \frac{\mathbf{u}^2}{2RT} + \frac{\mathbf{u}^2\mathbf{v}^2}{2(RT)^2} - O(\mathbf{C}^3) \right] = \frac{e^{-\frac{\mathbf{v}^2}{2RT}}}{(2\pi RT)^{\frac{3}{2}}} f^{eq'} . \tag{189}$$

To discretize f^{eq} in a velocity space, we now can use the Gauss-Hermite quadrature for the integration of Eq.(189) in a velocity space, which gives

$$\int f^{eq}\, d\mathbf{C} = \int_{-\infty}^{+\infty} \frac{e^{-\frac{\mathbf{v}^2}{2RT}}}{(2\pi RT)^{\frac{3}{2}}} f^{eq'} d\mathbf{C} \approx \sum_{i=1}^{n} w_i\, f_i^{eq'}, \tag{190}$$

where w_i are the weights. We can see that $w_i f_i^{eq'} = f_i^{eq} \approx f_i^0$, which can be written by

$$f_i^0 \approx f_i^{eq} = w_i f_i^{eq'} = w_i \rho \left[1 + \frac{\mathbf{u}\mathbf{v}}{RT} - \frac{\mathbf{u}^2}{2RT} + \frac{\mathbf{u}^2\mathbf{v}^2}{2(RT)^2} \right], \tag{191}$$

and in which, Eq.(189) has been used. On the other point, Eqs.(177) and (190) indicate the following properties of f_i^{eq}:

$$\sum_{i=1}^{n} f_i^{eq} = \rho. \tag{192}$$

Now it is time to nondimensionalize the equilibrium distribution function in terms of velocity using $\mathbf{c} = \mathbf{v}/\mathbf{U}$ and $\mathbf{u} \leftarrow \hat{\mathbf{u}} = \mathbf{u}/\mathbf{U}$ (to have simple notations), Eq.(191) becomes

$$f_i^{eq} = w_i \rho \left[1 + \frac{\mathbf{u}\mathbf{c}}{RT} - \frac{\mathbf{u}^2}{2RT} + \frac{\mathbf{u}^2\mathbf{c}^2}{2(RT)^2} \right]. \tag{193}$$

Note that w_i in Eq.(193) depends on the discrete set of velocities and we will discuss it in Section 5.2.2.

From discussions in above, $f_i^0 \approx f_i^{eq}$ is an important approximation and those functions are interchangeable variables, therefore Eq.(187) can also be given as

$$f_i(\mathbf{x} + \delta x, t + \delta t) - f_i(\mathbf{x}, t) = \frac{\delta t}{\tau_v}\left(f_i^{eq} - f_i\right) + \delta t F_i \frac{\mathbf{c} - \mathbf{u}}{RT} f_i^{eq} , \tag{194}$$

where velocities in the force term are nondimensionalized. The equilibrium distribution function in Eq.(194) is given by Eq.(193), while the force term still requires the adequate approximation.

Force term in a discretized form

Discretization applied in Eq.(182) is a simple finite difference approximation for the discrete Boltzmann equation in time and space. Another discretizing method is the integration of equation between definite time ranges. For simplicity, we can denote the collision and force terms in Eq.(181) as

$$K_i = \frac{\hat{f}_i^0 - \hat{f}_i}{\hat{\tau}_v} - \hat{F}_i \frac{\partial \hat{f}_i}{\partial c_i}\,, \tag{195}$$

and its integration from $t = 0$ to $t = \delta t$ gives

$$\int_0^{\delta t} K_i(\mathbf{x}, t) = \frac{\delta t}{2}\big(K_i(\mathbf{x} + \delta x, t + \delta t) - K_i(\mathbf{x}, t)\big)\,, \tag{196}$$

where the trapezoidal rule has been used. It is possible to replace the right hand side of Eq.(187) with Eq.(196) and the discretized Boltzmann equation becomes

$$f_i(\mathbf{x} + \delta x, t + \delta t) - f_i(\mathbf{x}, t) = \frac{\delta t}{2}\Lambda_i(\mathbf{x}, t)$$

where $\Lambda_i = K_i(\mathbf{x} + \delta x, t + \delta t) - K_i(\mathbf{x}, t)$ and which gives the distribution function as

$$f_i = f_i + \frac{\delta t}{2}\Lambda_i(\mathbf{x}, t)\,. \tag{197}$$

At this point, the moments of the distribution function in Eqs.(141) and (143) become

$$\rho = \int f_i\, d\mathbf{c} = \int \left(f_i + \frac{\delta t}{2}\Lambda_i(\mathbf{x}, t)\right) d\mathbf{c} = \int f_i d\mathbf{c} + \int \frac{\delta t}{2}\Lambda_i(\mathbf{x}, t) d\mathbf{c}_{\lceil=0} = \rho\,, \tag{198}$$

$$\rho u_i = \int \mathbf{c}_i f_i\, d\mathbf{c} = \int \mathbf{c}_i \left(f_i + \frac{\delta t}{2}\Lambda_i(\mathbf{x}, t)\right) dc_i = \int \mathbf{c}_i f_i\, d\mathbf{c} + \frac{\delta t}{2}\hat{F}_i\,. \tag{199}$$

Sums over the finite set of velocities of integrals in Eqs.(198) and (199) are constructed through the Gauss-Hermits quadrature as,

$$\rho = \sum_i^n f_i\,, \tag{200}$$

$$\rho \mathbf{u} = \sum_i^n \mathbf{c}_i f_i + \frac{\delta t}{2} F_i\,, \tag{201}$$

where $\hat{F}_i$, a macroscopic external acceleration of the force, denoted as F_i. Note that the sum of the weight, $\sum w_i$, derived from the Gauss-Hermits quadrature is unit, thus it is neglected in Eqs.(200) and (201). We saw that above sums are the same as given sums in Eq.(108). The additional term in the velocity momentum comes from the force term. We stated that the force term in the discretized Boltzmann equation needs the approximation to have a discrete form. Introducing the approximated equilibrium distribution function in Eq.(193) into the force term yields

$$A_i = F_i \frac{\mathbf{c}_i - \mathbf{u}}{RT} f_i^{eq} = w_i \rho \left[\frac{\mathbf{F}(\mathbf{c}_i - \mathbf{u})}{RT} + \frac{(\mathbf{F} \cdot \mathbf{u})\mathbf{c}_i{}^2}{(RT)^2} - \frac{3(\mathbf{F} \cdot \mathbf{c}_i)\mathbf{u}^2}{2(RT)^2} + O(\mathbf{C}^3) \right], \tag{202}$$

which is called a moment-expansion scheme (Guo & Shu, 2013). The modification of this scheme with a relaxation time gives the He-Shan-Doolen force scheme, which is defined by

$$A_i = \left(1 - \frac{1}{2\tau}\right) w_i \rho \left[\frac{\mathbf{c}_i \cdot \mathbf{F}}{RT} - \frac{\mathbf{u} \cdot \mathbf{F}}{RT} + \frac{\mathbf{c}_i{}^2(\mathbf{u} \cdot \mathbf{F})}{(RT)^2} \right], \tag{203}$$

where **F** is the acceleration of external force. Now the discretized Boltzmann equation is

$$f_i(\mathbf{x} + \mathbf{c}_i \delta x, t + \delta t) - f_i(\mathbf{x}, t) = \frac{\delta t}{\tau_v}\left(f_i^{eq} - f_i\right) + \delta t A_i\,, \tag{204}$$

where the viscosity is realized as

$$\nu = \delta t \left(\frac{1}{w} - \frac{1}{2}\right) RT = \delta t \left(\tau_v - \frac{1}{2}\right) c_s^2. \tag{205}$$

After the temporal evolution of Eq.(204), we will compute the macroscopic density and velocity with the distribution functions according to Eq.(200) and (201), respectively. There are many versions of force terms (Guo & Shu, 2013) embeddable in the discretized Boltzmann equation and the simplest one is

$$A_i = w_i \frac{\mathbf{c}_i \cdot \mathbf{F}}{RT}\,, \tag{206}$$

which is originated from the LGA and is constructed based on the minimal force term properties:

$$\int d\mathbf{v} \mathbf{F} \frac{\partial f}{\partial \mathbf{v}} = \sum_i^n A_i = 0, \text{ and} \tag{207}$$

$$\int d\mathbf{v} \mathbf{F} \frac{\partial f}{\partial \mathbf{v}} \mathbf{v} = \sum_i^n A_i \mathbf{c}_i = \rho \mathbf{F}\,. \tag{208}$$

If the force scheme in Eq.(206) is considered in a simulation, the macroscopic density and velocity are computed by

$$\rho = \sum_i^n f_i \text{ and } \rho \mathbf{u} = \sum_i^n \mathbf{c}_i f_i\,, \tag{209}$$

respectively. Numerical works in this thesis use the force scheme given in Eq.(203), which also satisfies the constraints given in Eqs.(207) and (208).

A numerical procedure to solve the discretized Boltzmann equation (Eq.(204)) is split into two numerical steps called streaming and collision. Those steps are seen from a general pattern of the discretized Boltzmann equation and are suitable to handle boundary conditions for a complicated geometry. In appendix A, we show how the lattice Boltzmann equation derives the NSE. But now we will proceed to show what kind of lattices can be used to the numerical simulations in the standard approach. The word standard refers the original LBM without any alteration, modification or numerical ingredients.

5.2.2 Lattices for discretized Boltzmann equation

We have found two things, the weights and the finite velocity sets, from the discretization of the Boltzmann equation in Section 5.2.1. Now we shall define them for specific cases of, let us say, lattices. In a physical space, velocity vectors of a group of particles/molecules direct to everywhere. We can approximately sort them to the certain directions depending on the dimension, e.g. we can find how many particles with velocity in a given finite space approximately direct to the cardinal directions in the two dimensional spaces. Those cardinal directions are considered as a selected finite velocity set, as illustrated in **Figure 34**.

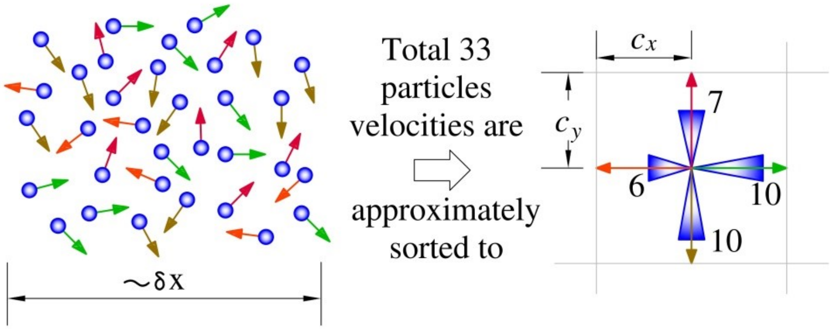


Figure 34. Example of selecting the finite velocity set: Four ordinal directions are selected as finite velocities and each of which a certain number of particles belongs to. For an instant, in a given range, 6 particles velocities are directed to west direction, approximately.

But, choosing the finite set of velocities is not arbitrary. A selected velocity set must satisfy hydrodynamic laws and Galilean invariant, and results with this must be the same as what would be obtained from the Boltzmann equation in velocity space. Generally, the finite velocity set is referred as a lattice. For the each direction of lattice, we need to define the weights and velocities.

Constraints of the lattice weights

In Section 5.1.4, two moments of equilibrium distribution functions are derived in Eqs.(177) and (178), which can be given in discretized forms as follows:

$$\sum_i f_i^0(\mathbf{x},t) = \rho(\mathbf{x},t)\,, \tag{210}$$

$$\sum_i \mathbf{c}_i f_i^0(\mathbf{x},t) = \rho(\mathbf{x},t)\mathbf{u}(\mathbf{x},t)\,. \tag{211}$$

All moments, including moments up to the third order moments of the equilibrium distribution functions, will be used to define the lattice weights. To illustrate a procedure to derive constraints on the weights, we will use above the zeroth and first order moments. In order to analyze the weights, let us introduce Eq.(193) into Eq.(210):

$$\rho = \sum_i f_i^{eq} = \sum_i \rho w_i \left[1 + \frac{\mathbf{uc}}{RT} - \frac{\mathbf{u}^2}{2RT} + \frac{\mathbf{u}^2\mathbf{c}^2}{2(RT)^2}\right]$$

$$= \rho\left[\underbrace{\sum_i w_i}_{=1} + \frac{\mathbf{u}}{RT}\underbrace{\sum_i w_i \mathbf{c}_i}_{=0} + \frac{\mathbf{u}^2}{2RT}\left(\frac{1}{RT}\underbrace{\sum_i w_i \mathbf{c}_i^2}_{=RT} - \sum_i w_i\right)\right],$$

where the counterpart of the square brackets must be 1. Assuming the weights are constant, the above moment is true, only if

$$\sum_i w_i = 1\,, \tag{212}$$

$$\sum_i w_i \mathbf{c}_i = 0 \;\text{ and} \tag{213}$$

$$\sum_i w_i \mathbf{c}_i^2 = RT\,. \tag{214}$$

Those are the first three of constraints for the lattice weights and finite set of velocity vector. From Eq.(211), we can write:

$$\rho\mathbf{u} = \sum_i \mathbf{c}_i f_i^{eq} = \sum_i \mathbf{c}_i \rho w_i\left[1 + \frac{\mathbf{uc}}{RT} - \frac{\mathbf{u}^2}{2RT} + \frac{\mathbf{u}^2\mathbf{c}^2}{2(RT)^2}\right]$$

$$= \rho\left[\underbrace{\sum_i \mathbf{c}_i w_i\left(1 - \frac{\mathbf{u}^2}{2RT}\right)}_{=0} + \frac{\mathbf{u}}{RT}\underbrace{\sum_i \mathbf{c}_i^2 w_i}_{=RT} - \frac{\mathbf{u}^2}{2(RT)^2}\underbrace{\sum_i \mathbf{c}_i^2 \mathbf{c}_i w_i}_{=0}\right],$$

which gives the fourth constraints of w_i and $\mathbf{c}_i$ as

$$\sum_i \mathbf{c}_i^2 \mathbf{c}_i w_i = 0\,. \tag{215}$$

In order to write the higher order constraints in a convention form, let us introduce following summation conventions:

$$u_\alpha c_{i\alpha} = \sum_\alpha u_\alpha c_{i\alpha} = u_x c_{ix} + u_y c_{iy} + u_z c_{iz} = \mathbf{u}\cdot\mathbf{c}_i\,, \tag{216}$$

$$\delta_{\alpha\beta} = \begin{cases} 1 & \text{if } \alpha = \beta \text{ and} \\ 0 & \text{if } \alpha \neq \beta \end{cases}, \tag{217}$$

where α and β are the indexes indicating components, $\delta_{\alpha\beta}$ is the Kronecker delta function. All constraints of w_i and $\mathbf{c}_i$ are

$$\sum_i w_i = 1\,,\;\; \text{(recalled from Eq. (212))} \tag{218}$$

$$\sum_i w_i c_{i\alpha} = 0\,, \text{(recalled from Eq. (213))} \tag{219}$$

$$\sum_i w_i c_{i\alpha} c_{i\beta} = RT\delta_{\alpha\beta}\,, \text{(recalled from Eq. (214))} \tag{220}$$

$$\sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} = 0\,, \text{(recalled from Eq. (215))} \tag{221}$$

$$\sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} = (RT)^2 \left(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\gamma\beta}\right) \text{ and} \tag{222}$$

$$\sum_i w_i \, c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} c_{i\varepsilon} = 0\,. \tag{223}$$

The latter two constraints would be derived from the second and third order moments of the equilibrium distribution function (Viggen, 2014). The even moments of w_i and $\mathbf{c}_i$ are isotropic tensors, whereas the odd moments are vanished due to the even symmetry (Wolf-Gladrow, 2000).

Grid and lattice

As mentioned above, a finite velocity set discretized from the continuous velocity of a particle is considered as a lattice. The geometric domain of the flow problem governed by the discrete Boltzmann equation has also gotten to discretized into a grid depending on the dimension. As is descended from its ancestor, the standard LBM usually uses a uniform structured grid for the discretization. Like other conventional methods, each cell of the grid in the LBM has the temporal physical values from macroscopic scale. However, the main variable in the LB computation is the distribution function, which has discretized into the finite velocity set with a lattice in Section 5.2.1. A spatial address of the distribution function allows us to let coincide the lattice on the grid for the LBM.

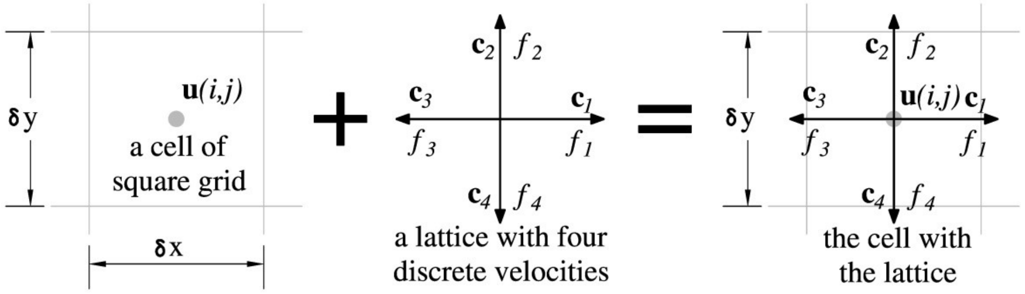


Figure 35. An example of compounding the computational grid (generally called lattice) for the LBM: Cartesian grid with regular lattice.

The computational grid (lattice) in the LBM is composed from the alignment of the grid and lattice, the procedure of which is depicted in **Figure 35**. Generally, the lattices for the LBM are notated as DdQq, where D is an initial of dimension, d is a number of dimensions, Q is an initial of qualifiers and q is a number of discrete velocities (Qian, et al., 1992). For instance, the lattice depicted in **Figure 35** can be D2Q4, however, this arrangement of the lattice is barely used for fluid simulations. We shall discuss about representative lattice arrangements for each spatial dimension, namely D1Q3, D2Q9 and D3Q19, and shall define the lattice weights and velocity sets for them.

One dimensional lattice: D1Q3

To solve one dimensional fluid problem by the LBM, we can use many lattices with different arrangements, such as D1Q2, D1Q3, and D1Q5. Depending on the number of finite velocities, lattices can be classified as lower and higher order lattices. The most popular one is D1Q3 lattice arrangement, which has three velocities, as shown in **Figure 36** (a).

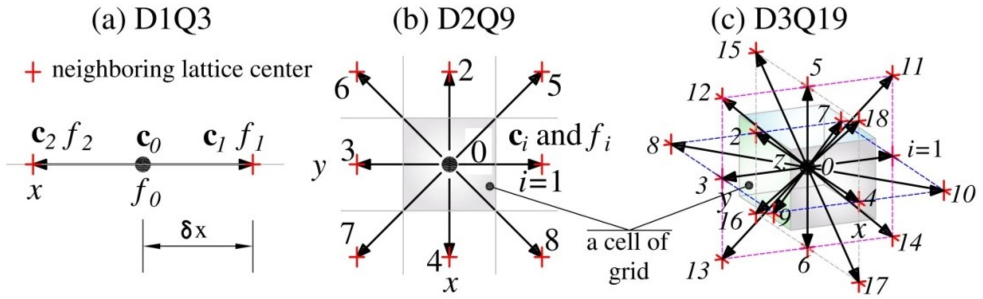


Figure 36. Widely used lattice arrangement for LBM in (a) one-, (b) two-, and (c) three dimensional spaces.

All formal lattices used for the LBM have two types of velocities: non-zero velocities and zero-velocity. Zero-velocity vector and associated distribution function define particles being at rest on the lattice. Non-zero velocities direct and reach up neighboring lattice center. For D1Q3, non-zero velocities, $\mathbf{c}_{i\neq 0}$, are equal to each other and can be defined as $\mathbf{c}_1 = \mathbf{c}_2 = \delta x/\delta t$. Similarly, the weights for those velocities can be denoted as w_0 and $w_1 = w_2 = w_x$, where x is horizontal velocities. The constraints, given in Eqs.(218-223), for the weights are:

$$\sum_i w_i = w_0 + 2w_x = 1\,,\quad \text{(from Eq. (218))}$$

$$\sum_i w_i c_{i\alpha} c_{i\beta} = w_0 \mathbf{c}_0^2 + w_x \mathbf{c}_1^2 + w_x \mathbf{c}_2^2 = 2w_x \left(\frac{\delta x}{\delta t}\right)^2 = RT(1)\ \text{, (from Eq. (220))}$$

and

$$\sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} = w_0 \mathbf{c}_0^4 + w_x \mathbf{c}_1^4 + w_x \mathbf{c}_2^4 = 2w_x \left(\frac{\delta x}{\delta t}\right)^4 = (RT)^2(3)\ \text{. (from Eq. (222))}$$

We have three unknowns for D1Q3 arrangement, and above expressions give a system of equation

$$\begin{cases} w_0 + 2w_x = 1 \\ 2w_x \left(\dfrac{\delta x}{\delta t}\right)^2 = RT \\ 2w_x \left(\dfrac{\delta x}{\delta t}\right)^4 = 3(RT)^2 \end{cases}, \tag{224}$$

which gives solutions of:

$$w_0 = \frac{2}{3}, w_{1,2} = w_x = \frac{1}{6} \text{ and } RT = \left(\frac{\delta x}{\delta t}\right)^2 \frac{1}{3}\,. \tag{225}$$

We know that the RT comes from the approximation of the Maxwell-Boltzmann distribution function and is the product of the absolute temperature and ideal gas constant. Since the RT is defined as a constant parameter, a flow system is enforced to be an isothermal process. Defining the RT from the EOS derived in Eq.(155) and introducing the result in Eq.(225) gives

$$RT = \frac{p}{\rho} \cong \frac{\partial p}{\partial \rho} = {c_s}^2 \to \left(\frac{\delta x}{\delta t}\right)^2 \frac{1}{3} \ , \tag{226}$$

where c_s is the speed of sound by definition and it is called the lattice speed of sound. The equilibrium distribution function can be rewritten with the speed of sound as

$$f_i^{eq} = w_i \rho \left[1 + \frac{\mathbf{uc}}{{c_s}^2} - \frac{\mathbf{u}^2}{2{c_s}^2} + \frac{\mathbf{u}^2\mathbf{c}^2}{2{c_s}^4}\right]. \tag{227}$$

One dimensional numerical example by the LBM with D1Q3 lattice arrangement is given to the comparison to the finite difference method in Section 5.2.4.

Two dimensional lattice: D2Q9

Two dimensional fluid flows often carried out on D2Q9 lattice arrangement, as shown in **Figure 36** (b), which has 9 velocities including the zero-velocity. Other lower and higher order lattices, e.g. D2Q4 and D2Q12, respectively, can be used for the LB computations. However, some of them not satisfy the conservation laws in case of fluid flows, while others require high computational costs. From the arrangement, there are three types of velocity vectors, let say zero-velocity, axis velocity and diagonal velocity. They can be denoted as $w_o, w_1 = w_2 = w_3 = w_4 = w_a$ and $w_5 = w_6 = w_7 = w_8 = w_d$, where a means axis and d means diagonal. In addition, the RT can be different, thus we need to construct four of equations from the constraints. The equations are:

$$\sum_i w_i = w_0 + 4w_a + 4w_d = 1\,,$$

$$\sum_i w_i c_{i\alpha} c_{i\beta} = 2\underbrace{\sum_i w_i c_{ix} c_{iy}}_{=0} + \sum_i w_i c_{ix} c_{ix} + \sum_i w_i c_{iy} c_{iy} = RT(2) \to$$

$$\to 2\left(2w_a\left(\frac{\delta x}{\delta t}\right)^2 + 4w_d\left(\frac{\delta x}{\delta t}\right)^2\right) = 2RT \ \to \left(\frac{\delta x}{\delta t}\right)^2 (2w_a + 4w_d) = RT\,,$$

and

$$\sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\delta} = 4\sum_i w_i c_{ix}^2 c_{iy}^2 + \left(\sum_i w_i c_{ix}^4 + \sum_i w_i c_{iy}^4\right) = 4(RT)^2 + 6(RT)^2,$$

which can be extracted as

$$\sum_i w_i c_{ix}^2 c_{iy}^2 = (RT)^2 \text{ and } \sum_i w_i c_{ix}^4 + \sum_i w_i c_{iy}^4 = 6(RT)^2.$$

Those expressions give

$$4w_d\left(\frac{\delta x}{\delta t}\right)^4 = (RT)^2 \ \text{ and}$$

$$2(2w_a + 4w_d)\left(\frac{\delta x}{\delta t}\right)^4 = 6(RT)^2 \text{ , respectively.}$$

Gathering derived equations for the weights in the system becomes

$$\begin{cases} w_0 + 4w_a + 4w_d = 1 \\ \left(\frac{\delta x}{\delta t}\right)^2 (2w_a + 4w_d) = RT \\ 4w_d \left(\frac{\delta x}{\delta t}\right)^4 = (RT)^2 \\ (2w_a + 4w_d)\left(\frac{\delta x}{\delta t}\right)^4 = 3(RT)^2 \end{cases}, \tag{228}$$

which yields solution of

$$RT = \left(\frac{\delta x}{\delta t}\right)^2 \frac{1}{3}, w_0 = \frac{4}{9}, w_a = \frac{1}{9} \text{ and } w_d = \frac{1}{36}. \tag{229}$$

The speed of sound defined in the 1D space is the same as the speed of sound defined in the 2D space. Therefore the speed of sound for a regular lattice used in fluid simulations by the LBM can be

$$c_s = \frac{\delta x}{\delta t}/\sqrt{3}\ . \tag{230}$$

All computational works in this thesis are performed in the 2D space using D2Q9 lattice arrangement. Implementation of the regular LBM is the same for all dimensional cases and is discussed in Chapter 6.

Three dimensional lattice: D3Q19

For three dimensional computations D3Q15, D3Q19 and D2Q27 arrangements are often used. More convenient one is D3Q19, which has 19 velocities as shown in **Figure 36** (c). Three dimensional lattice is composed from several planes of two dimensional lattice. Accordingly, there are three unknowns of the weighting: w_0, $w_a = w_{1\sim6}$ and $w_d = w_{7\sim18}$, where a and d meant to be axis and diagonal. We need to construct a system with four equations using the same procedures presented in previous examples and the system results following parameters:

$$w_0 = \frac{1}{3}, w_{1\sim6} = \frac{1}{18}, w_{7\sim18} = \frac{1}{36} \text{ and } RT = \left(\frac{\delta x}{\delta t}\right)^2 \frac{1}{3}. \tag{231}$$

Once the lattice is defined according to isotropic properties, the discretized Boltzmann equation on that lattice becomes the so-called lattice Boltzmann equation.

Velocity vectors for lattices

A discrete velocity set must hold the same constraints of the weights. In a use of regular lattice, velocity vectors can be defined from the lattice arrangement. For example, the discrete velocities in D2Q9 lattice can be estimated as

$$\mathbf{c}_i = \begin{cases} (0,0) & i = 0 \\ c_x\left(cos\left[(i-1)\frac{\pi}{2}\right], sin\left[(i-1)\frac{\pi}{2}\right]\right) & i = 1\sim4 \\ \sqrt{2}c_x\left(cos\left[(i-5)\frac{\pi}{2}+\frac{\pi}{4}\right], sin\left[(i-5)\frac{\pi}{2}+\frac{\pi}{4}\right]\right) & i = 5\sim8 \end{cases}, \tag{232}$$

where $c_x (= \delta x / \delta t)$ is the lattice speed and δx is the grid spacing. In the regular lattice arrangement, $\delta x = \delta y$ and the lattice spacing and lattice time step are assumed to be unit. Simply, the unity of the lattice space and time step give the unit velocity vectors as shown in **Table 2**.

Table 2. Velocities for D1Q3, D2Q9 and D3Q19 with $\delta x = 1$ and $\delta t = 1$.

i	$\mathbf{c}_0$	$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$	$\mathbf{c}_4$	$\mathbf{c}_5$	$\mathbf{c}_6$	$\mathbf{c}_7$	$\mathbf{c}_8$	$\mathbf{c}_9$	$\mathbf{c}_{10}$	$\mathbf{c}_{11}$	$\mathbf{c}_{12}$	$\mathbf{c}_{13}$	$\mathbf{c}_{14}$	$\mathbf{c}_{15}$	$\mathbf{c}_{16}$	$\mathbf{c}_{17}$	$\mathbf{c}_{18}$
D1Q3	0	1	-1																
D2Q9	0	1	0	-1	0	1	-1	-1	1										
	0	0	1	0	-1	1	1	-1	-1										
D2Q19	0	1	0	-1	0	0	0	1	-1	-1	1	1	-1	-1	1	0	0	0	0
	0	0	1	0	-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

Because the lattice Boltzmann equation is a dimensionless equation, its discretization parameters, such as lattice spacing and lattice time step, do not have physical unit. In some reference, units from lattice assigned for them as $\delta x = 1\ lu$ and $\delta t = 1\ lt$, where lu is lattice unit and lt is the lattice time (Michael & Daneal T, 2006).

5.2.3 Boundary conditions

In a limited space, boundary conditions are used to implicate characteristic of regions that do not consider in a computation. Boundary conditions must express the correct phenomena on the boundary, which is a surface that created by cutting of a region to select a computational domain. In fluid flows, a common boundary is a solid surface of the container, in which a fluid flows. The surface interacting with the fluid flow should be modelled as a boundary condition. In conventional method, such as finite difference and finite volume method, a boundary condition is given by the expressions having the macroscopic variables. In the LBM, however, boundary conditions are usually given in terms of the distribution function. The solid surfaces are considered either slip or non-slip surfaces depending on the roughness of the materials. In the LBM, the slip and non-slip boundaries are modeled by using a bounce forward and bounce back scheme (Guo & Shu, 2013) (A.A.Mohamad, 2011) (Jansen & Krafczyk, 2011), respectively. If a solid surface is moving relatively with a fluid flow, the Galileo's invariant need to be considered for the boundary condition (Alexander, 2008). Among the many boundary conditions with the different accuracy in the LBM, we present the widely used boundary conditions herein.

Bounce-back boundary condition

A bounce-back boundary condition is used to model rough surface, where the tangential velocity is zero due to the friction of the wall. Bounce-back boundary conditions are originated from the LGA (Wolfram, 1986) (Michael & Daneal T, 2006) and can be used to model both the stationary and moving surface. The main concept to fulfill this condition is to bounce back the distribution function on the surface to the direction that the distribution function comes from. Several versions of bounce back boundary conditions can be found in the literatures. We will review only two of them, namely half-way and full-way bounce back boundary condition, as shown in **Figure 37**.

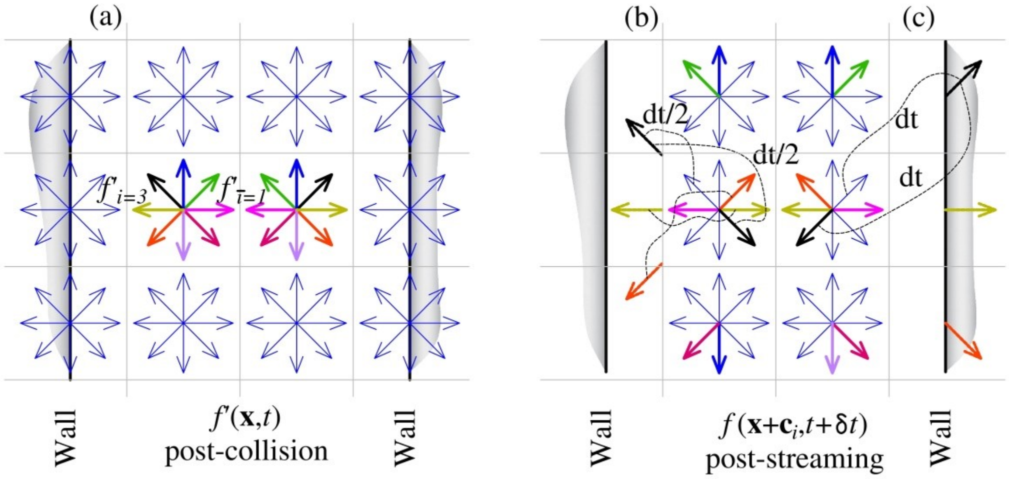


Figure 37. Bounce-back boundary condition: (a) post collision state, (b) half-way bounce-back scheme and (c) full-way bounce-back scheme.

After a collision process, the streaming process takes place to transfer the distribution functions to their designated directions. For a cell, which adjacent to the wall, as shown in **Figure 37** (a), some distribution functions, for example f_1, remain unknown during the streaming process. The unknown distribution functions of the fluid cell adjacent to the solid cell can be defined by the distribution functions that would transfer into the wall as follows

$$\text{Half}-\text{way: } f_{\bar{\imath}}(\mathbf{x}, t+\delta t) = f_i'(\mathbf{x}, t)\,, \tag{233}$$

$$\text{Full}-\text{way: } f_{\bar{\imath}}(\mathbf{x}, t+\delta t) = f_i'(\mathbf{x}+\mathbf{c}_i\delta t, t) \text{ or}$$
$$f_{\bar{\imath}}(\mathbf{x}, t+\delta t) = f_i(\mathbf{x}, t+\delta t)\,, \tag{234}$$

in which,

$$f_i'(\mathbf{x}, t) = f_i(x, t) + \frac{\delta t}{\tau_v}\left(f_i^{eq} - f_i\right) + \delta t A_i \tag{235}$$

is the distribution function after the collision process. Note that $\bar{\imath}$ is the inverse direction to *i*. The main difference between half-way and full-way scheme is time, as shown in **Figure 37** (b) and (c). In this thesis, we mainly use the half-way bounce back boundary condition to merit its simple implementation. In the literature, the half-way bounce back boundary condition is known as the mid-way (A.A.Mohamad, 2011), mid-plane (Michael & Daneal T, 2006) or mid-link bounce back scheme.

Bounce-forward boundary condition

A smooth surface can be modeled by using a bounce forward scheme, which reflects the distribution function as a mirror. The bounce forward boundary condition in the LBM is realized as a slip boundary condition. In this boundary condition, the tangential velocity of fluid does not become zero. Two types of the scheme can be used on a surface, as shown in **Figure 38**.

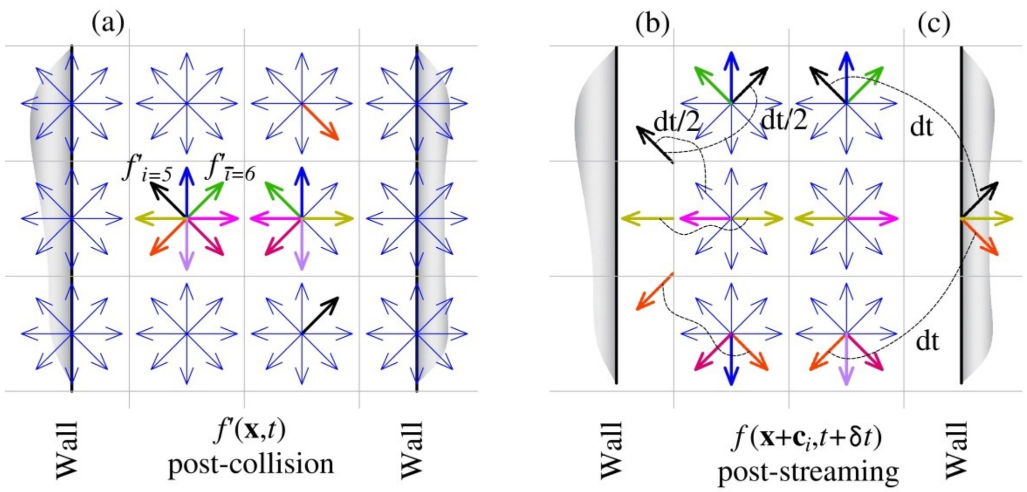


Figure 38. Bounce forward boundary condition: (a) post collision state, (b) half-way scheme and (c) full-way scheme.

The missing distribution functions on the fluid node adjacent to the solid surface are assigned in

$$\text{Half} - \text{way: } f_{\bar{\imath}}(\mathbf{x}, t+\delta t) = f_i'(\mathbf{x} - \mathbf{c}_i \mathbf{N}\delta t, t)\,, \tag{236}$$

$$\text{with } \mathbf{N} = \{\mathbf{n} = 0, \boldsymbol{\tau} = 1\},$$

where **n** and **τ** are the normal and tangential unit vectors on the surface. While the distribution functions on the solid surface can be defined in

$$\text{Full} - \text{way: } f_{\bar{\imath}}(\mathbf{x}, t+\delta t) = f_i(\mathbf{x}, t)\,, \tag{237}$$

where $\bar{\imath}$ is the mirrored direction to *i*. The full-way bounce forward condition has streaming and collision on the solid surface and collision on the surface performs the process of boundary condition.

Neumann boundary condition: The velocity is given

If the spatial derivation of the velocity on the boundary is prescribed, the velocity boundary condition can be assigned on the boundary. This boundary condition is known as the Zou/He boundary condition and can be used to model an inflow and outflow boundary. For D2Q9 lattice, constraints to construct the velocity boundary condition are the zeroth and first order discrete moments of the distribution function (Eq.(209)):

$$\rho = \sum_{i=0}^{8} f_i\,, \qquad \rho\mathbf{u} = \sum_{i=0}^{8} \mathbf{c}_i f_i \tag{238}$$

and the bounce-back rule for non-equilibrium part of the distribution function (A.A.Mohamad, 2011)

$$f_{\bar{\imath}}(x,t) - f_{\bar{\imath}}^{eq}(x,t) = f_i(x,t) - f_i^{eq}(x,t)\,. \tag{239}$$

Let the velocity on the boundary be

$$\mathbf{u} = \{u_x, u_y\}\,. \tag{240}$$

To demonstrate the derivation of the velocity boundary condition, the north surface of the two dimensional domain is considered as an inflow boundary, as shown in **Figure 39**.

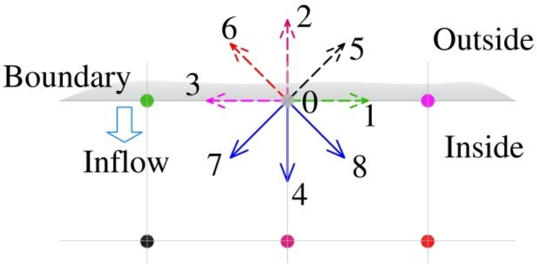


Figure 39. Boundary lattice on the north boundary of the computational domain.

Equations (238) and (239) with the known velocity give a following system of equations

$$\begin{cases} \rho_{in} = f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 & (a) \\ \rho_{in} u_x = f_1 - f_3 + f_5 - f_6 - f_7 + f_8 & (b) \\ \rho_{in} u_y = f_2 - f_4 + f_5 + f_6 - f_7 - f_8 & (c) \\ f_4 = f_2 - f_2^{eq} + f_4^{eq} & (d) \end{cases}, \tag{241}$$

where ρ_{in}, f_4, f_7 and f_8 are unknowns after a streaming process, as shown in **Figure 39**. Finding $f_4 + f_7 + f_8$ from (a) and (c), then equating them gives the first unknown, density

$$\rho_{in} = \frac{1}{1+u_y}[f_0 + f_1 + f_3 + 2(f_2 + f_5 + f_6)] \,. \tag{242}$$

In the determination of density, we notice that $f_2 + f_5 + f_6$ in parenthesis of Eq.(242) is the summation of the inverse distribution functions for the unknown distribution functions and $f_0 + f_1 + f_3$ is the summation of the distribution functions which are tangential to the boundary. Now we can define the second unknown, f_4, from (d) using the equilibrium distribution function formulation given in Eq.(193):

$$f_4 = f_2 - \frac{2}{3}\rho_{in} u_y \,, \tag{243}$$

where $-\frac{2}{3}\rho_{in} u_y$ is the only term remained from $f_4^{eq} - f_2^{eq}$. Finding f_8 from (b) and substituting it with Eq.(243) into (c) yields the third unknown

$$f_7 = f_5 + \frac{1}{2}(f_1 - f_3) - \frac{1}{6}\rho_{in} u_y + \frac{1}{2}\rho_{in} u_x \,, \tag{244}$$

where f_5 is the opposite distribution function to the unknown f_7 and, f_1 and f_3 are the tangential directed distribution functions on the boundary. Similarly, finding f_7 from (b) and substituting it with Eq.(243) into (c) yields the fourth unknown

$$f_8 = f_6 - \frac{1}{2}(f_1 - f_3) - \frac{1}{6}\rho_{in} u_y - \frac{1}{2}\rho_{in} u_x \,, \tag{245}$$

where f_6 is the opposite distribution function to unknown f_8 and $f_1 - f_3$ is the subtraction of the tangential directed distribution functions on the boundary. Those are telling us the boundary condition formulae have a similarity. We can apply those similarity conditions for the determination of the formulae for the velocity boundary conditions on the other boundaries. We give the velocity boundary conditions with their schemes on the boundaries in **Figure 40** to **Figure 42**.

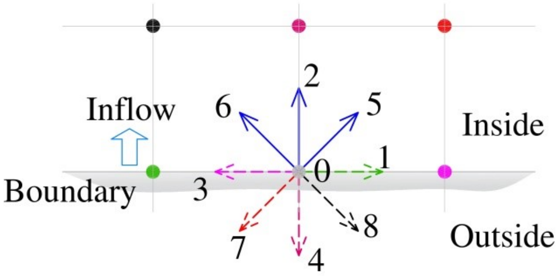


Figure 40. Boundary lattice on the south.

The velocity boundary condition on the southern boundary, **Figure 40**, can be defined as:

$$\rho_{in} = \frac{1}{1-u_y}[f_0 + f_1 + f_3 + 2(f_4 + f_7 + f_8)] \tag{246}$$

$$f_2 = f_4 + \frac{2}{3}\rho_{in}u_y \tag{247}$$

$$f_5 = f_7 - \frac{1}{2}(f_1 - f_3) + \frac{1}{6}\rho_{in}u_y + \frac{1}{2}\rho_{in}u_x \tag{248}$$

$$f_6 = f_8 + \frac{1}{2}(f_1 - f_3) + \frac{1}{6}\rho_{in}u_y - \frac{1}{2}\rho_{in}u_x\,. \tag{249}$$

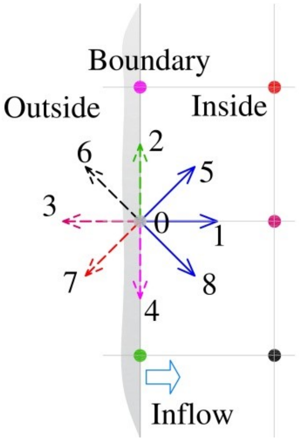


Figure 41. The boundary lattice on the west.

The velocity boundary condition on the west boundary, **Figure 41**, can be defined as

$$\rho_{in} = \frac{1}{1-u_x}[f_0 + f_2 + f_4 + 2(f_3 + f_6 + f_7)] \tag{250}$$

$$f_1 = f_3 + \frac{2}{3}\rho_{in}u_x \tag{251}$$

$$f_5 = f_7 - \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{in}u_x + \frac{1}{2}\rho_{in}u_y \tag{252}$$

$$f_8 = f_6 + \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{in}u_x - \frac{1}{2}\rho_{in}u_y\,. \tag{253}$$

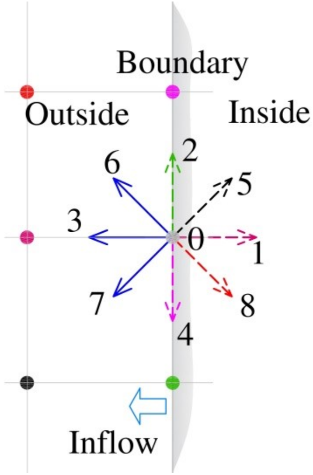


Figure 42. Boundary lattice on the east. Dashed distribution functions are known because the streaming process.

The velocity boundary condition on the east boundary, **Figure 42**, of the domain can be defined as

$$\rho_{in} = \frac{1}{1+u_x}[f_0 + f_2 + f_4 + 2(f_1 + f_5 + f_8)] \tag{254}$$

$$f_3 = f_1 + \frac{2}{3}\rho_{in}u_x \tag{255}$$

$$f_7 = f_5 - \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{in}u_x - \frac{1}{2}\rho_{in}u_y \tag{256}$$

$$f_6 = f_8 - \frac{1}{2}(f_2 - f_4) - \frac{1}{6}\rho_{in}u_x + \frac{1}{2}\rho_{in}u_y\,. \tag{257}$$

The sign of the terms with velocity, $\frac{1}{6}\rho_{in}u_x$ and $\frac{1}{2}\rho_{in}u_y$, mimics the direction of the unknown distribution function. For instance, in Eq.(257), an unknown is directed in the north-west direction and its vertical component has a positive sign while the horizontal component has a negative sign.

Dirichlet boundary condition: The pressure is given

Giving a pressure on the boundary sometimes leads a stable simulation in the LBM. As we see in Section 5.1.3, the pressure can be defined by the density through Eq.(155). With the prescribed density, we can construct missing distribution functions on the boundary as well as the velocity. The pressure boundary condition is usually given at the outlet boundary and we will derive it at the north boundary using **Figure 43**.

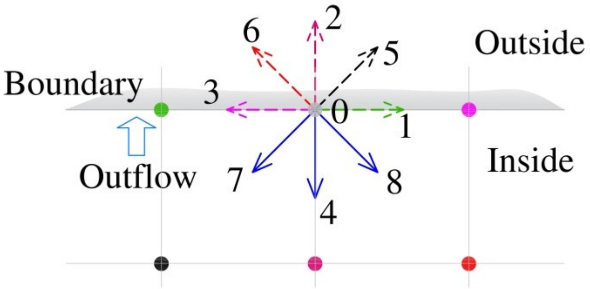


Figure 43. Boundary lattice on the north as outlet boundary.

A procedure to derive the pressure boundary condition is the same as that of the velocity boundary condition. But the velocity components are unknown and should be computed from the density. In addition, we have the three unknown distribution functions and all together we have four unknowns. If we assume that the tangential velocity on the boundary is adjusted to zero, we can formulate the pressure boundary condition using a system with the four equations. In case of non-zero tangential velocity on the boundary, the additional constraint can be given by Eq.(239) using the tangential directed distribution functions. The system of equations can be

$$\begin{cases} \rho_{out} = f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 & (a) \\ \rho_{out} u_x = f_1 - f_3 + f_5 - f_6 - f_7 + f_8 & (b) \\ \rho_{out} u_y = f_2 - f_4 + f_5 + f_6 - f_7 - f_8 & (c)\ . \\ f_4 = f_2 - f_2^{eq} + f_4^{eq} & (d) \\ f_1 - f_1^{eq} = f_3 - f_3^{eq} & (e) \end{cases} \tag{258}$$

To define u_y, the first unknown, $f_4 + f_7 + f_8$ should found from (a) and (b). Equating the results of $f_4 + f_7 + f_8$ gives the vertical component of the velocity vector

$$u_y = \frac{f_0 + f_1 + f_3 + 2(f_2 + f_5 + f_6)}{\rho_{out}} - 1\,. \tag{259}$$

The horizontal component of velocity vector can be computed from (e) as

$$u_x = (f_3 + f_1)\frac{3}{2}\,. \tag{260}$$

The third unknown, f_4 , is readily defined from (d) as before

$$f_4 = f_2 - \frac{2}{3}\rho_{out} u_y\,. \tag{261}$$

Now, we proceed to find f_8 from (b). Substituting the result with Eq.(261) into (c) gives the fourth

$$f_7 = f_5 + \frac{1}{2}(f_1 - f_3) - \frac{1}{6}\rho_{out} u_y - \frac{1}{2}\rho_{out} u_x\,. \tag{262}$$

Similarly, defining f_8 from (b) and substituting it with Eq.(261) into (c) gives the last unknown as

$$f_8 = f_6 - \frac{1}{2}(f_1 - f_3) - \frac{1}{6}\rho_{out} u_y + \frac{1}{2}\rho_{out} u_x\,. \tag{263}$$

The pressure boundary condition on the other boundaries can be formulated by the symmetry properties using the five equations (Eqs.(259)-(263)) above.

Outflow boundary conditions

Beside the pressure boundary condition, more simple boundary conditions can be assigned at the outflow boundary. These boundary conditions are adequate for the case that the velocity and pressure are unknown. One of them is the second order extrapolation boundary condition (the Neumann boundary condition) and it can be written by

$$f_i(\mathbf{x}_b, t) = 2f_i(\mathbf{x}_b + \mathbf{c}_i\mathbf{N}, t) - f_i(\mathbf{x}_b + 2\mathbf{c}_i\mathbf{N}, t)\,, \tag{264}$$

where $\mathbf{x}_b$ is a coordinate of the boundary cell and $\mathbf{N} = \{\boldsymbol{n} = 1, \boldsymbol{\tau} = 1\}$ is a unit vector on the boundary.

Another widely used boundary condition is the zero-gradient boundary condition and it can be given as

$$f_i(\mathbf{x}_b, t) = f_i(\mathbf{x}_b + \mathbf{c}_i\mathbf{N}, t)\,, \tag{265}$$

which is the first order accuracy.

Periodic boundary condition

A periodic boundary condition often used for the modeling infinite characteristic of the domain. It means that the periodic boundary condition provides opportunity to simulate flow in small domain of the large system.

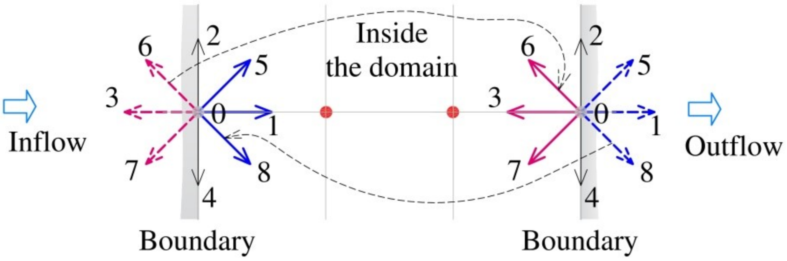


Figure 44. Periodic boundary condition on the inflow and outflow boundaries.

For instance, a periodic boundary condition is required on the inflow and outflow boundaries of the flow through infinite channel, as shown in **Figure 44**. The periodic boundary condition on the inflow boundary

$$f_i(\mathbf{x}_o, t + \delta t) = f_i(\mathbf{x}_N, t)\,, \quad i = 1, 5, 8 \tag{266}$$

and on the outflow boundary

$$f_i(\mathbf{x}_N, t + \delta t) = f_i(\mathbf{x}_o, t)\,, \quad i = 3, 6, 7 \tag{267}$$

where $\mathbf{x}_o$ and $\mathbf{x}_N$ are the coordinates of the inflow and outflow boundaries.

5.2.4 Stokes second problem: 1D flow

Consider that there are two parallel infinite plates that bound a fluid between them and the upper plate moves as the sinusoid oscillation as shown in **Figure 45**. The oscillating plate generates a laminar flow between the two infinite plates. The problem is called Stokes second problem (G. K. Batchelor, 1967) because the analytical solution of this problem was founded by George Gabriel Stokes in 1851. In the literatures, the problem is referred as oscillating Couette flow or Rayleigh-Stokes flow. We shall solve the laminar flow between two plates by using the FDM and LBM to compare numerical procedures of methods.

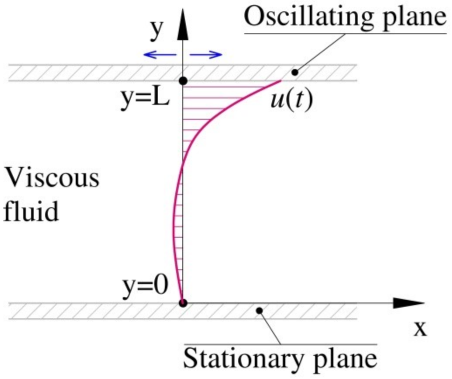


Figure 45. Schematic illustration of the Stokes second problem.

Let us write the NSE (Eq.(16)) in the x-direction

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=-\frac{1}{\rho}\frac{\partial P}{\partial x}+D\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right). \quad (268)$$

With the characteristic of the flow, depicted in **Figure 45**, spatial derivatives for the x- and z-direction can be neglected and only the time and the y-directional derivation remain:

$$\frac{\partial u}{\partial t}=-\frac{1}{\rho}\frac{\partial P}{\partial x}+D\frac{\partial^2 u}{\partial y^2}, \quad (269)$$

which is the governing equation of the Stokes second problem. The pressure term can be neglected or can be assigned as a constant term. The coefficient D can be viscosity for momentum or diffusion for heat transfer. Initial and boundary conditions on the plates are

$$u(y,0)=0 \text{ for } y>0 \text{ and} \quad (270)$$

$$u(L,t)=U \text{ for } t>0 \quad (271)$$

respectively. The velocity of the harmonic motion of the upper plane is given as

$$U(t) = A\, sin\left(\frac{2\pi}{T} t\right) , \tag{272}$$

where A ($= 1.0$ ms^{-1}) is the amplitude of harmonic motion, T (= 5 s) is the wavelength or period and t is time.

Simple Euler method

Taking the Euler approximation for the time derivative and the central difference scheme for the second order spatial derivative of Eq.(269) gives the finite difference equation for the problem:

$$u_j^{n+1} = u_j^n + \Delta t D \frac{u_{j+1}^n - u_{j-1}^n - 2u_j^n}{\Delta y^2} + \alpha \Delta t , \tag{273}$$

where u_j^{n+1} is the velocity in time advance $n + 1$ at j cell, Δt (=0.001 s) is the time step, Δy (=0.2 m) is a grid spacing, D (=20 m^{2}s^{-1}) is the diffusion coefficient and α is the constant pressure term.

Lattice Boltzmann method

We use well explained D1Q3 model for this problem. There is no discretization of governing equation in Eq.(269), instead we shall define the lattice Boltzmann equation with an appropriate equilibrium distribution function. The to-be-solved lattice Boltzmann equation is

$$f_i^{n+1}(j + c_{yi}\delta t) = (1 - w_v) f_i^n(j) + f_i^{eq}(j) w_v , \tag{274}$$

where $f_i^{n+1}(j)$ is the distribution function in time advance on j cell, c_{yi} is the y-component of discrete velocity of lattice, δt is the lattice time step, w_v is the relaxation parameter. Time step δt for the lattice is assumed to be unit to maintain the unit discrete velocity as well as the unit lattice spacing. Time step and spacing of the lattice are different from the computational time step, Δt, and grid spacing, Δy. The equilibrium distribution function, $f_i^{eq}(j)$, can be given as

$$f_i^{eq}(j) = w_i u(j) , \tag{275}$$

where w_i is the weights for the lattice. Please have a look at **Figure 36** (a) and refer to Section 5.2.2 to get the weights and discrete velocities. Equation (274) must be solved by two steps called the collision and streaming process, which are presented in the code given in **Table 3**. The relaxation parameter can be calculated from the diffusivity D according to Eq.(205), like

$$w_v = \frac{1}{c_s^2 D \dfrac{\Delta t}{\Delta y^2} + 0.5} , \tag{276}$$

where $c_s^2 = 2$. The time step Δt for the LBM can be defined from a velocity scaling between the physical and the LB variables as

$$U_R = U_L \frac{\Delta y}{\Delta t} , \tag{277}$$

where U_L is the dimensionless boundary velocity at the upper plate. As seen so far, all variables in the LBM must be dimensionless and after the simulation, they must be scaled appropriately just like Eq.(277). A velocity for new time step can be computed with the distribution functions

$$u_L^{n+1} = \sum_{i=0}^{3} f_i^{n+1}(j) \ . \tag{278}$$

Comparisons

With the same grid spacing, the LBM gives a stable simulation with $\Delta t = 0.01$, while the FDM uses $\Delta t = 0.001$. It means that the LBM method can be faster than the FDM. The simple Euler code was shorter than the LB code, as given in **Table 3**.

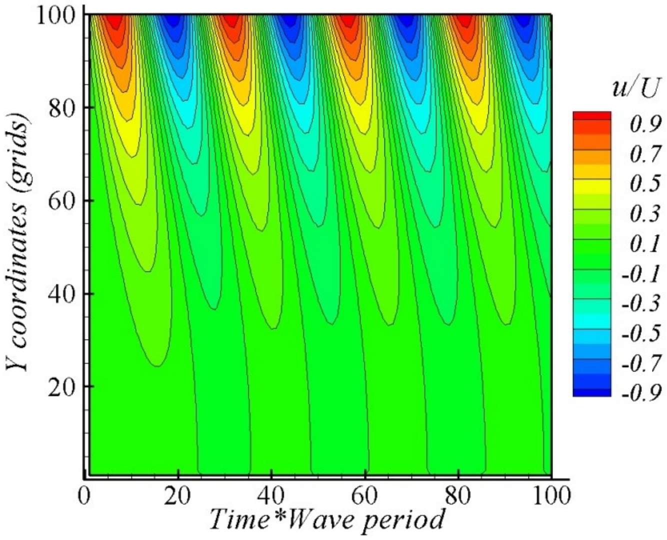


Figure 46. Numerical results for the Stokes second problem by Euler method (color) and LBM (isoline).

It is because that the LBM uses an additional variable named the distribution function except the physical variables of the problem and an additional scaling operation except the obligatory computation in numerical procedure.

As seen in the computer code, the Euler method uses three nodal stencils, while the LBM uses two nodal stencils for streaming and single node for other operations. This is an inherent advantage of the LBM for parallel computations.

The computation of the LBM took shorter time than that of the Euler method and the results are compared to each other in **Figure 46** and **Figure 47**. Total computational time was 20 s and velocity variables for each second are presented in **Figure 46**, where the color fields for the Euler method and the iso-line for the LBM. At specific times, all measured velocity profiles are compared in **Figure 47**, where we can observe some little discrepancies. Reducing time step in both methods increases the numerical accuracy.

Table 3. The comparison of main parts of the code for the Stokes second problem

Simple Euler method (FDM)

```
! initial conditions
    do j=1,m
      u(j)  =0. ! zero clear
      un(i)=0. ! zero clear
    end do

! main loop
    do n=1,nt !time starts

      time=n*dt

! boundary condition
      u(1  )=0.0
            un(1)=0.0
      u(m)=amp*sin(2.*pi*time/wp)
            un(m)=u(m)

    do j=2,m-1

    rh=cc*(u(j+1)+u(j-1)-2.*u(j))+alfa*dt
    un(j)=rh+u(j)    ! next time

    end do ! j

      u=un   ! for next time

    end do ! time
```

Lattice Boltzmann method (LBM)

```
!initial conditions
    do j=1,m
      u(j)=0.0
      do k=0,2
        f(k,j)=w(k)*u(j)
      end do
    end do

! main loop
    do i=1,nt !time starts
! streaming
    do j=m,1,-1
      f(1,j)=f(1,j-1)
      f(2,m-j)=f(2,m-j+1)
    end do
! boundary condition
      ul=amp_L*sin(2.0*pi*i/per_L)
      f(2,m)=ul*(w(1)+w(2)+w(0))-f(1,m)-f(0,m)
      f(1,1)=-f(2,1)
! macroscopic variable
    do j=1,m
      u(j)=f(1,j)+f(2,j)+f(0,j)
! collision
      f(0,j)=(1.-ome)*f(0,j)+ome*w(0)*u(j)
      f(1,j)=(1.-ome)*f(1,j)+ome*w(1)*u(j)
      f(2,j)=(1.-ome)*f(2,j)+ome*w(2)*u(j)
    end do

    end do ! time
```

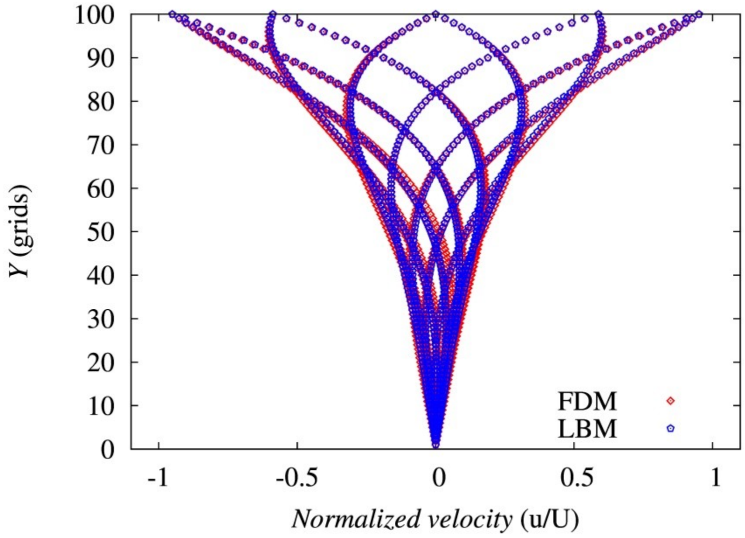


Figure 47. Comparison of the velocity profiles for the Stokes second problem measured at intervals of 0.1*T* and predicted by the FDM and LBM.

5.3 Lattice Boltzmann method for scalar transports

In the previous sections, we have discussed how the LBM solves fluid flows based on the mesoscopic modelling. Many problems of fluid flow involving transport or diffusion of another matter in a fluid encounter in engineering practices, such as heat transfer and sediment transport. Such problems are modeled by the scalar transport equation, e.g. Burgers' equation, ADE or CDE, or heat transfer equation, while fluid flow is modeled with the Navier-Stokes equation in conventional methods, as discussed in Part 1. This paradigm is used to model such a problem in the LB methods. In other words, while aforementioned LBM solves the flow field, another set of LB model, which we discuss in this section, needs to take a care for the transport phenomena.

First attempts for the scalar transport in the LBM were based on the LGA, where the macroscopic equations can be derived the contribution of the some magic and the LGA, itself, had a complexity to model such a problem (Wolf-Gladrow, 2000). Depending on the transporting matter, different approaches developed after 1990s. For instance, two types of approach, namely a multi-speed approach and a double-distribution function approach, have been simultaneously developed for heat transfer in fluid flows. While, for the convection-diffusion problem, double-distribution function or the coupled model with the conventional method was proposed (Guo & Shu, 2013). A multi-speed lattice Boltzmann was introduced as the direct extension of the general lattice Boltzmann method exploiting the higher order moments of the distribution function and exploring the thermal possibility of the Boltzmann equation. In contrast, a double-distribution function approach uses the second lattice Boltzmann equation to model scalar transport process and keeps the isothermal LBM for the fluid flow. One of the first double-distribution function models was proposed by Bartoloni et al in 1993 (Bartoloni, et al., 1993) to model the Rayleigh-Benard convection. In the model, the temperature was solved by the second LBE as a passive scalar and no temperature effect was modeled for the fluid flow. Another model of a double-distribution function approach was developed by Wolf-Gladrow in 1995 for the diffusion of substances in an arbitrary number of dimensions, which was the simplest LBM for diffusion problems (Wolf-Gladrow, 1995). More detailed research was done by Elton et al in 1995 (Elton, et al., 1995) exploring consistency, convergence, stability and efficiency of the LBM for the macroscopic scalar transport equations. The double-distribution function model was further extended by Shan in 1997 (Shan, 1997) to model the Rayleigh-Benard convection by considering the Boussinesq approximation on convection (generally discussed in Section 5.1.1), which was traced from their previous research about multi-component flow modeling with the LBM (Shan & Chen, 1993). Further contributions made to widen the application of the model and improve the accuracy, such as a double-distribution function method on an irregular lattice (Van der Sman & Ernst, 2000) and coupled methods with conventional method. A comprehensive review of the double-distribution function model for a passive scalar transport can be found in (Zhou, 2009) and (Huang, et al., 2011).

In this section, we will discuss the widely used double-distribution function model for scalar transport because of its applicability to the both problems in sediment transport and heat transfer, as objectives of this thesis. The macroscopic diffusion and CDE (ADE) will be derived from the LBE through the multi-scale expansion analysis in Appendix 2.

5.3.1 Lattice Boltzmann model for scaler transport

If we put tracing dye to a small river, dye will be transported by flowing water into downriver and will spread out to invade all the cross-section of the river flow as diffusion. In the nature, instead of dye there are number of transporting substances in rivers like waste, sediment and dissolved matter or heat. Physical interaction with flow field and quantity of substances in a fluid is governed by the ADE:

$$\frac{\partial \phi}{\partial t} + u_\alpha \frac{\partial \phi}{\partial x_\alpha} = D \frac{\partial^2 \phi}{\partial x_\alpha^2}\,, \tag{279}$$

where ϕ is the dependent variable of a substance and D is the diffusion coefficient, which can be affected by the molecular diffusion, turbulent mixing, etc. The word advection can be interchanged with the convection depending on the use of Eq.(279). This ADE is a combination of the simple-wave equation and the diffusion equation, but mathematically it has the properties of the latter one (Vreugdenhil, 2012). If we introduce a variable $K = D - ux$ for the x-component of Eq.(279), we get the diffusion equation

$$\frac{\partial \phi}{\partial t} - K \frac{\partial^2 \phi}{\partial x^2} = 0\,, \tag{280}$$

which means the diffusion in a frame of reference moving with the mean flow. On the other point, if the flow velocity is zero: no motion of fluid is observed in the system, Eq.(279) gives diffusion equation, directly

$$\frac{\partial \phi}{\partial t} = D \frac{\partial^2 \phi}{\partial x_\alpha^2}\,. \tag{281}$$

The dependent variable of the substance can have a unit of concentration for sediment or pollutant wastes and temperature in the heat transfer. The advection, diffusion and advection-diffusion terms are described in **Figure 48**.

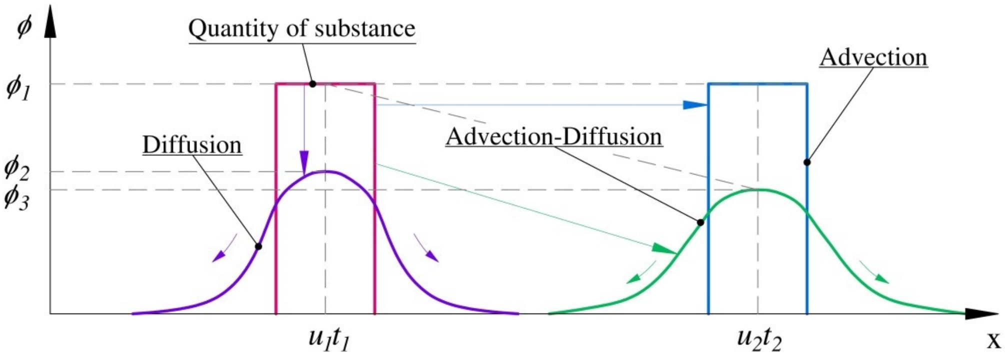


Figure 48. Diffusion, advection (convection) and the advection-diffusion phenomena.

The LBE for a scalar transport problem is the same as the LBE without the force term used in a fluid flow in Eq.(204),

$$g_i(\mathbf{x} + \delta x, t + \delta t) - g_i(\mathbf{x}, t) = \frac{\delta t}{\tau_s}\left(g_i^{eq} - g_i\right), \tag{282}$$

where $g_i(\mathbf{x}, t)$ is the distribution function for a scalar variable, τ_s is the dimensionless relaxation time regarding with the diffusion coefficient, D. The distribution function to accommodate the advection-diffusion and diffusion effect can be

$$g_i^{eq} = \phi w_i \left[1 + \frac{\mathbf{c}_i \cdot \mathbf{u}}{c_s^2}\right] \text{ and } g_i^{eq} = \phi w_i, \tag{283}$$

respectively, where θ is the dimensionless quantity for the dependent variable, which can be computed by

$$\phi(\mathbf{x}, t) = \sum_{i=0}^{N} g_i(\mathbf{x}, t) \ , \tag{284}$$

where N is the number of velocities in a lattice. The speed of sound, c_s, takes value of $1/\sqrt{2}$ for D1Q2, D2Q4 and D3Q6, and $1/\sqrt{3}$ for D1Q3, D2Q5, D2Q9 and D3Q15. Based on the symmetry properties of the lattice, D2Q4 and D3Q6 models can produce the same results like the use of D2Q9 and D3Q15 lattices, respectively. For the simplicity that comes from the shareable properties with fluid flow modeling by the LBM, we use D2Q9 for the two-dimensional scalar transport problem. For D2Q9 arrangement, the relaxation time can be determined by

$$D = \delta t \frac{\Delta x^2}{\Delta t}\left(\tau_s - \frac{1}{2}\right) c_s^2 \ , \tag{285}$$

which is just the same as Eq.(205). If the diffusion coefficient is dimensionless, then space and time steps are removed from the relation in Eq.(285).

For a non-passive substances transporting with a fluid flow, the effect of the substances on the fluid flow must be accounted as a force term for the governing equation of fluid. For instance, during the heat transport, the density deviates as temperature changes. The density deviation generates the local inertia force for the fluid and it can be approximated by the Boussinesq approximation for the LBM, as using the same concept described in Section 2.1.1. The force induced by the density variation due to the temperature changes is presented in Section 5.6.2.

5.3.2 Boundary conditions for scaler transport

Boundary conditions for scalar field evolution in the LBM are essential to get a good result from the simulation. From boundary conditions discussed for a flow field in Section 5.2.3, the outflow and periodic boundaries can be applied directly, if the boundary requires such treatments. We shall discuss some widely used boundary conditions in following contexts.

Dirichlet-type boundary conditions - 1

We derive a boundary condition when the scalar value on the boundary, which can be a solid surface or an inflow/outflow, is known. A schematic illustration of the boundary is given in **Figure 49** and we will not discuss the derivation or statement of the conditions for other boundaries.

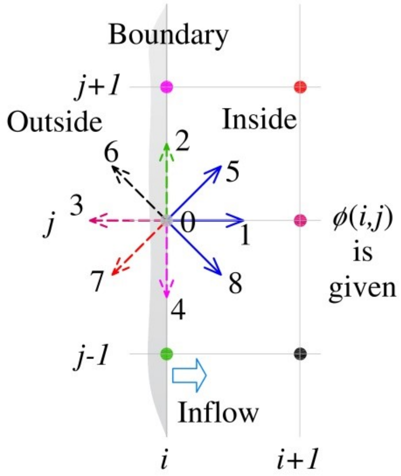


Figure 49. Inflow boundary at the west. Scalar variable at inflow boundary is given.

It is obvious for the boundary lattice

$$\sum_{i=0}^{N=8} g_i(i,j) = \phi(i,j)\,, \tag{286}$$

where $\phi(i,j)$ is the known scalar variable. From **Figure 49**, the missing distribution functions after the streaming process are g_1, g_5 and g_8, which can be defined with the weights as

$$g_1 = w_1\phi_r\,, g_5 = w_5\phi_r \text{ and } g_8 = w_8\phi_r\,, \tag{287}$$

where ϕ_r is the residual value to satisfy Eq.(286) after the evaluation of the boundary condition. With Eq.(287), the residual value can be calculated as

$$\phi_r = \frac{\phi - (g_0 + g_2 + g_3 + g_5 + g_6 + g_7)}{w_1 + w_5 + w_8}\,. \tag{288}$$

Using this residual value, we can construct the missing distribution functions with Eq.(287) (Michael & Daneal T, 2006).

Dirichlet boundary condition - 2

When the scalar value of the boundary, the same boundary in **Figure 49**, is given, we can derive the boundary condition based on the detailed flux conservation equation (A.A.Mohamad, 2011). For the normal to the boundary, the detailed flux conservation is

$$g_1^{eq} - g_1 + g_3^{eq} - g_3 = 0\,, \tag{289}$$

which gives the unknown distribution function g_1 as

$$g_1(\mathbf{x}_o, t+\delta t) = w_1\phi + w_3\phi - g_3 = \phi(w_1 + w_3) - g_3\,, \tag{290}$$

where the equilibrium distribution function, $g_i^{eq} = w_i\phi$, was used. Using the first expression in Eq.(283) gives the same result. The other two distribution functions are defined as

$$g_5(\mathbf{x}_o, t+\delta t) = \phi(w_5 + w_7) - g_7 \text{ and} \tag{291}$$

$$g_8(\mathbf{x}_o, t+\delta t) = \phi(w_8 + w_6) - g_6\,. \tag{292}$$

Dirichlet boundary condition - 3

Another possible way to derive a Dirichlet boundary condition is to use a bounce-back condition for the non-equilibrium part of distribution functions. For the diagonal distribution functions, the bounce-back to the non-equilibrium distribution function, using **Figure 49**, is

$$g_5 - g_5^{eq} = g_7 - g_7^{eq} \; . \tag{293}$$

The unknown, g_5, is

$$g_5 = g_7 + g_5^{eq} - g_7^{eq} \; . \tag{294}$$

Substituting the distribution function for the advection-diffusion problem into Eq.(294) yields

$$g_5 = g_7 + 2w_5\phi\frac{\mathbf{c}_5\mathbf{u}}{c_s^2} \; . \tag{295}$$

Similarly, we can state the boundary condition as

$$g_{\bar{\imath}}(\mathbf{x}_o, t+\delta t) = g_i(\mathbf{x}_o, t) + 2w_{\bar{\imath}}\phi\frac{\mathbf{c}_{\bar{\imath}}\mathbf{u}}{c_s^2} \; . \tag{296}$$

Dirichlet boundary condition - 4

A boundary condition based on the Ladd's boundary condition (Zhang, et al., 2012) is given as

$$g_{\bar{\imath}}(\mathbf{x}_o, t+\delta t) = -g'_i(\mathbf{x}_o, t) + 2w_i\phi \, , \tag{297}$$

which is a similar boundary condition to the Dirichlet boundary condition – 2, but uses a post-collision distribution function.

Neumann boundary condition - 1

We only give the derivation of the boundary condition on **Figure 49** and the boundary conditions on the other boundaries can be defined by using the symmetry condition. For the Neumann boundary conditions, the scalar flux at the boundary is prescribed and given as

$$-D\frac{\partial\phi}{\partial x} + u_x\phi = q \, , \tag{298}$$

where the first term is a diffusion flux, the second term is an advection flux and the flux value q is given.

The first Neumann boundary condition can be derived when the advection flux is ignored on the boundary. Thus, the finite difference approximation of Eq.(298) without the advection flux is

$$-D\frac{\phi(i+1,j) - \phi(i,j)}{\Delta x} = q \, , \tag{299}$$

Defining $(i+1,j) = g_1(i+1,j)/w_1$, $\phi(i,j) = g_1(i,j)/w_1$ and substituting into Eq.(299) yields

$$g_1(i,j) = g_i(i+1,j) + \frac{q}{w_1 D} \, , \tag{300}$$

where the flux and diffusion coefficients is dimensionless and spacing Δx is omitted. The boundary condition can be written as

$$g_{\bar{\imath}}(\mathbf{x}_o, t+\delta t) = g_{\bar{\imath}}(\mathbf{x}_o + \mathbf{c}_{\bar{\imath}} \cdot \mathbf{n}, t) + \frac{q}{w_{\bar{\imath}} D} \,. \tag{301}$$

Neumann boundary condition - 2

If the diffusion flux is zero, the advection flux is remained to be a flux at the boundary:

$$u_x \phi = q \,. \tag{302}$$

We can state that the advection flux can be defined by the moments of the distribution functions

$$\sum_{i=0}^{N=8} g_i \mathbf{c}_i \cdot \mathbf{n} = q \,. \tag{303}$$

Reusing **Figure 49** and substituting Eq.(287) into Eq.(303) gives

$$\phi_r = \frac{q + g_3 + g_6 + g_7}{w_1 + w_5 + w_8} \,, \tag{304}$$

which can be embedded in Eq.(287) that gives the unknown distribution functions. If we prescribe $q = 0$, we will get the zero flux boundary condition.

Neumann boundary condition - 3

If the diffusion and advection fluxes are considered and an resulting expression after the finite difference approximation for Eq.(298) is

$$\phi(i,j) = \frac{q - D\phi(i+1,j)}{u_x - D} \,, \tag{305}$$

which leads the boundary condition on the west boundary

$$g_{\bar{\imath}}(\mathbf{x}_o, t+\delta t) = \frac{q - D g_{\bar{\imath}}(\mathbf{x}_o + \mathbf{c}_{\bar{\imath}} \cdot \mathbf{n}, t)}{u_x(\mathbf{x}_o, t) - D} \quad \text{, where } \bar{\imath} = 1, 5, 8. \tag{306}$$

Adiabatic boundary condition

If the scalar gradient near the boundary is zero, an adiabatic boundary condition is required. The condition,

$$\frac{\partial \phi}{\partial x} = 0 \,, \tag{307}$$

can be approximated using the Euler method:

$$\phi(i+1,j) - \phi(i,j) = 0 \,. \tag{308}$$

Recalling the zeroth order moment of the distribution function for Eq.(308), the condition becomes

$$\sum_{i=0}^{N=8} g_i(\mathbf{x}_o, t+\delta t) = \sum_{i=0}^{N=8} g_i(\mathbf{x}_o + \mathbf{c}_{\bar{\iota}} \cdot \mathbf{n}, t) \,, \tag{309}$$

where $\mathbf{n}$ is the normal vector to the boundary. Simply, an adiabatic boundary condition is fulfilled by copying the all distribution functions on the adjacent lattice into the boundary lattice.

Anti-bounce-back boundary condition

To maintain the zero scalar value on the boundary, an anti-bounce-back boundary condition is served as

$$g_{\bar{\iota}}(\mathbf{x}_o, t+\delta t) = -g_i(\mathbf{x}_o, t) \,, \tag{310}$$

which is a case of the Dirichlet boundary condition - 2, if the known scalar value is zero on the boundary. Note that all boundary conditions presented above is based on **Figure 49** and the most of them is stated for the west boundary as the inflow/outflow.

5.3.3 Example of scalar transport problem

Let's have a simple numerical example for the scalar transport by the LBM. Consider now the release of a quantity of salt in a channel, 400 m in length, in which there is a flow defined by a sinusoidal wave. With the given condition, let us compute the advection and diffusion of the salt concentration through the channel. We will use the simple FDM and LBM with D1Q3 lattice in the one dimensional channel.

Finite Difference method

The one dimensional FDE for the advection-diffusion problem reads

$$\phi_i^{n+1} = \phi_i^n + \Delta t D \frac{\phi_{i+1}^n - 2\phi_i^n + \phi_i^n}{\Delta x^2} - \Delta t u_x \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x} \,, \tag{311}$$

where ϕ_i^n is the concentration of salt, D (=2.0 m^{2}s) is the diffusion coefficient and u_x is the channel velocity computed from Eq.(272). We choose the time step to be $\Delta t = 0.01$ s to ensure the stable simulation and the grid spacing is $\Delta x = 4.0$ m. The total simulation time is given as 1000 s, which includes 5 phases of the velocity field. An initial condition for the concentration of salt is

$$\phi^{n=1}(\mathbf{x}) = \begin{cases} 1 & 180\text{ m} < \mathbf{x} < 220\text{ m} \\ 0 & \text{otherwise} \end{cases} \,. \tag{312}$$

Lattice Boltzmann method

The lattice Boltzmann equation for the scalar field is Eq.(282) and we can write it in streaming and collision steps as follows

$$\text{Collision: } g'_i(\mathbf{x}, t) = g_i(\mathbf{x}, t) + \frac{1}{\tau_s}\left(g_i^{eq} - g_i\right) , \tag{313}$$

$$\text{Streaming: } g_i(\mathbf{x}+\delta x, t+\delta t) = g'_i(\mathbf{x}, t)\,. \tag{314}$$

where τ_s will be defined by using Eq.(276). Using the same spacing for space discretization, we find the time step $\Delta t = 0.4$ s for the LBM. The equilibrium distribution function is given by the first expression in Eq.(283).

Comparisons

The results comparison is given in **Figure 50**. In the first figure, the horizontal axis shows the channel length while the vertical axis shows the time advances, where we have selected 5 different times and have plotted the concentration of them in the second figure.

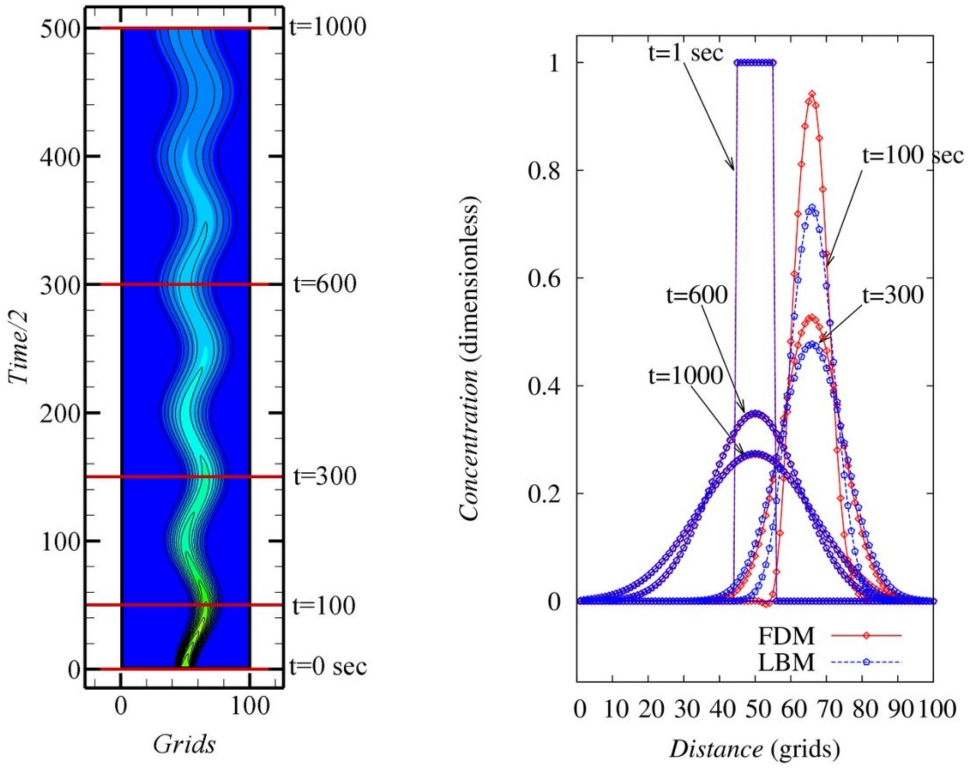


Figure 50. Solution of the simple advection-diffusion problem by the FDM and LBM.

Color gradient shows the result of the FDM and solid lines show the result of the LBM in the first plot. Interestingly, the results of the two methods were different until $t = 400$ s as shown in the profile of the $t = 100, 300$ in the second plot and after that the results matched each other until the end of the simulation. The FDM was overestimated and instability was observed on the left edge of the distribution, as seen at $t = 100$. To get a more accurate result, we can reduce time step in the FDM. The higher order FDMs are available, but those are supposed to be not adequate to be compared with the second order accuracy of LBM. If we use a bigger time step for the FDM, e.g. the time step as the same as one used for the LBM, the simulation will be unstable. The LBM gives rather stable and faster simulation than the DFM on the same grid discretization.

5.4 Turbulence and stability

Here we discuss about a numerical stability of the LBM with turbulent models or extensions of the LBM, because the turbulence has been a challenge of stability for the LB computation.

5.4.1 Numerical stability of lattice Boltzmann method

Numerical stability is a challenging aspect of the LB models. Particularly, the standard LBM, which we have discussed so far, is suffering from instability in a simulation at a high Reynolds number. Numerical stability is sometimes related to a numerical accuracy, since the standard LBM is a second order accurate method and the LB equation can recover the NSE in a limited condition of a low Mach number. The certain two reasons to become instability of LB computations can be the discretization and the collision operator. Although, the LBM uses a flexible Boltzmann equation with the linear collision operator, the discretization of the Boltzmann equation uses a finite difference approximation in space and time. Therefore, the computation of LBE has a characteristic of the finite difference scheme and error of the space discretization. The second reason is the definition of the relaxation time, which is defined in a result of the multi-scale expansion to the macroscopic equations. The relation between the relaxation time and molecular viscosity can be written as

$$\nu_R = \frac{\Delta x^2}{\Delta t}\left(\tau_\nu - \frac{1}{2}\right) c_s^2\ , \tag{315}$$

where ν_R [m^{2}s^{-1}] is the molecular viscosity of a fluid, Δt [s] and Δx [m] are the grid spacing and time step. To ensure the stability in the first glance, the relaxation time τ_ν must be greater than 0.5. The lower limit of relaxation time, $\tau_\nu > 0.5$, leads a zero lattice viscosity of fluid, as shown in **Figure 51**.

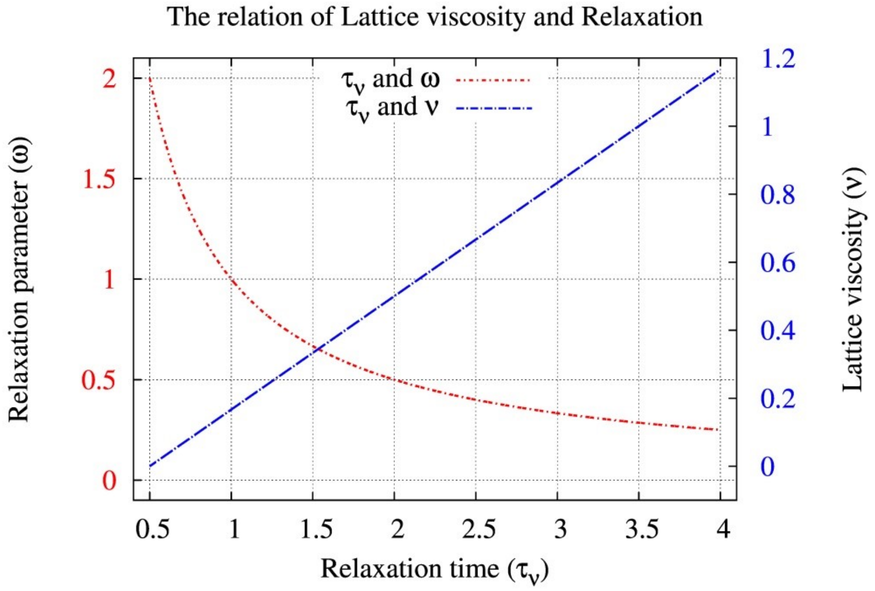


Figure 51. Relation between relaxation time and lattice viscosity/relaxation parameter.

In respect to a relaxation parameter, $w = \frac{1}{\tau_\nu}$, the stability limit is $0 < w < 2$. A turbulent nature of flow is often found in a low viscosity, which leads instability for the numerical scheme (Sterling & Shiyi, 1996). Let us see how the stability can be related to the other parameters in Eq.(315). Dividing Eq.(315) by UL, being characteristic velocity and length, respectively, is

$$\frac{\nu_R}{UL} = \frac{\Delta x}{\Delta t}\left(\tau_\nu - \frac{1}{2}\right)\frac{c_s}{U}\frac{\Delta x}{L}c_s\,, \tag{316}$$

where the left hand side is the Reynolds number, Re, U/c_s is the Mach number, Ma, $L/\Delta x$ is the grid number, N, and $\Delta x/\Delta t$ is the unit velocity as U/U_L. Assigning the appropriate terms in Eq.(316) and writing it in incompressible condition become

$$Ma = Re\frac{U}{U_L}\frac{c_s}{N}\left(\tau_\nu - \frac{1}{2}\right) < 0.2\,, \tag{317}$$

where Re and c_s are constants and U and U_L are proportional to each other. To ensure the incompressible condition in Eq.(317) at a certain Re number, we have options to increase the grid number as a decrease of velocity and relaxation time or their combination. However, the relaxation time has the lower bound of magnitude. The lattice viscosity can positively be conditioned by the lattice Boltzmann equation (Qian, et al., 1992) as

$$\nu \geq \mathrm{Max}\left[0,\ \underset{for\ f_i > f_i^{eq}}{\mathrm{Sup}}\left(\frac{f_i - f_i^{eq}}{f_i} - \frac{1}{6}\right)\right]. \tag{318}$$

Practically, the velocity, U_L, used in the LBM simulation is suitable to be less than 0.2 (A.A.Mohamad, 2011). Similarly, another stability condition requires the mean flow velocity in computation to be below a maximum stable velocity that is a function of several parameters (Chen & D.Doolen, 1998), including sound speed, the relaxation time, and the wave number, which defined by the linear von Neumann analysis for the LB equation (Sterling & Shiyi, 1996). Based on the same concept, adaptive time step and grid refinement techniques had been introduced to perform stable simulations with the LBM (Thürey, et al., 2006). Using a small time step or fine grid for a discretization, as well as an adaptation of grid, can be said the direct numerical simulation (DNS) with the LBM for turbulent flows.

There are many other methods to improve stability of the LBM, for instance the use of an irregular grid or the higher order finite difference scheme for a discretization (McNamara, et al., 1995), the use of two relaxation time (TRT) or multiple-relaxation time (MRT) for the collision term, contributions of turbulence models and conditions of Entropy or Boltzmann $\boldsymbol{H}$-theorem and the use of different collision operator based on moments or microscopic modeling, like the discrete unified gas-kinetic scheme (Guo, et al., 2013).

5.4.2 Stabilization techniques with relaxation times

Among the other stabilization techniques, we will briefly discuss the techniques based on the relaxation time, which causes the diversity for LBMs. It is stated that two-relaxation time (TRT) and multi-relaxation time (MRT) schemes for the collision term brings not only stability, but also accuracy and efficiency (Luo, et al., 2011) over the single-relaxation time scheme (SRT), which is the BGK model. Please note that the BGK and SRT are the same models in this context.

The MRT scheme is developed as the same time as the BGK model introduced and is involved by the all possible degrees of freedom to optimize the LBE. Unlike the BGK model, which uses the same relaxation rate for a collision, the MRT uses multiple relaxation times and different eigenvalue with the collision matrix (Guo & Shu, 2013). Successful application of the MRT spreads over the problems which cannot be solved by the SRT (d'Humières, 2002) without the improvement.

The TRT scheme decomposes the BGK collision operator into the solution of the symmetry and anti-symmetry components (Ginzburg, 2005). Like the SRT, fluid viscosity, both the bulk and molecular viscosity, is related to the symmetry eigenvalue (relaxation parameter), while the coefficient of the diffusion tensor is respected with the anti-symmetry eigenvalue. The eigenvalues considered in the TRT model are the most important two relaxation rates (Luo, et al., 2011). If the symmetry and anti-symmetry eigenvalues are equal to each other, the scheme reduces the SRT scheme. If the additional collision freedom is involved for the TRT, one derives the MRT scheme (Ginzburg, et al., 2008). None of those models, the TRT and MRT, is elaborated in further and is used for the simulations presented herein.

5.4.3 Turbulence modeling with Large Eddy Simulation

As we stated that solving a turbulent flow with the LBM often leads the lower limit of the relaxation time. There are two possible ways to model turbulent flows with the standard LBM (Hou, et al., 1996). The first way might be the revisions into the small scale fluctuation in the LGA and the idea to use it in large scale resolved flows by the LBM. Unfortunately, no attempt has been made in this way, since it is difficult to connect the small scale and large scale dynamics based on the microscale particle nature. The second way is to introduce or adapt the traditional turbulence model into the LBM. Based on the similar properties, the sub-grid scale models can be easily introduced into the LBM. The basic concept of the sub-grid scale model is considered the models, which account the effect of the small scale eddies in resolved large scales. Smaller scales in a fluid flow are ignored in the model during the computation to reduce the cost with the help of a spatial filtering function for the variables. Among the models, the simple Smagorinsky model (Deardorff, 1970), described in Section 2.2, uses a positive eddy viscosity to represent the small scale energy damping and is successfully introduced in the framework of the standard LBM. From the numerical analysis point of view, the Smagorinsky model in LBM (Hou, et al., 1996) brings the stable numerical scheme rather than resolved turbulent flow.

In the some literature, the LBMs incorporated with the LES model or k-epsilon models are termed as the extended LBM due to its extension to the high Reynolds number flow (Liu, et al., 2008), (Chen, et al., 2003). The distribution function and the equilibrium distribution function are related to the large scale flow and the resolved solution by the standard LBM is available to define the local unresolved small scale effects on the fluid flow, which is represented by the effective eddy viscosity. In order to define the eddy viscosity term and to evaluate the effect of small scale eddies on the fluid flows, the filtered distribution function is introduced as

$$\bar{f_i}(\mathbf{x}) = \int f_i(\mathbf{x}) G(\bar{\mathbf{x}}, \mathbf{x}') dx' \ , \tag{319}$$

where G is the spatial filtering function depending on the filter width in space. It is stated that the multi-scale expansion with the filtered distribution function, $\bar{f}_i(\mathbf{x})$, reduces the NSE with the total viscosity, including the eddy viscosity, and the macroscopic variables resulting $\bar{\rho} = \sum_i \bar{f}_i$ and $\overline{\rho\mathbf{u}} = \sum_i \bar{f}_i \mathbf{c}_i$. Eventually, it is equivalent to replace the filtered variables with the unfiltered variables and again the lattice Boltzmann equation reads

$$f_i(\mathbf{x} + \mathbf{c}_i\delta t, t + \delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau_{tot}}\left(f_i - f_i^{eq}\right) + \delta t A_i\,, \tag{320}$$

where τ_{tot} is the total relaxation time with respect to the addition of the molecular viscosity and the eddy viscosity (Hou, et al., 1996). The total viscosity can be found by the Smagorinsky model as

$$\nu_{tot} = \nu + \nu_{ed} = \nu + C\Delta^2|\bar{S}|\,, \tag{321}$$

where ν_{ed} is the eddy viscosity, C ($= C_s^2$ in Eq. (31)) is the Smagorinsky constant, Δ is the filtering width, which can be identified as a mesh resolution or a geometric average of grids spacing, and $|\bar{S}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}$ is the magnitude of the large scale strain rate tensor. The total viscosity gives the total relaxation time as the follow,

$$\tau_{tot} = \frac{1}{c_s^2}\left(\nu + C\Delta^2|\bar{S}|\right) + \frac{1}{2}. \tag{322}$$

The local magnitude of the strain tensor can be easily defined by the non-equilibrium stress tensor and the molecular viscosity as

$$|\bar{S}| = \frac{\sqrt{\nu^2 + 18C\Delta^2\left(P_{\alpha\beta}P_{\alpha\beta}\right)^{1/2}} - \nu}{6C\Delta^2}\,. \tag{323}$$

The second order moments of the non-equilibrium part of the distribution functions is

$$P_{\alpha\beta} = \sum_i c_{i\alpha}c_{i\beta}\left(f_i - f_i^{eq}\right)\,. \tag{324}$$

In the different form (Liu, et al., 2008), the total relaxation time can be defined as

$$\tau_{tot} = \frac{1}{2}\left(\sqrt{\tau_\nu^2 + \frac{18}{\Delta x^2\rho}C\Delta^2\left(2P_{\alpha\beta}P_{\alpha\beta}\right)^{1/2}} + \tau_\nu\right), \tag{325}$$

where Δx is the grid spacing and τ_ν is the relaxation time with respect to the molecular viscosity. Conceptually, the Smagorinsky constant, C, can be defined by the resolved Reynolds stress term using the test filter width, however, the recommended values for it is often used. In traditional methods, for instance in Part 1, the recommended values for the constant is $C = 0.01\sim0.04$, while for the LBM, this values is suggested as $C = 0.0025\sim0.0625$ in channel with the vegetation (Gac, 2014), $C = 0.0169\sim0.0289$ for the near wall turbulence (Wang, et al., 2014) and $C = 0.006$ for the air flow (Fernandino, et al., 2009). Our research with Eq.(322) shows the constant likely to be $C = 0.01 - 0.03$ (Ayurzana, 2016) in water flow.

5.4.4 Advanced fluid-kinetic model: Entropic LBM

One of the promising methods in the stabile LB turbulent computation is the entropic LBM (ELBM) of Karlin's group (Gu, et al., 2014), in which, the distribution functions are straightened to satisfy the maximum condition of the entropy at the every time step of a simulation. But this ELBM is based on the advanced fluid-kinetic theory for the low-dissipative hydrodynamic flow (Karlin, et al., 2014), where the Gibbs' principle with the maximizing the entropy is used to derive a lattice kinetic theory for the turbulent flow. There are several versions of the entropic LBM reported (Karlin, et al., 2014).

Based on the Boltzmann ***H***-theorem (Theorem 2 in Section 5.1.4), a simple entropic stabilizer was proposed to stabilize a LBM simulation at a high Reynolds number in an incompressible flow. The entropy of the LBE in D2Q9 lattice can be written as

$$S[f] = -\sum_{i=0}^{8} f_i \, ln\left(\frac{f_i}{w_i}\right), \tag{326}$$

where f_i is the density distribution function, w_i is the weight and $S[f]$ is the entropy. Further, we write the f_i as a component of the velocity moments:

$$f_i = k_i + s_i + h_i, \tag{327}$$

where k_i is the kinematic part of populations, which depends only on the locally conserved fields, s_i is the shear part of populations, which depends on the stress tensor and h_i is the higher order moments of populations as a linear combination of the remaining higher order moments. Introducing deviations $\Delta s_i = s_i - s_i^{eq}$and $\Delta h_i = h_i - h_i^{eq}$, the extremum condition of the $S[f]$ take the form:

$$\sum_{i=0}^{8} \Delta h_i \ln\left(1 + \frac{(1-\beta\gamma)\Delta h_i - (2\beta - 1)\Delta s_i}{f_i^{eq}}\right) = 0, \tag{328}$$

where β can be defined from the kinematic viscosity as $\nu = c_s^2\left(\frac{1}{2\beta} - \frac{1}{2}\right)$ and the maximum of stabilizer γ can be computed as

$$\gamma^* = \frac{1}{\beta} - \left(2 - \frac{1}{\beta}\right)\frac{\langle \Delta s | \Delta h \rangle}{\langle \Delta h | \Delta h \rangle} \text{ with } \langle X | Y \rangle = \sum_{i=0}^{8} \frac{X_i Y_i}{f_i^{eq}}. \tag{329}$$

Non equilibrium part of the distribution function can be written as

$$f_i - f_i^{eq} = \Delta s_i + \Delta h_i, \tag{330}$$

and it is used for the calculation of higher order moments Δh_i. The deviations of shear part of the populations are defined by following velocity moments:

$$\begin{cases} \Delta s_0 = \rho\left(4uv\widetilde{\Pi}_{xy} - \left[\dfrac{u^2 - v^2}{2}\right]\widetilde{N}\right), \\ \Delta s_{1,3} = \dfrac{\rho}{2}\left(\left[\dfrac{1 + c_{ix}u + u^2 - v^2}{2}\right]\widetilde{N} - [2c_{ix}v + 4uv]\widetilde{\Pi}_{xy}\right), \\ \Delta s_{2,4} = \dfrac{\rho}{2}\left(\left[\dfrac{-1 - c_{iy}v + u^2 - v^2}{2}\right]\widetilde{N} - \left[2c_{iy}u + 4uv\right]\widetilde{\Pi}_{xy}\right), \text{ and} \\ \Delta s_{5,6,7,8} = \dfrac{\rho}{4}\left(\left[4uv + c_{ix}c_{iy} + 2c_{iy}u\right]\widetilde{\Pi}_{xy} + \left[\dfrac{-u^2 + v^2 - c_{ix}u + c_{iy}v}{2}\right]\widetilde{N}\right), \end{cases} \tag{331}$$

where $\widetilde{\Pi}_{xy}$ and $\widetilde{N}$ are the central moments and cx_i and cy_i are the discrete velocities of the lattice. The central moments are defined by the natural moments and velocities as

$$\widetilde{\Pi}_{x,y} = \sum_{i=0}^{8} f_i c_{ix} c_{iy} - uv \text{ and } \widetilde{N} = \sum_{i=0}^{8} f_i c_{ix}^2 - \sum_{i=0}^{8} f_i c_{iy}^2 - (u^2 - v^2). \tag{332}$$

After defining the stabilizer with above formulae, the post collision state of the distribution function is computed in the form:

$$f_i' = f_i - \beta(2\Delta s_i + \gamma^* \Delta h_i) \tag{333}$$

and the streaming is done by copying the distribution functions to their designated directions as usual. The determination of the macroscopic variables, density and velocity, is the same as the standard LBM, as well as the equilibrium distribution function.

5.5 Free surface Lattice Boltzmann method

After the brief review of a free surface flow modeling, we will provide the solution of the LB modeling for the free surface flow. The implementation of the described model will be found in Section 6.1 with the general numerical algorithm of the LBM.

5.5.1 Review for free surface flow modeling

One of the primary subjects of this work, as well as hydraulics, is a free surface flow. The problem of fluid flow with the presence of the free surface often occurs in the civil engineering field, whether in designing and construction stages of hydraulic structures, which interacts with fluid flow, or in natural disasters such as flood inundation, storm surges and Tsunamis. In the past, many afford of mathematical and numerical models on free surface predictions for a flow had been made since the accurate freely moving interface for two immiscible fluids had practically needed. The numerical methods can be classified in terms of numerical techniques and fluids of interest, as shown in **Figure 52** (a) and (b), respectively.

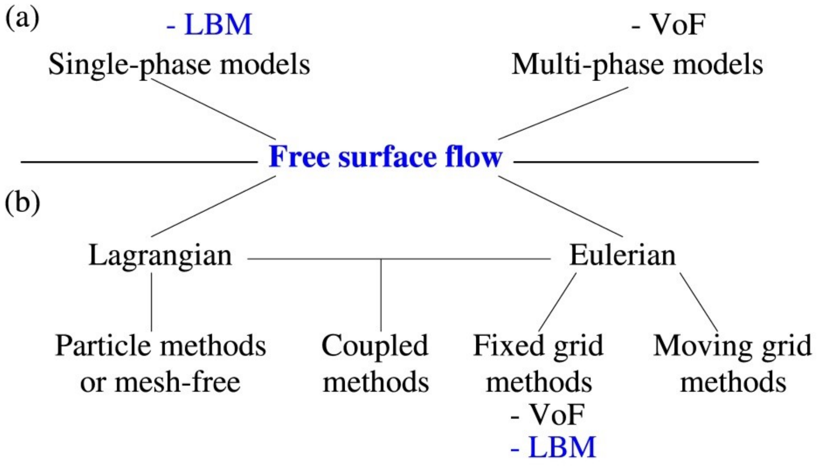


Figure 52. Classification of free surface flow models: (a) based on the involving fluids in a model and (b) based on the technique used in a solution.

In terms of fluids involved in the modeling, the method can be single- and multi-phase modeling. For single-phase free surface modeling, the primary fluid, e.g. the water is considered and the secondary fluid, e.g. the air, is neglected with the existence of the precise boundary condition on the interface between the primary and secondary phases. The boundary condition is called the free-surface boundary condition. Examples of the single-phase free surface methods are the mesh-free methods, such as smoothed particle hydrodynamics (SPH) (Gingold & Monaghan, 1977), (Monaghan, 1994) and the element-free Galerkin method (EFG) (Belytschko, et al., 1994), and the fixed grid methods, such as the shallow water approximation (Sielecki & Wurtele, 1970) and the LBM (Körner, et al., 2005). For the multi-phase flow, two or more fluids are considered and interfaces of those flows are determined as a free surface under the influences of those fluids on it. Famous representatives of multi-phase models are the Volume of Fluid (VoF) (Hirt & Nichols, 1981) in Eulerian approach and multi-phase particle methods (Monaghan & A, 1995).

In the aspect of numerical techniques used in a model, the methods can be grouped into the Lagrangian, the Eulerian approaches or the coupled scheme of them. The particle based discretization for the NSE in a frame of the Lagrangian approach developed many mesh free methods such as the SPH, moving particle semi-implicit method (MPS) (Koshizuka & Oka, 1996), (Koshizuka, et al., 1998), finite point method (FPM) (Onate, et al., 1996), the EFG and so on (Li & Liu, 2002). In the Lagrangian viewpoint, the free surface is tracked by the particles creating a free surface; the motion of those particles can be governed by the following equation (Lin & Liu, 1999) in general:

$$\mathbf{u}_p = \frac{d\mathbf{X}(t)}{dt}, \tag{334}$$

where **X** is a position vector which provides the coordinate of a fluid particle on the free surface at time *t*. The SPH and MPS are more attractive than others and have been shown the ability to solve a free surface flow as matured techniques in a wide range of space scale, more precisely from the bubble (Das & Das, 2009) in the laboratory scale to the near field Tsunami in the real field scale (Debroux, et al., 2001). While the Eulerian approach for a free surface modeling takes as much responsible as the Lagrangain approach. Although the very first numerical modeling for a free surface flow introduced by Harlow and Welch (Harlow & Welch, 1965) was the Lagrangain-Eulerian coupled technique, the following methods, such as the solution algorithm for VoF (SOLA-VoF) (Nichols, et al., 1980), height function method (Hirt & Nichols, 1981), level set method (Sussman, et al., 1994) and vortex based method (Baker Gregory, et al., 1982), are fully described on the grid which are whether fixed or moving grids. A comprehensive review on the Eulerian approach for the free surface flow can be found in (Lin & Liu, 1999) or (Scardovelli & Stéphane, 1999). The Eulerian approach for tracking the free surface is to update the free surface location based on the following equation:

$$\frac{\partial \vartheta}{\partial t} + \mathbf{u} \cdot \nabla \vartheta = 0\,, \tag{335}$$

where ϑ is a scalar quantity carrying the material information. For example, the quantity, ϑ, in the VoF method is the fractional volume of fluid (to simply, the fluid fraction value), $\vartheta = \mathrm{F}$, and the equation conserves the mass for two fluids which are considered. While, the density function, ϕ, is considered as the quantity, $\vartheta = \phi$, in Eq.(335) for the level set method.

As shown in **Figure 52**, the LBM for a free surface flow is a single-phase fixed grid approach and is first introduced by Korner and Singer (Körner & Robert F, 2000) to model the metal foaming process. Besides that, several approaches for the LBM to handle a free surface flow has been made by the single- and multi-phase modeling. Briefly, during the early stage of the LBM development, a two-phase immiscible fluid model is introduced in (Gunstensen, et al., 1991) based on the discrete immiscible lattice-gas model using the collision rule for two phases, colored with red and blue, to obtain surface tension between the two fluids. Further developments based on this method are known as color-gradient model for multi-phase flow and the model have the possibility to handle the free surface flow. Another popular multiple phase LBM was introduced by Shan and Chen (Shan & Hudong, 1993). The model is celebrated as the Shan-Chen model and became the origin of other multi-phase and multi-component models (Huang, et al., 2015). The basic idea of the Shan-Chen type model is to use distinct distribution functions for each fluid and to evaluate the repulsive force on the interface between fluids, which leads the phase separation. Other multiple-phase models in the LBM, such as the phase-field model (He, et al., 1999) and the free energy model (Swift, et al., 1996), are applicable for the free surface flow, when the density ratio is adjusted. Very good review for the multi-phase LBM can be found in (Li, et al., 2016) and (Huang, et al., 2015). After that, the featured LBM for the free surface flow is the single-phase color method modified from the color-gradient model by Ginzburg and Steiner (Ginzburg & Konrad, 2003), in which the collision is carried out only on the active cells filled fully or partially with fluid and the model maintain a sharp interfacial front with the help of an anti-diffusion algorithm for re-coloring scheme. Borrowing the idea of earlier marker-and-cell method, the coupled model of the LBM and the front tracking method is proposed in (Lallemand, et al., 2007) for the free surface flow including the dynamics of gas phase. Recently, based on the same concept, the coupled algorithms of LBM with the VoF or Level set technique are proposed in (Janssen & Krafczyk, 2010) and (Rüde & Thürey, 2004). The coupled method of Janssen et al, (Janssen & Krafczyk, 2010) uses an additional advection equation for the VoF carried out with a classical finite volume method, while the free surface is reconstructed by a piecewise linear interface reconstruction. Korner et al (Körner, et al., 2005), (Rüde & Thürey, 2004) formulated the VoF or Level set method in terms of mesoscopic way and in the model, the free surface is tracked by the flux-based advection scheme. A general discretization of the geometry of the coupled techniques is depicted in **Figure 53**. In the discretization, the interface cells and fluid cells are considered as active cells if the model is single-phase. The interface cells define the free-surface.

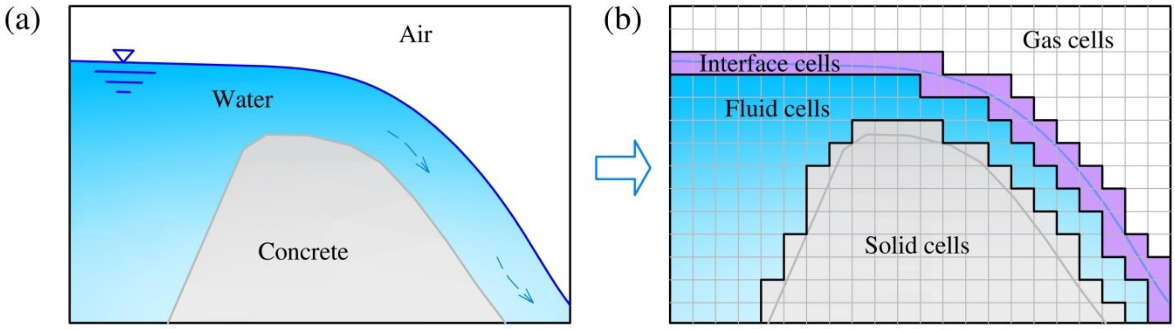


Figure 53. Geometry discretization of the VoF based LBM methods: (a) reality and (b) discretization.

In the following sections, we give the detailed explanation of the so-called free surface-lattice Boltzmann method of Korner et al (Körner, et al., 2005) and (Thürey, et al., 2006).

5.5.2 Free surface representation

The free surface algorithm for a use in the LBM was first introduced by Körner and Singer (Körner & Robert F, 2000), (Körner, et al., 2005) for the simulation of metal foaming and was later corrected for and tested on the two- and three-dimensional free surface flows by Thürey (Nils, 2007) and (Thürey, et al., 2006). Since the free surface in the LBM can be described using the same concept applied to the VOF method (Hirt & Nichols, 1981), each cell has a volume fraction value of a fluid that is expressed as the ratio of the mass to the density of the cell, i.e., $\epsilon = m/\rho$. Depending on the volume fraction value of the liquid, each cell is marked by flags as an indication of the materials in the computational cell, such as F for fluid (water), G for gas (air), W for solid, and IF for interface cells, as shown in **Figure 54** (b).

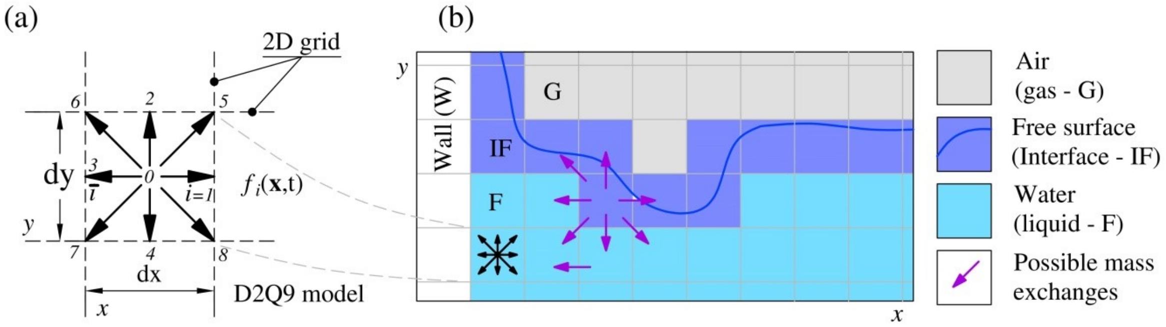


Figure 54. General scheme of free surface-lattice Boltzmann method: (a) D2Q9 lattice arrangement on the 2D grid and (b) materials in the domain and free surface representation.

The free surface is represented as chained single-layered interface cells having an arbitrary volume fraction value of 0 to 1, and the evolution of the free surface is tracked by mass calculations of the interface cells and cells other than solid and gas cells, which have no water fraction content. The scalar quantity for this method is the volume fraction of fluid, which in turn, the mass in a cell. Equation (335) in this case becomes

$$\frac{\partial m}{\partial t} + \mathbf{u} \cdot \nabla m = 0\,. \tag{336}$$

In contrast to the VOF method, the mass value of the cell is directly updated by mass exchange with neighboring cells at each time step of a computation (Körner, et al., 2005). So that the mesoscopic time and space discretization for Eq.(336) yields,

$$m(\mathbf{x}, t + \delta t) = m(\mathbf{x}, t) + \sum_{i=1}^{8} \Delta m_i(\mathbf{x}, t + \delta t)\,, \tag{337}$$

where $\mathbf{x}$ is the space vector, t is the current time, δt is the time step, and i denotes the lattice direction (**Figure 54** (a)). Mass exchange Δm_i is allowed for interface cells with neighboring F or IF cells, but does not allow for interface cells with neighboring G or W cells, as shown in **Figure 54** (b). Mass exchange between IF and F cells is easily defined by the difference between coming and leaving distribution functions before the streaming step, as

$$\Delta m_i(\mathbf{x}, t + \delta t) = s_e = \; f_{\bar{\imath}}(\mathbf{x} + \mathbf{c}_i, t) - f_i(\mathbf{x})\,. \tag{338}$$

We shall describe the collision and streaming steps for the time evolution of distribution functions, as well as their connections with the free surface-LBM in Section 6.1. However, the mass exchange between interfaces must have a special contribution by the volume fractions of the mass exchanging pairs, as follows:

$$\Delta m_i(\mathbf{x}, t + \delta t) = s_e \frac{\epsilon(\mathbf{x} + \delta t \mathbf{c}_i, t) + \epsilon(\mathbf{x}, t)}{2}, \tag{339}$$

where the difference between distribution functions, the flux exchange s_e, captures the neighboring status of the cell pair. The neighboring status of a cell reveals what types of cells exist in the surrounding space, as shown in **Figure 55** (b). Except the standard cell, other cell status for the interface cell is assumed as invalid cells, which need to be corrected during the excess mass distribution.

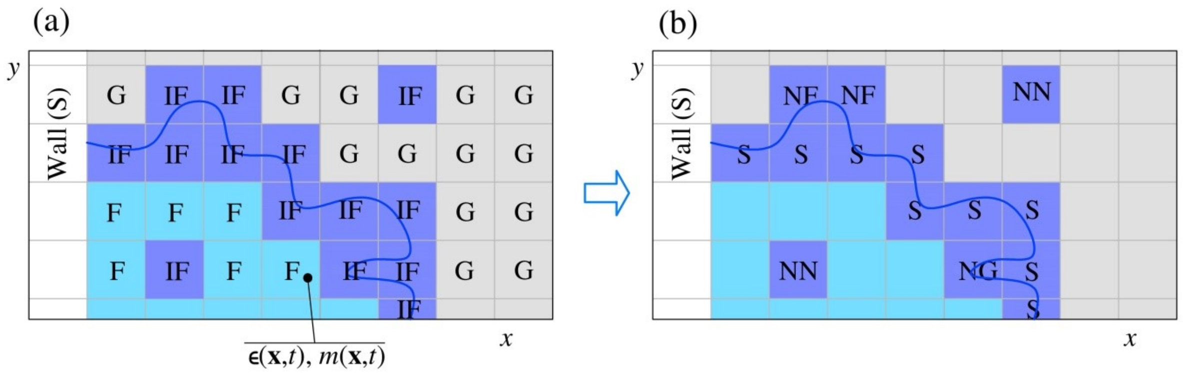


Figure 55. Flags used in the free surface-lattice Boltzmann method: (a) temporal state of cells as flags and (b) neighboring states for interface cells; S – standard cells, NF – no fluid neighbor, NG – no gas neighbor and NN – no interface neighbor.

Appropriate values of s_e based on the neighboring status prevent undesired IF cell from being filled or emptied (Thürey & Rüd, 2009). The value of s_e can be defined from **Table 4** based on the neighboring status.

Table 4. Substituting s_e of Eq.199 with the appropriate term given here forces the undesired interface cells to fill or empty. In this table $\mathbf{x}_{nb}$ denotes the position of the neighboring cell: $\mathbf{x}_{nb} = \mathbf{x} + \delta t\mathbf{c}_i$ (Nils, 2007)

Neighboring status of current and neighboring cell	Standard cell at $\mathbf{x}_{nb}$	No fluid neighbors at $\mathbf{x}_{nb}$	No gas neighbors at $\mathbf{x}_{nb}$
Standard cell at $\mathbf{x}$	$f_{\bar{i}}(\mathbf{x}_{nb}, t) - f_i(\boldsymbol{x})$	$f_{\bar{i}}(\mathbf{x}_{nb}, t)$	$-f_i(\boldsymbol{x})$
No fluid neighbors at $\mathbf{x}$	$-f_i(\boldsymbol{x})$	$f_{\bar{i}}(\mathbf{x}_{nb}, t) - f_i(\boldsymbol{x})$	$-f_i(\boldsymbol{x})$
No empty neighbors at $\mathbf{x}$	$f_{\bar{i}}(\mathbf{x}_{nb}, t)$	$f_{\bar{i}}(\mathbf{x}_{nb}, t)$	$f_{\bar{i}}(\mathbf{x}_{nb}, t) - f_i(\boldsymbol{x})$

The neighboring status of cells also helps to correct the evolution and the advection of interface cell, properly. The free surface is discretized as interface cells, so that the motion of the free surface must be represented by the transformation/advection of interface cells.

5.5.3 Interface advection

After the mass is updated over the entire domain using Eq.(337), the streaming and collision steps for the fluid flow (Eq.(204)) are performed in order to obtain new macroscopic variables (Eq.(200) and (201)) on the active cells. Since the density of the cell is updated by Eq.(200), the interface cell might be transformed into a G or F cell based on the following criteria:

$$\begin{aligned} IF \ \rightarrow \ F \text{ when } m(\mathbf{x}, t+\delta t) > (1+k)\rho(\mathbf{x}, t+\delta t) \text{ or} \\ IF \ \rightarrow \ G \text{ when } m(\mathbf{x}, t+\delta t) < (-k)\rho(\mathbf{x}, t+\delta t), \end{aligned} \tag{340}$$

where k (= 10^{-3}) is the additional offset value for the emptied or filled threshold ignoring a cell, which were previously treated. Depending on the filled or emptied status of the IF cell, the flags of neighboring G or F cells should be changed and the cells should obtain appropriate mass according to the excess mass distribution:

$$m(\mathbf{x} + \delta t\boldsymbol{c}_i) = m(\mathbf{x} + \delta t\mathbf{c}_i) + m^{ex}\left(\frac{\eta_i}{\eta_{total}}\right), \tag{341}$$

where m^{ex} is the positive or negative excess mass of the filled or emptied IF cell, and η_{total} is the sum of all weights η_i, each of which is computed by the normal vector **n** on the free surface as follows:

$$\begin{aligned} \eta_i &= \begin{cases} \mathbf{n}\cdot\mathbf{c}_i & \text{if } \mathbf{n}\cdot\mathbf{c}_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for the filled cells, and} \\ \eta_i &= \begin{cases} -\mathbf{n}\cdot\mathbf{c}_i & if\ \mathbf{n}\cdot\mathbf{c}_i < 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for the emptied cells.} \end{aligned} \tag{342}$$

Depending on the filled or emptied status of the current cell (IF cell), further changes of the neighboring cells flag are determined. Examples of the flag changes are given in **Figure 56** (b) and (c). Actually, the flags of the changed cells, i.e., the emptied or filled IF cells and their neighboring G or F cells, are not allowed to change their real flags before the excess mass distributed. Based on Eq.(340), these cells will have temporal transition flags during the excess mass distribution. Moreover, the distribution functions of the newly generated interface cells having temporal transition flags, which changed from G to the IF (shown in **Figure 56** (b)), can be initialized with the equilibrium distribution functions, as follows:

$$f_i(\mathbf{x}, t) = f_i^{\ eq}(\rho^{av}, \mathbf{u}^{av}) \,, \tag{343}$$

where $f_i^{\ eq}$ is the equilibrium distribution function for the fluid flow. The average macroscopic variables, density ρ^{av} and velocity vector $\mathbf{u}^{av}$, in Eq.(343) are computed using macroscopic variables of surrounding non-gas cells and must be assigned as the variables of the newly generated cells. Now, truly, the flags of cells are assigned as G, F or IF to interface cells leaving the temporal flags. Then the estimation of the mass values takes place for the next time step.

5.5.4 Initial and boundary condition

Initially, if water (fluid) is at rest and the depth of water is considerable for a computation, the hydrostatic pressure condition must be given in the initial condition to balance the force field. This initial condition (Bogner, 2009) can be expressed in terms of density derived from the barometric formula using $\rho_0 = 3P_0$:

$$\rho(z) = \rho_0 e^{3\mathbf{g}y} \,, \tag{344}$$

where y is the depth of water in an initial state and P_0 is the reference pressure at the free surface. This initial condition also provides an excellent calculation for the hydrostatic pressure.

Right after the streaming step in the fluid flow, the free-surface boundary condition must be imposed on the interface cells in order to recover the distribution functions that would be streamed from cells for which the following condition holds:

$$\epsilon(\mathbf{x} + \mathbf{c}_i) = 0 \text{ or } \mathbf{n} \cdot \mathbf{c}_i > 0, where\ \mathbf{n} \ = \frac{1}{2}\begin{pmatrix} \epsilon(\boldsymbol{x}_{k-1,j}) - \epsilon(\boldsymbol{x}_{k+1,j}) \\ \epsilon(\boldsymbol{x}_{k,j-1}) - \epsilon(\boldsymbol{x}_{k,j+1}) \end{pmatrix}. \tag{345}$$

To derive the free-surface boundary condition, the detailed flux conservation equation can be used. In that derivation, the free-surface boundary condition assumes that the fluid has a much lower kinematic viscosity than the gas state (Thürey, et al., 2005). Consequently, the free-surface boundary condition is expressed in terms of the following distribution function:

$$f_{\bar{\imath}}{}'(\mathbf{x}, t + \delta t) = f_i^{\ eq}(\rho_A, \mathbf{u}) + f_{\bar{\imath}}^{\ eq}(\rho_A, \mathbf{u}) - f_i(\mathbf{x}, t), \tag{346}$$

where ρ_{A} is the gas density implicitly acting as an air pressure onto the free surface and the velocity $\mathbf{u}$ is defined by using Eq.(201) in the previous time step or by initial condition at first. Reconstructing the distribution functions by the free-surface boundary condition is shown in **Figure 56** (a).

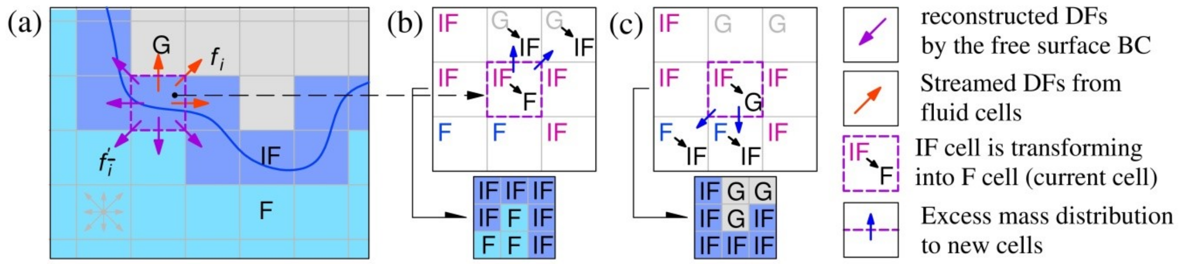

(a)
G
f_i
$f_{\bar{i}}$
IF
F
(b)
IF G G
IF IF
IF IF IF
F
F F IF
IF IF IF
IF F IF
F F IF
(c)
IF G G
IF IF IF
G
F F IF
IF IF
IF G G
IF G IF
IF IF IF
reconstructed DFs by the free surface BC
Streamed DFs from fluid cells
IF F
IF cell is transforming into F cell (current cell)
Excess mass distribution to new cells

Figure 56. Free surface movement: (a) free surface boundary condition, (b) IF cell is being filled and neighboring G cells recognized to change into IF cells and (c) IF cell is being emptied and neighboring F cells recognized to change into IF cells.

This is only a boundary condition for the free surface. The bounce back scheme (Eq.(234)) is imposed on the motionless wall interacting with IF cells to express the interaction of the free surface with the solid surface.

5.6 Liquid-solid phase transitions in free surface flows

Hereafter, we discuss a particular LB solution for the liquid-solid phase transition in a free surface flow as a proposed model. Boundary conditions for the moving solid interface resulted by the phase transition will be discussed. The implementation of the proposed LBM is found in Section 6.2.

5.6.1 Review of LB phase change modeling

A phase change (transition) is a physical phenomenon which shows the changes of uniform physical properties of a matter. The phase change is often happened with the changes of the thermal energy in the system. The thermal energy of the system can be explained by the entropy of or simply the temperature field of the system. Among communities, conducting numerical studies on phase changes by applying conventional methods, fixed spatial grid and front tracking methods are extensively used in the confined domain (Hu & Argyropoulos, 1996), (Virag, et al., 2006) without the free-surface condition. Recent models for a use in the conventional method are effective, but cumbersome and require several systems of equations to solve flows and phase changes, as well as adaptive or moving grids to clearly define melting/solidification front and iterative techniques (Danaila, et al., 2014) to solve nonlinear equations. However, phase changes in a natural convection flow remain a primary focus of these studies. But there exist a plenty of phase change problems in open channels like the ice in the rivers. Particle-based methods, which have an inherent ability to represent the free surface, are beginning to be applied to melting and solidification problems in free surface flows (Iwasaki, et al., 2010).

Skipping the advances in the continuum thermodynamic modeling, a brief review for the application of LBM for the heat transfer is roughly emphasized here. The modeling of the heat transfer in the LBM is introduced among with the other development of the standard LBM at almost the same time. When Alexander et al., (Alexander, et al., 1993) introducing the first multi-speed LBM for the thermodynamics extending the isothermal models explained in Section

5.2, Bartoloni et al., (Bartoloni, et al., 1993) introduced the double distribution function approach based on the idea of conventional numerical method for the thermodynamics. The later one was an appropriate approach to be used for phase transition problems. Soon after, Wolf-Gladrow (Wolf-Gladrow, 1995) proved that the lattice Boltzmann modeling of heat transfer can be more attractive and advances for the numerical method by deriving the LBE for the diffusion equation. Including the advection term to the equilibrium distribution function in the LBE for diffusion can lead the advection-diffusion solution in the LBM. This formulation for the ADE is modeled as a scalar field in the LBM having the second set of distribution function. The double distribution function approach, explained in the Section 5.3, is used to model phase transition problem in this thesis. The model adopted in this thesis was firstly introduced by Jiaung et al., (Jiaung, et al., 2001) and later demonstrated by Hubert et al., (Huber, et al., 2008).

As of this writing, the liquid-solid phase change models in the LBM have not yet been applied to a free surface water flow in the fields of hydraulics. The free surface LBMs coupled with the phase transition in material science in (Attar & Körner, 2011) and (Ammer, et al., 2014) are the pioneering works in this particular interest.

5.6.2 Liquid-solid phase change modeling

A liquid-solid phase change problem is often referred to as a Stefan problem, and basic modeling approaches have been presented in (Alexiades, 1992) and a number of other studies. When applying the LBM to a phase change problem, the complexities encountered by conventional methods are a large extent eliminated. In the framework of the LBM, the problem of heat transfer with phase changes can generally be solved by several approaches, such as a phase field method or an enthalpy-based method (Chatterjee & Chakraborty, 2006). These methods have been successfully tested and improved through application to metal melting/solidification problems (Semma, et al., 2008), but have not yet been applied to free surface water flows. As stated above, the numerical model uses two distribution functions expressed through the lattices on a fixed grid: one for a flow field and the other for heat transport, as shown in **Figure 63**. In the proposed method, the local enthalpy is updated non-iteratively with a temperature field.

In the modeling of heat transfer with phase transition, the temperature field is considered to be an essential variable and can be calculated by the following thermal lattice Boltzmann equation with latent heat of fusion (Jiaung, et al., 2001):

$$g_i(\boldsymbol{x}+\boldsymbol{c}_i\delta t, t+\delta t) - g_i(\boldsymbol{x},t) = -\frac{\left(g_i(\boldsymbol{x},t) - g_i^{eq}(\boldsymbol{x},t)\right)}{\tau_h} - w_i \frac{L_h}{c_p}\left(l_f(\boldsymbol{x}, t-\delta t) - l_f(\boldsymbol{x},t)\right), \quad (347)$$

where $g_i(\boldsymbol{x},t)$ is the distribution function of the temperature field, $\tau_h (= 3\alpha + 1/2\,)$ is the dimensionless relaxation time with respect to the thermal diffusivity α, L_h is the dimensionless latent heat of fusion, c_p is the specific heat capacity of water or ice and l_f is the liquid fraction defining liquid-solid region in a domain, shown in **Figure 57**. The liquid fraction value of 1 represents the liquid region in a domain, whereas 0 represents the solid region. Between 0 and 1 shows the interface zone in a single layer of cells. At the interface between a liquid and solid, $l_f(\mathbf{x})$ takes a value of between 0 and 1, where a "mushy" zone (Voller & Prakash, 1987) may be observed when conduction dominates heat transfer (Ayurzana & Hosoyamada, 2016) and can be defined when the solidus and liquidus temperature are distinguished (Jiaung, et al., 2001).

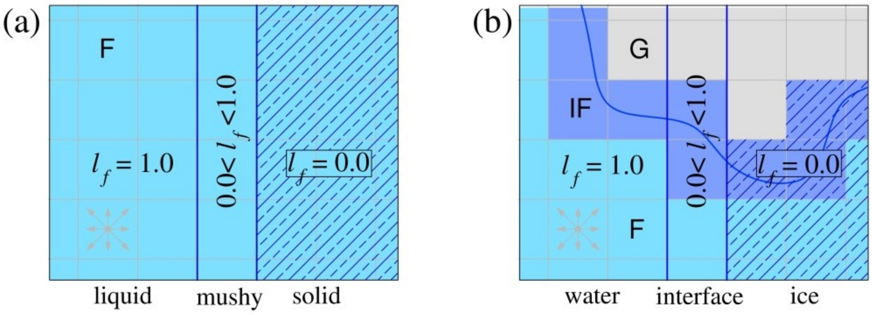


Figure 57. Liquid and solid regions in a computational domain: (a) single phase flow domain, (b) multiple-phase flow domain (for instance ice in free surface flow)

The specific heat capacity and thermal diffusivity must be defined appropriately in the computational cell depending on the cell type:

$$\begin{gathered}\alpha = \left(1 - l_f(\mathbf{x})\right)\alpha^{ice} + l_f(\mathbf{x})\alpha^{water} \text{ and}\\ c_p = \left(1 - l_f(\mathbf{x})\right)c_p^{ice} + l_f(\mathbf{x})c_p^{water},\end{gathered} \tag{348}$$

where the superscripts ice and water indicate the thermal diffusivities and specific heat capacities of ice and water, respectively. The specific heat capacity of water in lattice form can be obtained from the Stefan number,

$$St = \frac{c_p^{water}\Delta\theta}{L_h}, \tag{349}$$

and is related to the specific heat capacity of ice as

$$\frac{c_p^{R,ice}}{c_p^{R,water}} = \frac{c_p^{ice}}{c_p^{water}}. \tag{350}$$

The Stefan number in the simulations of the present study can be defined by Eq.(349) either using real values or lattice values. The equilibrium distribution function of the temperature field can be given as

$$g_i^{eq} = w_i\theta\left[1 + \frac{\boldsymbol{c}_i \cdot \mathbf{u}}{c_s^2}\right] \text{ with } \theta = \sum_{i=0}^{8} g_i, \tag{351}$$

and the macroscopic temperature T can be converted into a dimensionless temperature θ as follows:

$$T = \frac{T_{max} - T_{melt}}{\theta_{max} - \theta_{melt}}(\theta - \theta_{melt}) + T_{melt}. \tag{352}$$

After the dimensionless temperature evolution, the local enthalpy, obtained by

$$En = c_p\theta + l_f(\mathbf{x}, t - \Delta t)L_h, \tag{353}$$

can be used to linearly interpolate the liquid fraction,

$$l_f(\boldsymbol{x}) = \begin{cases} 1 \quad for\; En > En_s + L_h = En_l \\ 0 \quad for\; En < En_s = c_p\theta_{melt} \\ \dfrac{En - En_s}{En_l - En_s} \; for\; En_s \leq En \leq En_s + L_h \end{cases}, \tag{354}$$

and the liquid fraction defines the liquid (water) and solid (ice) phases in a domain. The model does not require iteration for the local enthalpy, since it was reported that the enthalpy update without iteration has negligible effects (Huber, et al., 2008). And also the model uses exact thermal properties for phases, which are often neglected in existing numerical methods, i.e., the thermal properties of the liquid are used for both the solid and liquid phases.

The force **F** experienced by the density difference (variation) induced by the temperature difference can be defined with the acceleration of gravity according to non-Boussinesq approximation (Tong & Koster, 1993) for the buoyance as

$$\mathbf{F} = \mathbf{g}(1 - \alpha_V(\theta - \theta_o)^2\,), \tag{355}$$

where **g** is the dimensionless acceleration due to gravity, α_V is the thermal volume expansion of water, and θ_o is the dimensionless reference temperature at the maximum density of water. The value of $\mathbf{g}\alpha_V$ can be defined in terms of the Rayleigh number (Ra) definition as

$$Ra = \frac{\boldsymbol{g}\beta(T_{hot} - T_{cold})H^3}{\nu\alpha}. \tag{356}$$

For the heat transfer module, Eq.(347) can recover the following dimensionless macroscopic equation using the multiscale expansion:

$$\frac{\partial\theta}{\partial t} + u_\alpha\frac{\partial\theta}{\partial x_\alpha} = c_s^2\left(\tau_h - \frac{\Delta t}{2}\right)\frac{\partial^2\theta}{\partial x_\alpha^2} - \frac{L_h}{c_p}\frac{\partial l_f}{\partial t} \tag{357}$$

where the heat source term, $\frac{L_h}{c_p}\frac{\partial l_f}{\partial t}$, is directly derived from the last term in the right hand side of Eq.(347) (Jiaung, et al., 2001). A general procedure to derive the macroscopic equation from the LBE is given in Appendix A2.

5.6.3 Models for moving liquid-solid interface

In order to evaluate a moving melting/solidification front, a liquid fraction value can be used to reflect DFs on a surface of a solid zone. The effect of the solid region can be simulated easily in the LBM using the probabilistic boundary condition or the immersed boundary condition. The probabilistic or partial-bounce back boundary conditions are introduced to model porous media flow. The three types of method of the probabilistic boundary conditions, namely outgoing bounce-back (Dardis & McCloskey, 1998), post-collision bounce-back (Thorne & Sukop, 2004), and pre-collision bounce-back (Walsh, et al., 2009) are compared and evaluated in (Walsh, et al., 2009). The first two probabilistic boundary conditions do not require additional modification for the determination of macroscopic variable, however, in some cases, the mass conservation is not ensured as reported in (Walsh, et al., 2009). However the later one, the pre-collision bounce-back shows better performance in terms of accuracy and efficiency, one need to consider the modification for the estimation of the velocity field (Walsh, et al., 2009). Based on porous media flow simulation and the new simulation step (e.g. the outgoing bounce-back (Dardis & McCloskey, 1998)) can follow after the normal collision process,

$$f_i^{**}(\mathbf{x}, t+\delta t) = f_i^{*}(\mathbf{x}, t) + \frac{1}{\tau}\left(f_i^{eq}(\mathbf{x}, t) - f_i^{*}(\mathbf{x}, t)\right), \quad 0 \le i \le 8$$

as

$$f_i(\mathbf{x}, t+\delta t) = f_i^{**}(\mathbf{x}, t+\delta t) + n_s[f_{\bar{i}}^{**}(\mathbf{x}+\mathbf{c}_{\boldsymbol{i}}\delta t) - f_i^{**}(\mathbf{x}, t)], \quad 1 \le i \le 8, \qquad (358)$$

where $n_s (= 1 - l_f)$ is a solid index, $f_i^{*}(\boldsymbol{x}, t)$ is the post streaming DF from the previous time, $f_i^{**}(\boldsymbol{x}, t+\delta t)$ is the post collision DF before the porous step, and $\bar{\imath}$ subscript is the opposite direction to i. A numerical procedure of the partial-bounce back boundary treatments for the both problems, porous media flow and liquid-solid moving boundary, is the same as given above.

Similarly, the immersed boundary condition also can be used to impose no-slip velocity boundary condition on a liquid-solid interface and the first application was carried out in (Huang, et al., 2013). More details about the adoption and application of liquid-solid phase transition in a free surface flow (Ayurzana & Hosoyamada, 2016), (Ayurzana & Hosoyamada, 2017) are discussed in Section 6.2 and in Chapter 7 and 8.

5.7 Immersed boundary method

5.7.1 Immersed Boundary LBM

A fluid flow is always bounded by solid surface or interacted with the solid body. If a solid is moving, a numerical solution, even a mathematical modeling for the effect of a moving boundary in a fluid flow becomes complicated. In the early development of the conventional numerical method, the treatment of a moving body or surface was required the continuous grid generation for each simulation step and additional solution process (Guo & Shu, 2013). As the immersed boundary method is introduced, the pre-existing these difficulties in numerical method was eliminated. Because the boundary condition on the solid surface, which does not need to be fit conform a Cartesian grid, can be imposed as the modification of the fluid governing equation (Peskin, 1977). In general a surface of solid does not intersect with the grid node, so some of the computational cells will be cut. The effect of cut cells can be treated in several ways, such as a continuous forcing approach, a discrete forcing approach and cut-cell methods (Mittal & Iaccarino, 2005), (Bandringa, 2010). Early work of the immersed boundary method coupling with the LBM had done by Noble and Torczynski (Noble & Torczynski, 1998), who had introduced the immersed boundary modification into the discretized LBE without losing the generality. The next attempt to adapt the conventional numerical concept for immersed boundary into the LBM was done in (Feng & Michaelides, 2004), where the restoring force term was estimated and added to the LBE. Inspired by the simplicity of Noble and Torczynski's solution for an immersed boundary method, the further developments on that solution were extendedly studied by Strack and Cook (Strack & Cook, 2007) and others. The immersed boundary modification is reasonably formulated from the mass and momentum balance perspective. With the immersed boundary modification, the discretized Boltzmann equation in Eq.(204) is rewritten as

$$f_i(\boldsymbol{x}+\mathbf{c}_i\delta t, t+\delta t) - f_i(\boldsymbol{x},t) =$$
$$-\frac{\delta t(1-\beta)}{\tau_{tot}}\left(f_i(\boldsymbol{x},t) - f_i{}^{eq}(\boldsymbol{x},t)\right) + \beta f_i{}^{m}(\boldsymbol{x},t) + \delta t A_i, \tag{359}$$

where $\mathbf{c}_i$ is the discrete unit velocity in the i direction, τ_{tot} is the dimensionless relaxation time with respect to the lattice viscosity ν and is adjusted with the sub-grid scale turbulent model (Hou, et al., 1996) explained in Section 5.4.3, and β is the parameter given by (Strack & Cook, 2007)

$$\beta(s_f,\tau) = \frac{s_f(\tau-0.5)}{(1-s_f)+(\tau-0.5)}, \tag{360}$$

in which $s_f(\mathbf{x},t)$ is the solid fraction value of the cell, which takes a value between 0 and 1. Solid fraction values of 0 and 1 represent the fluid (water) and solid (ice), respectively, as shown in **Figure 57**. In Eq.(360), the total relaxation τ_{tot} can be used instead of the relaxation time τ. The immersed boundary modification can be used for not only dynamic separation of solid (ice) and liquid (water) phases, but also for a moving body (e.g. moving ice) in a fluid flow. An additional collision term f_i^m is for cells partially or fully covered by a solid, i.e., ice cell, is given as

$$f_i{}^{m}(\mathbf{x},t) = f_{\bar{i}}(\mathbf{x},t) - f_i(\mathbf{x},t) + f_i{}^{eq}(\rho,\mathbf{u}_s) - f_{\bar{i}}{}^{eq}(\rho,\mathbf{u}), \tag{361}$$

where $\mathbf{u}_s$ is the velocity of the moving solid, which set to 0 for some study, i.e., the ice is fixed. It is visible that the additional collision in Eq.(361) is based on the concept of a bounce-back for the non-equilibrium part of the distribution function.

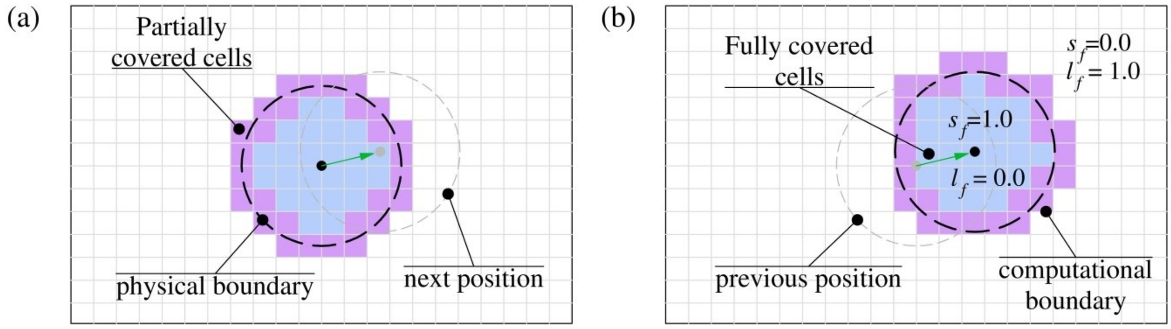


Figure 58. Immersed body discretization on Cartesian grid at two subsequent of time: (a) initial position, (b) next position.

5.7.2 Immersed boundary method for phase change

To apply the immersed boundary modification into a liquid-solid phase transition is straight work except the liquid fraction value need to be inserted instead of the solid fraction value. Those two variables are contrasted to each other as shown in **Figure 58**. For liquid-solid phase transition problem, the Eq.(360) becomes

$$\beta(l_f,\tau) = \frac{(1-l_f)(\tau-0.5)}{l_f+(\tau-0.5)}, \tag{362}$$

in which $l_f(\mathbf{x},t)$ is the liquid fraction value of the cell, which takes a value between 0 and 1. Liquid fraction values of 0 and 1 represent ice (solid phase) and water (liquid phase), respectively, as shown in **Figure 57**.

5.7.3 Arbitrary shaped floating body simulation

A moving solid body in the fluid flow can be effectively simulated with the immersed boundary (IB) modification into the discrete Boltzmann equation (see Eq.(359)). Noble and Torczynski's version of the IB method generates the smooth hydrodynamic force for a fluid flow and solid motion. We firstly proposed an algorithm to use the immersed boundary LBM to the floating object in the single phase free surface LBM (Ayurzana & Hosoyamada, 2018). Floating ice in free surface flow is an arbitrary shaped floating body because body shape changes over time due to its phase change. In the explanation, we use an ice body in example of floating solid. First, the force and the torque acting on an ice covering n cells are simply modified as:

$$\begin{gathered}\mathbf{F}_f = s\frac{\Delta x^2}{\Delta t}\sum_n \beta_n \sum_{i=0}^{8} f_i^m \mathbf{c}_i \\ \mathbf{T}_f = s\frac{\Delta x^2}{\Delta t}\sum_n (\mathbf{x}_n - \mathbf{x}_c) \times \left(\beta \sum_{i=0}^{8} f_i^m \mathbf{c}_i\right)\end{gathered}, \tag{363}$$

where s is the submerged volumetric percent of the floating ice body (**Figure 59** (c)), $\mathbf{x}_n$ is the coordinate of the current cell, $\mathbf{x}_c$ is the coordinate of the central cell of the floating body, as shown in **Figure 59** (b). The hydrodynamic force acting on the submerged part depends on the situation of the body position.

The motion of the ice body is resolved by the equation of motion:

$$\begin{gathered}m\frac{d^2x}{dt^2} = \mathbf{F}_f + \mathbf{F}_b \\ I\frac{dw}{dt} = \mathbf{T}_f\end{gathered}, \tag{364}$$

where m is the ice mass [kg], I is the moment of inertia of the body [kg m^2], w is the angular velocity [rad s^{-1}], $\mathbf{F}_b = \mathbf{g}(\rho_{ice}A_{ice} - \rho_{wat}A_{sub})$ is the buoyance force in two-dimensional space [N], A_{sub} is the area of the submerged part the floating body [m^2]. The equation of motion should be solved by the FDM explained in Section 3.2.2 and the implementation for the floating body can be found in Section 6.5.

If phase changes of water take place around the interface between ice and water, mass and shape of the ice body will be changed over time. The changes of the body will effect on the value of every physical variables of the body at each time step. To enable body shape changes over time or to simulate arbitrary shaped body, the algorithm must use the body shape of previous time step for the computations in the current time step. In addition, the staircase discretization of a body may generate an error and the accumulation of the error leads unphysical body shape changes for a moving body. To avoid this unphysical body shape change and to reduce error accumulation on the body shape, an integer center of mass for the moving body is introduced in Eq.(363) and it must be residing on the lattice node in order to discretize ice body into the computational domain. The integer center of mass $\mathbf{x}_c$ should be obtained from the real center of mass $\mathbf{x}_c^r$ defined by Eq.(364), see **Figure 60** (a). Moreover, the movement of the ice body requires a special care for the cell types. When the ice body moves to a new position, previous cells, which are being released from the occupied cells by the ice body, need to be converted into appropriate cell types (**Figure 60** (b)).

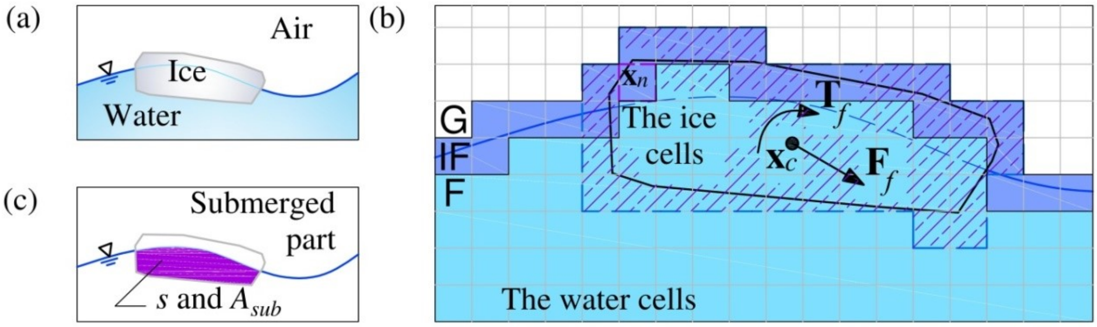


Figure 59. Floating ice on the free surface: (a) a general view, (b) discretization into 2D grid and (c) the representation of the submerged part implying displaced water by the floating ice.

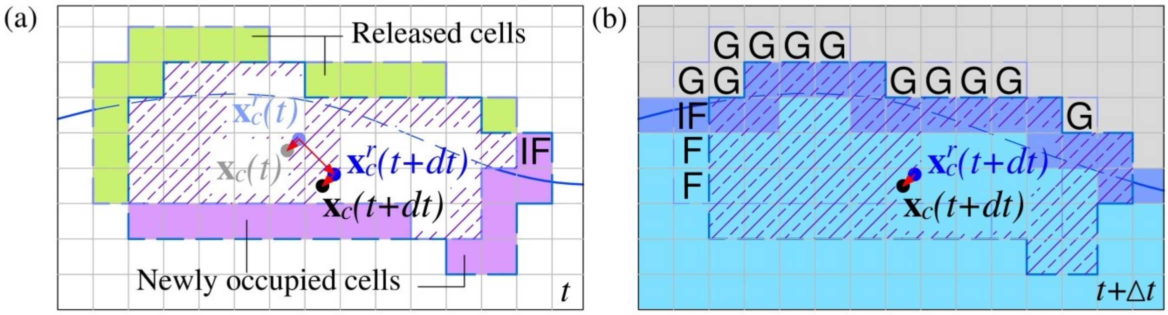


Figure 60. Movement of ice body: (a) Released and newly occupied cells and the centers, (b) letter flags on the cells show their changes due to the movement of ice body.

As the ice moves, newly occupied cells by the ice body are correctly initialized with the thermal and physical variables of the ice at the current time step. For instance, a temperature field of ice must be translated and rotated with the ice body movement to maintain the fields inside the ice. The same numerical procedure should be applied for the thermal distribution functions, as well as physical and numerical variables unique for the ice cells. This translation and rotation of the variables can be done by storing variables into temporal polar coordinate fields originated at the integer center of the ice body. Then the variables initialization for the cells of a new position of moving ice body uses the stored variables in the polar coordinate after the ice movement. This is most critical implementation of the moving body algorithm, since the phenomena inside the solid body are a major for the computation.

If gas cells are newly occupied by the ice body discretization, shown in **Figure 60** (a), they mainly become IF cells and distribution functions for them need to be reinitialized with Eq.(343). The mass of a released IF cell, which are going to be G cell, need to be distributed among the neighboring IF cells as Eq.(341) in order to ensure the mass conservation. This algorithm for a floating object is also applicable for the fully immersed body simulation, such as bed forming and sedimentation.

6 NUMERICAL IMPLEMENTATION

So far, we have discussed the theory and methods of different LB models from different interest of application. Here we shall give the pseudo codes to explain the implementation of these LBM.

6.1 Free surface flow modeling by the LBM

Before the implementation of the free surface LBM, the widely applied implementation for the discretized Boltzmann equation, the collision and streaming scheme, should take a place.

6.1.1 General implementation of lattice Boltzmann method

As often addressed, the very attractive feature of the LBM is its simplicity of the implementation. The LBM inherited the collision-streaming paradigm from the ancestor the LGA (Guo & Shu, 2013) (A.A.Mohamad, 2011) (Alexander, 2008). The collision-streaming paradigm is a base for the different implementation versions depending on the computer language in use (Latt, 2007). In the collision-streaming paradigm, the discretized lattice Boltzmann equation Eq.(204) is divided into two equations, namely

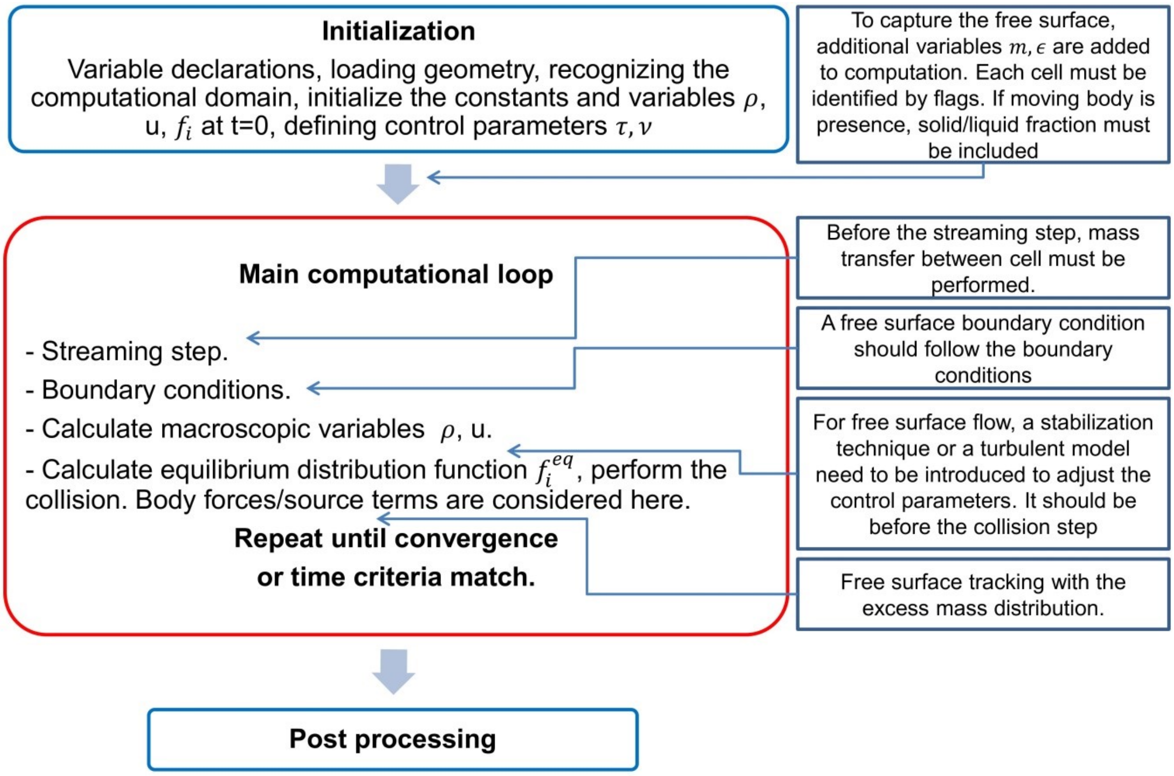
$$\text{Collision: } f'_i(\mathbf{x},t) = f_i(\mathbf{x},t) + \frac{1}{\tau_s}\left(f_i^{eq} - f_i\right) + A_i \text{ and} \tag{365}$$

$$\text{Streaming: } f_i(\mathbf{x}+\delta x, t+\delta t) = f'_i(\mathbf{x},t)\,. \tag{366}$$

There are two fashions for the collision-streaming paradigm, depending on which step (equation) is solved first in algorithmically, so called collision-streaming fashion and streaming-collision fashion. They are conceptually identical, however, they leads different result in some cases. For instance, the results depend on the initialization for the distribution functions. Generally, the following two initializations can be used for the initial value of distribution functions as

$$f_i|_{t=0} = w_i\rho \text{ or } f_i|_{t=0} = f_i^{eq}\,. \tag{367}$$

It is obvious from Eq.(367) that the initial condition for velocity field will define the differences in the results. The most studies use the collision-streaming fashion with the second initialization of Eq.(367) for the distribution function. In this thesis, we use the streaming-collision fashion with the shifted initialization with Eq.(367) depending on the problem. We do not use two distribution functions for f_i and f'_i, since the streaming step is coded to be not overwrite or destroy data from the storage. An algorithmic scheme of the general computational procedure for the LBM is given in **Figure 61** with the inclusion of the free surface algorithm.



Initialization

Variable declarations, loading geometry, recognizing the computational domain, initialize the constants and variables ρ, u, f_i at t=0, defining control parameters τ, ν

To capture the free surface, additional variables m, ϵ are added to computation. Each cell must be identified by flags. If moving body is presence, solid/liquid fraction must be included

Main computational loop

- Streaming step.
- Boundary conditions.
- Calculate macroscopic variables ρ, u.
- Calculate equilibrium distribution function f_i^{eq}, perform the collision. Body forces/source terms are considered here.

Repeat until convergence or time criteria match.

Before the streaming step, mass transfer between cell must be performed.

A free surface boundary condition should follow the boundary conditions

For free surface flow, a stabilization technique or a turbulent model need to be introduced to adjust the control parameters. It should be before the collision step

Free surface tracking with the excess mass distribution.

Post processing

Figure 61. General computational algorithm for the LBM with inclusions of a free surface algorithm.

As shown in **Figure 61**, the LB numerical algorithm including equations is very simple. A computation is straightforward and no iteration is required. The macroscopic variables will be defined from the distribution functions, which are the main variables in a LB simulation. The macroscopic variables are used for the computation of equilibrium distribution function. In some case, it is adequate to compute the equilibrium distribution function separately from the collision. In the following, we give the implementation of the above algorithm with the pseudocode using the Fortran 77 programming language.

6.1.2 Implementation of free surface LBM

To solve a free surface flow, the general flow of the algorithm with the equations of the LBM is not destroyed. Instead the inclusions from the free surface flow modeling is inserted where the procedure takes place in, as shown in **Figure 61**.

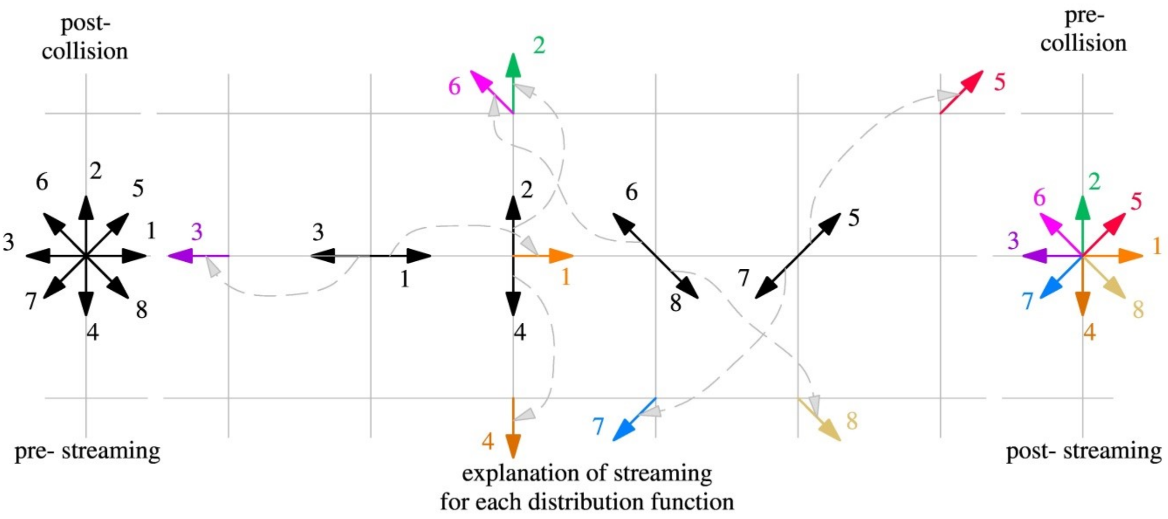


Figure 62. Explanation of streaming process for D2Q9 lattice arrangement. The distribution function of 0 direction, f_0, is not shown here because f_0 don't leave its location.

Generally, two subroutines, `massev` and `change`, are added to the basic algorithm of the LBM, as seen in **Code 1**. The pseudo-code is devoted to explain the basics of the algorithm and is possible to extend it to an in-house serial code. Note that the pseudo-code listed in the thesis is not advanced or efficient implementation for the free surface-lattice Boltzmann method.

Code 1: Main program of the free surface lattice Boltzmann simulation

```
      program free_surface
      include 'paramc.h'
      open(*,file='****.dat')   ! new files for result printing
! Initialazation
      compute dimensionless number
      compute viscosity or relaxation parameter
      call initial !where initialize the variables and distribution functions
                   !as well as flagging for each cell
! main computational loop
      DO kk=1,mtotal  ! discrete time step (lattice time step)
         time=kk*dt  ! dt is physical time step in second
      call massev    ! mass evaluations on cells
      call streaming ! streaming for the distribution functions (DFs)
      call fluidbound! boundary conditions for DFs
      call densvel   ! evaluation of macroscopic variables
      call subgrid   ! sub-grid scale model for turbulent
      call collision ! collision for the DFs
      call result    ! printing results
      call change    ! updating of cell information for the free surface
             check the time criteria
      END DO
! end of the main computational loop
      stop
      end
! end of the main program
```

Every global variables in the code are declared in the header file, `paramc.h`, so that subroutines don't need to mention the arguments used in it. The computation of the control parameters, such as viscosity or relaxation parameter, is a less complicated issue and will be described in Section 6.3, because it is related to the scaling of the macroscopic variables. For simplicity, we describe each subroutine embedded in **Code 1** by giving their pseudocodes.

In the initialization, all variables in the code need to get initial values as well as constants like lattice weighting and discrete velocities. This subroutine, **Code 2**, contains the realization of the geometry of a computational domain with the boundaries. The main implementation of a free surface model is the marking cells by assigning the flags as material, depicted in **Figure 54**. Flags can be logical, character or integer depending on programming design, since the code does not use them for an arithmetical operation. Generally the flagging makes much easier to implement operations for the LBM as well as to implement boundary conditions. After initializing the macroscopic variables, the initialization of the distribution function takes place with Eq.(367). Another important thing for the implementation is to use another set of flagging to possess the status about the neighboring information.

Code 2: Pseudocode for initialization for LB simulation.

```
   subroutine initial
   include 'paramc.h'
   integer icc(0:n,0:m) !temporal memory

   open(**,file='gridData.dat') !loading the grid data
   do j=m,0,-1
   read(**,*) (icc(i,j),i=0,n)
   end do
   ! lattice constants
   w(:)=(/4./9.,1./9.,1./9.,1./9.,1./9.,1./36.,1./36.,1./36.,1./36./)
   cx(:)=(/0.0,1.0,0.0,-1.0,0.0,1.0,-1.0,-1.0,1.0/)
   cy(:)=(/0.0,0.0,1.0,0.0,-1.0,1.0,1.0,-1.0,-1.0/)
! density. velocity, fluid fraction, mass, flag and DFs are initialized.
   do i=0,n
   do j=0,m
   rho(i,j), p(i,j), tau(i,j)=1./ome ! initial values
     ! use Fig.25 to mark cells by flag
   if(icc(i,j).eq.free surface) then
    flag(i,j), u(i,j), v(i,j), e(i,j), ma(i,j) !for interface cell (IF)
   else if (icc(i,j).eq.fluid) then
    flag(i,j), u(i,j), v(i,j), e(i,j), ma(i,j) !for fluid cell (F)
   else if (icc(i,j).eq.solid) then
    flag(i,j), u(i,j), v(i,j), e(i,j), ma(i,j) !for solid cell (S)
   end if
       ! initialization for the DFs
       do k=0,8
       f(k,i,j)=w(k)*rho(i,j) !Eq.(367)
       enddo
   end do
   end do
! initial neighborhood imformation
   do i=0,n
   do j=0,m
    assign the neigboring information on the cell ! use Figure 55
   end do
   end do

   return
```

```
end
```

After the initialization, the main loop takes place until the convergence or total time meets with time criteria. In the main loop, **Code 1**, totally seven main subroutines are included to accomplish the tasks for the free surface flow modeling (`massev` and `change`), streaming-collision fashion for the LBM (`streaming`, `fluidbound`, `densvel` and `collision`) and turbulent modeling (`subgrid`).

A subroutine for the mass exchange between the cells is not a trivial thing. It is carried out by using the concept explained in Section 5.5.2. Generally, the mass exchanges between the fluid cells (F cell) can be ignored, because they have the same values of fraction in terms of free surface. An attention must be paid to the interactions between the cells that have unequal values of fraction. In other words, interface cells (IF cells) must be a center for the implementation. An IF cell exchange mass with F cells and IF cells, which exist in adjacent to the current cell. If the current cell is F cell, the mass exchange with the adjacent IF cell is simply calculated by Eq.(338). However, if the current cell is IF cell, we need to consider the neighboring status for the current cell and the mass exchanging pair using **Table 4**. Then the total mass of cell is updated with Eq.(337) for all IF and F cells.

Code 3: Subroutine to compute the mass exchanges between cells

```
subroutine massev
include 'paramc.h'
real delm(1:8,0:n,0:m) ! mass change in specific direction
do i=0,n  !computational domain
do j=0,m

if(flag(i,j).eq.1) then       ! F cell
       do k=1,8
       delm(k,i,j)=use Eq.338
   end do
  ma(i,j)=ma(i,j)+asum        ! Eq.198
else if(flag(i,j).eq.2) then ! IF node
   do k=1,8
       delm(k,i,j)=use Eq.339 with Table 4
   end do
  ma(i,j)=ma(i,j)+asum        ! Eq.337
end if

end do
end do

return
end
```

The streaming step is simple for D2Q9 lattice arrangement, as shown in **Figure 62**. In the result of streaming process, some set of distribution functions will be missed at boundaries. Those missed distribution functions are reconstructed by the boundary conditions explained in Sections 5.2.3 and 5.5.4 for the fluid flow.

Code 4: Subroutine for the streaming step

```
subroutine streaming
```

```
      include 'paramc.h'
! streaming
         use Figure 62
      return
      end
```

In a subroutine for boundary conditions (**Code 5**), we give a full implementation of few boundary conditions which can be used for the two dimensional channel flow: an inlet boundary on the left, blocked by solid at the bottom and the top surfaces and an outlet boundary on the right. As mentioned above, the single-phase free surface LBM ignores the gas cells for every operation. In addition, the gas and solid cells are only used to impose the boundary conditions.

Code 5: Subroutine to impose boundary conditions

```
      subroutine fluidbound
      include 'paramc.h'
      nx=0.
      ny=0.

! bounce-back boundary condition for obstacle Eq.233
      do i=0,n
      do j=0,m
          if(flag(i,j).eq.0) then
             do k=1,8
             op=opp(k)
             if(flag(i+cx(k),j+cy(k)).eq.2
     &      .or.flag(i+cx(k),j+cy(k)).eq.1) then
             f(k,i+cx(k),j+cy(k))=f(op,i+cx(k),j+cy(k))
             endif
             enddo
          endif
      enddo
      enddo
! Velocity boundary condition for inlet Eq.250-253
      do j=1,m
      if(flag(0,j).eq.1.or.flag(i,j).eq.2) then
      uo=known velocity
      vo=known velocity
      rhow=f(0,0,j)+f(2,0,j)+f(4,0,j)+2.*(f(3,0,j)+f(6,0,j)+f(7,0,j))
      rhow=rhow/(1.-uo)
      f(1,0,j)=f(3,0,j)+2.*rhow*uo/3.
      f(5,0,j)=f(7,0,j)+rhow*uo/6.+0.5*rhow*vo
      f(8,0,j)=f(6,0,j)+rhow*uo/6.-0.5*rhow*vo
      endif
      enddo
! Outflow boundary condition for outlet
      do j=0,m
      if(flag(n,j).eq.1.or.flag(n,j).eq.2) then
      f(3,n,j)=f(3,n-1,j)
      f(6,n,j)=f(6,n-1,j)
      f(7,n,j)=f(7,n-1,j)
      endif
      enddo
! free surface boundary for interface
```

```
do i=0,n
do j=0,m
if(flag(i,j).eq.2) then
   use Eq.346
endif
enddo
enddo

return
end
```

After successful treatment of streaming, including boundary conditions, a subroutine (**Code 6**) to calculate the macroscopic variables will take place in the code. The density, velocity and pressure, if necessary the vorticity, the streamlines and the shear stress, can be calculated in this subroutine.

Code 6: Subroutine to calculate the macroscopic and hydrodynamic variables.

```
subroutine densvel
include 'paramc.h'

do j=0,m
do i=0,n
do k=0,8
   rho(i,j)=use Eq.200
   if Eq.112 is used
    u(i,j)=use Eq.201
    v(i,j)=use Eq.201
   else if Eq.206 is used
    u(i,j)=use Eq.209
    v(i,j)=use Eq.209
   end if
end do
p(i,j)=rho(i,j)*RT !Pressure Eq.155
end do
end do

return
end
```

After the calculation of the macroscopic variables, the collision step must be performed in terms of numerical procedures. If a flow is a turbulent or unsteady, one should incorporate with a turbulent model or a stabilization technique to secure the successful simulation. In our case, we use the sub-grid scale turbulent model to adjust relaxation time to locally defined relaxation time to capture the turbulent structure, in addition to stable a simulation. The implementation of the sub-grid scale model is given in **Code 7**.

Code 7: Subroutine for the sub-grid scale turbulent model

```
subroutine subgrid
include 'paramc.h'
cc=suggested value    ! smagorinsky constant

do i=0,n
do j=0,m
```

```

       if(flag(i,j).eq.1.or.flag(i,j).eq.2) then !only for F and IF
         do k=0,8
         compute feq(k,i,j)
         compute Pab with Eq.324
         end do
         compute Sbar with Eq.323
         cmpute tau(i,j) with Eq.322 or Eq.325
       end if

      end do
      end do

      return
      end
```

In the collision step (**Code 8**), the most of variables in the code are used. For instance the density and velocities are used to calculate the equilibrium distribution functions. The distribution functions after streaming and boundary conditions are used to derive the post collision distribution function for next time step. The local relaxation time defined by the sub-grid scale turbulent model is also used for the collision operator.

Code 8: Subroutine for the collision step.

```
      subroutine collision
      include 'paramc.h'

! force on fluid
      do i=0,n
      do j=0,m
      if(flag(i,j).eq.1.or.flag(i,j).eq.2) then
         compute force with Eq.203 or Eq.206
      end if
      end do
      end do
! collision step
      do i=0,n
      do j=0,m
      if(flag(i,j).eq.1.or.flag(i,j).eq.2) then
         do k=0,8
         compute feq(k,i,j) with Eq.193
         perform collision f(k,i,j) with Eq.365
         end do
      end if

      end do
      end do

      return
      end
```

Now it is time to update flags for each cell based on the fluid fraction value which in turns relates to the cell mass. A subroutine to do cell updates will be the last unit, where a free surface motion is tracked with the movement of IF cells and mass distribution to the new cells are made. The mass conservation is ensured here.

Code 9: Subroutine to perform the free surface movement related cell information update

```
      subroutine change
      include 'paramc.h'
      real sumw(0:n,0:m),mex(0:n,0:m) !sum of weight to distribute an excess mass
      real averh,aveuu,aveuv          !an average density and velocities
      integer tempflag(0:n,0:m)       !temporal flag

! temporary flag setting on each cell
      do j=0,m
      do i=0,n
       if(flag(i,j).eq.2) then ! only for IF cell
          use Eq.340 to assign temporal flags
       end if
      end do
      end do
! changing the neighboring flags
! bacause of ordering current cell flag change
      do j=0,m
      do i=0,n
         if(tempflag(i,j).eq.IFF) then ! IF to F cell
           do k=1,8
             use Figure 56
             compute averh, aveuu, aveuv ! average variables of
                                            !surrounding non-gas cells
          f(k,i+cx(k),j+cy(k))=use Eq.343
             flag(i+cx(k),j+cy(k))=2 ! newly generated cell
           end do
         end if
      end do
      end do
      do j=0,m
      do i=0,n
         if(tempflag(i,j).eq.IFG) then ! IF to G cell
           do k=1,8
             use Figure 56
             compute averh, aveuu, aveuv ! average variables of
                                            !surrounding non-gas cells
          f(k,i+cx(k),j+cy(k))=use Eq.343
             flag(i+cx(k),j+cy(k))=2 ! newly generated cell
           end do
         end if
      end do
      end do
! excess mass determination and distribution
      do j=0,m
      do i=0,n
         use Eq.341 and 342
      end do
      end do
! volume fraction change
      do j=0,m
      do i=0,n
         compute e(i,j) from ma(i,j)
      end do
      end do
```

```
! temporary flag is changed to major flag
      do j=0,m
      do i=0,n
         flag(i,j) is assigned based on
         tempflag(i,j) which will be deleted
      end do
      end do
! set neigboring information on IF cell
      do j=0,m
      do i=0,n
         assign the neigboring information on the cell ! use Figure 55
      end do
      end do

      return
      end
```

From the subroutines embedded in **Code 1**, we don't give the subroutine called `result`. This subroutine will have not only printing the results, but also scaling the macroscopic variables to the dimensional variables using the scales. The scales and the aforementioned control variables are tied to the dimensionless numbers. We will discuss about it in Section 6.3.

6.2 Coupled algorithm for free surface and phase change modeling

6.2.1 Direct integration

As described in Section 5.6, the computational model use two set of distribution functions; one for fluid flow, another one for scalar field which is a temperature field in our case, as shown **Figure 63** (b). When the two modules use the same time step, the modules are integrated as a direct integration. Fortunately, the implementation of a coupled algorithm for a free surface flow and heat transfer modeling with phase change in direct integration is quite straightforward. Additional variables devoted for heat transfer and phase transitions will be inserted to subroutines as additions. For instance, the liquid fraction value, l_f, is assigned to F and IF cells to distinguish a cell either water or ice. If a cell is an ice, it should be treated as a solid cell, but having the distribution functions to continuously define the macroscopic variables. Just like the free surface algorithm, the distribution functions of the scalar field are only calculated on the F and IF cells.

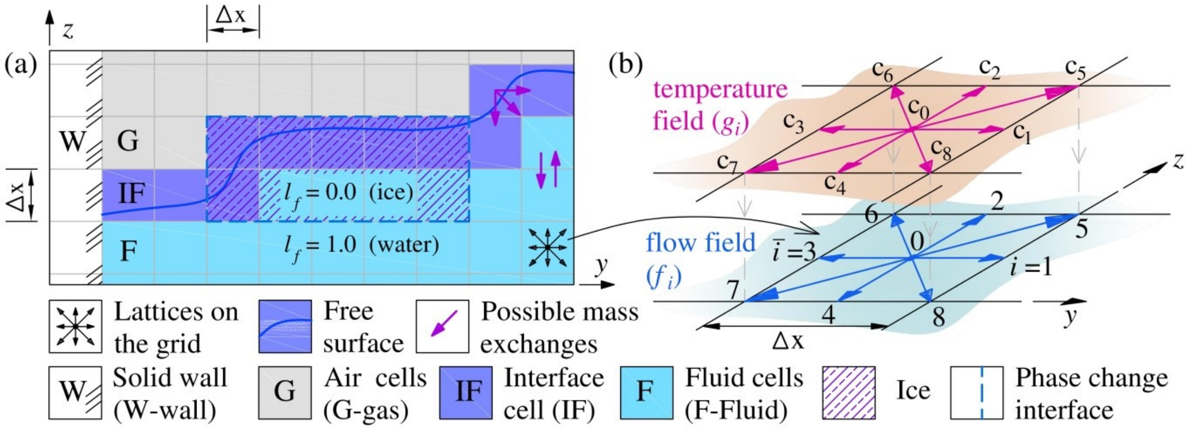


Figure 63. Schematic description of the coupled numerical model: (a) free surface representation with cell types and the possibilities of mass exchange between cells, (b) a description of two sets of distribution functions model for fluid flow and temperature field.

The numerical model for a liquid-solid phase transition in a free surface flow will have totally ten subroutines, the seven of which is described in Section 6.1. The rest of three subroutines are added to the evolution of temperature field and the estimation of phase transition. The main program is given in **Code 10** for the numerical model for a liquid-solid phase transition in a free surface flow.

Here, we only give the specific subroutines for the scalar field. We just mention tips or hints for the modifications or additions due to a scalar field to the fluid flow modeling subroutines. In the `initial` subroutine, the variables like `th(i,j)`, `lf(i,j)` and `En(i,j)`, and the distribution functions of the scalar field are initialized. In the mass exchanging subroutine `massev`, the mass exchange with an ice IF cell must be disallowed, as shown in **Figure 63** (a). It is the same for ice F cells. An ice IF cell means that the cell is IF, but having $l_f = 1.0$ for its liquid fraction value. The streaming step is the same for a fluid flow and scalar field. Only one subroutine `streaming` can work for two sets of the distribution function, since the lattice arrangement for two modules is the same.

Code 10: Main program for the liquid-solid phase transition in free surface flow modeling

```
      program phase_free_surface
      include 'paramc.h'
      open(*,file='****.dat')   ! new files for result printing
! Initialization
      compute dimensionless number
              for fluid flow and scalar field
      compute viscosity or relaxation parameter
              for fluid flow and scalar field
      call initial !where initialize the variables and distribution functions
                   !as well as flagging for each cell
      define time cycling !if code uses different time steps
! main computational loop
      DO kk=1,mtotal  ! discrete time step (lattice time step)
         time=kk*dt   ! dt is physical time step in second
      call massev     ! mass evaluations on cells
      call streaming ! streaming for fluid flow
      call fluidbound! boundary conditions for fluid flow
      call densvel   ! evaluation of density and velocity
      call subgrid   ! subgrid scale model for turbulent
      call collision ! collision for fluid flow
      call coll_scal ! collision for scalar field
      call scalbound ! Boundary conditions for scalar field
      call streaming ! streaming for scalar field
      call scalcalcu ! evaluation of scalar variable
      call result    ! printing results
      call change    ! updating of cell information for the free surface
             check the time criteria
      END DO
! end of the main computational loop
      stop
      end
```

For the subroutine `collision`, we must consider the external force (Eq.(355)) induced by a scalar field, for instance, it is a buoyance force in a fluid flow because of the relation between density and temperature. Furthermore, a treatment for the moving boundary condition, i.e. the interface between liquid (water) and solid phase (ice), for the fluid flow must be implemented using either the probabilistic boundary condition or the immersed boundary method explained in Section 5.6.2 and 5.7, respectively. The treatment of boundary condition, **Code 11**, can follow the collision step directly, if the probabilistic bounce-back condition is used.

Code 11: Addition to the collision subroutine to account liquid-solid interaction

```
! probabilistic boundary condition after collision, Eq.358
      do i=0,n
      do j=0,m
       amda=1.0-lf(i,j)
          do k=1,8
          op=opp(k)
          f(k,i,j)=f(k,i,j)+amda*(f(op,i+cx(k),j+cy(k))-f(k,i,j))
          end do
      end do
      end do
```

Generally, after the streaming-collision fashion for fluid flow, the collision-streaming fashion for a scalar field, which can be the streaming-collision fashion just like a fluid flow counterpart, takes a place. The collision step for the scalar field is much easier than that of the fluid flow module. However, the evolution of the distribution function requires the liquid fraction values at current and old time steps to account the latent heat source.

Code 12: Subroutine of the collision step for the evolution of temperature distribution function

```
      subroutine coll_scal
      include 'paramc.h'

      do i=0,n
      do j=0,m
          if(flag(i,j).eq.1.or.flag(i,j).eq.2) then
          define thermal properties for the cell
          using Eq.348 and Eq.285
      do k=0,8
      geq(k,i,j)= use Eq.283
        g(k,i,j)= use Eq.347
      end do
          endif
      end do
      end do

      return
      end
```

The boundary conditions for the scalar field must be separately treated as given in **Code 13**.

Code 13: Subroutine for the boundary conditions for heat transfer and phase transition

```
      subroutine scalbound
      include 'paramc.h'
! adiabaric condition on solid surface
      do i=0,n
         do j=0,m
             if(flag(i,j).eq.F and IF cell) then
                 do k=1,8
                  if(flag(i+cx(k),j+cy(k)).eq.S cell) then
             g(opp(k),i,j)=g(opp(k),i-cx(k),j-cy(k)) !Eq.309
                  end if
                 end do
             end if
         end do
      end do
! Constant temperature boundary condition at inlet.
      do j=0,m   !Dirichlet Boundary - 2
         if(flag(0,j).eq.F and IF cell) then
             g(1,0,j)=twh*(w(1)+w(3))-g(3,0,j)
             g(5,0,j)=twh*(w(5)+w(7))-g(7,0,j)
             g(8,0,j)=twh*(w(8)+w(6))-g(6,0,j)
         end if
      end do
! East boundary condition, outlet
      do j=0,m   !Eq.265
         if(flag(n,j).eq.F and IF cell) then
      g(3,n,j)=g(3,n-1,j)
      g(6,n,j)=g(6,n-1,j)
      g(7,n,j)=g(7,n-1,j)
         end if
      end do
! boundary condition on the free surface
      do i=0,n
      do j=0,m
         if(flag(i,j).eq.IF cell) then
         give boundary condition depending on the problem
         endif
      end do
      end do

      return
      end
```

After the boundary conditions, recovering the missed parts during the streaming, the scalar field and phase-transition are defined as shown in **Code 14**. A phase-transition is defined by the interface between water and ice having a value of $0.0 < l_f(x) < 1.0$. In turns, the liquid fraction value is used to impose the boundary condition in the collision subroutine at the moving interface, i.e. the melting front.

Code 14: Subroutine to calculate the temperature and to account the phase transition

```
      subroutine scalcalcu
      include 'paramc.h'
! temperature calculation
      do j=0,m
      do i=0,n
          do k=0,8
          th(i,j)= use the second equation in Eq.351
          end do
      end do
      end do
! liquid fraction value update
      do i=0,n
      do j=0,m
         if(flag(i,j).eq.F and IF cell) then
         compute En(i,j) with Eq.353
         compute lf(i,j) with Eq.354
         end if
      end do
      end do
      return
      end
```

Now it is almost at the end of implementation for the direct time integration. To finish the initialization of the scalar related variables, more importantly the thermal distribution functions need to be reinitialized in change subroutine as like

$$g_i(\mathbf{x},t) = g_i^{eq}(\theta^{av},\mathbf{u}^{av})\,, \tag{368}$$

where θ^{av} and $\mathbf{u}^{av}$ are the average macroscopic variables of surrounding non-G cells in respect with the current cell.

6.2.2 Sub-cycling integration

The interaction between the free surface flow module and the heat transport with the phase change module is such that the temperature difference produces a buoyance force in the flow field, and the flow field affected by the buoyance force forms a temperature field in the domain. Although the buoyance force is negligible in a turbulent flow, it must be included in a computation. The lattice viscosity is related to the lattice thermal diffusivity of a fluid as $\alpha^{\text{water}} = \nu/\text{Pr}$, where Pr is the Prandtl number, so that the relation between the computational modules is maintained. However, depending on the choice of grid spacing and time step, the modules can be integrated in two different time scales, which results the sub-cycling in time integration, as follows:

$$n_s = \left\lfloor \frac{\Delta t_h}{\Delta t_f} \right\rfloor + 1, \tag{369}$$

where $\lfloor\cdot\rfloor$ is the floor operator to convert a real number to an integer, $\mathit{\Delta} t_h$ is the time step of the heat transport module, and $\mathit{\Delta} t_f$ is the time step of the fluid flow module. Since the grid spacing is defined by the selected grid number, the time steps can be determined from the following relations:

$$\mathbf{g}_R \;=\; \mathbf{g}\frac{\Delta x}{\Delta t_f^2} \text{ and } \alpha_R^{ice} \;=\; \alpha^{ice}\frac{\Delta x^2}{\Delta t_h^2}, \tag{370}$$

where the subscript R indicates the real physical value of the acceleration of gravity or the thermal diffusivity of ice. The ratio $\alpha_R^{ice}/\alpha_R^{water} \;=\; \alpha^{ice}/\alpha^{water}$ is used to find the dimensionless thermal diffusivity for ice in the second relation of Eq.(370). A sub-cycling in the time integration improves the numerical stability of the computation because the relaxation times in the modules can be adjusted. In the implementation, Eq.(369) need to be embedded into the main program in **Code 10**.

6.3 Scaling and parameterization

A Lattice Boltzmann world is a dimensionless world. A physical world is scaled to the lattice Boltzmann world. Vice versa, the results obtained by the LBM must be scaled to the physical results. It is caused that the continuous Boltzmann equation is nondimensionlized (see Section 5.2.1) to derive the LBE. Additionally, the velocity space in mesoscopic scale is discretized into unit discrete velocities.

Parameterization is a work to define the LB control variables and scaling factors. The determination of the LB dimensionless control parameters such as a characteristic velocity U_{LB}, characteristic length L_{LB}, lattice viscosity ν and relaxation time τ, is related to the numerical stability and accuracy. The most of the parameters have physical counterparts with the unit. The connection between the LBM and the physical parameters can be given by applying the scaling factors such as grid spacing Δx, a time step Δt and density scaling $\Delta\rho$:

$$U_R = U_{LB}\frac{\Delta x}{\Delta t}\,, \qquad L_R = L_{LB}\Delta x\,,\;\; \nu_R = \frac{\Delta x^2}{\Delta t}\nu\,, \qquad \tau_R = \Delta t\,\tau\,, \tag{371}$$

where R subscript indicates the physical variable, initial of the word “Real”. The grid spacing can be calculated with the selected number of grids for the characteristic physical length. A difficult task is to define the time step that ensures the numerical stability in a computation.

Generally, the dimensionless number characterizing the problem is used to define some of the lattice parameters. For example, using the Reynolds number similarity between the physical and lattice world, one can define either the lattice velocity or real viscosity:

$$Re_R = Re_{LB} \rightarrow Re_R = \frac{\mathrm{U}_{LB}L_{LB}}{\nu}\,. \tag{372}$$

For the open channel flow, the Froude number can be used to define the lattice gravity, $\mathbf{g}_{LB}$,

$$Fr_R = Fr_{LB} \rightarrow Fr_R = \frac{\mathrm{U}_{LB}}{\sqrt{\mathbf{g}_{LB}L_{LB}}}\,. \tag{373}$$

For the heat transfer, the Rayleigh number and the Prandtl number can be used for the determination of the product of lattice gravity and a volume expansion coefficient of fluid, $\mathbf{g}\beta$, and the relation between thermal diffusivity and viscosity, respectively:

$$Ra_R = Ra_{LB} = \frac{\mathbf{g}\beta(\theta_{hot} - \theta_{cold})L_{LB}^2}{\nu\alpha} \ ; \ Pr_R = Pr_{LB} = \frac{\nu}{\alpha} . \tag{374}$$

We use following paths to define the parameters. If the Reynolds number and the characteristic velocity and length are known, Eq.(372) gives the lattice viscosity, which further defines the relaxation time as

$$\nu = c_s^2(\tau - 0.5) . \tag{375}$$

Then selecting the grid spacing and the lattice velocity considering the low Mach number condition, usually $U_{LB} < 0.1$, the time step can be defined as

$$\Delta t = \frac{U_{LB}\Delta x}{U_R} . \tag{376}$$

Another common way is to choose the relaxation time and then defining the viscosity and the time step after finding the velocity from the Reynolds number. If the gravity is presented as an external force, one can select the lattice gravity $\mathbf{g}_{LB} < 10^4$. With the lattice gravity, the time step can be defined as (Nils, 2007)

$$\mathbf{g}_R = \frac{\Delta x}{\Delta t^2}\mathbf{g}_{LB} \ \rightarrow \ \Delta t = \sqrt{\frac{\mathbf{g}_{LB}\Delta x}{|\mathbf{g}_R|}} \ , \tag{377}$$

which is derived from Eq.(373) using the first two relations in Eq.(373). If the Froude number is known, the lattice gravity is computed from Eq.(373), directly. But the time step derived from the Eq.(376) or Eq.(377) will not satisfy the relation $\nu_R = \frac{\Delta x^2}{\Delta t}\nu$. If the computation is unstable with the defined time step, the gravity or the characteristic velocity can be reduced till the stable simulation performs.

The results from the LBM can be scaled through the following equations (Dupuis, 2002):

$$\text{Density: } \rho_R = \Delta\rho\rho_{LB} , \tag{378}$$

$$\text{Velocity: } \mathbf{u}_R = \mathbf{u}_{LB}\frac{\Delta x}{\Delta t} , \tag{379}$$

$$\text{Acceleration: } \mathbf{a}_R = \mathbf{a}_{LB}\frac{\Delta x}{\Delta \mathrm{t}^2} , \tag{380}$$

$$\text{Pressure/Stress: } P_R = P_{LB}\Delta\rho\frac{\Delta x^2}{\Delta t^2} , \tag{381}$$

$$\text{Force: } \mathbf{F}_R = \mathbf{F}_{LB}\Delta\rho\frac{\Delta x^2}{\Delta t} , \tag{382}$$

where the density scaling can be $\Delta\rho = 1000$ [kg m^{-3}] for water.

6.4 Immersed boundary modification

The immersed boundary method described in Section 5.7 is only contributed into the collision step. Three more variables, parameter β, equilibrium distribution function for moving solid surface $f_i^{\,eq}(\rho, \mathbf{u}_s)$ and additional collision term f_i^m, are implemented as shown in **Code 15** that is the modified version of **Code 8**.

Code 15: A collision subroutine for the immersed boundary modified collision step

```
      subroutine collision
      include 'paramc.h'
      bb=0.     ! B parameter in IB
      feqib=0.  ! initialization for temporal Dfs for solid
      omef=0.   ! initialization for additional term
! force on fluid
      do i=0,n
      do j=0,m
      if(flag(i,j).eq.1.or.flag(i,j).eq.2) then
         compute force
           do k=0,8
             compute feq(k,i,j)
             compute feqib(k,i,j) with Eq.193
           end do
      end if
      end do
      end do
! collision step
      do i=0,n
      do j=0,m
      if(flag(i,j).eq.1.or.flag(i,j).eq.2) then
         compute bb with Eq.362
         do k=0,8
         compute omef(k,i,j) with Eq.361
         perform collision f(k,i,j) with Eq.365
         end do
      end if

      end do
      end do

      return
      end
```

The motion of the moving surface, i.e. it is a melting front in a liquid-solid phase transition problem, does not require extra implementation in the code. However, a body immersed in the fluid is moving through a fluid flow, different case from the liquid-solid phase transition of passively fixed ice, one must implement the equation of motion for the moving immersed body.

6.5 Moving body simulation algorithm

The motion of an immersed body can be resolved by the equation of motion given in Eq.(364). The implementation of the equation of motion can be done in either dimensional or dimensionless form. If the motion of the body is tracked by dimensional world, the LB variables must be scaled into the physical variables using Eqs.(378-382). Otherwise, in a dimensionless form, results after a calculation of the dimensionless equation of motion are scaled into physical results. As applications of the immersed boundary motion by the IB-LBM founds in the many literatures (Feng & Michaelides, 2004) (Strack & Cook, 2007) (Feng, et al., 2007) (Fukumoto, 2015), the implementation of the IB-LBM is simple and straightforward. We use the following finite difference equations for the time advance of the velocity and position in dimensional form:

$$\begin{cases} \mathbf{u}_R^{n+1} = \mathbf{u}_R^n + \dfrac{\Delta t \mathbf{F}_R}{m} \\ w_R^{n+1} = w_R^n + \dfrac{\Delta t \mathbf{T}_R}{I} \end{cases} \text{and} \quad \begin{cases} \mathbf{x}_R^{n+1} = \mathbf{x}_R^n + \Delta t \mathbf{u}_R^{n+1} \\ \theta_R^{n+1} = \theta_R^n + \Delta t \mathrm{w}_R^{n+1} \end{cases}, \tag{383}$$

where

$$\mathbf{F}_R = \frac{1}{2}(\mathbf{F}_R^n + \mathbf{F}_R^{n-1})\,. \tag{384}$$

In addition, the solid flagging for the immersed boundary should be used to capture the successive motion of the immersed body. If a body is allowed to be floating, the calculation of the submerged part will take a place to get correct the hydrodynamic forces.

7 NUMERICAL EXAMPLES AND VALIDATIONS

Here we show validations for the LBM using simple, widely solved problems.

7.1 Fluid flow simulations

7.1.1 Flow past a square cylinder

To demonstrate simple fluid flow interacting with solid surfaces, we solved a flow past a square obstacle placed in a channel. The numerical simulations have been carried out in a domain that corresponding to (Ochoa & Fueyo, 2004) and shown in **Figure 64**. This domain was tested by several authors in turbulence flows with the LES, DNS and even experiments. In the current study, an obstacle with side length of D=20 nodes is chosen and the simulations were performed in the dimensionless form. The Reynolds and Strouhal number can be defined as;

$$Re = \frac{3u_o D}{\tau - 0.5}, \qquad St = \frac{fD}{u_o}, \tag{385}$$

where u_o and f are an inflow velocity and the vortex shedding frequency, respectively. For bluff bodies, important parameters are the drag, lift and pressure coefficients. They are estimated with the *rms* and *mean* values:

$$C_d = \frac{2|F_x|}{\rho u_o^2 D}, \; C_i = \frac{2F_y}{\rho u_o^2 D}, \; C_p = \frac{2(p - p_o)}{\rho u_o^2}, \tag{386}$$

where $p(=\frac{\rho}{3})$ and F are the pressure and forces acting on the obstacle surface, p_o is the reference pressure. The forces on a single node can be calculated by momentum exchange with the surrounding all possible fluid nodes as follow (Ladd, 1994);

$$F(x,t) = c_i[f_i(x,t) - f_{\bar{\imath}}(x + c_i \Delta t, t)], \tag{387}$$

where $\bar{\imath}$ is the opposite direction of i. If one uses the immersed boundary method, the force will be defined by using Eq.(363), but we don't.

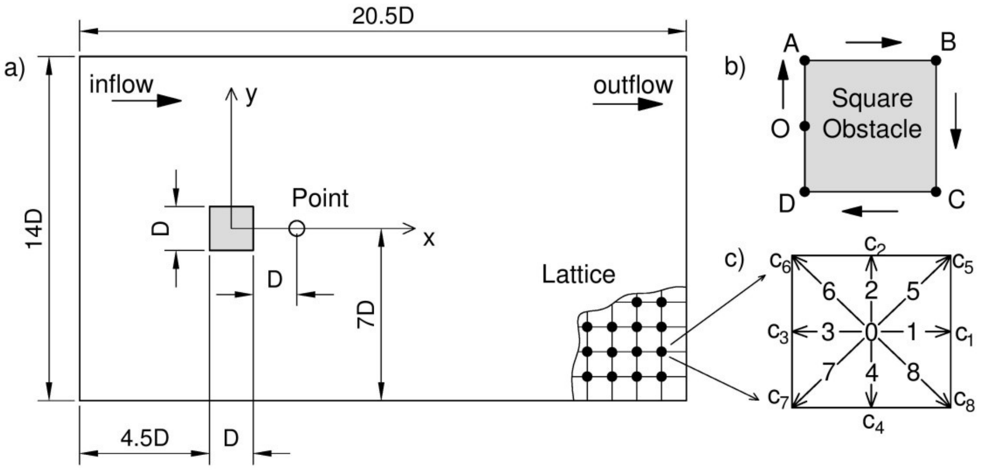


Figure 64. Schematic illustration of a computational domain and a lattice: (a) domain dimensions and a position of an obstacle, (b) the pressure measurement around the obstacle, (c) a lattice arrangement.

First, we run a simulation at the Re=100 and the results were compared (see **Figure 65**) to the other numerical results in (Breuer, et al., 2000). The results did not agree exactly with the other studies, because the boundary conditions used in the simulation were different. But a general characteristic of the velocity profiles was the same as the results, seen in **Figure 65**. In (Breuer, et al., 2000), the drag coefficient and the Strouhal number were estimated as 1.35 and 0.14, while in our simulation, they were 1.27 and 0.15, respectively. These similar results show that our numerical code properly works in laminar flows. In **Figure 66**, a comparison between the vorticity fields and velocity fields are given. The detailed discussion can found in (Ayurzana, 2016).

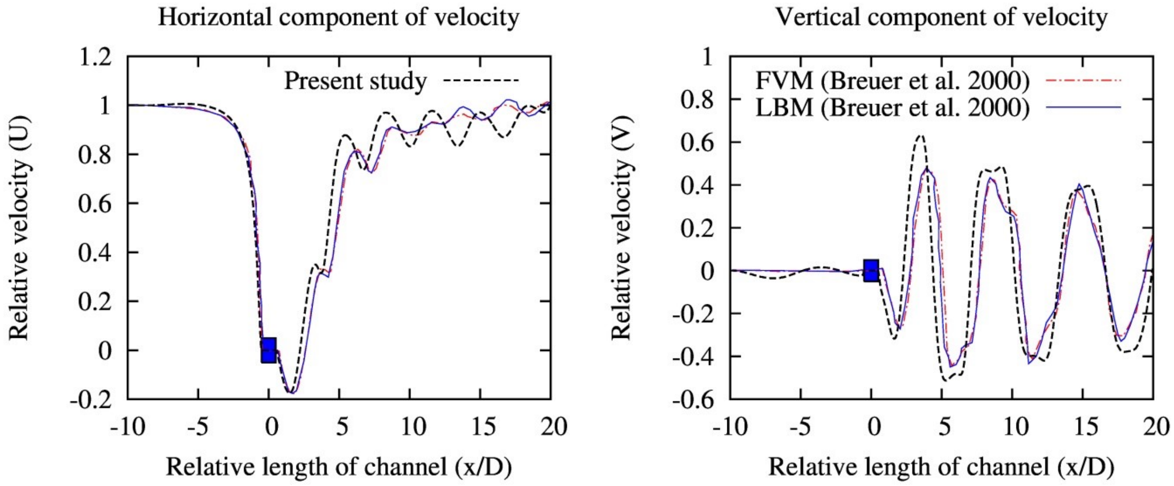


Figure 65. Profiles of horizontal and vertical velocity component in the streamwise direction obtained by the extended LBM are compared with the results in (Breuer, et al., 2000).

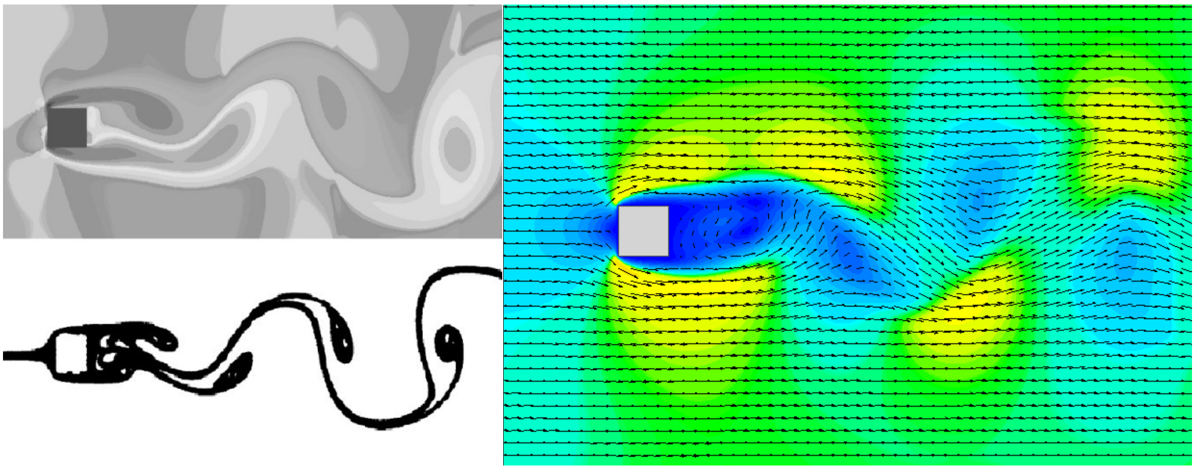


Figure 66. On the left, the vorticity around the square obstacle at Re=100. The lower one is the result of (Breuer, et al., 2000). In the right, the velocity vector is plotted.

Next study of interest was a high Reynolds number flow simulation with the turbulent model explained in Section 5.4.3. For a flow past bluff bodies, flows with the Reynolds numbers higher than 300 are considered in turbulent. It is very interesting to check the effect of the turbulence model on the result by the standard LBM. In order to explore the effect of the turbulent model, the standard LBM (StLBM) and the extended LBM (ExLBM) in the same condition is applied for flows past a square obstacle at Re=300. It is conceptually expected that the flow predictions by

StLBM and ExLBM must be the same at a certain time. However, the results show that the velocity field was the difference at the same time. Then we found the similar velocity profiles to check the time difference due to the turbulence model in the LBM. **Figure 67** and **Figure 68** shows the velocity profiles at time T=24000 for the StLBM and at T=23780 for the ExLBM. The time difference induced by the turbulent model was 220 time steps. In other words, the ExLBM was delayed by some time compared to the StLBM. It is because that the total relaxation time is locally increased by the turbulent eddy viscosity, as given in Eq.(322).

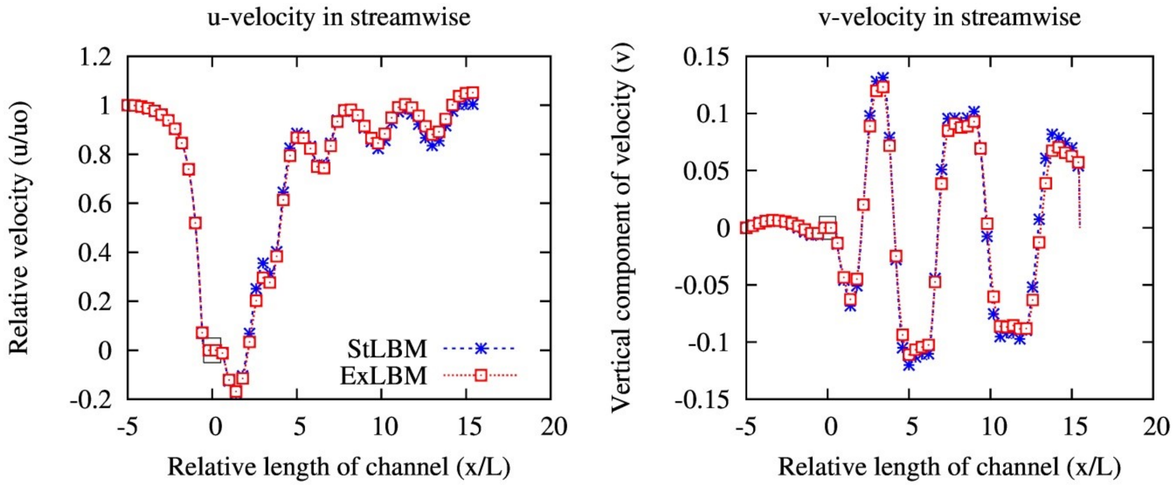


Figure 67. Comparison of velocity profiles in the streamwise direction computed by the StLBM and the ExLBM.

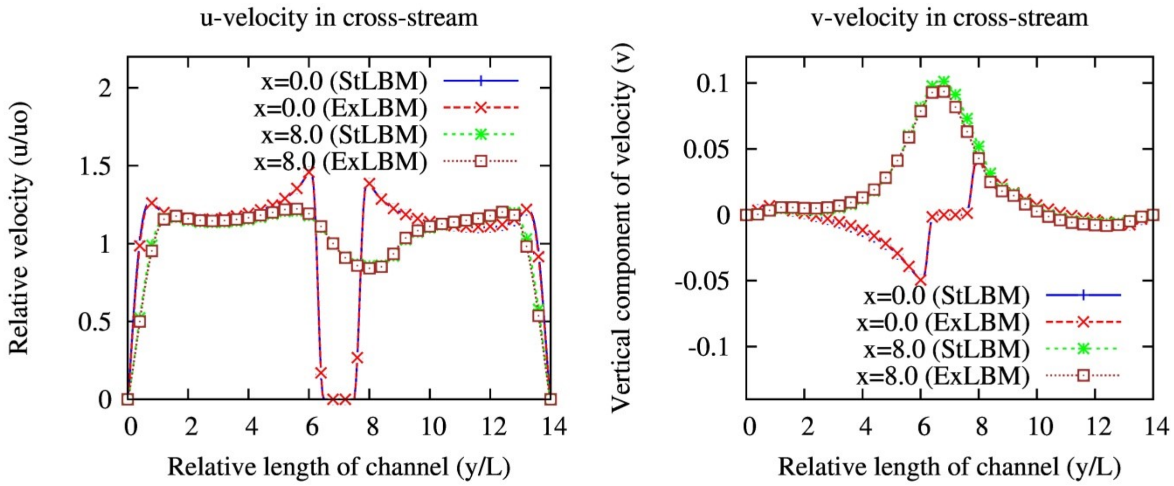


Figure 68. Comparison of velocity profiles in the cross-stream direction computed by the StLBM and the ExLBM.

It should be noted that the flow pattern and its magnitude were almost the same in the results by the StLBM and the ExLBM at different times that means the LBM can produce better result with the sub-grid scale turbulent model.

To validate our code properly in turbulent regimes, we run next simulations with Re = 22000 and compared the pressure coefficient to the result of (Liu, et al., 2008) shown in **Figure 69**. In our simulations, the pressure coefficient grows up more than the result of (Liu, et al., 2008) at the back face of the obstacle. Except this overestimates, the values of the other points were agreeing with the results of (Liu, et al., 2008). The streamlines around the obstacle is compared with the result of (Liu, et al., 2008) and that was also in good agreement. The drag, lift coefficients and the Strouhal numbers were determined as $C_d^{rms} = 3.06$, $C_l^{rms} = 2.28 \times 10^{-2}$, and St = 0.126 in the current study while these parameters in (Liu, et al., 2008) were $C_d^{rms} = 2.1$ and St=0.134 , respectively. The recirculation of eddies observed in upper and lower part of the obstacle reported in (Liu, et al., 2008) can also be resolved with the use of finer grid in our simulation.

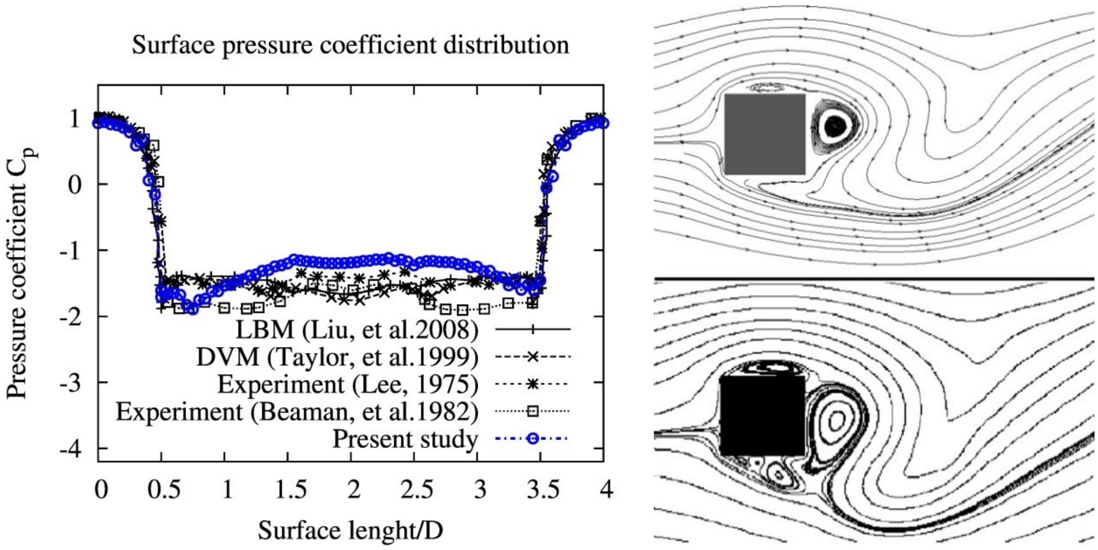


Figure 69. Figure on the left shows the pressure coefficients on the obstacle surface from different studies [10] and a starting point of measurement is O shown in **Figure 64**. Figure on the right shows the streamlines comparison, the upper one is the current study while the lower one is the result of (Liu, et al., 2008) at Re=22000

As used the same concept, we measured the vertical velocity as a time function at the control point after the obstacle (see **Figure 64**) and the results are compared with the results obtained from conventional methods (Liu, et al., 2008) incorporated in the Smagorinsky model in **Figure 70**.

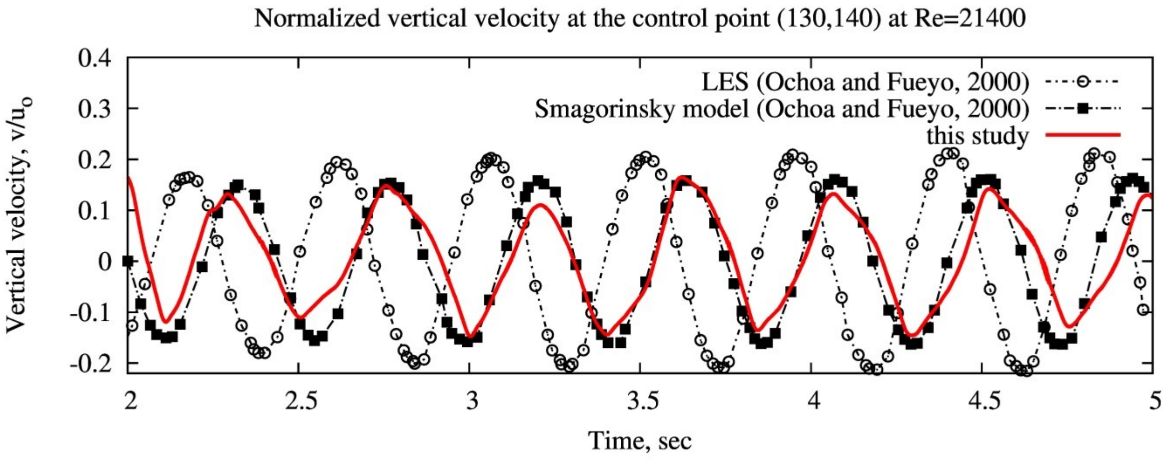


Figure 70. Time series of the vertical velocity on the observation point behind the obstacle by the different models at Re=21400.

In (Ochoa & Fueyo, 2004), the conventional high order computational framework, called PHEONICS, was used to solve 2D&3D simulations with the different models of turbulent, e.g. LES and Smagorinsky model. The code was also applied for that flow of Re = 21400 and the results are compared in **Figure 70**. From the comparison in **Figure 70**, the results by the present LBM and (Ochoa & Fueyo, 2004) are in good agreement while the result by the LES (Ochoa & Fueyo, 2004) has higher amplitudes for the vertical velocity profile. It reveals that the results are hugely dependent from the model used in the turbulent. With the high Reynolds number flow, two cases have been compared and the good agreement found. One of the growing aspects in the research tendency among the LBM is the turbulence. We had reached the flow past a square obstacle of up to Re = 3.82x10^5 where we observed the some oscillation in the vicinity of the obstacle edge. It is notable that a combination of chosen values for viscosity and inlet velocity effects for the results and numerical oscillations.

7.1.2 Doubly periodic shear flow

Here, we discuss the comparison between the standard and entropic LBM (ELBM), which is described in Section 5.4.4. The results were reported in (Ayurzana, et al., 2017). A doubly periodic shear layer flow is often considered as a benchmark (Karlin, et al., 2014), (Brown, 1995) case of an under-resolved simulation of smooth flows with sharp features. Here, a shear flow solved by the standard LBM and the entropic LBM in resolutions of N=128, 256, 512, and 1024. Initial conditions for a flow field are given by

$$\begin{cases} u = \begin{cases} u_0 tanh\left(k\left(\frac{y}{N}-0.25\right)\right), y \leq N/2, \\ u_0 tanh\left(k\left(0.75-\frac{y}{N}\right)\right), y > N/2, \end{cases} \\ \qquad v = \delta u_0 \sin\left(2\pi\left(\frac{x}{N}+0.25\right)\right), \end{cases} \tag{388}$$

where $k(= 80)$ is the parameter controlling the width of the shear layer and $\delta(= 0.05)$ is the parameter creating small perturbation of velocity in the y-direction, which initiates a Kelvin-Helmholtz instability. The turning over time of the shear layer is defined as $t_c = N/u_0$, where $u_0(= 0.04)$ is the initial velocity defining the Reynolds number for flow as $Re = u_0 N/\nu$. In numerical test, the Reynolds number was set at 30000. The simulation results of vorticity at t_c by the two methods are shown in **Figure 71** in order of grid resolutions. The simulation of the standard LBM was failed before the convergence of the under-resolution at a low resolution of domain, for instance, unstable appears at t = 2200 before the vortex roll-ups in the interface of shear layers, shown in the top figure of **Figure 71** (a).

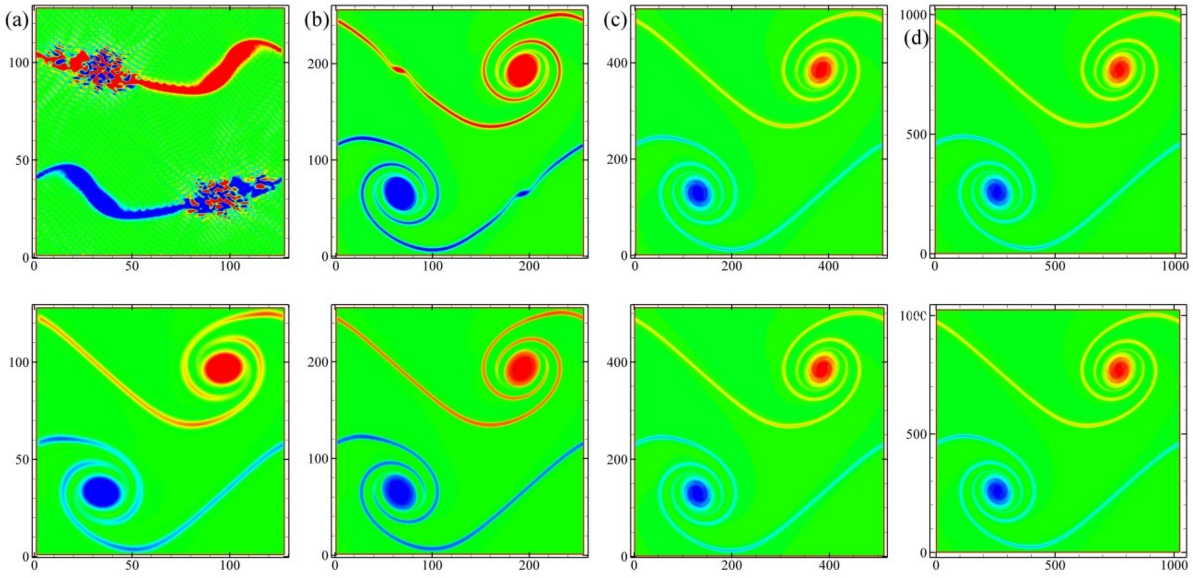


Figure 71. Doubly periodic shear layer flow simulations on the different grid resolutions. Top figures were produced by using the standard LBM, while bottom figures were produced by using the entropic LBM. Vortex roll-ups appears at (a) tc=3200, (b) tc=6400, (c) tc=12800 and (d) tc=25600.

As an increase of grid resolution, the relaxation time of the standard LBM increases and the computation become rather stable. The computation successfully survived on the further cases of simulations (except N=128) in the case of standard LBM. However, as shown in the top of **Figure 71** (b), two small additional roll-ups created at linear parts of the vortex field, which are the source of unstable solutions. As expected shapes of the vortex, the ELBM produces the flow field at the even smaller resolution of the grids and the additional roll-ups, emerging from unstable of the numerical solution does not appear in any case of simulations. This stable simulation can be performed at very small kinematic viscosity due to the coarse grid and high Reynolds number. The stabilizer in the ELBM is a self-adaptive local parameter and impacts on the resolving of the flow fields. The distribution of the stabilizer and its value range do not depends on the Reynolds number, as shown in **Figure 72**.

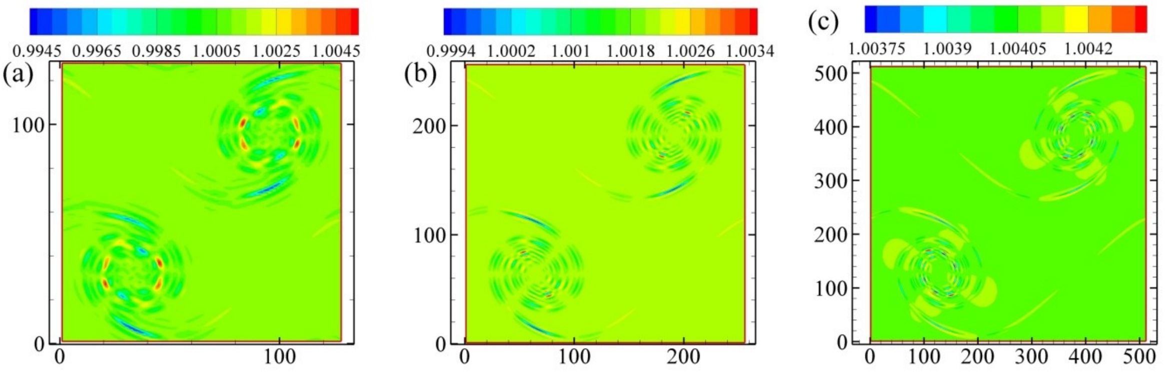


Figure 72. Snapshot of the temporal and spatial distribution of the stabilizer, γ^*.

For the cost of computations, the ELBM code was 2 to 4 times slower than the StLBM code. However, the ELBM can produce unconditionally stable simulations for various flow regimes, in particular, for turbulent flows. The good agreement between results by the two LBM methods with fine grids shows good validations for the methods.

7.2 Heat transfer simulations

To validate the LBM code extended to the scalar field evaluation, the natural convection flow due to a heat transfer in a square enclosure is considered as a benchmark problem. A configuration of the problem is simple, often appears in natural convection flows among literatures, and consists of a square enclosure with a hot wall at the left, a cold wall at the right and heat insulated walls at the top and bottom, as shown in **Figure 73**. For a fluid flow, all walls were modeled as no slip walls. For heat transfer, the Dirichlet boundary condition-2 was used on the heated wall while the anti-bounce-back boundary condition was imposed on the cold wall. The insulated walls were modeled by the adiabatic boundary condition. All prescribed boundary conditions for heat transfer is given in Section 5.3.2.

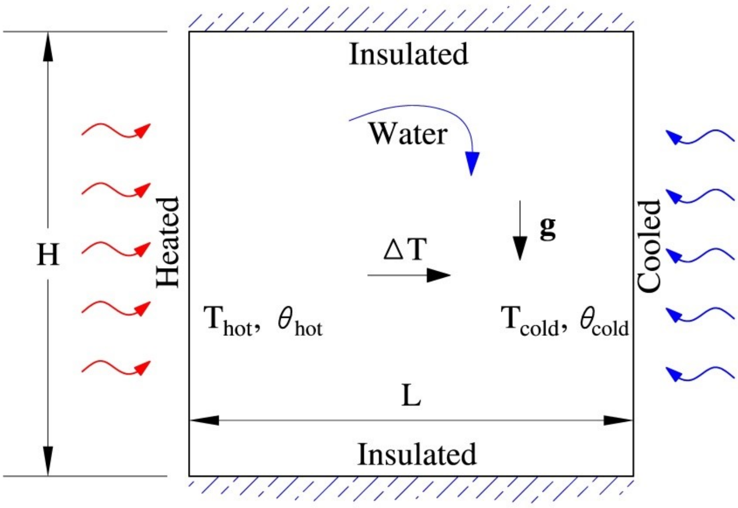

Insulated
Water
H
Heated
ΔT
g
Cooled
$\mathrm{T_{hot}}$, θ_{hot}
$\mathrm{T_{cold}}$, θ_{cold}
L
Insulated

Figure 73. Schematic of the flow configuration for natural convection of water.

The natural convection problem can be characterized by the Rayleigh, Prandtl and the Nusselt numbers,

$$Nu = \frac{hL}{k}. \tag{389}$$

Here, h is the convective heat transfer coefficient and k is the thermal conductivity of a fluid. For this simulation, Rayleigh numbers of $Ra = 10^3 \sim 10^7$ had been considered and the Prantdl number was fixed at $Pr = 11.58$. The temperature difference between two side walls (hot and cold) was fixed at 8°C for all runs. From the Ra number in Eq.(356), one can find a term $\mathbf{g}\beta$ for a an estimation of the force term in Eq.(204) and the determination of grid spacing and time step to obtain the real physical values from the LBM results at each time step. We also used following dimensionless variables for coordinates and velocities as $X = \mathbf{x}/L, Y = \mathbf{y}/H, U = \mathbf{u}/(\alpha/L\,\sqrt{(Ra\,Pr)})$. The dimensionless temperature was calculated by Eq.(352) and the dimensionless temperature difference were $\Delta\theta = 0$. A main part of this section is reproduced from one of our work in (Ayurzana & Hosoyamada, 2016). The temperature field and the stream functions for five different Rayleigh number flows are plotted in **Figure 74**.

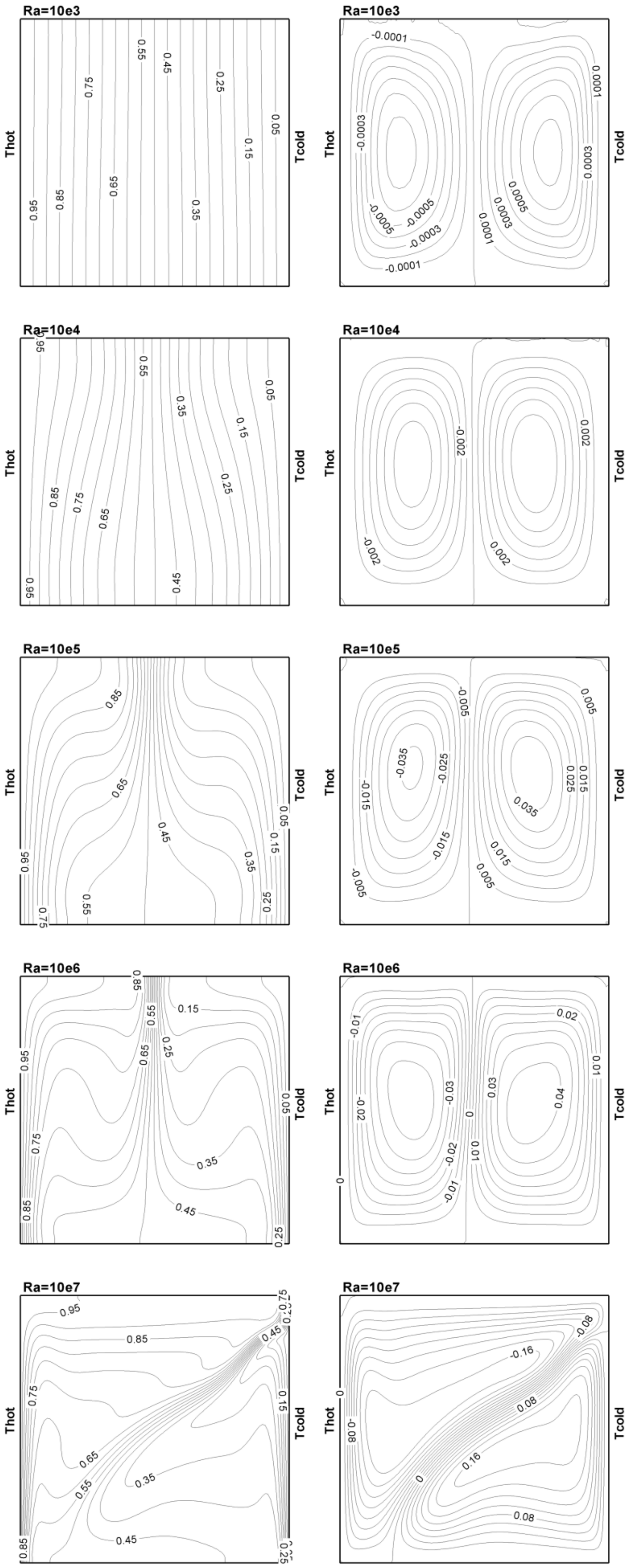


Figure 74. Evolution of the temperature field (in the left) and the stream function (in the right) for $\Delta T = 8°\mathrm{C}$

As the characteristic of the natural convection flow, a flow field is separated by the density inversion plane like two rolling regions. But it is not the character of the flow with the Ra = 10^7, where the density inversion plane is placed in the diagonal of the domain. It should be noted that we could not find the comparative data of the natural convection flow of water with the Ra = 10^7 in order to evaluate our results at the same Ra number. The temperature field and the stream function for another case of the Ra were in good agreement with the result reported in (Tong & Koster, 1993). In addition, the Nusselt numbers had been compared with those taken from (Tong & Koster, 1993), plotted in **Figure 75** and the comparison in values has been reported in **Table 5**.

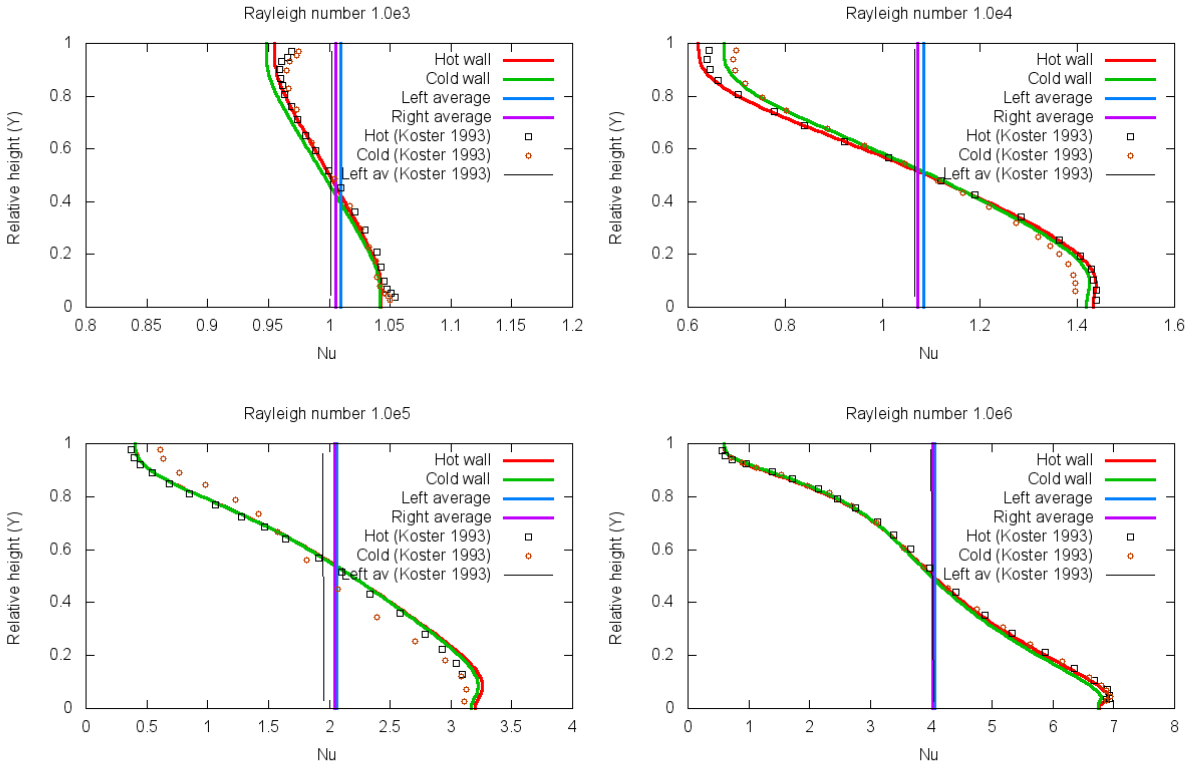


Figure 75. Comparison of the Nusselt numbers

Table 5. Summary of the numerical results at the various Ra and comparison with the results from (Tong & Koster, 1993), which are underlined.

Ra	ΔT (°C)	$\overline{Nu}$	Ψ_{max} (x10^3)	Ψ_{min} (x10^{-3})	U_{max}
10^3	8.0	1.005 1.001	1.138 1.136	1.152 1.206	0.008 0.008
10^4	8.0	1.071 1.066	3.913 3.451	3.715 3.749	0.034 0.025
10^5	8.0	2.005 2.005	4.881 4.909	4.371 5.364	0.049 0.040
10^6	8.0	4.119 4.120	6.115 5.601	6.135 5.645	0.057 0.057
10^7	8.0	9.1981	7.612	8.110	0.145

Our simulations provided the expected behavior of the flow field and the results were in good order of accuracy. From the comparison, it can be noticed that the numerical code has been validated successfully and can be applied for heat transfer problems with the presence of various flow regimes. The limitation of application to the LBM for a heat transfer problem can be related to the numerical instability emerged from whether the BGK approximation and the fixed Prantdl number. The instability can be alleviated using a stabilization method, such as the MRT, TRT or adaptive time-step. Since the relaxation time of scalar field increases with the decrease of Prantdl number, the stability condition can be improved using the varied Prantdl number with temperature. Also, a wide range of temperature could lead to instability because of the small set of velocities. The range of the temperature will be discussed in Chapter 8.

7.3 Phase change simulations in enclosure

For phase changes in this section, we only consider water and ice, their liquid-solid changes in a domain.

7.3.1 Phase change induced by heat conduction: the Stefan problem

Here we consider the Stefan problem, melting of a slab of ice with a length of 0.1 m, to validate the proposed LBM for the liquid-solid phase changes in free surface flows. The analytical solution (Alexiades, 1992) of this problem is given as

$$X(t) = 2\chi\sqrt{\alpha_R^{water} t} \text{ and} \tag{390}$$

$$T(x,t) = T_{max} - (T_{max} - T_{melt}) \frac{erf\left(x/2\sqrt{\alpha_R^{water} t}\right)}{erf(\chi)} \tag{391}$$

with the transcendence function for χ,

$$\chi e^{\chi^2} erf(\chi) = \frac{St}{\sqrt{\pi}}, \tag{392}$$

to find positions of liquid-solid interface and temperature distributions in liquid region at times, respectively. Initially, the temperature of the ice slab was at the melting temperature of T_{melt}=0°C. One side of the ice is insulated, while the other is abruptly set at the T_{max}=25°C at t=0 and it is maintained for all times t > 0 in the simulation. We set imaginary thermocouples in the slab at lengths of 0.01, 0.03, 0.05, and 0.09 m and measures the temperature in time evolution. Simply, we chose N=100 grid for the length of the slab in both the analytical and numerical solution. For the numerical solution, we use D1Q3 lattice arrangement (A.A.Mohamad, 2011) for Eq.(347) and the relaxation time is obtained from the relation

$$\tau_h = 3\alpha_R^{water} \frac{\Delta t_h}{\Delta x^2} + 0.5, \tag{393}$$

where Δt_h is the time step, which is set as Δt_h=1.0 s and Δx (=0.001 m) is the grid spacing. The constant temperature boundary condition (Alamyane & Mohamad, 2010) in Section 5.3.2 is applied to the heated side of the slab, while the second order extrapolation boundary condition (A.A.Mohamad, 2011) for scalar field is imposed on the other side. The temperature is calculated by Eq.(352) using the dimensionless temperature computed by the LBM, and then the melting front, the liquid-solid interface, is defined by the liquid fraction value using Eq.(354).

The total melting time was defined as 34.01 hours by the analytic and the present LBM (1). The comparisons of the results by the analytical and numerical methods are given in **Figure 76** and **Figure 77**. The color map of **Figure 76** is the temperature distribution estimated by the analytical solution. The result of the present LBM (1) uses the relaxation time defined by Eq.(393), whereas the LBM (2) uses the adjusted relaxation time. The best fit of the melting fronts found for the analytical solution and the LBM (2). However the total melting time with the LBM (2) was lasted for 34.72 hours. The numerically defined melting fronts in **Figure 76**, as well as temperature profiles at different times and measurement positions in **Figure 77**, shows the discrepancy in the middle of the simulation time. The good agreement has ben observed before 8.5 hours and after 30 hours in the experiment, as shown in **Figure 76** and **Figure 77**. The temperature profiles at specific times, which are the times the melting front reaches the imaginary thermocouples, with the analytical solution show the linear in space while the profiles defined by the LBM show the deviating in space. The maximum errors of the LBM compared to the analytical solution are reported in **Table 6**. The Stefan problem gives the validation for the phase transition of ice in tiny volume ignoring the fluid flow, as well as the free-surface condition.

Table 6. Maximum errors of the LBM compared to the analytical solution

Cases	Relaxation time, τ_h	Maximum error (%)		
		of melting front	of temperature profiles at times	of temperature profile at positions
LB model (1)	1.727	4.04	5.59	5.00
LB model (2)	1.755	1.32	7.80	4.53

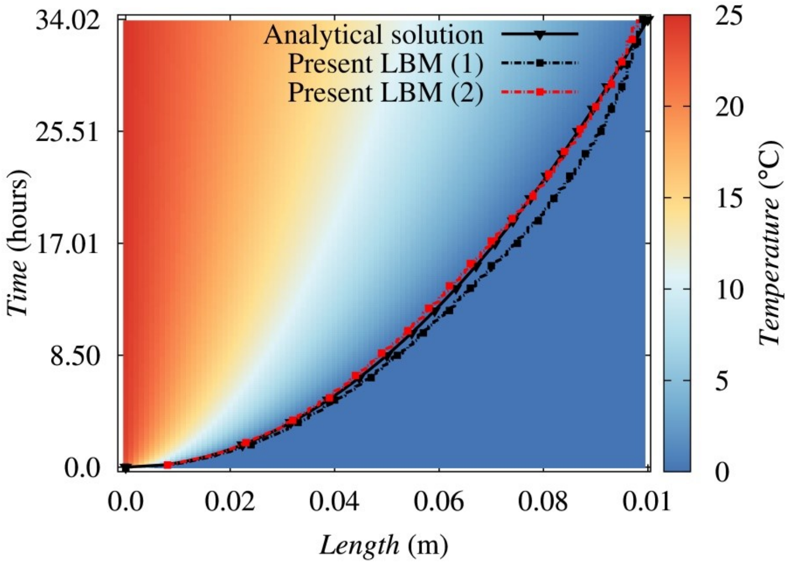


Figure 76. Time history of the temperature distribution and the melting front locations by the analytical and numerical methods.

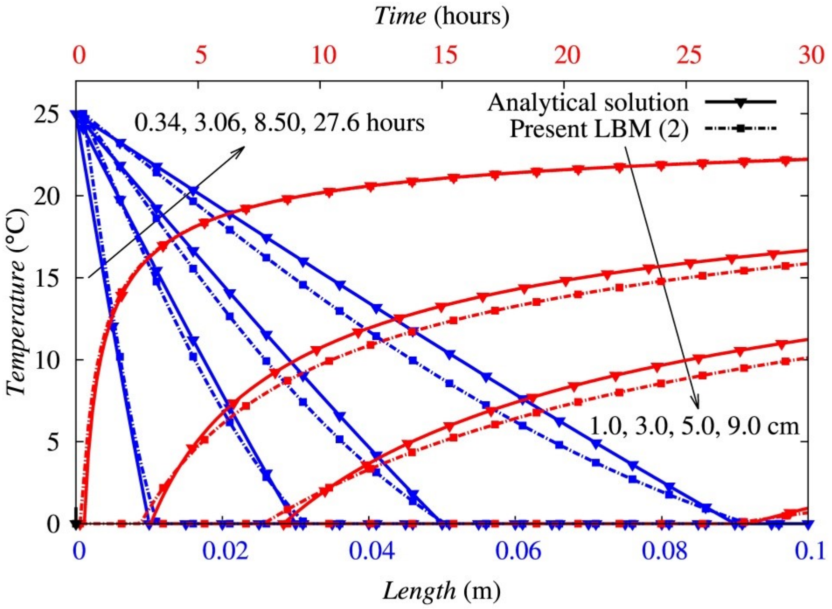


Figure 77. Temperature profiles at different times and at different positions: blue lines for the temperature distributions for times and red lines for the temperature distributions for positions.

7.3.2 Ice melting by convective flow

We applied the method to an ice melting simulation in a rectangular enclosure, which had been studied experimentally and numerically in (Virag, et al., 2006) and (Arid, et al., 2012). The problem configuration is given in **Figure 78** (a). The results of this study were reported in (Ayurzana & Hosoyamada, 2016) and it is a combination problem of Section 7.2 and Section.7.3.1.

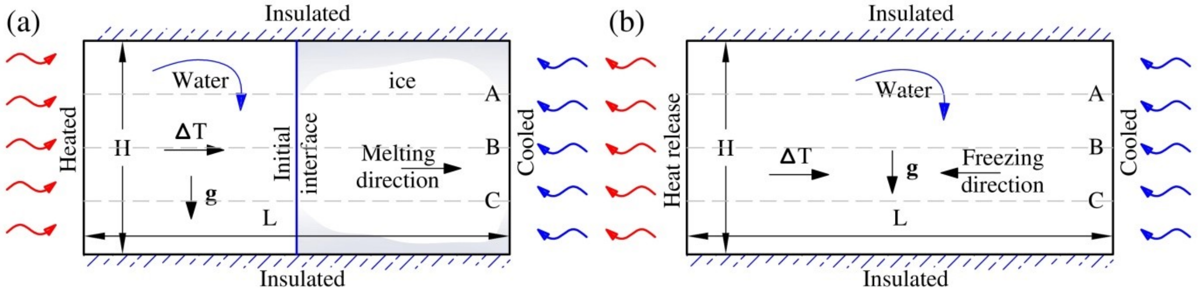


Figure 78. Computational domains for the ice melting (a) and water freezing problem (b).

In the phase change problems, the Stefan number $St = c(T_{hot} - T_o)/L_f$ is used to control the simulation and the Fourier number $Fo = t\alpha/L^2$ is used to analyze the results by means of the time dependent heat transfer. Except the numerical model described in Section 5.6, there is very simple approach to evaluate the liquid fraction value. For that method, the liquid fraction is computed (Semma, et al., 2008) as

$$L_f = \begin{cases} 1 & \text{for } \theta > \theta_{melt} + \varepsilon \\ 0 & \text{for } \theta < \theta_{melt} - \varepsilon \\ \dfrac{(\theta - \theta_m + \varepsilon)}{2\varepsilon} & \text{for } \theta_{melt} + \varepsilon \le \theta \le \theta_{melt} + \varepsilon \end{cases} \tag{394}$$

and we call it as the method without an enthalpy update. We had simulated melting front by two different ways; without an enthalpy update (Eq.(394)) and with an enthalpy update (Eq.(354)), respectively. They had shown slightly distinct melting fronts, shown in the first plot of **Figure 79**. The closer results with the results of (Arid, et al., 2012) have provided from the way with the enthalpy updates and the melting fronts have been compared with (Arid, et al., 2012) and (Virag, et al., 2006) in the second plot of **Figure 79**. For melting front profiles, notable discrepancies with (Arid, et al., 2012) have been observed on the first four profiles in the second plot of **Figure 79**. However, at the same Fourier numbers during the simulation, the average melting front positions and shapes have the same positions and tendencies. Melting processes take place more intensively at the bottom of the enclosure rather than at the top.

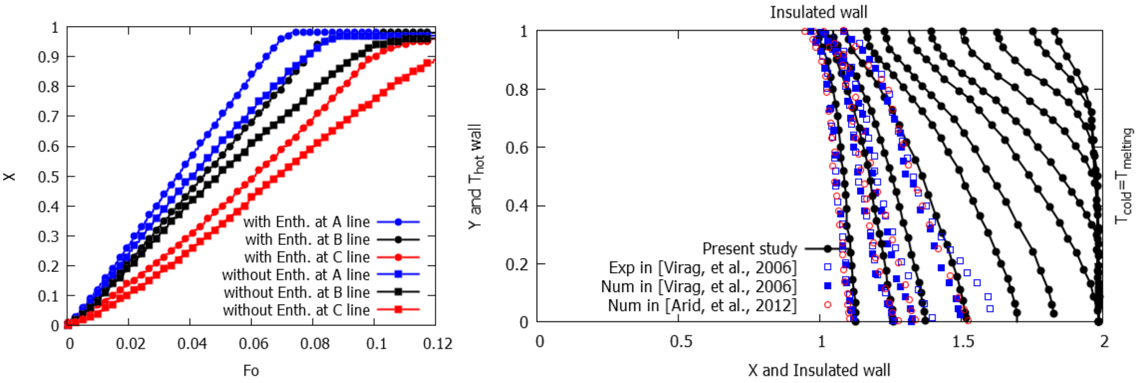


Figure 79. The first plot represents the melting front measurements at three horizontal lines: Y=0.75 (A), 0.5 (B), 0.25 (C line) by different evaluation of phase change explained above. In the latter plot, comparison of melting front evolutions at the different Fourier numbers; Fo=0.0089, 0.017, 0.023, 0.032, 0.044, 0.053, 0.065, 0.074, 0.086, 0.098, 0.11, 0.119 etc, are represented.

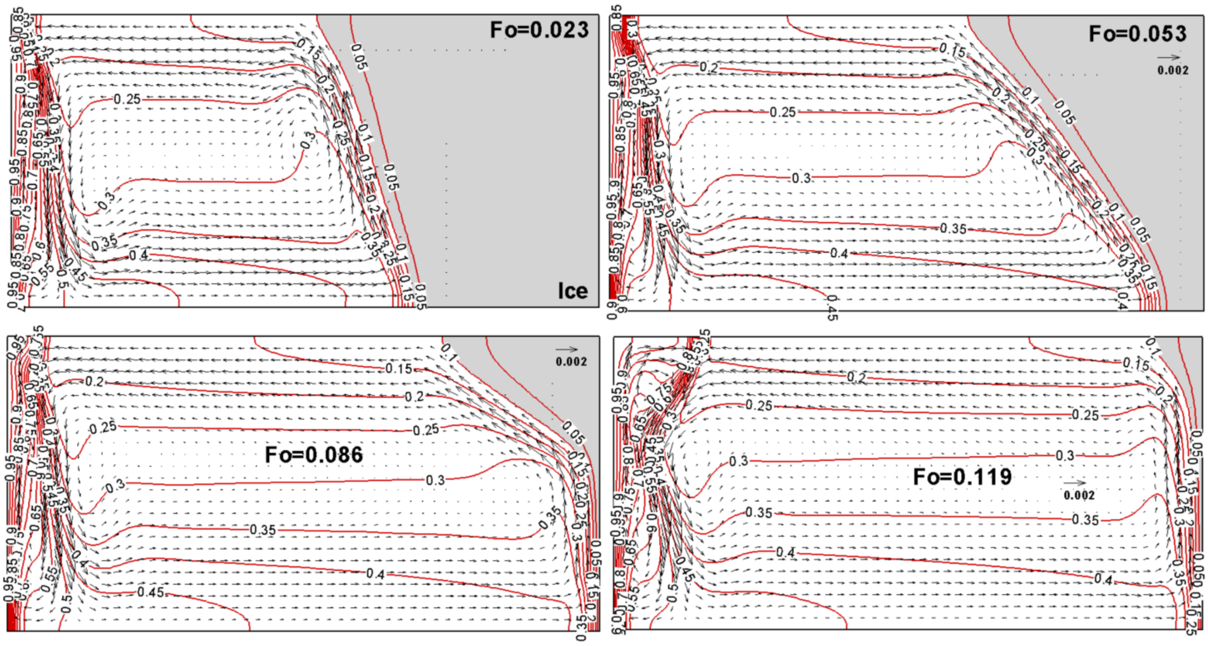


Figure 80. Velocity vector and temperature field at three different Fo number.

The melting front reached the right wall at Fo=0.065. It took Fo=0.12 to finish the melting process of ice. The temperature profiles and velocity vector are given in **Figure 80** at the selected Fo numbers. Overall predictions with the LBM were qualitatively correct with respect to other numerical results. Note that the results are non-dimensional.

7.3.3 Water freezing in a rectangular enclosure

We simulated a freezing process on the same configuration (see **Figure 78** (b)) with the preceding melting simulation in order to examine the performance of the LBM for a freezing phenomenon and to find the spent time to complete a phase change in half of the enclosure. It is obvious that freezing and melting are opposite phenomena, which means that they do not take the same time nor shows same interface shape between the liquid and solid state. Initially, the water in domain was at the temperature $T = 8°\mathrm{C}$. After the simulation started, the temperature at the right wall abruptly maintained with temperature $T_{cold} = 0°\mathrm{C}$ and kept thereafter. It is observed that the freezing takes place intensively after the convection flow damped down substantially in the enclosure. At the beginning, the high Nusselt number in **Figure 81** (a) shows the convection dominated heat transfer at the left wall. The great decrease of the Nusselt number is evidence of the convection flow disappearing. The convection flow was very weak after Fo=0.4, because there was a small amount of temperature difference in domain, as shown in **Figure 81** (b). The conduction rate of heat transfer is dominated after Fo=0.3 rather than the convection rate of heat transfer, which can be seen in **Figure 82**. Two ways to treat the phase change, with the enthalpy update and without the enthalpy update, have shown special distinct at Fo=0.56 and Fo=0.93 in **Figure 82** (b) and it might depend on the enthalpy updates with previous liquid fraction values. In this simulation, the freezing processes are about 12 times slower than the melting processes to complete the task that the half region of enclosure must melt or froze.

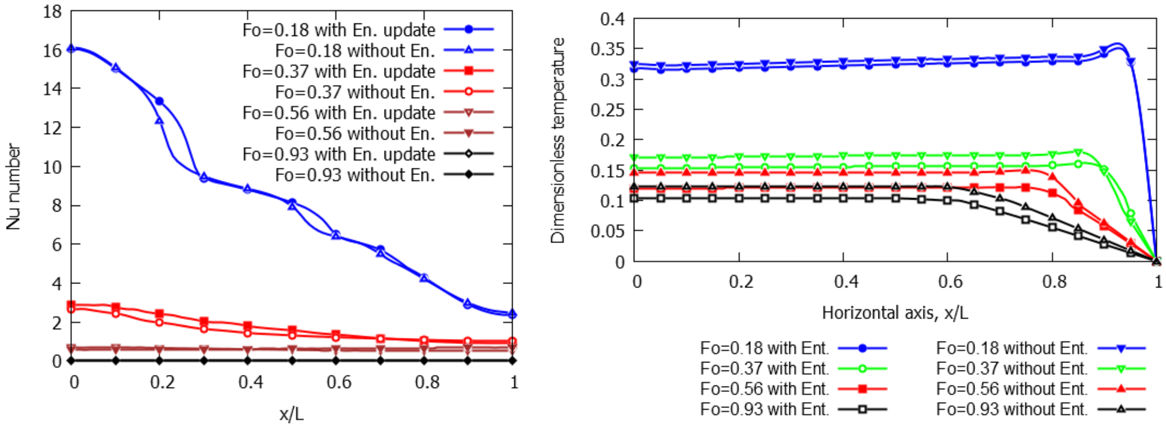


Figure 81. Results at the different Fo number by two different approaches in the LBM: (Left) The Nusselt number at left wall and (Right) the temperature profiles through the line A.

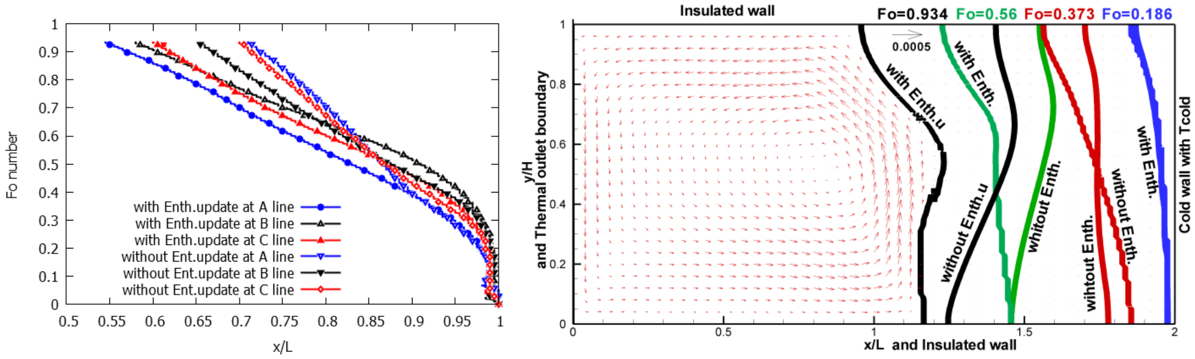


Figure 82. The first plot shows the freezing front measurements at the different Fo number at the same lines of earlier melting simulation. The second plot shows the freezing fronts by two different phase change treatment and the velocity vector of the case of the enthalpy updated treatment at Fo=0.934.

Based on the above melting and freezing problems solved by the LBM, the phase transition treatment with the enthalpy update scheme might be more stable and accurate than without the enthalpy update scheme.

7.3.4 Ice melting from the bottom

Here we solved other numerical simulation of ice melting in a square enclosure which 0.2 m in the each side. Initially, the enclosure was filled with ice in a ready-to-melt case, i.e. the temperature of ice was t_{melt} = 0°C and the latent heat was removed. In the bottom side, the constant heat is maintained while the walls in right/left were assumed to be insulated from heat. The top wall is maintained with t_{melt}. Heat transferred by conduction in the initial stage of a simulation. After the creating sufficient space for the convection flow by the conduction dominated melting, the convection flow has started by the density difference in the melted zone. A density influence by the temperature is calculated by a force term expressed by the non-Boussinesq approximation in the numerical procedure. Depending on the Rayleigh number (Ra), the number of fingering and its height was different. Each finger has shown the flow circulation in it. In **Figure 83** (a), the melting interfaces of Ra=10^7 were shown with the velocity vector at the time 11.91 minutes. The convection flow created several fingering and they joined to the two big circulation flows immediately. Those two big circulations dominated for further heat transport and melted ice until the top boundary. If the Rayleigh number is low, the conduction heat transfer is dominated and took long time to start the convection heat transport. The maximum velocity induced by heat was around 0.01 m/s, while the maximum Nusselt number at the bottom wall was around Nu=40, shown in **Figure 83** (b). After the ice melted until the top boundary, the temperature field in the middle area of the enclosure was almost constantly distributed, as shown in **Figure 84**. The numerical results were what we expected and the same as the results of researcher in the same field, which has not yet officially been published elsewhere (Esfahani, et al., 2016).

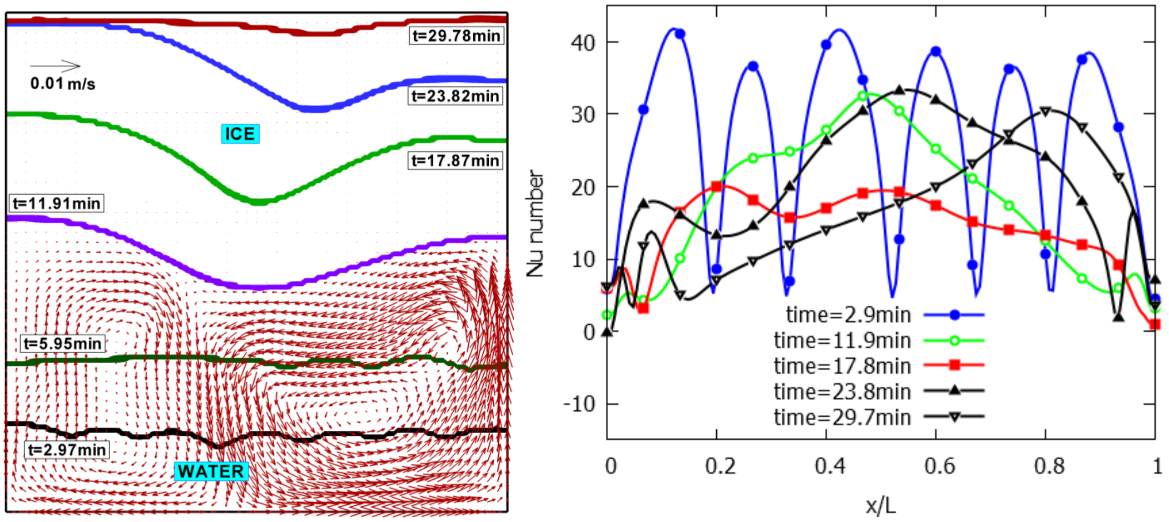


Figure 83. (a) Melting fronts of the ice melting from the below and (b) the measured Nusselt number at the bottom wall at the different melting times.

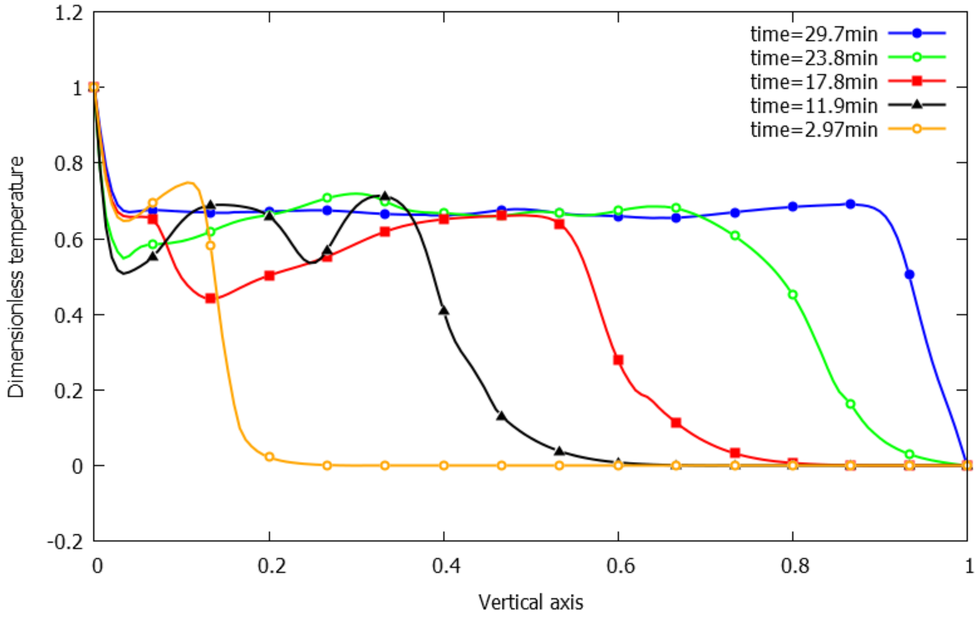


Figure 84. The temperature profiles at the different melting time.

7.4 Free surface flow simulations

Here, the model described in Sections 5.5 and 6.1 is applied to free surface flows to demonstrate the application of the LBM. The results are reported in (Ayurzana, et al., 2016).

7.4.1 Dam break analysis

First, we applied the Free-surface LBM on a dam break benchmark problem to validate the performance of the algorithm. The simulated results compared against the experimental results conducted on the same geometrical configuration shown in **Figure 85**. For the wall, a slip boundary condition was imposed. To validate numerical simulation, we measured a dimensionless waterfront position X^* at a dimensionless time T^* as,

$$T^* = t\sqrt{ng/L}\,, X^* = \frac{x}{L} \tag{395}$$

and the time evolution of water depth at specific points (A and B) depicted in **Figure 85**. In Eq.(395), $n = (H/L)$ is the aspect ratio of a water column, H and L are the initial height and width of the water column, x is the waterfront displacement at time t. We conducted two numerical simulations on grids of 200x400 and 300x600 to investigate the grid resolution independence. The time steps $\Delta t_{400} = 0.00007$ s and $\Delta t_{600} = 0.00006$ s were used, respectively. The parameters used in the simulations are determined through the parameterization formulas explained in Section 6.3.

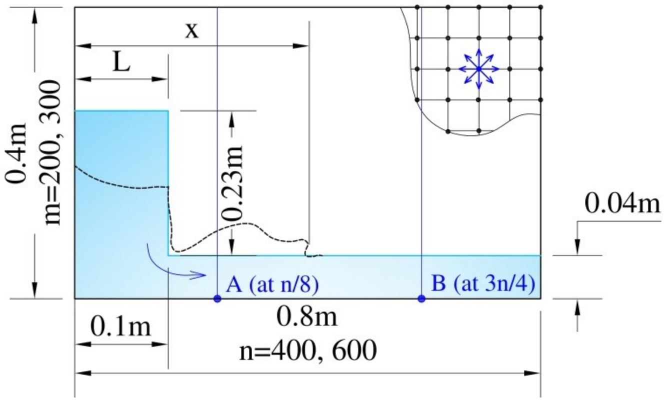


Figure 85. Schematic sketch of a dam break problem with a wet bottom. A lattice on the grid is depicted on the upper right corner of the scheme. Only dimensions of the height m and width n for the numerical tests had been attached to the scheme.

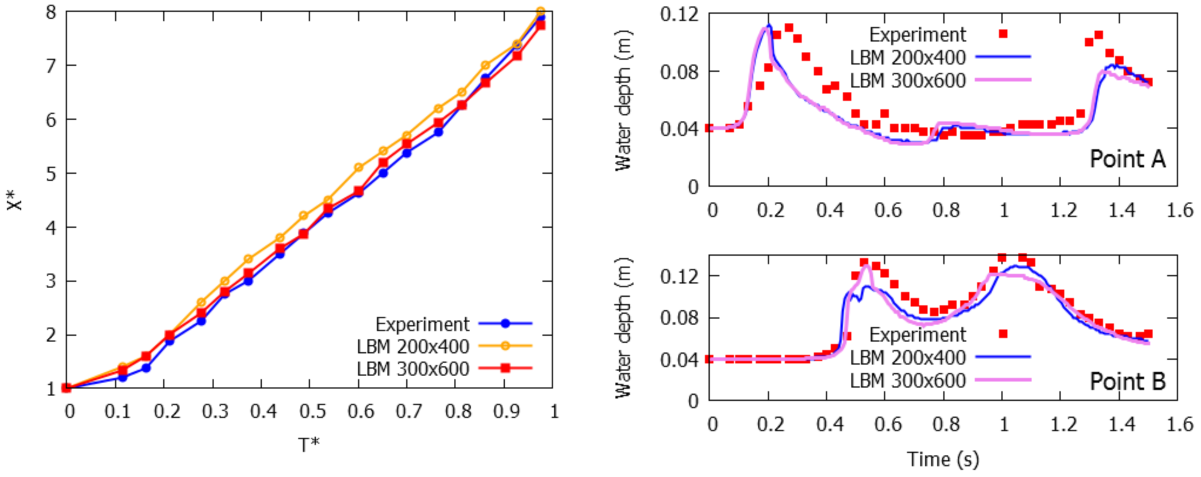


Figure 86. Comparison of the melting front displacement and time evolutions of water level at the control point A and B in the experimental and numerical tests.

It seemed that the grid resolution has a slight influence on the numerical results since the curve of the case of 300x600 has been plotted very nearly with the experimental one the first plot in **Figure 86**. In the numerical experiment, a plate gate, separating the water column from the wet bottom in the tank, had not yet been included. The effect of the gate removing in the lab experiment appears with water depth evolution on the point A at time 0.2 s to 0.4 s on the second plot in **Figure 86**, where experiment's time had been delayed. Except some offsets, the time evolution of water depths has the same tendency with experimental one in **Figure 86** (b). Some discrepancies are considered as some deficiencies of data extracted from the video frame in the experiment. Because a normal video camera had been used to capture the phenomena in the laboratory experiment, the image has some perspective representation, which can be seen in **Figure 87** (a). For the sake of convenience to printed document, we used edge detecting effect on the each frame of the image and black lines to express ideal water surfaces avoiding doubt with perspective surfaces in **Figure 87** (a). The free surface shapes for three cases are in good similarity except flying water droplets and splashes on the wall, as compared in **Figure 87**. The water splash on the wall and flying droplets are difficult to be captured in a small scale LBM simulation since the interface between water and air phase is expressed by a continuous single layer of IF cells. Based on the validation process, it can be claimed that the single phase simulation of LBM for free surface problem has a substantial capability.

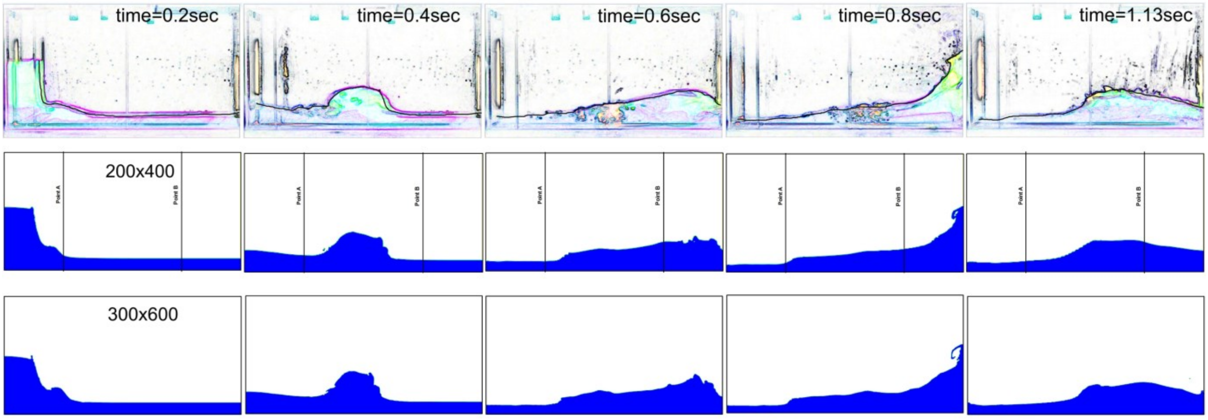


Figure 87. Time sequence image comparison of experimental (a-upper) and numerical dam break tests (b-middle and c-lower) with the wet bed.

7.4.2 Flow over a weir

Weirs are well studied structures theoretically and experimentally, but less effort has been made by numerical studies because of perfection and priority. Matured weirs measure flow discharge very precisely, if a best fit discharge coefficient curve has determined accurately. Among with the advances in the numerical simulation, there exist many opportunities to develop brand-new weir or flume. In this study, we simulated flows over a sharp-crested rectangular weir in the two dimensional space to determine the discharge coefficient and flow pattern over the weir. Weirs and spillways have the same hydraulics manner for an inflow and outflow in terms of boundary conditions. We impose the Zou/He boundary condition at the inlet and the zero gradient open boundary condition (described in Section 5.2.3) at the outlet. The geometry for simulations is given in **Figure 88** (a).

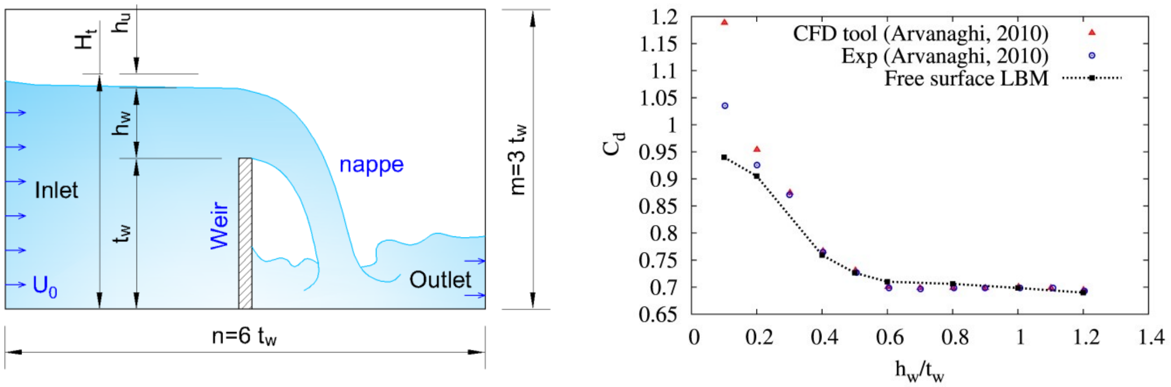


Figure 88. Dimension for flows over a sharp-crested weir and comparison of the discharge coefficient defined by the LBM, CFD tool and experiment for the different ratio of h_w/t_w.

The discharge equation for a sharp-crested weir (Henderson, 1996) in unit width, assumed here as unit discharge, can be simplified as,

$$q = \frac{2}{3} C_d h_w^{3/2} \sqrt{2g}\,, \tag{396}$$

where C_d and h_w are the discharge coefficient and the static head over the crest. Practically, the discharge coefficient depends on many parameters such as a flow characteristic, channel geometry and the ratio of crest height to static head. Ignoring the channel geometry effect on the discharge, we examined the discharge coefficients of several discharge cases and compared to the results given in (Arvanaghi & Navid, 2013), where the study had conducted a physical experiment and a simulation by the commercial CFD tool, Fluent. Since this section is devoted to demonstrate the free surface LBM in open channel hydraulics, we only perform several numerical simulations on the configuration with crest height $t_w = 0.15$ m, and the defined discharge coefficients are plotted with the results taken from (Arvanaghi & Navid, 2013) in **Figure 88** (b). We had used the following parameters for time step and grid spacing, $\Delta t = 0.000074$ s and $\Delta y = 0.0025$ m. In our simulation, it is observed that the ratio of the crest height to the static head on the crest was a main parameter to indicate flow characteristics of sharp-crested weir. If it exceeds over a unit, depending on the downstream situation, a submerged flow condition can be observed. A closed flow circulation was created between weir and nappe when case of $h_w/t_w > 0.8$, because the outlet boundary was the first order zero gradient boundary condition. When the ratio become $h_w/t_w < 0.2$, the nappe flow had totally adhered to the weir surface.

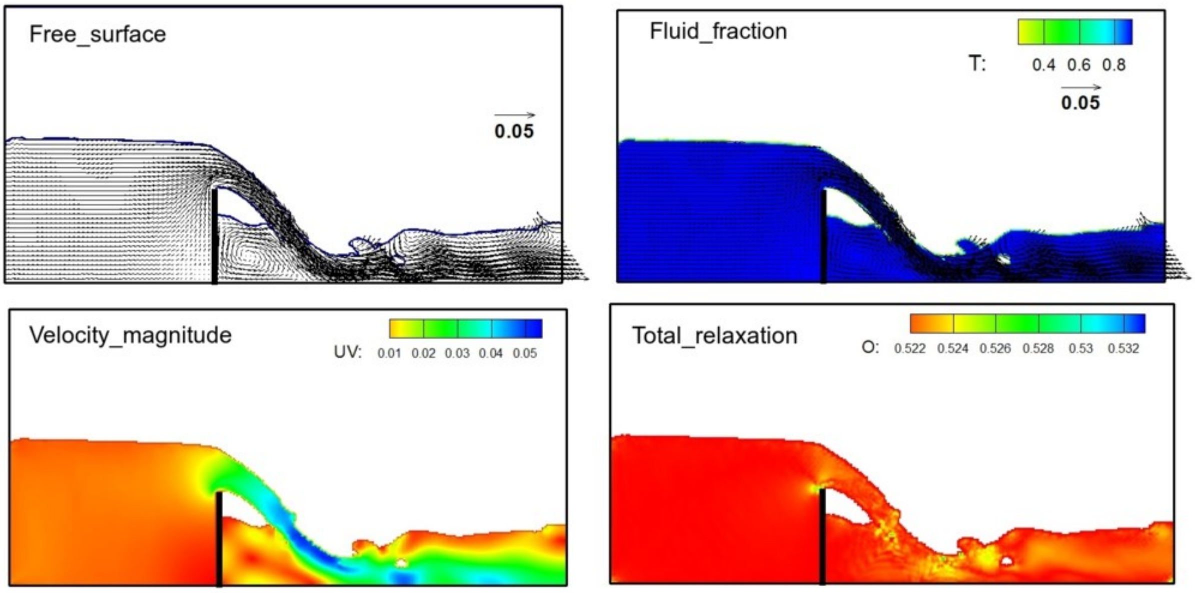


Figure 89. The results of the flow over a sharp-crested weir: the flow discharge is $q = 0.044\ m^2/s$.

7.4.3 Spillway example

Spillways are important structures for dam safety and a designing and operation of this structure is quite difficult. A water flow with high energy acting through the spillway surface causes damage for a structure like erosion and corrosion. Furthermore, the water flow with undissipated energy erode the river bed at downstream of such structures. Spillways or significant hydraulic structures are mainly designed by the physical model, which has the scale effect and requires a cost and time. Here, we model small scale stepped spillways to demonstrate the free surface LBM to use in the investigation of the important hydraulic structures. Two different step configurations are considered to evaluate hydraulic performances, i.e. big-stepped and small-stepped spillway as depicted in **Figure 90**.

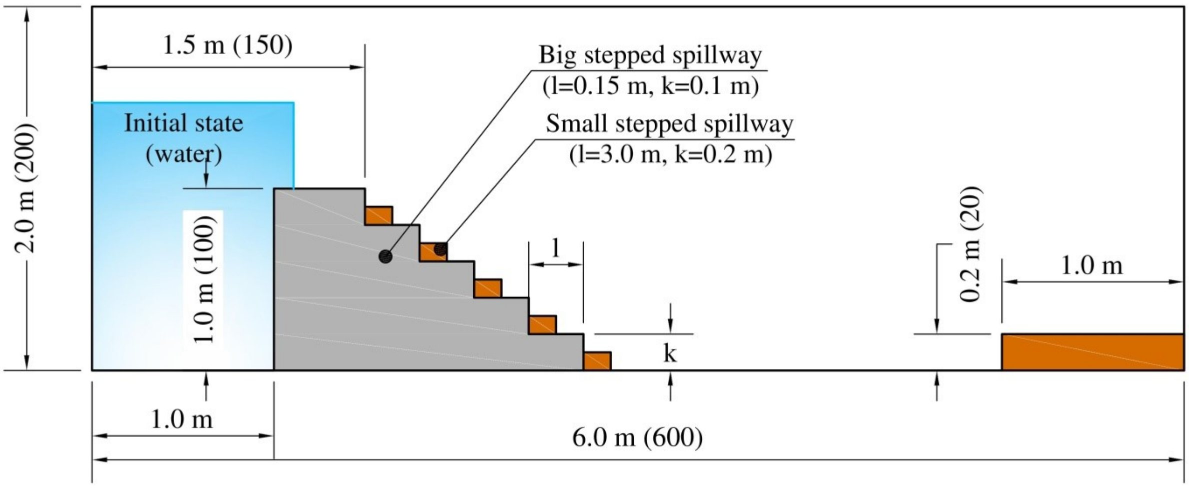


Figure 90. Two cases of stepped spillway simulation. Spillway heights are 1.0 m and a stilling basin is included to dissipate energy in the simulation.

This study did not intend to dive into the detailed investigation of stepped spillway. Therefore, we measured the water surface and average velocity through the stream-wise direction to expose the energy dissipation. Depending on an approaching energy, it is appeared that the first step, usually designed smaller than the other steps, has a big void in backward in the both simulation cases, which means that the step height must meet with the design procedure for this step. For the spillway with big steps, the nappe flow started from the first step and continued until the last step with a quick appear of skimming flow in the middle, shown in **Figure 91**. Whereas, the stepped spillway with small steps shows a good performance on reducing energy (see **Figure 92**), where the skimming flow condition were dominated.

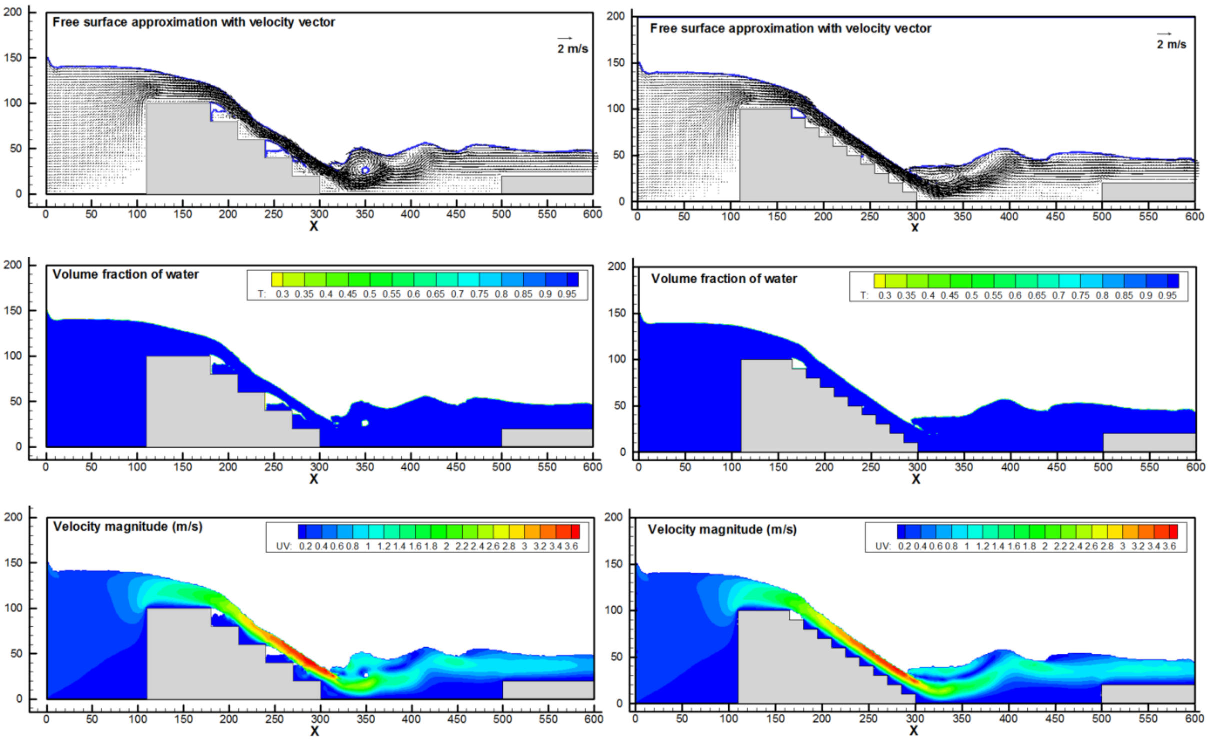


Figure 91. The first column of figures shows the results of the big stepped spillway (0.2 m in height and 0.3 m in length of a step), while the second column of figure shows the result for the small-stepped spillway (0.1 m in height and 0.2 m in length).

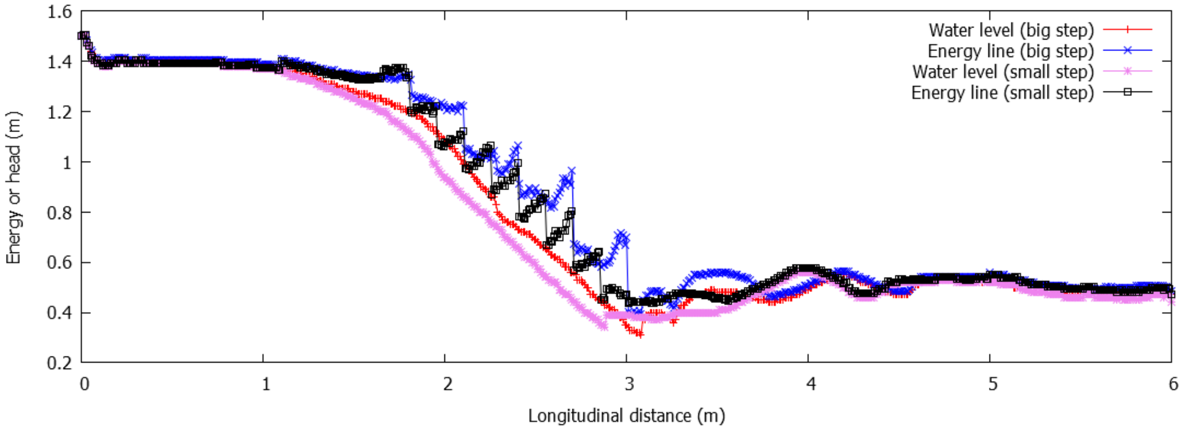


Figure 92. Energy head over the spillway at two simulation cases. The energy is defined as the addition of the static head and the velocity head.

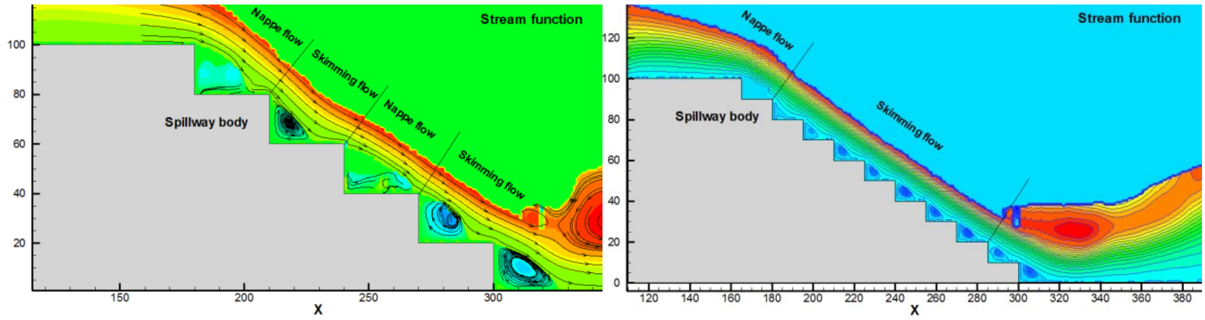


Figure 93. Stream functions for two cases: big-stepped spillway and small-stepped spillway.

The flow regimes on the steps for the two cases are roughly determined in **Figure 93**. The skimming and nappe flows were chained one after another for the big-stepped spillway, as seen in **Figure 93** (a). The nappe flow is considered as the cause of the cavitation on a spillway surface. Thus, the big-stepped spillway is not adequate in terms of hydraulic performance. The best flow regime is skimming flow (Rajaratnam, 1990), which can be seen in small stepped spillway as shown in **Figure 93** (b).

7.5 Liquid-solid phase transitions in free surface flows

7.5.1 Melting of an ice cube in ambient temperature

In order to validate the proposed numerical procedure in the free-surface condition, we carried out a brief laboratory experiment. The melting of an ice cube prepared in a freezer was compared with the results of the LB simulation. We used a commercially available infrared thermal imaging camera to measure the temperature distribution in a captured frame. An ice cube having sides of 4.5 cm was placed on a smooth wooden surface having lower thermal diffusivity and lower reflection of heat. In the heat transport module of the numerical model, the wooden surface was modeled as an adiabatic wall and the constant-temperature boundary condition was imposed on the water/ice surface interacting with the surrounding air. The temperature was maintained as a constant like the room temperature on the boundary, as in the experiment. In the fluid flow module, the wooden surface under the ice cube was assumed to be a no-slip wall, whereas the free-surface boundary condition (Eq. (346)) without surface tension was assumed for the water/ice surface interacting with the air. We used 60 grids for one side of the ice cube, the grid spacing was $\Delta x = 7.5\times10^{-4}$ m, and the time steps were $\Delta t_f = 6.91\times10^{-3}$ s and $\Delta t_h = 7.603\times10^{-2}$ s according to Eq.(370). These time steps provided sub-cycling at $n_s = 12$ with Eq.(369), so that the heat transport module is performed once every twelve steps of the fluid flow module. The time sequence of the thermal image is shown in **Figure 94** (a), followed by the corresponding numerical results in (b) and (c) of **Figure 94**.

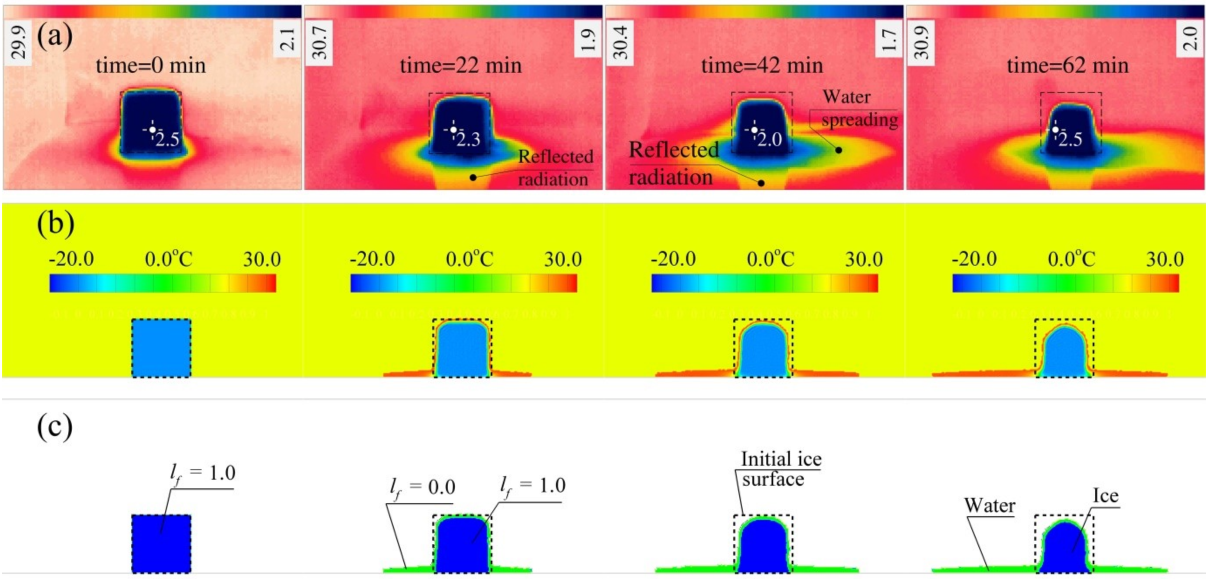


Figure 94. Time sequences of the experimental and numerical results for an ice cube melting in ambient air. (a) Infrared images of the ice cube and ambient condition; (b) numerically determined temperature field; (c) numerically determined ice and water phases.

Ice melts from the bottom at a low rate in the experiment, although no melting occurs in numerical simulation because of the given boundary condition. As shown in **Figure 94**, the difference in height of the melting ice cube after 62 min was 3.56 mm, whereas the numerical value was higher. The top of the ice cube became rounded in the numerical simulation, whereas, in the experimental test, the top remained approximately flat. The reason for this difference in shape might be related to the velocity of flowing water on the surface of the ice cube. The water thickness flowing on the ice surface in the numerical test, indicated as lf = 0 in **Figure 94** (c), was observed to be much thicker than that in the experiment. Since we used a coarse grid for discretization, the numerical model requires at least two double-layer grids to simulate the surface of the water covering the ice. The average water surface temperature on the ice, as determined by the thermal camera, was approximately 2.2°C in the experiment, as shown in Fig. **Figure 94** (a), which agreed with the result of the numerical simulation for the ice-water interface.

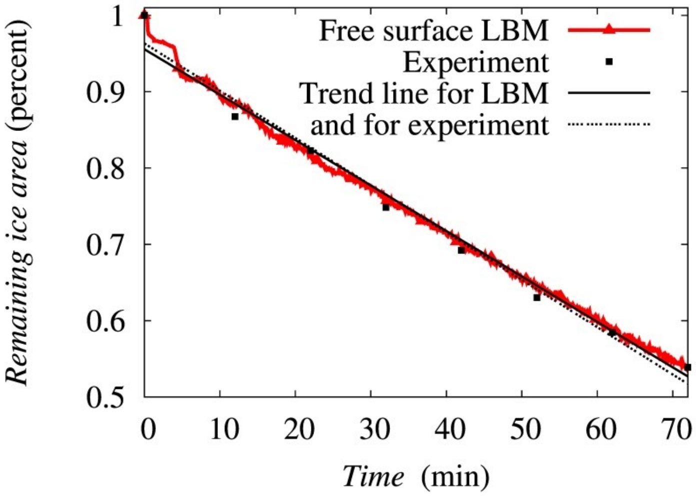


Figure 95. Experimentally and numerically determined remaining ice area.

For the sake of generality, we show the remaining ice percentage with respect to melting time in **Figure 95**, which shows the accuracy of the numerical model. The melting rate of ice was nearly linear, and the numerical results were in good agreement with the experimental results, with the exception of the initial oscillation in the numerical results. Similar studies were conducted by (Faizal & Septiawan, 2014) for an ice cube in still water and by (Tan, 2014) for an ice cube melting in ambient air. Both of these studies used particle-based methods, and their results were less continuous and exhibited a step-like tendency over time.

7.5.2 Ice melting by pouring water

An ice melting by pouring water is more realistic and dynamic experiment and the results were reported in (Ayurzana & Hosoyamada, 2017). In the evaluation, we melted an ice cube by pouring water on it using the 2D numerical simulation. The results were compared with a 3D laboratory experiment carried out using the same configuration so as to verify the numerical model. A glass with an ice cube was placed on a plate and water was poured in the center of the top surface of the ice. The ice cube size was 4.5×4.5×4.5 cm. The temperature of the poured water was 30°C, while the initial ice temperature was -30°C. The air temperature was taken as 25°C and a constant temperature boundary condition was applied at the inlet boundary and the free surface. In terms of heat transport, the walls and bottom surface were assumed to be adiabatic in the simulation. The bounce forward boundary condition, which reflects the distribution function at the wall like a mirror, described in Section 5.2.3, was used to impose a slip boundary at the glass wall. Since it is difficult to capture the temperature distribution inside the ice experimentally, the evolution of the ice melting front against time was used as a metric to compare the numerical results. Because of the high temperature difference between the ice and the poured water, the ice melting process took over 1 minute. The temporal change in the melting front and the remaining ice area were monitored by processing the images recorded using a camera during the experiment.

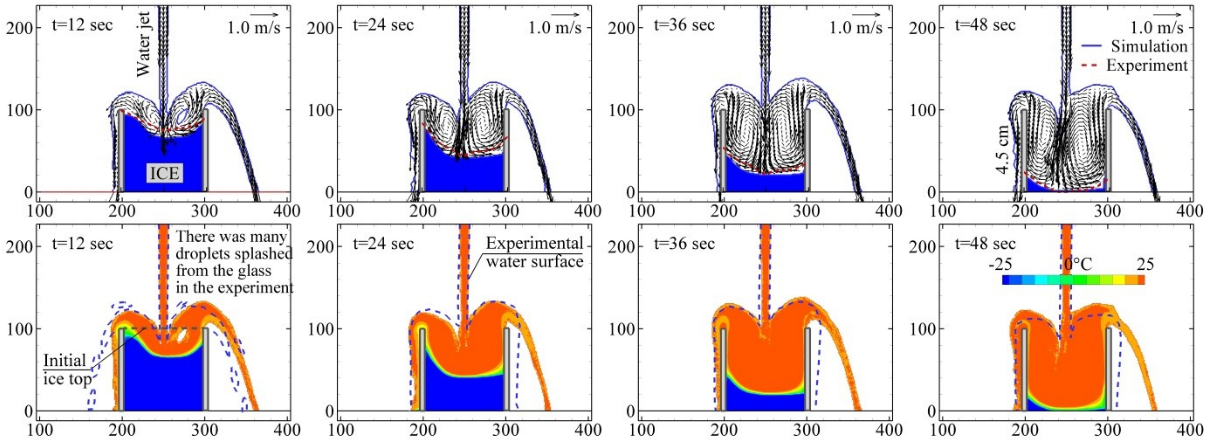


Figure 96. Temporal states of liquid-solid phases (a) and temperature fields during the ice melting by pouring water (b)

In **Figure 96** (a), the variation in experimental and numerical melting front with free surfaces and velocity vectors is shown at different time sequences. Experimental and numerical melting fronts are almost the same, as seen in **Figure 96** (a). In the experiment, the water overflowed from the glass in all directions. However, in the simulation, the water overflowing from the left side is modeled as being attached to the side, while the water overflowing from the right side is computed as a weir flow as shown in **Figure 96** (a). The melting front on the right side is more active than that on the left side in the simulation; this is also the case in the experiment. The heat transport on the left side was lower than that on the right side, as shown in **Figure 96** (b).

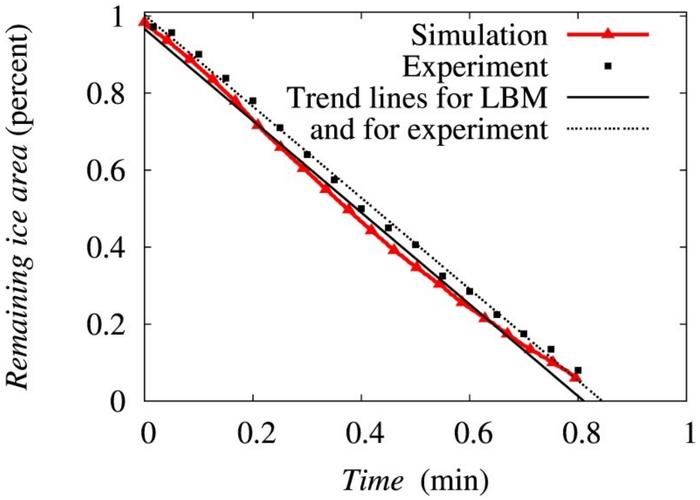


Figure 97. Experimentally and numerically determined remaining ice percentage

The ratio of the remaining ice area to the initial ice area for each of the cases is defined and compared in **Figure 97**. The melting rate of the experiment and the simulation was nearly linear in comparison with the best-fit lines. The melting rate in the simulation was higher than that obtained from the experiment. The slight difference in the melting front evolution and the overflowing of water from the glass can be attributed to the dimensional difference. The temporal changes in the melting front and the flow of water over the side of the glass show the asymmetric properties in the simulation. The asymmetric melting front obviously leads to asymmetric water overflow. Another reason for the asymmetric flow might be that the surface tension of the water is not considered in the free surface boundary condition. It seems that the surface tension affects the water overflow in the lab experiment. The asymmetry of the melting front also can be related to the free surface boundary condition. However, the overall predictions of the numerical model were in reasonable agreement with the experiment.

8 THE LBM TO APPLICATIONS

So far, we have discussed the model, its implementation and validations. This chapter is devoted to particular problems in practice.

8.1 Ice bed melting

8.1.1 Short range of temperatures

In the literature, forced convections with turbulent flows for liquid-solid phase change problems are solved rarely. We aimed to simulate a phase change problem in turbulent regime and it was achieved with the ice melting process by a flow over sharp-crested weir. The free-surface LBM method has been used to simulate flow over a weir and those results are given in Section 7.4.2. Here, we extended the weir flow simulation to the phase change treatment with heat transfer neglecting the latent heat. The geometry of the simulation is given in **Figure 98** (a).

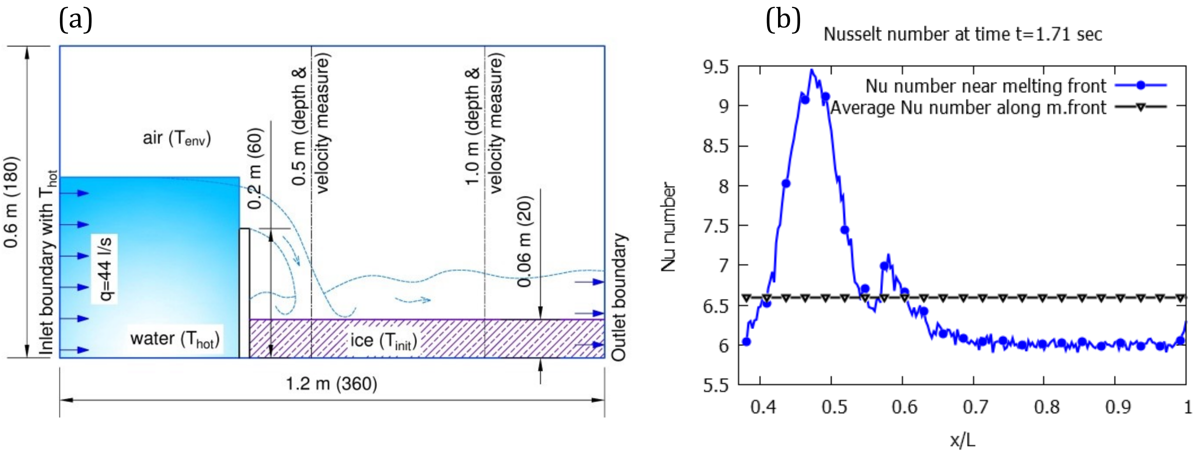


Figure 98. Ice bed melting by weir flow. Temporal and average value of the Nusselt number on the melting front.

Ice with thickness of 0.06 m and temperature of 0°C is located in downstream of a 0.2 m height weir. Water (flow q=44 l/s) with temperature of 10°C is released from upstream of the weir to downstream. The surrounding air temperature set to be 1.5°C. The latent heat was ignored for this short ranged temperature consideration. After the melting of t=6.83 s, the downstream ice had been completely melted in our simulation. It can be inferred that melting intensity is proportional to velocity magnitude, in turn the turbulent intensity (Ettema, et al., 1982), since the nappe flow provides the main contribution to transfer heat, shown in **Figure 99**.

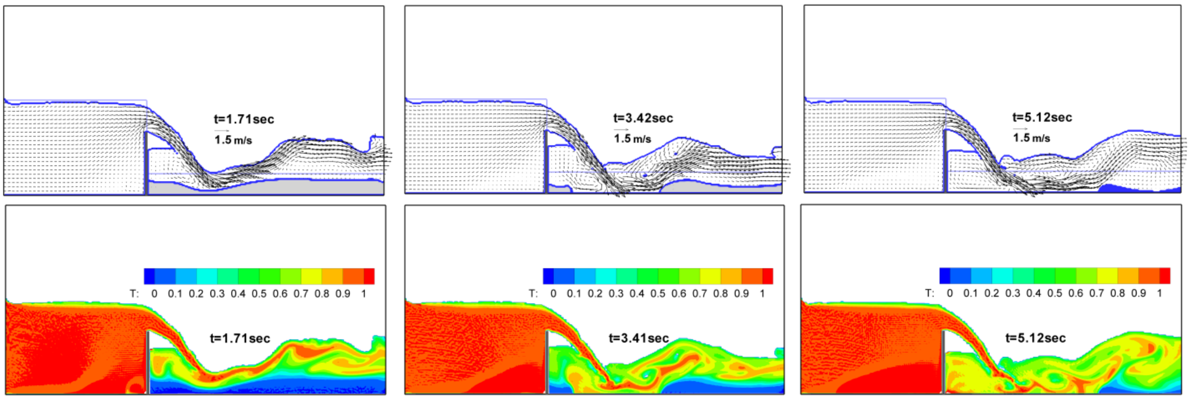


Figure 99. Free surface and ice-water interface at top and temperature field at the bottom have been shown at selected times within the simulation. The initial state is illustrated as a straight grey line.

We measured the Nusselt number on the melting front at t=1.71 s, as given in **Figure 98** (b). The average Nu was 6.59, while the maximum Nu was 9.5, which observed where the nappe jet hits on the ice surface. This simulation was simulated by employing the direct integration approach for the fluid flow and heat transfer with phase change modules. The extension of temperature range with the direct integration approach leads the instability for the simulation. So that, we were advised to use the sub-cycling integration, when the temperature range is wide and the latent heat is present.

8.1.2 Extended range of temperatures

Previously, ice bed melting by the flow over weir was simulated for ice that was ready to melt, i.e., the ice temperature was set to 0°C, and the latent heat was ignored. In this section, we included a latent heat source term in heat transport and extended the temperature range. The initial and boundary condition is indicated by the problem geometry, as shown in **Figure 100** (a). Initially, temperatures of -30°C, 20°C, and 30°C were set for ice, air, and water, respectively. The inlet and outlet boundary of the flow field was imposed with a velocity boundary condition, whereas the wall and the surface of the weir were modeled as slip walls. Water at a temperature of 30°C was supplied to the inlet, where the thermal boundary was given by the Dirichlet boundary condition, explained in Section 5.3.2. The second-order extrapolation boundary condition was imposed on the outlet boundary for the heat transport module. The other walls and the surface of the weir were assumed to be adiabatic. The weir flow by the free surface LBM was carefully investigated and validated in Section 7.4.2, and we herein used the weir flow with the Froude number of Fr = 0.13. In order to examine grid independence, we considered two grid resolutions, namely, h = 60 and h = 80, where h is the grid number used for the weir height, as shown in **Figure 100**.

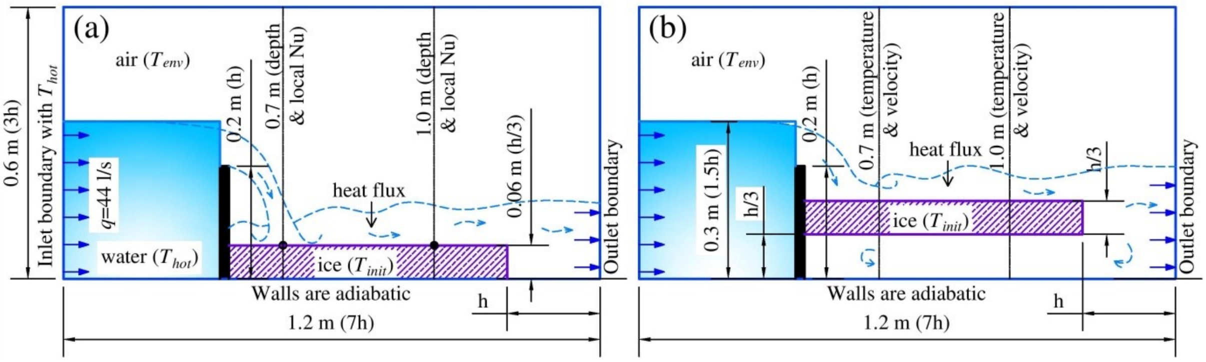


Figure 100. Numerical domains for the (a) ice bed and (b) ice cover melting by the flow over the weir.

The relaxation times for the grid resolutions for the flow field were chosen as τ_v= 0.526 and 0.534, respectively, which is adjusted to τ_{tot} by the sub-grid scale model (Section 5.4.3). The relaxation times for the heat transfer module was determined by the similar relation as Eq.(393) using the thermal diffusivity of water, which can be connected to the lattice viscosity by the Prandtl number. The ratio of the remaining ice area to the initial ice area was measured and is shown with respect to melting time in **Figure 101**.

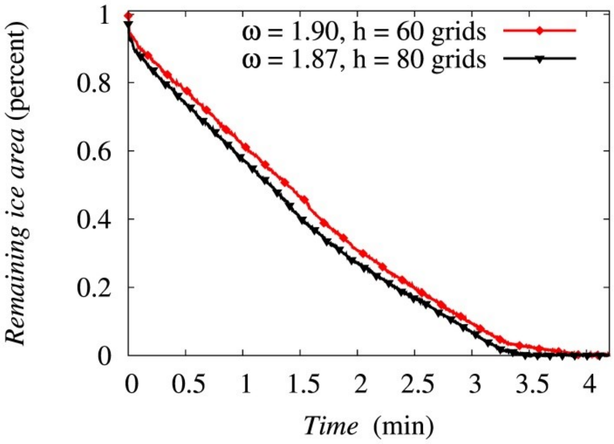


Figure 101. Percentage of remaining ice area for different grid sizes and parameters as determined through numerical simulation.

Depending on the parameters, both grid resolution and the selected relaxation times, the total times of melting differed by approximately 0.6 min, and in case of h = 60, the ice lasted 3.8 min. Based on these considerations, choosing appropriate parameters grounded in their physical relations is more important than grid resolutions. The melting rates for these two cases have similar melting rate tendencies, as shown in **Figure 101**, but after approximately 3.3 min melting rates are changed due to the low melting rate of the ice located directly behind the weir. Between the nappe entrance and the weir, where the flow is partially circulated, the convective heat transfer between the water and the ice was small due to the low velocity in this region.

Heat transfer between water and ice can clearly be explained in terms of the Nusselt number. The local Nusselt number at a point on the melting front is defined as (Mohamad & Kuzmin, 2010):

$$Nu_L = \left|\frac{\partial\theta}{\partial y}\right|_{y=interface}, \tag{397}$$

and the average Nusselt number along the ice-water interface was calculated as

$$Nu_{av} = \frac{1}{H}\sum_{k=1}^{k=H} Nu_L\, M, \tag{398}$$

where H is the grid number for the length of the ice, and M is the grid number for the depth of the water above the ice. The local Nusselt number and ice depth are plotted with respect to the elapsed time in **Figure 102**.

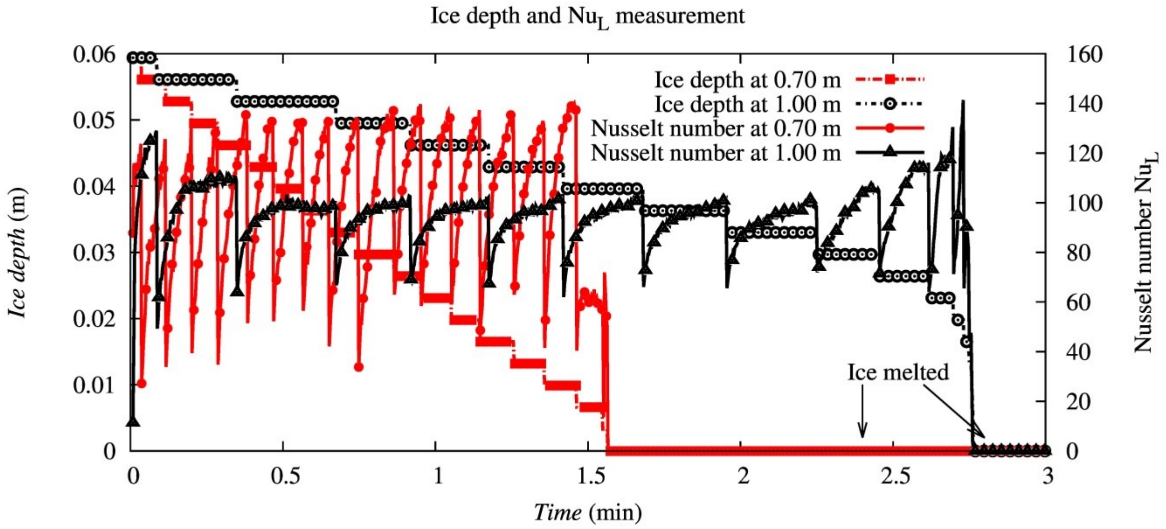


Figure 102. Time series of ice depth and local Nusselt number measurement.

The local Nusselt number at 0.7 m fluctuates with a higher frequency and amplitude than that at 1.0 m, and the tendency in both cases is to increase with time until decreasing suddenly at the location at which the ice depth decreases. Generally, this tendency is due to the fact that heat increases in ice and water near the interface, and the sudden drop is due to the disappearance of the ice. As the frequency increases, ice at approximately 0.7 m from the origin quickly melted because the convective heat transport at this position is high. The heat transfer coefficient expressed in terms of Nu_{av} in **Figure 103** was approximately $h_c = Nu_{av}k_R^{ice}/L_m \approx$ 1,090.8 WK^{-1}m^{-2}, where L_m is the characteristic depth of water on the ice. The temperature field and ice/water phase with a free surface at three different times are plotted in **Figure 104**, where the interaction of the flow structure and the thermal behavior of phases are shown clearly. The total melting time of the ice bed was about 3.25 minute and the melting rate of ice mass was around 42.46 kg/min.

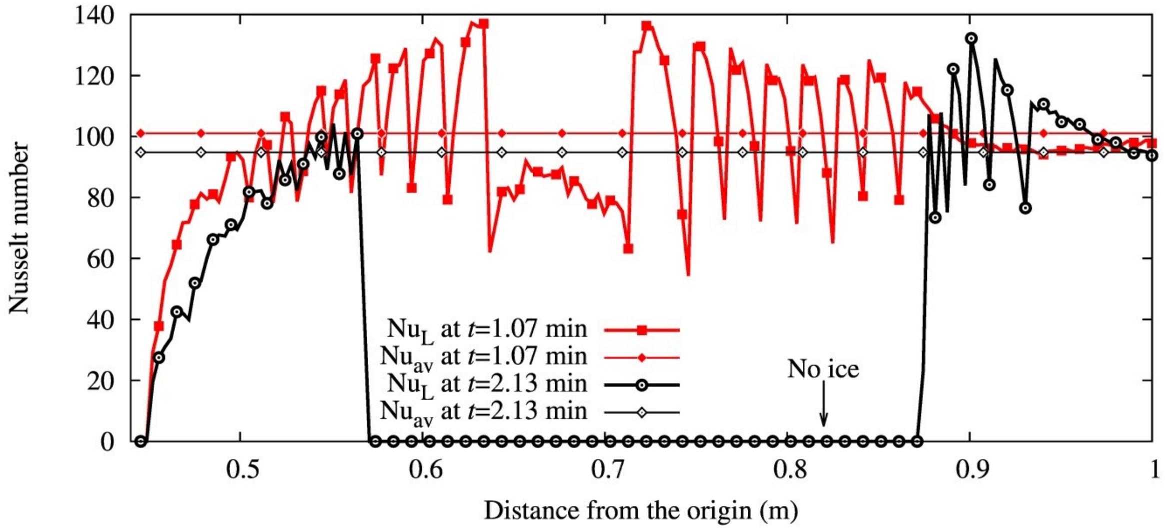


Figure 103. Local and average Nusselt numbers near the interface between ice and water

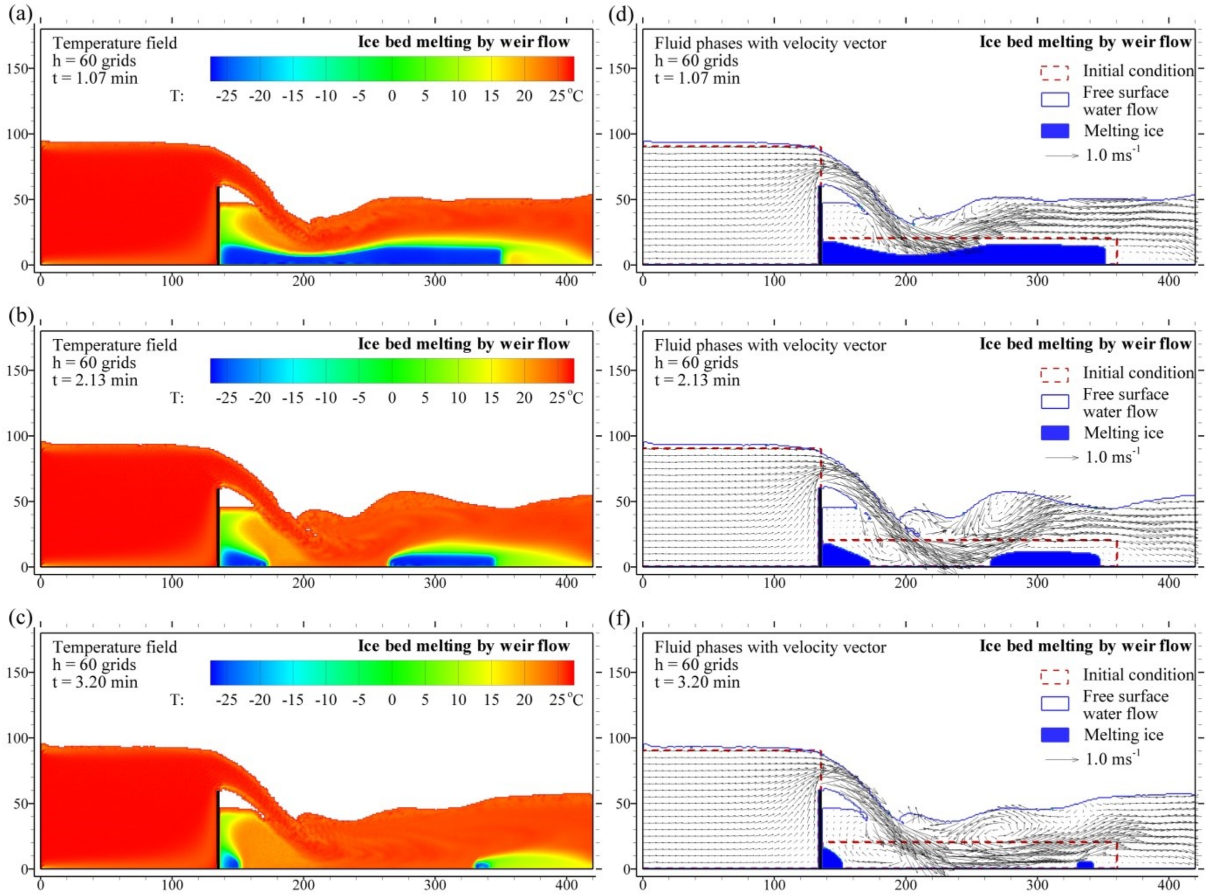


Figure 104. Temperature and vector fields at three different times obtained through simulation of ice bed melting.

8.2 Characteristic of melting of an immersed ice sheet

The motion of ice in the model will be studied in Section 8.4, where we analyzed floating ice on the water surface (Ayurzana & Hosoyamada, 2018). However, in this section, the ice does not break up and move freely. An ice sheet melting simulation was conducted in order to determine the applicability of the numerical model to real field problems, in which ice in a river or a reservoir is mixed with free surface flows. In this case, the fixed position of the ice will help in the visualization of the freezing of water. As shown in **Figure 100** (b), the condition and geometry of the ice sheet are such that heat is absorbed by the bottom of the ice and the water is expected to freeze downward due to water being trapped under the ice sheet, where a natural convection flow may dominate. Excluding the outlet velocity condition, all of the parameters and given conditions were the same as in the ice bed melting simulation.

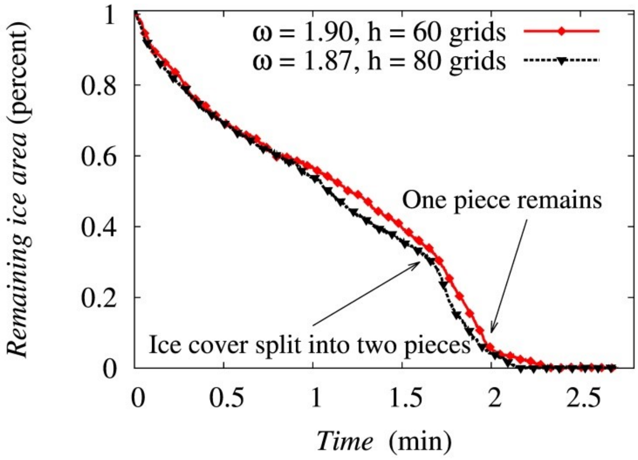


Figure 105. Percentage of remaining ice area for different grid sizes and parameters as determined through numerical simulation.

Figure 105 shows the melting rate of ice for the two simulation cases, h = 60 and 80, and a discrepancy between the time courses appears at around 1.0 min into the simulation. However, a comparison between those two cases of the general melting shape of ice sheet revealed no significant differences.

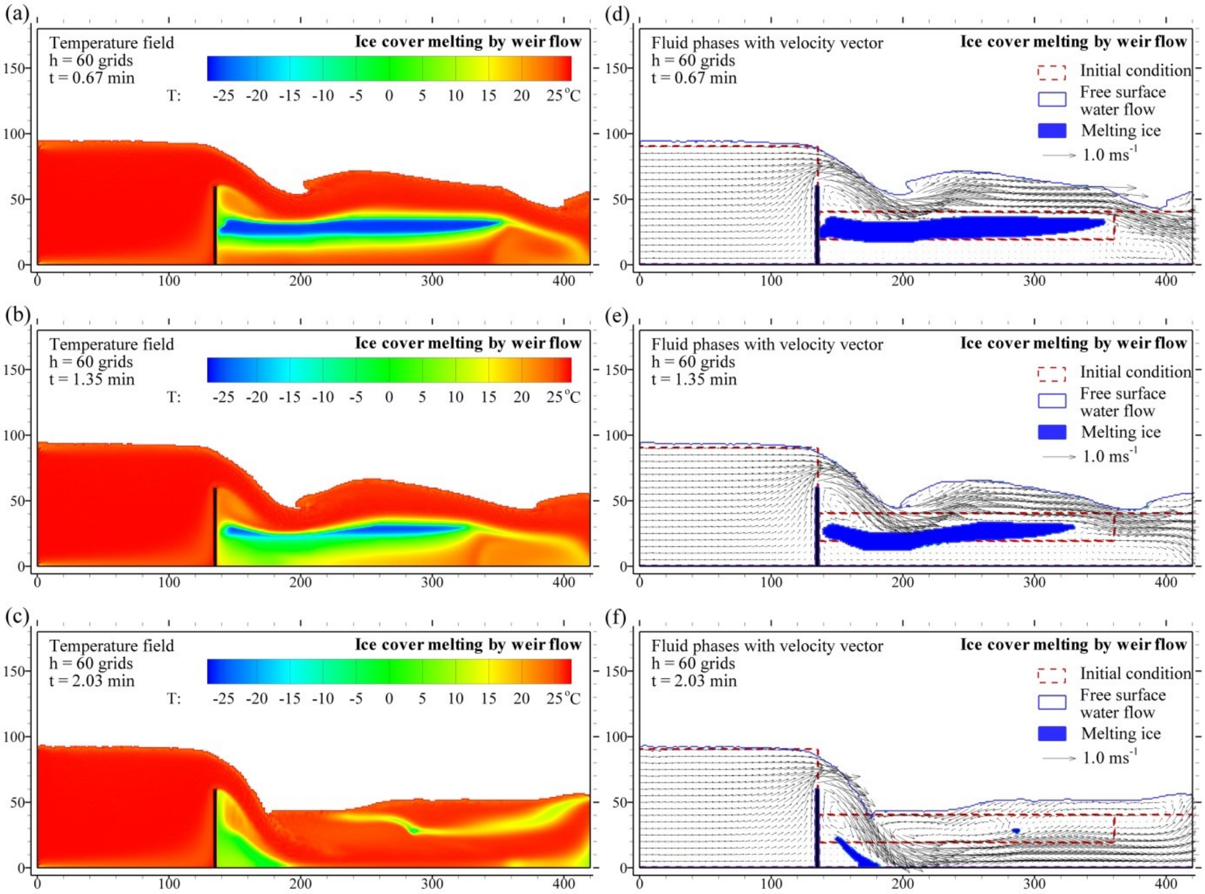


Figure 106. Temporal change of temperature and flow vector field distributions at three different times obtained through simulation of ice sheet melting.

Figure 106 shows the general melting process of the ice sheet in the simulation. The ice sheet absorbs heat from water flowing over it, and the cold water flows out of the outlet boundary. At the same time, beneath the ice sheet, heat can only be transported by conduction, as shown in **Figure 106** (a), until a natural convection flow form in the water region beneath the ice sheet, because the water in this region is trapped by the circulating water flow near the outlet boundary, as shown in **Figure 106** (d). The circulation flow near the outlet boundary transports heat and momentum into the closed region beneath the ice sheet. Another circulation flow was observed where the water flows over the weir. This circulation carries heat to the ice behind the weir. As water flows over the ice sheet, melting occurs and the ice is gradually eroded by overtopping flow (**Figure 106** (b)). The melting rate of the upper surface of the ice sheet was higher than in other parts of the melting ice. This situation continues until a natural convection flow form in the water region beneath the ice sheet. As expected, freezing occurred on the bottom surface of the ice sheet due to convection. However, an opening eventually formed near the back face of the weir through which water could flow into the region of water trapped by the circulation flow, as shown in **Figure 106** (e). The water flow then surrounds the ice sheet and melts the ice from all sides. The outlet circulation flow was an active heat transporter and melted and sharpened the tail of the ice sheet, as shown in **Figure 106** (a) and (e). Due to erosion, the ice sheet was split into two pieces by the overtopping flow after approximately 1.6 min, which is indicated by the recurved shape of the lines in **Figure 105**. The piece near the outlet boundary quickly melted because it was surrounded by an active flow field.

The small piece of ice remaining can be seen in **Figure 106** (f), and its effect on the temperature field can be seen in **Figure 106** (c). The piece of ice near the weir eventual extended downward to the bottom boundary, as shown in **Figure 106** (f). **Figure 107** shows the temperature and velocity profiles at 0.7 m and 1.0 m at various times. **Figure 107** shows that the velocity and temperature profiles have the same tendency because massive amounts of heat are transported by convection in turbulent flows. The vertical temperature gradient can be high where the flow velocity is high, as indicated near the upper surface of the ice sheet in **Figure 107** (a) and (b). Unrealistic velocity and temperature decreases in the middle of the profiles appear at 1.0 m in **Figure 107** (c) because of the piece of ice remaining in the flow, which cannot move with the water flow. As such, this piece of ice influenced the velocity field as an obstacle, resulting in a cooler temperature distribution.

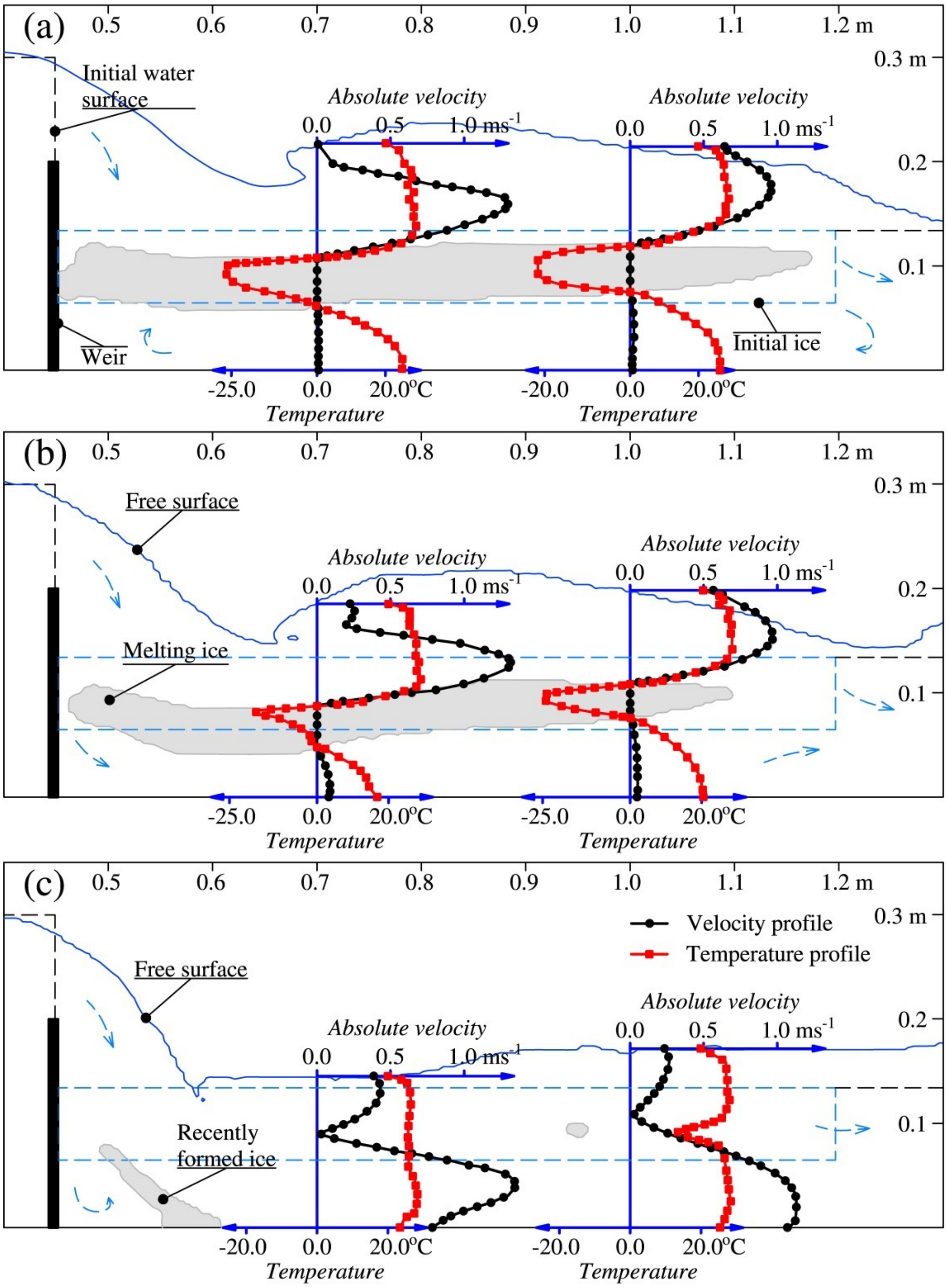


Figure 107. Vertical profiles of temporal velocity and temperature at three different times: (a) t = 0.67 min, (b) t = 1.35 min, and (c) t = 2.03 min.

8.3 Open water forming mechanism in downstream of hydropower plant

8.3.1 Ice problems of small hydropower plants in cold region

Let us discuss about the ice problem in small hydropower plants in Mongolia as representative of cold region. Off-the-grid small hydropower plants (SHPPs) have been promoted as an efficient source to fulfill the electricity demands in the rural areas of Mongolia (Infrastructure, 2002). A total of 11 SHPPs has been commissioned since 1986 (Boldbaatar, 2013) and a number of plants have been planned for future construction. However, except for two of the plants, the others can be operated only for a short duration. The rest are unable to operate during the winter, from the time the river ice freezes until the ice breaks. In this section, we consider one of the SHPPs, namely the Tosontsengel hydropower plant, seen in **Figure 108**. This is located in the village of Tosontsengel, which is one of the coldest places in Mongolia.

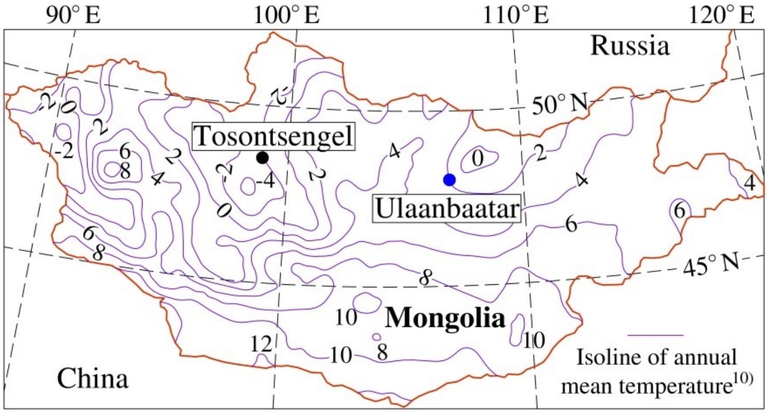


Figure 108. Location of the Tosontsengel hydropower plant

The SHPP operates on the Ider River, which has a flow under the ice cover during the winter (forum, 2013). At this time, the plant faces a serious ice problem in the downstream area, as seen in **Figure 109**, where the ice ridge and jam is created in downstream.

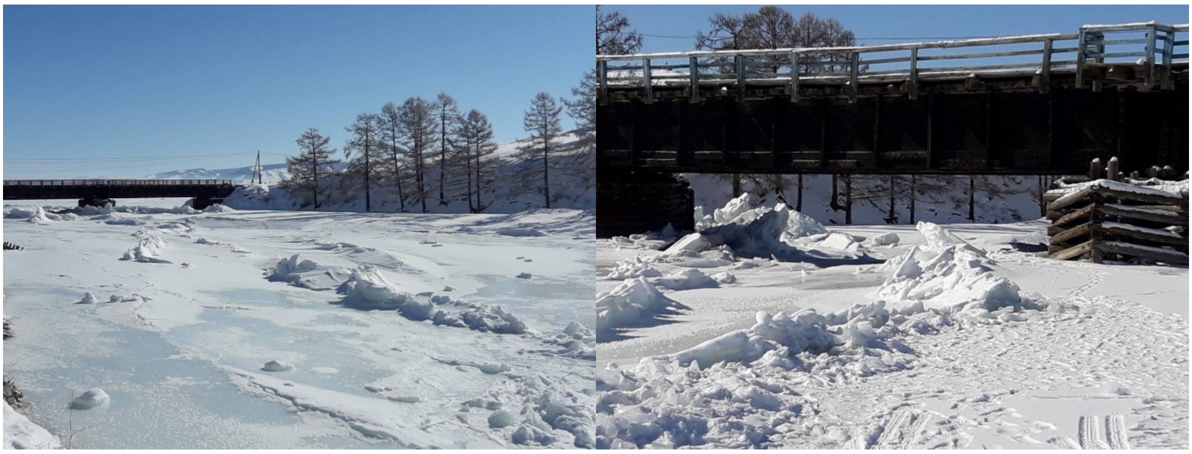


Figure 109. Downstream ice conditions: (a) from the power house to the bridge in downstream and (b) near the bridge, March 16, 2017.

Also, the ice problem in the outlet channel of the turbine not only reduces energy production, but also decreases the cross-sectional area of the waterway, which causes bed erosion in the unlined channel.

In general, studies on the ice problem and the possible countermeasures (Gebre, et al., 2013) are based on on-site measurements. There are very few studies that consider the prediction or modeling of the ice problem. In physical terms, the behavior of ice in the outlet of a hydropower plant in a cold region can be modeled as an interaction between ice and the free surface flow.

This involves liquid-solid phase changes since warm water is released from the turbine to the outflow channel changing the ice condition.

8.3.2 Problem statement

In the winter of 2007, a year after its commissioning, the Tosontsengel hydropower plant experienced ice jam and ice ridge formation in the downstream area between the outlet structure and a bridge that is located around 130 m downstream of the plant, shown in **Figure 109**. An engineer working in the plant recalls, "Power production did not meet the expected operational plan and the efficiency was lower. In addition, ice heavily affected the bridge pier as an excess load. That year, workers in the plant suspected that fluctuations in the water discharge and the low speed of water in the outlet created the downstream ice problem." The next year, they experimented by passing water downstream through the turbine without operating it. In other words, the water discharge through the turbine was the same as the river flow rate. Consequently, the ice jam reduced greatly, but small ice ridges were still observed downstream. Since then, the plant authorities decided to use this mechanism in the winter. When passing water through a stationary turbine, the water surface is not enclosed by ice for a distance of 4 to 5 m from the outlet. This is termed open water, and it exists during the winter. Forming of open water at specific rivers (Ettema & Zabilansky, 2001), (Prowse & Beltaos, 2002) has been well documented rather than open water forming in the downstream channel of hydropower plant. Open water in downstream might be the key factor for the condition of ice in downstream. Understanding of the open water forming mechanism provides important knowledge for the controlling ice condition in downstream.

No research has been conducted for a way to deal with the ice and to operate the plant during the winter. However, workers are still seeking a way to produce power in the winter. In this study, we simulate the process of open water formation (see **Figure 110**) in the downstream. This occurs due to the release of warm water from the upstream through the turbine. It provides the possibility of applying the proposed numerical model to the ice phenomena in hydropower plants. Absence of field data and computational limitations make it difficult to model the ice problem on a large scale.

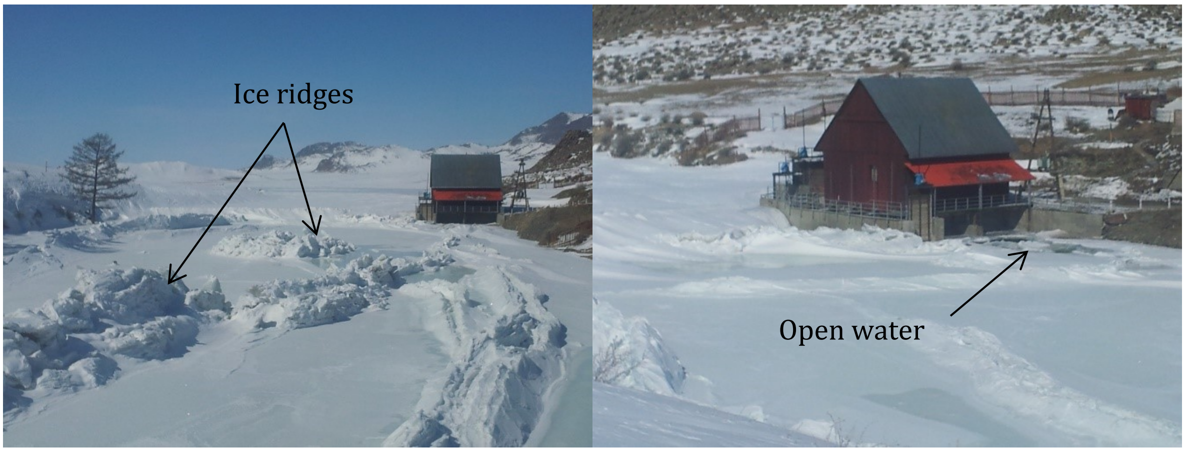


Figure 110. The Tosontsengel hydropower plant, March 24, 2014.

8.3.3 Numerical results

Figure 111 shows a longitudinal cross-section of the Tosontsengel hydropower plant, including the computational domain. The computational domain does not include the upstream and the spiral case of the turbine. Hence, the inlet boundary of the computational domain is at the beginning of the turbine tailrace. The domain extends to 32 m downstream. We use a grid spacing Δx=0.04 m and a time step Δt=4.11×10^{-4} s in direct numerical integration (see the Section 6.2.1). We assume that the downstream of the plant is fully covered with ice of thickness 0.3 m and the wicket gates of the turbine are closed initially. The temperature of the ice cover is taken to be -10°C in all simulations. The simulation starts with the opening of the wicket gates. At the inlet boundary, velocity and constant temperature conditions are imposed for the fluid flow and heat transport modules, respectively. Velocities at the inlet boundary are approximated from the capacity of the small turbine because of lack of winter flow data. At the outlet boundary, the second order extrapolation boundary condition was applied to the fluid flow and the heat transport modules. A typical bounce back boundary condition is used for the concrete wall and channel bed as a no-slip boundary condition.

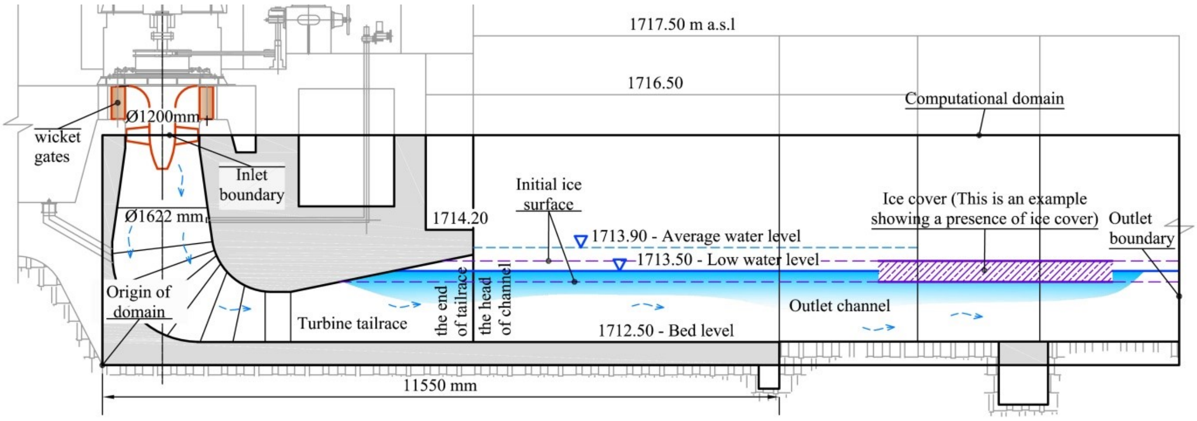


Figure 111. Longitudinal cross-section of the Tosontsengel hydropower plant and the computational domain with boundary conditions.

As observed in the field, there is a continuous ice cover through the downstream area. Hence, the ice cover is assumed to be stationary and there are no crushing phenomena in the simulation. In order to understand the formation of open water in the ice covered downstream, three different water temperatures, 2°C, 4°C, and 8°C, are considered as cases for the low discharge water in the simulations. As ice set at a low water level, the first task was to find the discharge that maintains the low water level in downstream. Because at this time, no winter flow discharge data were available. The first run was carried out with the small turbine capacity. The discharge was apparently high and the maximum velocity reached at 8 m s^{-1}, as shown in **Figure 112** (a). The discharge was then reduced until the lowest water level is obtained. We found the discharge of 0.87 m^{3}s^{-1} (**Figure 112** (b)) and this was used in three cases of run with the ice cover.

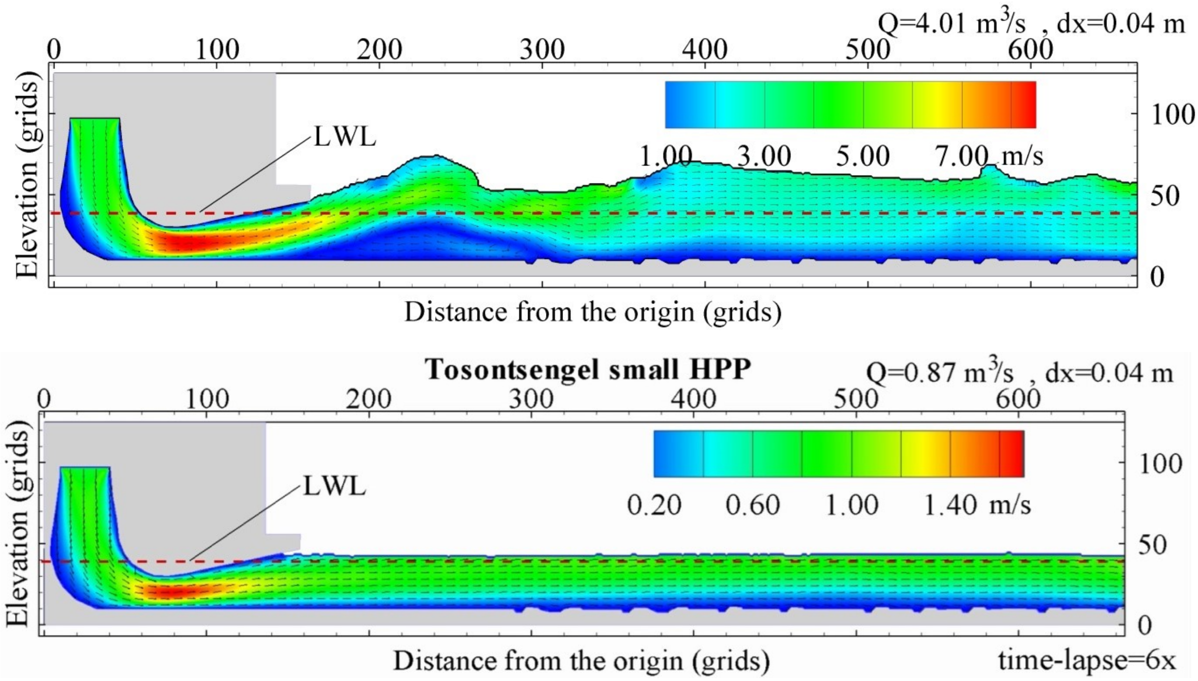


Figure 112. Temporal situations with the high (a-top) and low discharge (b-bottom) of the outlet channel of the Tosontsengel hydropower plant.

As shown in **Figure 113**, in case 1, the ice area evolution was stable except for an initial drop that could have been caused by initial pulses of inflow. Melting should have started near the end of the tailrace, where the velocity is the highest. However, open water was not formed during more than 10 minutes of simulation, as seen in **Figure 114** (a), and eventually the system went into equilibrium. This is because of the low heat transfer rate between ice and water at the bottom of the ice cover. Melting or freezing of the ice cover happens in the other two cases. As the temperature is lower, the melting rate was lower for case 2 in comparison with case 3 in **Figure 113**. With the same hydraulic conditions, open water was created at different times; this is seen as a transition of the remaining ice area in **Figure 113**. After the open water is formed, freezing occurs in case 2, because of the water overflow and the low discharge of water under the ice cover. In case 3, the ice area gradually decreases because the temperature of water is sufficient to melt ice on and under the ice cover. This is shown in **Figure 113** and **Figure 114** (c). The air temperature is shown in **Figure 114** (b) and is the same for all cases. As shown in **Figure 114**, the length of the open water was 0.18 m for case 2 and 0.4 m for case 3. The open water in case 3 was further extended to 2.08 m in length at 13 minutes and 4.0 m at 23 minutes.

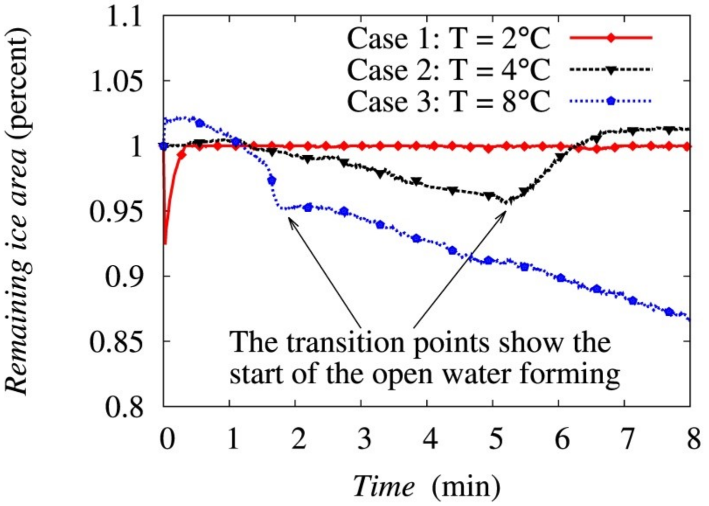


Figure 113. Numerically determined remaining ice area of the three temperature cases.

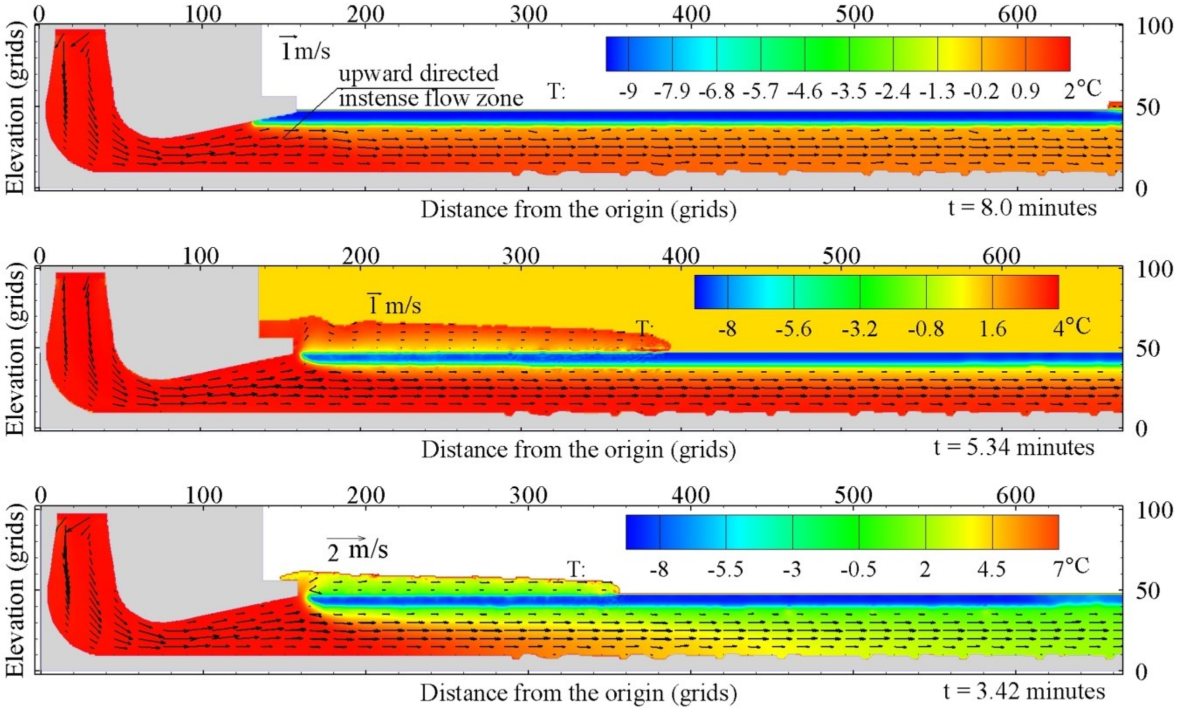


Figure 114. Numerically defined temperature field and flow field at different times and cases: (a) case 1, (b) case 2 and (c) case 3.

Hydraulically, the water flowing at the end of the tailrace tends to flow in a direction upward to the free surface. It then flows toward the downstream channel. The flow at the end of the tailrace, as pointed on **Figure 114** (a), is intensified when the downstream is covered with the ice and open water exists in the head of the downstream channel. In other words, the ice covered channel acts against the water coming from the turbine, like the flow through a conduit under pressurized conditions. This pressurized condition also initiates the ice cover cracks. The open water is under lower pressure and interacts with the air; hence, water can easily flow in an upward direction after leaving the tailrace, as seen in **Figure 114** (b) and (c). Therefore open water is maintained during winter.

The fact is that when the temperature is lower than 2°C and the discharge of the released water is approximately the maximum capacity of the turbine, the downstream ice cover does not melt. This implies that when water at a temperature near 2°C is flowing through the downstream channel and there exists open water in the downstream, ice will not melt nor extend if there is no overflow above the ice cover from the open water. Based on the flow characteristics obtained from the above simulation, if the flow direction to the head of the downstream channel is adjusted through the channel, it is possible to control downstream ice phenomena for different discharge conditions. One possible mechanical solution can be a controllable foil at the end of the tailrace, as shown in **Figure 115**. The foil length should be around 1.5 m to change successfully the upward directed streamflow into the stream wise direction and two ends must be sharpened. The foil should be controllable to change its attack angle to incoming flow depending on the flow rate. The adjusted flow will widen the cross section of the channel by melting ice cover from its bottom. Once the equilibrium state is reached under the ice cover in terms of thermodynamics, the ice condition will be stable and controllable in the outlet section of the power house. The detailed investigation of the foil and its material should be provided in order to evaluate its feasibility.

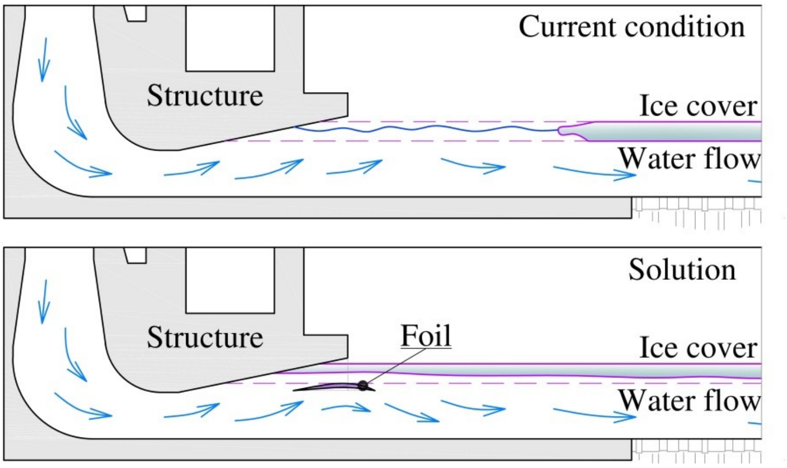


Figure 115. A foil installation in the turbine tailrace to adjust flow direction toward under ice cover.

8.3.4 Interpretation of open water forming mechanism

The end of the tailrace is a special region, where the temperature of water is almost the same as that of the inlet boundary. This is where the heat exchange between water and ice starts. The flow is also intensive and is directed at the ice cover. The hydraulic and thermal conditions that exist in this region form the basic elements for the formation of open water. The open water formation mechanism can be described by the following process, as shown in **Figure 116**, with an assumption of ice covered channel, initially and no crack for ice cover.

Initially, ice warming and melting processes take place (1) at the end of the tailrace. This one sided melting process ends at the initial opening (2) of the ice cover. The initial opening can appear adjacent to or near to the concrete wall, depending on the geometry of the tailrace. Generally, the process observed in the outlet of the turbine is exactly the same in an open channel flow, e.g. in natural rivers. Melting at more than two sides can occur for the ice cover, if water flows out through the initial opening and flows over the ice surface. The melting rate depends on the water temperature and the discharge. If the discharge and temperature are sufficient to melt ice, the open water will be extended. This extension process (3a) takes place until the equilibrium state of the open water is reached. In the other case of temperature and discharge, ice cover will be extended (3b) in depth and eventually the downstream (channel) will be covered by ice again. In this extension process, aufeis (naled ice) (Schohl & Ettema, 1986) forms in the downstream channel of the hydropower plant and critical condition become worse for the bridge in downstream. Fortunately, the low variance of winter discharge of the Ider River provides calm forming of aufeis. These extending processes directly lead consequent equilibrium processes after an uncertain time. If the discharge and water temperature are sufficient to compare with thermal conditions of ice and the environment, open water will persist for long time in terms of its equilibrium state (4a) maintaining its opening. The significant fluctuation of discharge and temperature, in addition an environmental condition, leads the new ice cover at the upper or lower level of initial ice cover instead of open water.

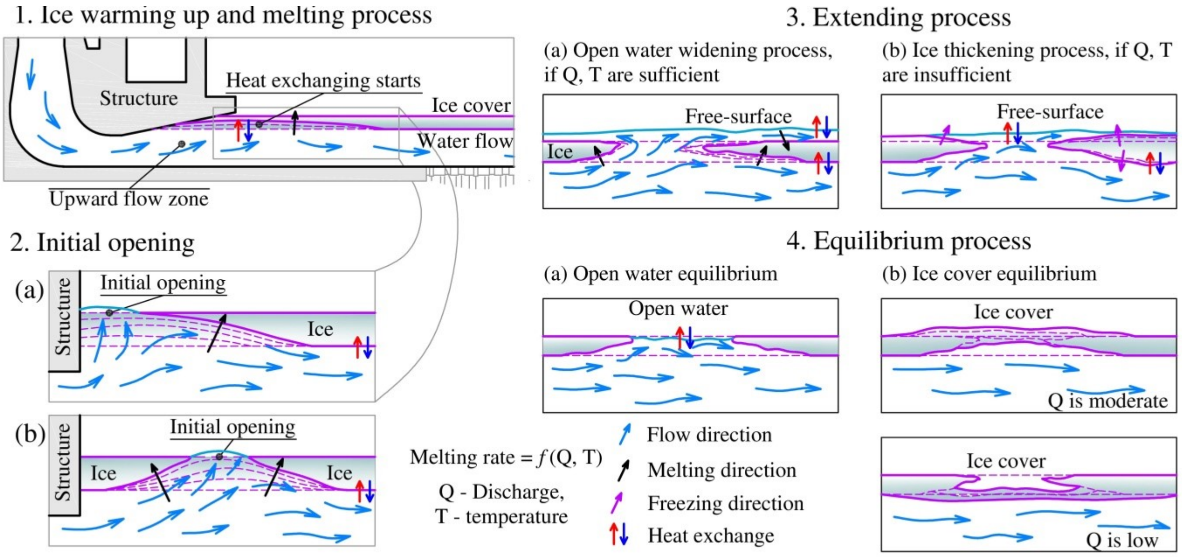


Figure 116. Open water forming mechanism in open channel flow based on the numerical investigation.

It is practically proved by the photo in **Figure 117** that the low discharge and low temperature lead open water enclosed by the ice again at lower levels in the Tosontsengel HPP.

The formation of open water in the downstream channel of the hydropower plant is successfully simulated to demonstrate the performance of the proposed LBM. The results indicate that the proposed model can be used for studying measures to control downstream ice in hydropower plants. It can also be applied to study the ice phenomena in open channel flows.

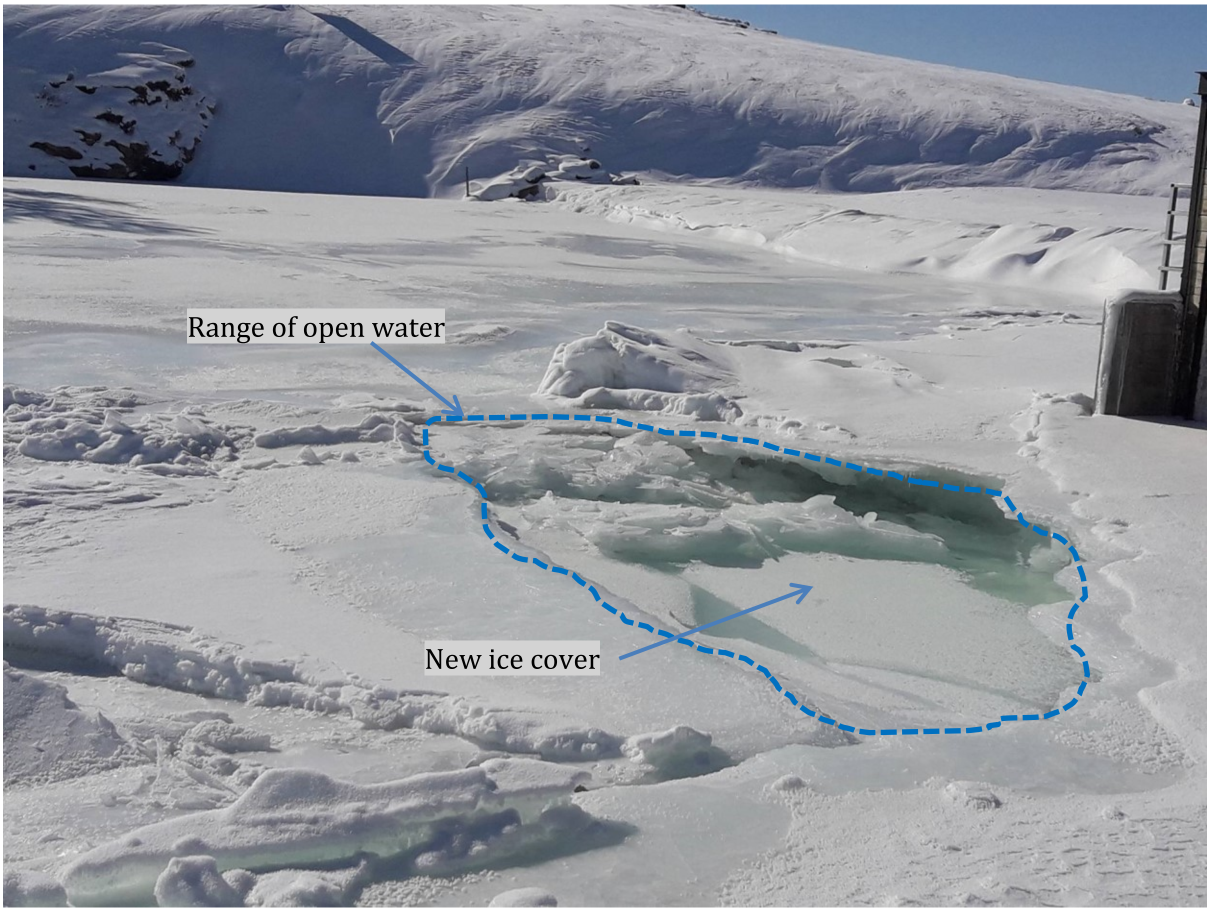


Figure 117. Open water next to the Tosontsengel HPP was enclosed by the ice, Feb 27, 2017.

8.4 Freely floating ice

The floating of ice bodies is an important part of a simulation of ice in a free surface flow. For simplicity, a simulation of an ice cylinder floating on the free surface is performed and compared with an experimental result in order to demonstrate the proposed algorithm for the floating body simulation. The ice cylinder with diameter D = 3 cm, initially positioned above the free surface at a distance h = D + D/2, falls into stagnant water and then floats on the water surface. The initial temperature of ice was -25°C, whereas the water temperature was 30°C. We tried to provide the same condition for the experiment and simulation. A grid spacing $\Delta x = 1.0\times10^{-3}$ m and a time step $\Delta t = 4.57\times10^{-5}$ s are used for the simulation. The simulation time was shorter than the melting of the ice cylinder.

In **Figure 118** (a), we compare the coordinate of the center of the ice cylinder measured as a function of time in the simulation and laboratory experiment. Overestimates of position can be observed at the turnings of the ice cylinder position, which might be caused by the underestimates of the hydrodynamic force in simulation. By contrast, an initial flow field might be affected on the ice cylinder movement in the experiment. Another source of the overestimates might be the choice of the Rayleigh number. **Figure 118** shows the numerical results at Ra = 10^4. When the Ra is greater than Ra = 10^4, the convective flow becomes active than the inertial flow induced by the movement of the ice body.

The temperature and velocity field shows symmetries except the splash during the ice cylinder plunging into the water in **Figure 118** (b). The shape of the cylinder at the end of the simulation was the same as the initial shape of the cylinder and the mass of the system was conserved during the simulation.

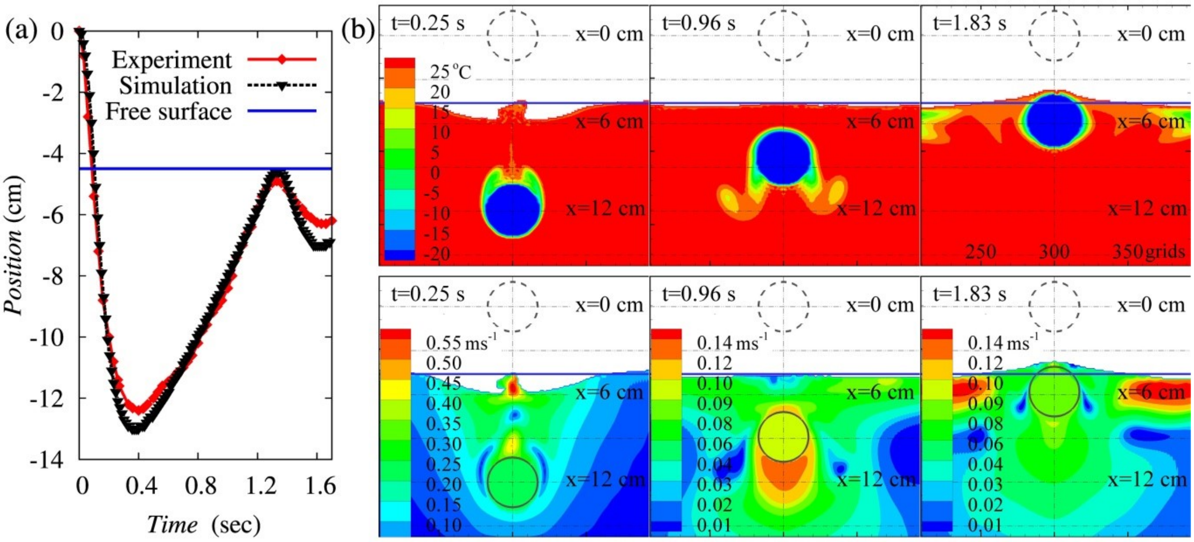


Figure 118. Falling and floating ice: (a) the comparison of the displacement of the ice cylinder measured in the simulation and lab experiment, (b) the computed temperature field (top) and flow field (bottom) at specific times.

Conclusively, the numerical results confirm that the proposed algorithm for the floating body satisfies the mass conservation, which has been difficult to be handled, and computes the involved free surface and heat transfer with reasonable accuracy. Nevertheless, shape changes of the floating body over time due to phase changes should be considered in further research. Finally, it can be said that the proposed model is not only capable of simulating ice dynamics in a free surface flow, but also capable of simulating dissolving related problems.

9 AN ATTEMPT TO THE LATTICE BOLTZMANN PARALLEL SIMULATION

The LBM is popular by it's inherit property of naturally suitable for a parallel simulation. We have made our attempt to the parallel simulation with the standard and Entropic LBM.

9.1 Parallel implementation of the Entropic LBM

In this study, the standard and entropic LBM are implemented in parallel way using the Fortran 90 programming language. A stable computation of the ELBM is gained by paying time and memory as the computational cost, which will be explained in Section 9.1.2. At the other point, engineering problems are often characterized as a large scale or long term phenomena in space and time. The LB solutions for these problems in central processing unit (CPU) based simulation ended up at difficult challenges because of the computational cost. The implementation of the ELBM on the graphics processing units (GPU) can solve the two important difficulties of computation: instability and high computational cost.

Nagaoka University of Technology (NUT) has a GPGPU system, which is available for students and staffs for their research activities (**Figure 119**).

Figure 119. General purposes graphics processing unit system at the NUT, Japan.

The GPGPU system is powered by 16 NVIDIA Tesla M2050 graphic processing units, which has the ability to perform 562 GFLOPs with double precision. Also CUDA Fortran version 11.3 has been installed on the GPGPU system. Grasping this technical potential and numerical method, we have made an attempt to implement an efficient parallel code for fluid flows on the GPGPU. In the CUDA programming, a code has to be devoted to two particular sections named a host code and a device code (Ruetsch & Fatica, 2011), as shown in **Figure 120**. A host code refers that the code executes on the host CPU machine, whereas a device code refers that the code executes in the GPU machine. The codes use memories on dedicated machines, i.e., variables used in the device code must use a GPU memory.

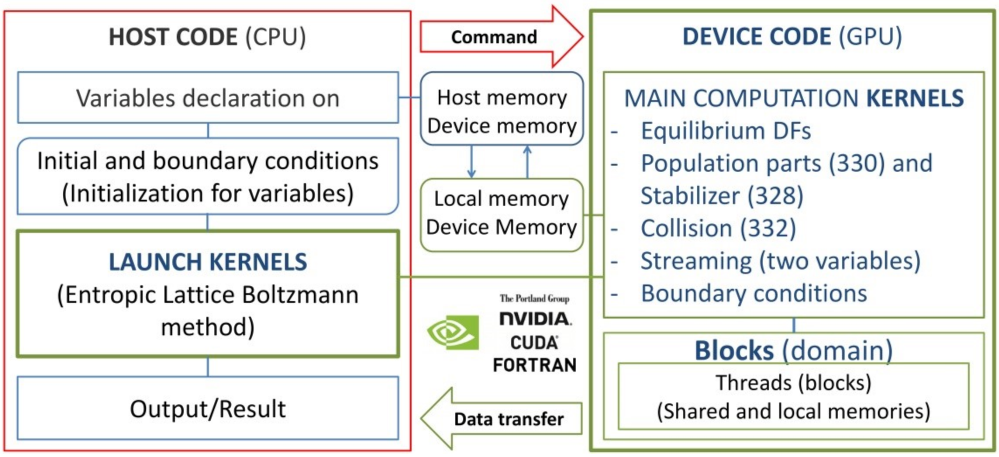


Figure 120. General architectural scheme of the implemented parallel code on the GPGPU system.

As the main program in the CUDA FORTRAN code is executed on the host, all variables in both host and device codes need to be declared specific memories of their use in the host code. Macroscopic variables, distribution functions and population parts are declared on the global memory (device memory) and could only be used in the device code. As the purpose of the ELBM code, initial conditions and loading geometric data are implemented on the host code, in which the device and host codes need to exchange data. So called kernels, subroutines in the Fortran language, implemented in the device code are launched from the host code. On the right hand side of **Figure 120**, the host code composing the kernels with used equations in numbers, which is performed on the predetermined blocks and threads in GPU architecture. An entropic part of the numerical algorithm tested as separate kernels, for instance a kernel for population part and a kernel for stabilizer. The separated kernels use low memory in consumption; however, the total computation time of two kernels was higher than their unified version in one kernel. The defining population parts and stabilizer is optimized to use low memory on the local memory of the GPU in one kernel, since the declaration of array in the kernel uses a lot of registrations, which might be caused the abortion of the kernels. Besides the streaming step, all required equations and formulae in the ELBM as well as the StLBM are easily implemented in a parallel way. To perform the streaming in an efficient way, two parallel distribution functions are used and they synchronize in the host code after the calculation of one time step. To improve performance of the parallel code, some optimization of array declaration as $f(x, y, i)$ for the distribution function instead of the regular array structure of the distribution function $f(i, x, y)$ is applied. Storing the constant parameters used in both codes in their memory save a small amount of the computational time. A single lattice is assumed to be a thread and a computational domain has divided as blocks horizontally.

9.1.1 Parallel performance of the ELBM code

The parallel code for the ELBM was implemented by the way that uses low memory and efficient computation, as explained above. We have solved flow past a cylinder in the laminar regime (Re = 200) to evaluate performance of the parallel code, since the StLBM can't survive at a high Reynolds number. The simulation domain formed as 300 grids in length and 100 grids in width. The simple bounce back boundary condition was imposed for the top wall, a bottom wall and a circular obstacle. The Zou/He boundary condition was applied with a characteristic velocity of u_0=0.1 for the inlet and the zero gradient boundary condition is used for the outlet boundary. The PGI Fortran compiler on the GPGPU system is used to perform parallel and serial computations to evaluate parallel performances. Time criteria to stop simulations was set at t = 50000 in all simulations.

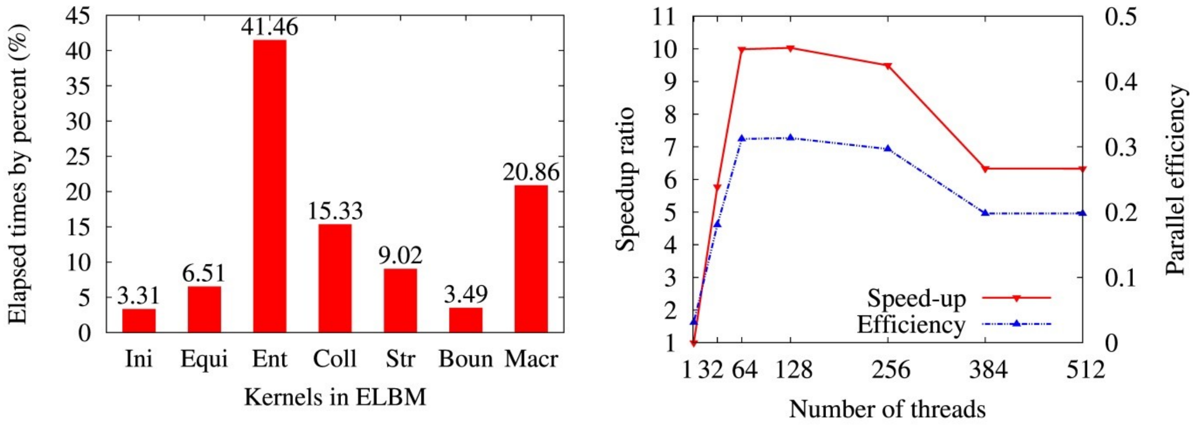


Figure 121. (a) Percent of elapsed times used by the kernels in single computational time of the ELBM in double precision. (b) Speedup ratio and parallel efficiency of the ELBM code with regards to the number of threads.

As investigating deeply the performance of the parallel ELBM code, an Entropic part (labeled "Ent" in **Figure 121** (a)), computing stabilizer and population parts, consumes around 41% of the all required time to perform kernels, shown in **Figure 121** (a). The second largest consumer was the defining macroscopic variables and synchronization of two groups of distribution functions (labeled "Macr" in **Figure 121** (a)), since the streaming process done by using the two groups of distribution functions.

The computational domain was divided as blocks in horizontally and threads in a block. A number of blocks can be chosen. **Figure 121** (b) shows speedup ratios and parallel efficiencies of the parallel computations on the GPGPU with respect to the number of threads used in the simulations. The parallel efficiency is determined as the ratio between the speedup ratios to the warp sizes of the GPU. The parallel code can speed up until 10 times faster than its serial version on the same machine. The highest speedup ratio was found in the 64 threads in a block. The computational efficiency would be convenient where the number of threads is lower than the maximum grid number of the domain. In other words, the number of threads should be smaller than the maximum number of grid in the horizontal or vertical direction.

Further, we examined the computational throughput and memory bandwidth for four cases of grid resolutions, shown in **Figure 122** (a). The number of thread was 512 in the simulations.

The computational throughput and memory bandwidth have a linear relation with the grid resolutions. Comparing to the maximum performance of the GPU, our code performs at the middle rate. Actually, a memory bandwidth was inefficient, uses a small amount of the possible memory bandwidth for the data transfer and management. Therefore, effective usage of the memory bandwidth in the parallel ELBM code needs to be explored. Elapsed time against the increase of grid resolutions plotted in **Figure 122** (b) shows the performances of the code on the different grid resolutions. Hence, the increase of the computational time has parabolic relation with the increases of grid resolution. Steepness of the parabola can be decreased if we increase the effective memory bandwidth in the code. To improve effectiveness of the code, possible kernels can be merged in such a way that uses low memory in the local memory of the device. Interestingly, the LBM index (Bailey, et al., 2009), which is an index to show the parallel performance of the LBM in the way that how much lattice is updated per second (LUPs), increased as the grid resolution increases in **Figure 122** (b).

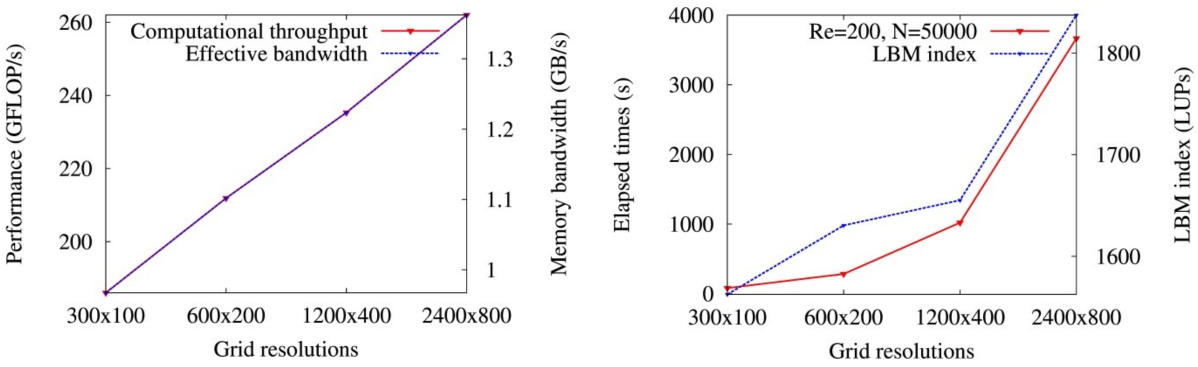


Figure 122. (a) Computational throughput and memory bandwidth against the grid resolutions. (b) The elapsed time and efficiency of ELBM computation against the grid resolutions.

9.1.2 High Reynolds number flow with the ELBM: flow past a circular cylinder

Flow past a bluff body has been widely studied both numerically and experimentally (Ong & Wallace, 1996) at high Reynolds numbers (Mittal & Balachandar, 1995) (Karabelas, 2010) (Rahman, et al., 2007) (Rajani, et al., 2016). The enhanced version of LBM using sub-grid scale (Hou, et al., 1996), (Liu, et al., 2008) and the direct numerical simulation like entropic approaches (Karlin, et al., 2014), (Ansumali, et al., 2004) are the main contributions to these studies. We have used a domain used in (Liu, et al., 2008) for our simulation. The boundary conditions are the same explained in Section 7.1.1. Preliminary test simulations were performed for the Re=1000 and Re=3900 in order to test the accuracy. We defined the Strouhal numbers and compared with the other studies conducted in the same condition using different techniques in **Table 7**.

Table 7. Strouhal numbers of flow around circular cylinder at different Reynolds number

Sources/methods of the results	Strouhal number
Re = 1000 – Present study	0.217
Extended LBM (using Smagorinsky sub-grid scale model) [12]	0.208
Experiment [13]	0.212
Standard k-epsilon model [5]	0.148
Re = 3900 – Present study	0.211
Extended LBM (using Smagorinsky sub-grid scale model) [12]	0.215
Experiment [13]	0.215
Standard k-epsilon model [5]	0.171
Re = 140000 – Present study	0.228
Extended LBM (using Smagorinsky sub-grid scale model) [12]	0.209

The Strouhal numbers defined by the ELBM were in good agreement with the results of experiments and other simulation methods. In **Figure 123**, the vorticity field of fully developed 2D turbulent flow at simulation time t = 10000 at Re = 140000 performed by the parallel ELBM is visualized.

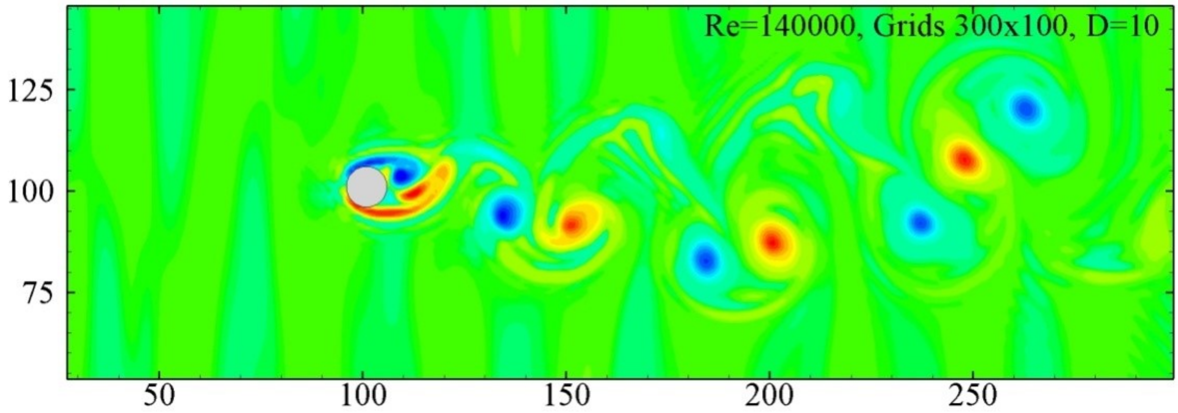


Figure 123. Vorticity field of flow past cylinder at Re=140000 on coarse grid.

A performance of the ELBM can clearly describe the Reynolds number effect on the cylinder. The flow field can be captured exactly comparing to the performance of the LBM extended by the Smagorinsky model. It was revealed that the Smagorinsky model for the LBM seems to overestimate the eddy viscosity (Ayurzana, 2016) at higher Reynolds number and it leads underestimates of velocity fields. Unlike it, the ELBM does not modify viscosity in the simulation and corrects velocity field by the maximum entropic condition. For coarse grid, the ELBM can resolve flow field and generate a vortex in the sub-grid. **Figure 124** shows the pressure coefficient distribution and the velocity magnitude in lattice form around the cylinder defined at Re = 140000. Based on the results obtained by the ELBM, the flow field can be computed with reasonable accuracy and stability at very high Reynolds numbers using the ELBM. It should be noted that the bounce back boundary condition might be affected on the accuracy in some case (Karlin, et al., 2014). In that case, proper boundary condition based on the kinetic theory must be used (Chikatamarla & Karlin, 2013).

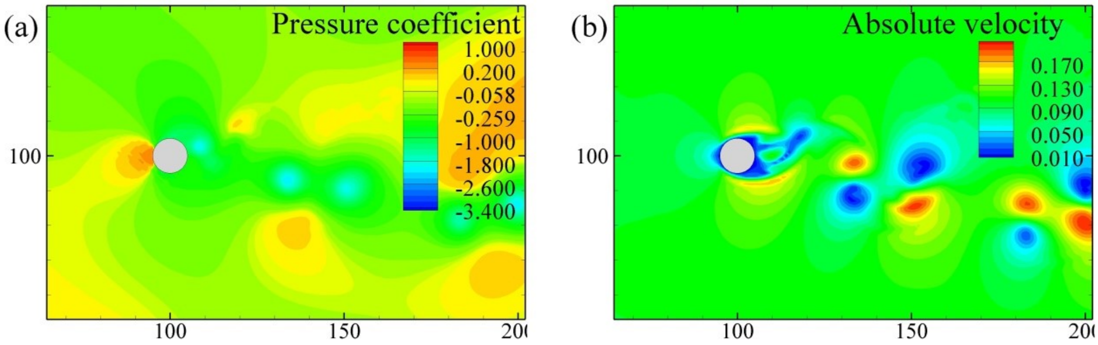


Figure 124. Pressure coefficient distribution and absolute velocity value at Re=140000

In summary, we have implemented an Entropic Lattice Boltzmann method in parallel operation using CUDA Fortran programming language on the NUT's GPGPU system. The entropic LBM is used to perform unconditionally stable simulations, while parallel computation is used to speed up simulations and improve efficiency of the code for the future large scale engineering simulations. The parallel code is aimed to be used for a large scale and long term simulation in engineering field in complicated geometry at high Reynolds number fluid flows. Further, we solved flow past a cylinder at Re=140000 to show the performance of the ELBM parallel code. To evaluate accuracy, the Strouhal number at Re=1000, 3900 are defined and compared with experiment and other numerical simulation results. The results were in good agreement. The parallel code of the ELBM is implemented in the combination of low memory usage and fast computation. Using the GPGPU for the ELBM, the computation can be performed 10 times faster than the serial code. Based on the stable and accurate computation of the ELBM, the parallel computation will be a very efficient method for CFD.

SUMMARY FOR PART 2

In this part, the mesoscopic numerical modeling called the LB models was extensively discussed with their applications. The numerical models aimed to solve ice problems in open channel flows which cover free surfaces, heat transfer, phase transitions and liquid-solid interactions. The numerical models were organized as the mesoscopic numerical framework as just like getting done in Part 1 for the macroscopic models.

The lattice Boltzmann model for the fluid and scalar transports were followed by the other models such as a free surface model and liquid-solid phase transition model. In the numerical implementation, the pseudo-potential codes for each model were described in great details. Numerical examples for each particular model were purposed to validate the model. The coupled models for, such as a natural convection, melting and freezing of water were also verified by the simple problems and experimental studies. The successfully validated numerical framework for the liquid-solid phase transition in free surface flows was applied to the open channel problems. The important application was the open water forming mechanism in an outlet channel of the small hydropower in Mongolia. Based on the numerical results, an idea of the mechanical solution to control ice in downstream of the hydropower plant is proposed.

The melting and freezing of ice in open channel flows was studied and shows the applicability of the proposed model for a short period of time. The application of the open water forming mechanism was resulted the explanation of the process of open water forming not only in the outlet of hydropower plant but also in natural rivers.

In the last, an attempt to the parallel computation with the LBM was investigated for single phase flows by the both standard and ELBM. The numerical procedure of the LBM best fit for the requirement of the parallel computation. With the parallel computation, we can get the 10 to 20 times faster simulations compared to the serial version of it. It is particularly crucial step for the application of the LBM to the real engineering problems.

The solution with the LB models of complex physics, ice in free surface flow, nominates the confidence that the LBM is applicable for various physical phenomena. We have a follow-up concept for particular problems described in the objectives of this research: sediment and ice. The concept is “all in one LBM” as depicted in **Figure 125**.

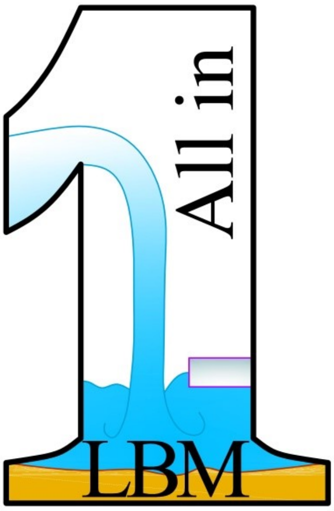


Figure 125. All in one LBM concept, where free surface flow pours into loose boundary with ice cover.

contains overall connections, conclusions to two parts of the research work and further considerations related to the subjects.

10.1 Conclusions

As specific objectives, we had learned to apply macro and mesoscopic models for particular problems that were sediment and ice in two parts, respectively. Each part has several contributions to the recent knowledge in research. For instance, in Part 1, the particle tracking method, which has superior to the traditional advection-diffusion based solution for sediment transport, is introduced. The particle tracking method treats each particle individual and describes local and global behavior of sediment transport based on the size of sediments. With respect to it, the settling velocity affected by the flocculation effect is also introduced to the particle tracking method as a simple model parameterized by an experiment. The flocculation model is applicable for an estuary where sediment discharged from river water to seawater. As a problem studied, the sediment transport in the estuary of the Ohkouzu diversion channel is investigated in the 2D and 3D space. In results, the characteristic of the spatial distribution of the sediment at different settling velocities and sizes was obtained. Concerns about the macroscopic models will be discussed with that of mesoscopic modeling in Chapter 10.3 as future works.

For Part 2, highlights are imbedded in the developed numerical model for a liquid-solid phase transition in a free surface flow, where many models are coupled to each other in first-time. For instance, the immersed boundary method is coupled with a free surface model and liquid-solid phase change treatment for water. The liquid-solid phase transition in a free surface flow considering the water itself is relatively new research work among the published articles. It is one certain contribution to the river ice research and unlocks the ice dynamics and thermodynamics. The free surface-immersed boundary LBM is applicable to the freely floating object in a free surface flow. Moreover, with the benefit of the scalar transport solution for liquid-solid phase changes, it is applicable for more complicated rarely studied subjects like a time-dependent arbitrary shaped floating object.

The sediment and ice modeling separately is achieved at certain degrees of satisfaction in this thesis. As the main objective, the research for both problems leaves the strong confidence that both modeling scales are possible for two problems can be solved simultaneously. Precisely, the lattice Boltzmann method is adequate for the small to mid-scale problems, while the macroscopic model is convenient for the mid to large scale problems like a river or lake. The lattice Boltzmann modeling can bring simple solutions for the sediment and ice as a generalized model. The particle tracking method proposed in the macroscopic modeling is combinational with the lattice Boltzmann modeling for the purpose of modeling particulate nature. Whether using the macro and mesoscopic modeling to the long-time, large-scale problem, one will naturally regard to high performance computing in an academic research.

10.2 Highlights and applications

The studies in two parts have highlighting points and the possible applications for different type of engineering problems. After listing highlights for two parts, we will commend applications of the models based on the concepts and methodology of models.

The macroscopic modeling of sediment transport in part 1

- demonstrates a new convenient benchmark problem for sediment and density currents, the lock-exchange with sediment. The benchmark problem has unique features to show interactions between currents induced by density or salt concentrations while showing the sediment transport and is suitable for the validation of new models.
- formulates a new particle tracking method (PTM), that treats each sediment particle individually in order to maintain the particle having independent and own velocity in fluid flows. The PTM is superior to the ADE in terms of the presentation of local and global dynamics of sediment and distribution analysis of sediment sizes.
- introduces a simple flocculation model, including the salting-out process that is well suited for the PTM and have an implicit effect on the settling rates of sediment particles. The flocculation model and its parameters were assured by the settling experiment.
- describes the conventional scheme for the sediment problem. This statement is made because employing the two different descriptive models for the same purpose at the same time gives interactive manipulations for modeling. For instance, the framework used in Part 1 (see **Figure 16**) has two sediment models and the results can be mutually assessed and compared each other that is rather trustful to use the single model for tricky problems.

The mesoscopic modeling of ice problems in part 2

- formulates the thermal-free surface-immersed boundary-lattice Boltzmann model from scratch. The model/method can be abbreviated as the T-FS-IB-LBM and it is a two-phase model.
- brings the novelties to the research field such as the LB solution for a liquid-solid phase transition in free surface flow, extended application of the IB formulation to phase transitions and floating body simulations etc.
- provides validation problems for a liquid-solid phase transition problems. With this statement, the author believes that the problems provided for validation can be the tests for new models designed for a liquid-solid phase transition.
- demonstrates a simple solution for the complex physics.
- introduces the LB solution to small to mesoscale hydraulics and ice problems. For instance, the T-FS-IB-LBM is partially applicable to the problems of fluid – solid interactions, free surface flows, porous media flows with free surfaces, scalar transports in free surface flows and combination of the above.

Possible applications of the PTM can be:

- sediment transport in river, reservoir and lakes

- wastewater treatment (flocculation or aggregation processes)
- salting-out process modeling of particulate matter
- and non-cohesive particle transport (sand and snow).

The model concept or its formulation of the T-FS-IB-LBM can be applied in

- metallurgy (metal modeling and foaming processes (Ammer, et al., 2014))
- ice problems in local and global (Esfahani, et al., 2016)
- volcanoes and lava flows
- solvent and solute
- and sediment and erosion/deposition.

10.3 Future recommendations

Two parts of research intersect on the two aspects: problem and coupling of modeling. As the problem, knowledges gained by separate studies for particular problems provide basic paths of modeling the problems together. In general, the sediment and ice considered at the same time in the same system are important and challenging subject. Background researches based on experiments and observations and their reviews have been provided extensively in (Ettema, et al., 2000), (Lau, et al., 1985), (Ettema & Daly, 2004), (Turcotte & Morse, 2013) and other articles. The mutual effects of ice and sediment (Ettema, 2002) should be studied and modelled by either macro or mesoscopic numerical modeling. One direct method for the problems is coupling of contributions of this dissertation. In other words, the flow and ice are modeled by the T-FS-IB-LB, while the PTM can take care of the sediments. Otherwise, the one can exploit the capability of the LB models for the problems obeying our proposed concept (see **Figure 125**).

As having the same aim, the two types of models, macro and mesoscopic modeling can be coupled with each other in different patterns. While using the macroscopic model for a transport phenomenon, the mesoscopic model can be used to solve a fluid flow (Hlushkou, et al., 2004). Also, existing knowledge from the FDM is usually applicable for the LBM (Junk, 2001). From the differences of underlying theory and procedure, macro and mesoscopic modeling carry irreplaceable properties with them. For instance, the LBM is suitable for complex shaped geometry, while the FDM may be comfortable for large scale computation. With this respect, the methods can serve their duty for their effective parts of a domain (Albuquerque, et al., 2004).

Herein we enumerate the bottlenecks in two parts as future works. In part 1, the PTM should receive more improvements in aspects of hindered settling, sediment particle-particle interactions, particle-fluid interactions and de-flocculation effects. Also the quantitative variables, e.g., deposition volume, bed and suspended loads are expected in the PTM. The poor connections to concentrate based evaluation should be explored if it is necessary. Even the particle is a representative particle in the model, the crowd of the particle effects for the fluid viscosity. So the relation between viscosity and particle density in the numerical framework should be expounded. To overcome the main disadvantage of the PTM, one can parallelize the model in the implementation (Mooney, 1951).

For the liquid-solid phase transition in free surface flows in Part 2, ice mechanical modeling such as cracking and crushing (Beltaos, 1990) must be included to be a more accurate modeling tool to be applied in the river ice dynamics. By virtue of the immersed boundary modification, the proposed model for the liquid-solid phase transition can be superior than the concentration based modeling of ice. The solution algorithm for the arbitrary shaped floating body should be more elaborated for general purpose and be extended to the multi-body interactions. The models are straightforward to the 3D formulation and computation. The algorithm for the ice melting shares the idea that can be used for the local erosion and scour (Ettema, 1980). Also, the LB model for the sediment transport (Masselot, 1998) needs to get new pace into the application.

As mentioned, the parallel implementations of the models are crucial to be effective and modern.

contains the connection between the lattice Boltzmann equation and the macroscopic governing equation such as the continuity equation, the NSE and the advection-diffusion equation.

A1 Derivation of the Navier-Stokes equation

We can simply check that the discrete Boltzmann equation can produce the macroscopic continuity and momentum equation using the moments of the distribution function. However, now, we will deal with the method by which the macroscopic equations can be reduced from the lattice Boltzmann equation. The method is celebrated as the Chapman-Enskog (Guo & Shu, 2013) (A.A.Mohamad, 2011) approximation that analysis the equation in multiple scales.

The distribution function can be approximated as an infinite series constructed by its perturbations with the small parameter ϵ:

$$f_i(\mathbf{x},t) = f_i^{(0)}(\mathbf{x},t) + \epsilon f_i^{(1)}(\mathbf{x},t) + \epsilon^2 f_i^{(2)}(\mathbf{x},t) + \cdots \ , \qquad (A.1)$$

where the small parameter is interpreted as (a) the small parameter, $|\epsilon| \ll 1$, expanding the function in a series, which can be assumed to be a Knudsen number, Kn, in case of Chapman-Enskog expansion and (b) the magnitude classifier for the perturbations, which can be a unit, $\epsilon = 1$, to be cancelled after the result derived (Wolf-Gladrow, 2000). In this expansion, the following properties are valid:

$$\sum_i f_i^{(1)}(x,t) = 0\,, \qquad \sum_i \mathbf{c}_i f_i^{(1)}(x,t) = 0\,, \ \text{ and} \qquad (A.2a)$$

$$\sum_i f_i^{(2)}(x,t) = 0\,, \qquad \sum_i \mathbf{c}_i f_i^{(2)}(x,t) = 0\,, \qquad (A.2b)$$

which means that the perturbations cannot contribute to the mass and momentum conservation. The lattice Boltzmann equation with a force term can be rewritten from Eq.(187) as follow

$$f_i(\mathbf{x} + \delta x, t + \delta t) = f_i(\mathbf{x},t) + \frac{1}{\tau_v}\left(f_i^{(0)} - f_i\right) + F_i \frac{\mathbf{c} - \mathbf{u}}{RT} f_i^{(0)}\,. \qquad (A.3)$$

The left hand side and the force term in Eq.(A.3) can be expanded into a Taylor series up to terms of second order of $\delta x = \mathbf{c}_i \delta t$:

$$f_i(\mathbf{x} + \mathbf{c}_i\delta t, t + \delta t) = f_i(x,t) + \delta t \frac{\partial f_i}{\partial t} + \delta t c_{i\alpha} \frac{\partial f_i}{\partial x_\alpha}$$
$$+ \frac{(\delta t)^2}{2}\left[\frac{\partial^2 f_i}{\partial t^2} + 2c_{i\alpha}\frac{\partial^2 f_i}{\partial t \partial x_\alpha} + c_{i\alpha}c_{i\beta}\frac{\partial^2 f_i}{\partial x_\alpha \partial x_\beta}\right] + O[(\delta t)^3] \qquad (A.4)$$

and

$$F_i \frac{\mathbf{c} - \mathbf{u}}{RT} f_i^{(0)} = \frac{\mathrm{c}_{i\alpha}}{RT} f_i^{(0)} F_\alpha(\mathbf{x},t)_{\lceil \text{to expand}}$$

$$= \delta t \frac{\mathrm{c}_{i\alpha}}{RT} f_i^{(0)} F_\alpha + \frac{(\delta t)^2}{2} \frac{\mathrm{c}_{i\alpha}}{RT} f_i^{(0)} \left[\frac{\partial F_\alpha}{\partial t} + c_{i\alpha} \frac{\partial F_\beta}{\partial x_\beta} \right] + O[(\delta t)^3]\,, \tag{$A.5$}$$

where we have used that the peculiar velocity is equivalent with the discrete velocity at an equilibrium state. Substituting the expansions in Eq.(A.3) gives

$$\begin{aligned} 0 = \frac{1}{\tau_v} f_i^{(0)} - \frac{1}{\tau_v} f_i - \delta t \frac{\partial f_i}{\partial t} - \delta t c_{i\alpha} \frac{\partial f_i}{\partial x_\alpha} \\ - \frac{(\delta t)^2}{2} \left[\frac{\partial^2 f_i}{\partial t^2} + 2 c_{i\alpha} \frac{\partial^2 f_i}{\partial t \partial x_\alpha} + c_{i\alpha} c_{i\beta} \frac{\partial^2 f_i}{\partial x_\alpha \partial x_\beta} \right] \\ + \delta t \frac{\mathrm{c}_{i\alpha}}{RT} f_i^{(0)} F_\alpha + \frac{(\delta t)^2}{2} \frac{\mathrm{c}_{i\alpha}}{RT} f_i^{(0)} \left[\frac{\partial F_\alpha}{\partial t} + c_{i\alpha} \frac{\partial F_\beta}{\partial x_\beta} \right] + O[(\delta t)^3]\,. \end{aligned} \tag{$A.6$}$$

Now, introducing two time scales and one spatial scale with the following scales,

$$\frac{\partial}{\partial t} \to \epsilon \frac{\partial}{\partial t^{(1)}} + \epsilon^2 \frac{\partial}{\partial t^{(2)}} \text{ and } \frac{\partial}{\partial x_\alpha} \to \epsilon \frac{\partial}{\partial x_\alpha^{(1)}}\,, \tag{$A.7$}$$

and the expansion in Eq.(A.1) to Eq.(A.6) and limiting up to order of ϵ^2 yields

$$\begin{aligned} 0 = \delta t \epsilon \frac{\partial f^{(0)}}{\partial t^{(1)}} + \delta t \epsilon^2 \frac{\partial f^{(1)}}{\partial t^{(1)}} + \delta t \epsilon^2 \frac{\partial f^{(0)}}{\partial t^{(2)}} + \delta t c_{i\gamma} \epsilon \frac{\partial f^{(0)}}{\partial x_\gamma^{(1)}} + \delta t c_{i\gamma} \epsilon^2 \frac{\partial f^{(1)}}{\partial x_\gamma^{(1)}} \\ + \frac{(\delta t)^2}{2} \epsilon^2 \frac{\partial^2 f^{(0)}}{\partial t^{(1)2}} + (\delta t)^2 c_{i\gamma} \epsilon^2 \frac{\partial^2 f^{(0)}}{\partial t^{(1)} \partial x_\gamma^{(1)}} + \frac{(\delta t)^2}{2} c_{i\beta} c_{i\gamma} \epsilon^2 \frac{\partial^2 f^{(0)}}{\partial x_\beta^{(1)} \partial x_\gamma^{(1)}} + w \epsilon f^{(1)} + w \epsilon^2 f^{(2)} \\ - \epsilon \delta t \frac{\mathrm{c}_{i\alpha}}{RT} f_i^{(0)} F_\alpha - \frac{(\delta t)^2}{2} \frac{c_{i\alpha}}{RT} f^{(0)} \epsilon^2 \frac{\partial F_\alpha}{\partial t^{(1)}} \\ - \frac{(\delta t)^2}{2} \frac{c_{i\alpha}}{RT} f^{(0)} \epsilon^2 c_{i\beta} \frac{\partial F_\alpha}{\partial x_\alpha^{(1)}} + O[(\delta t)^3] + O[\epsilon^3]\,. \end{aligned} \tag{$A.8$}$$

Sorting in order of ϵ and grouping the terms in G gives

$$0 = \epsilon G_i^0 + \epsilon^2 G_i^1 + O[(\delta t)^3] + O[\epsilon^3] \tag{$A.9$}$$

with groups:

$$G_i^0 = \frac{\partial f^{(0)}}{\partial t^{(1)}} + c_{i\gamma} \frac{\partial f^{(0)}}{\partial x_\gamma^{(1)}} + \frac{w}{\delta t} f^{(1)} - \frac{\mathrm{c}_{i\alpha}}{RT} f_i^{(0)} F_\alpha \text{ and} \tag{A.10a}$$

$$\begin{aligned} G_i^1 = \frac{\partial f^{(1)}}{\partial t^{(1)}} + \frac{\partial f^{(0)}}{\partial t^{(2)}} + c_{i\gamma} \frac{\partial f^{(1)}}{\partial x_\gamma^{(1)}} + \frac{\delta t}{2} \frac{\partial^2 f^{(0)}}{\partial t^{(1)2}} + \delta t c_{i\gamma} \frac{\partial^2 f^{(0)}}{\partial t^{(1)} \partial x_\gamma^{(1)}} + \frac{\delta t}{2} c_{i\beta} c_{i\gamma} \frac{\partial^2 f^{(0)}}{\partial x_\beta^{(1)} \partial x_\gamma^{(1)}} \\ + \frac{w}{\delta t} f^{(2)} - \frac{\delta t}{2} \frac{c_{i\alpha}}{RT} f^{(0)} \frac{\partial F_\alpha}{\partial t^{(1)}} - \frac{\delta t}{2} \frac{c_{i\alpha}}{RT} f^{(0)} c_{i\beta} \frac{\partial F_\alpha}{\partial x_\alpha^{(1)}}\,. \end{aligned} \tag{$A.10b$}$$

First, we analyze the first term in Eq.(A.9). The zeroth and first order discrete velocity moments of G_i^0 can be estimated as

$$\sum_i G_i^0 = \sum_i \left[\underbrace{\frac{\partial f^{(0)}}{\partial t^{(1)}}}_{\rho} + c_{i\gamma} \underbrace{\frac{\partial f^{(0)}}{\partial x_\gamma^{(1)}}}_{\rho u_\gamma} + \underbrace{\frac{w}{\delta t} f^{(1)}}_{0} - \underbrace{\frac{c_{i\alpha}}{RT} f_i^{(0)} F_\alpha}_{0} \right]$$

$$= \frac{\partial \rho}{\partial t^{(1)}} + \frac{\partial(\rho u_\gamma)}{\partial x_\gamma^{(1)}} \quad \text{and} \qquad (A.11a)$$

$$\sum_i c_{i\alpha} G_i^0 = \sum_i c_{i\alpha} \left[\frac{\partial f^{(0)}}{\partial t^{(1)}} + c_{i\gamma} \frac{\partial f^{(0)}}{\partial x_\gamma^{(1)}} + \frac{w}{\delta t} f^{(1)} - \frac{c_{i\alpha}}{RT} f_i^{(0)} F_\alpha \right]$$

$$= \underbrace{\frac{\partial}{\partial t^{(1)}} \sum_i c_{i\alpha} f^{(0)}}_{\rho u_\alpha} + \underbrace{\frac{\partial}{\partial x_\gamma^{(1)}} \sum_i c_{i\alpha} c_{i\gamma} f^{(0)}}_{P_{\alpha\beta}^0} + \underbrace{\frac{w}{\delta t} \sum_i c_{i\alpha} f^{(1)}}_{0} - \underbrace{\frac{1}{RT} \sum_i c_{i\alpha} c_{i\beta} f_i^{(0)} F_\alpha}_{\rho RT F_\alpha}$$

$$= \frac{\partial(\rho u_\alpha)}{\partial t^{(1)}} + \frac{\partial P_{\alpha\beta}^0}{\partial x_\gamma^{(1)}} - \rho F_\alpha \, , \qquad (A.11b)$$

where

$$P_{\alpha\beta}^0 = \rho u_\alpha u_\beta + \rho RT \delta_{\alpha\beta} \qquad (A.12c)$$

We have used Eqs.(210), (211),(A.2) and (178) for Eq.(A.11a) and Eqs.(211), (A.12c), (A.2) and the second term of (A.12c) for Eq.(A.11b). The first term in Eq.(A.9) readily gives the continuity equation with the zeroth order moment

$$\epsilon \frac{\partial \rho}{\partial t^{(1)}} + \epsilon \frac{\partial(\rho u_\gamma)}{\partial x_\gamma^{(1)}} = 0 \qquad (A.13)$$

and gives the Euler equation with the first order moment

$$\epsilon \frac{\partial(\rho u_\alpha)}{\partial t^{(1)}} + \epsilon \frac{\partial(\rho u_\alpha u_\beta)}{\partial x_\beta^{(1)}} + \epsilon \frac{\partial \rho}{\partial x_\beta^{(1)}} RT \delta_{\alpha\beta} - \epsilon \rho F_\alpha = 0 \, , \qquad (A.14)$$

which is the Navier-Stokes equation without the viscosity term.

Second, we analyze the second term in Eq.(A.9). The zeroth order discrete velocity moments of G_i^1 can be computed as

$$\sum_i G_i^1 = \underbrace{\frac{\partial}{\partial t^{(1)}} \sum_i f^{(1)}}_{0} + \underbrace{\frac{\partial}{\partial t^{(2)}} \sum_i f^{(0)}}_{\rho} + \underbrace{\frac{\partial}{\partial x_\gamma^{(1)}} \sum_i c_{i\gamma} f^{(1)}}_{0}$$

$$+ \underbrace{\frac{\delta t}{2} \frac{\partial^2}{\partial t^{(1)2}} \sum_i f^{(0)}}_{\rho} + \underbrace{\delta t \frac{\partial^2}{\partial t^{(1)} \partial x_\gamma^{(1)}} \sum_i f^{(0)} c_{i\gamma}}_{\rho u_\alpha} + \underbrace{\frac{\delta t}{2} \frac{\partial^2}{\partial x_\beta^{(1)} \partial x_\gamma^{(1)}} \sum_i c_{i\beta} c_{i\gamma} f^{(0)}}_{P_{\alpha\beta}^0}$$

$$+ \underbrace{\frac{w}{\delta t} \sum_i f^{(2)}}_{0} - \underbrace{\frac{\delta t}{2} \frac{1}{RT} \frac{\partial}{\partial t^{(1)}} \sum_i c_{i\alpha} f^{(0)} F_\alpha}_{0} - \underbrace{\frac{\delta t}{2} \frac{1}{RT} \frac{\partial}{\partial x_\alpha^{(1)}} \sum_i c_{i\alpha} c_{i\beta} f^{(0)} F_\alpha}_{\rho RT F_\alpha}$$

$$= \frac{\partial \rho}{\partial t^{(2)}} + \underbrace{\frac{\delta t}{2} \frac{\partial^2 \rho}{\partial t^{(1)2}}}_{(a)} + \underbrace{\delta t \frac{\partial^2 (\rho u_\alpha)}{\partial t^{(1)} \partial x_\alpha^{(1)}}}_{(b)} + \underbrace{\frac{\delta t}{2} \frac{\partial^2 P_{\alpha\beta}^0}{\partial x_\beta^{(1)} \partial x_\alpha^{(1)}}}_{(c)} - \frac{\delta t}{2} \frac{1}{RT} \frac{RT \partial \rho F_\alpha}{\partial t^{(2)}} \, , \qquad (A.15)$$

where

$$(a) \rightarrow \frac{\partial^2\rho}{\partial t^{(1)^2}} = \frac{\partial\left(\frac{\partial\rho}{\partial t^{(1)}}\right)_{\lceil Eq.A.13}}{\partial t^{(1)}} = -\frac{\partial^2(\rho u_\alpha)}{\partial t^{(1)}\partial x_\alpha^{(1)}} = -\frac{\partial}{\partial x_\alpha^{(1)}}\frac{\partial(\rho u_\alpha)}{\partial t^{(1)}}_{\lceil Eq.A.14}$$

$$= -\left[\frac{\partial\rho F_\alpha}{\partial x_\alpha^{(1)}} - \frac{\partial^2(\rho u_\alpha u_\beta)}{\partial x_\alpha^{(1)}\partial x_\beta^{(1)}} - \frac{\partial^2\rho}{\partial x_\alpha^{(1)}\partial x_\beta^{(1)}}RT\delta_{\alpha\beta}\right],$$

$$(b) \rightarrow \frac{\partial^2(\rho u_\alpha)}{\partial t^{(1)}\partial x_\alpha^{(1)}} = \frac{\partial}{\partial x_\alpha^{(1)}}\frac{\partial(\rho u_\alpha)}{\partial t^{(1)}}_{\lceil Eq.A.14}$$

$$= \frac{\partial\rho F_\alpha}{\partial x_\alpha^{(1)}} - \frac{\partial^2(\rho u_\alpha u_\beta)}{\partial x_\alpha^{(1)}\partial x_\beta^{(1)}} - \frac{\partial^2\rho}{\partial x_\alpha^{(1)}\partial x_\beta^{(1)}}RT\delta_{\alpha\beta} \text{ and}$$

$$(c) \rightarrow \frac{\partial^2 P^0_{\alpha\beta\,\lceil Eq.A.12c}}{\partial x_\beta^{(1)}\partial x_\alpha^{(1)}} = \frac{\partial^2(\rho u_\alpha u_\beta)}{\partial x_\alpha^{(1)}\partial x_\beta^{(1)}} + \frac{\partial^2\rho}{\partial x_\alpha^{(1)}\partial x_\beta^{(1)}}RT\delta_{\alpha\beta}\,.$$

Substituting (a), (b) and (c) in Eq.(A.15) yields

$$\sum_i G_i^1 = \frac{\partial\rho}{\partial t^{(2)}}\,, \qquad (A.16)$$

which ensures that there is no mass diffusion and mass is conserved. Before proceeding with the first order momentum, let us define $f_i^{(1)}$ from Eq.(A.10a) with a condition of $G_i^0 = 0$,

$$0 = \frac{\partial f^{(0)}}{\partial t^{(1)}} + c_{i\gamma}\frac{\partial f^{(0)}}{\partial x_\gamma^{(1)}} + \frac{w}{\delta t}f^{(1)} - \frac{c_{i\alpha}}{RT}f_i^{(0)}F_\alpha \rightarrow$$

$$\rightarrow f^{(1)} = \frac{\delta t}{w}\frac{c_{i\alpha}}{RT}f_i^{(0)}F_\alpha - \frac{\delta t}{w}\frac{\partial f^{(0)}}{\partial t^{(1)}} - \frac{\delta t}{w}c_{i\gamma}\frac{\partial f^{(0)}}{\partial x_\gamma^{(1)}}\,. \qquad (A.17)$$

The first order discrete velocity moment of G_i^1 is computed as

$$\sum_i c_{i\alpha}G_i^1 = \underbrace{\frac{\partial}{\partial t^{(1)}}\sum_i c_{i\alpha}f^{(1)}}_{(a)} + \underbrace{\frac{\partial}{\partial t^{(2)}}\sum_i c_{i\alpha}f^{(0)}}_{(b)} + \underbrace{\frac{\partial}{\partial x_\gamma^{(1)}}\sum_i c_{i\gamma}c_{i\alpha}f^{(1)}}_{(c)}$$

$$+ \underbrace{\frac{\delta t}{2}\frac{\partial^2}{\partial t^{(1)^2}}\sum_i c_{i\alpha}f^{(0)}}_{(d)} + \underbrace{\delta t\frac{\partial^2}{\partial t^{(1)}\partial x_\gamma^{(1)}}\sum_i c_{i\alpha}c_{i\gamma}f^{(0)}}_{(e)}$$

$$+ \underbrace{\frac{\delta t}{2}\frac{\partial^2}{\partial x_\beta^{(1)}\partial x_\gamma^{(1)}}\sum_i c_{i\beta}c_{i\gamma}c_{i\alpha}f^{(0)}}_{(f)} + \underbrace{\frac{w}{\delta t}\sum_i f^{(2)}c_{i\alpha}}_{(g)}$$

$$- \underbrace{\frac{\delta t}{2}\frac{1}{RT}\frac{\partial}{\partial t^{(1)}}\sum_i c_{i\alpha}c_{i\beta}f^{(0)}F_\alpha}_{(h)} - \underbrace{\frac{\delta t}{2}\frac{1}{RT}\frac{\partial}{\partial x_\alpha^{(1)}}\sum_i c_{i\alpha}c_{i\beta}c_{i\gamma}f^{(0)}F_\alpha}_{(i)}\,, \qquad (A.18)$$

where

$$(a) \rightarrow \frac{\partial}{\partial t^{(1)}}\sum_i c_{i\alpha}f^{(1)} = 0 \text{ according to Eq. (A. 2)}$$

$$(b) \rightarrow \underbrace{\frac{\partial}{\partial t^{(2)}} \sum_i c_{i\alpha} f^{(0)}}_{\rho u_\alpha} = \frac{\partial (\rho u_\alpha)}{\partial t^{(2)}}$$

$$(c) \rightarrow \underbrace{\frac{\partial}{\partial x_\gamma^{(1)}} \sum_i c_{i\gamma} c_{i\alpha} f^{(1)}}_{Eq.A.17} = \frac{\partial}{\partial x_\gamma^{(1)}} \sum_i c_{i\gamma} c_{i\alpha} \left[\frac{\delta t}{w} \frac{\mathrm{c}_{i\alpha}}{RT} f_i^{(0)} F_\alpha - \frac{\delta t}{w} \frac{\partial f^{(0)}}{\partial t^{(1)}} - \frac{\delta t}{w} c_{i\gamma} \frac{\partial f^{(0)}}{\partial x_\gamma^{(1)}} \right]$$

$$= \underbrace{\frac{\partial}{\partial x_\beta^{(1)}} \frac{\delta t}{w} \frac{1}{RT} \sum_i \mathrm{c}_{i\alpha} c_{i\beta} c_{i\gamma} f_i^{(0)} F_\alpha}_{0} \quad \underbrace{- \frac{\partial}{\partial x_\beta^{(1)}} \frac{\delta t}{w} \frac{\partial}{\partial t^{(1)}} \sum_i c_{i\alpha} c_{i\beta} f^{(0)}}_{(*)}$$

$$\underbrace{- \frac{\delta t}{w} \frac{\partial}{\partial x_\beta^{(1)}} \frac{\partial}{\partial x_\gamma^{(1)}} \sum_i c_{i\alpha} c_{i\beta} c_{i\gamma} f^{(0)}}_{(**)}$$

$$(d) \rightarrow \underbrace{\frac{\delta t}{2} \frac{\partial^2}{\partial {t^{(1)}}^2} \sum_i c_{i\alpha} f^{(0)}}_{\rho u_\alpha} = \underbrace{\frac{\delta t}{2} \frac{\partial^2 (\rho u_\alpha)}{\partial {t^{(1)}}^2}}_{Eq.A.11b} = \frac{\delta t}{2} \frac{\partial}{\partial t^{(1)}} \left[\rho F_\alpha - \frac{\partial P_{\alpha\beta}^0}{\partial x_\beta^{(1)}} \right]$$

$$(e) \rightarrow \underbrace{\delta t \frac{\partial^2}{\partial t^{(1)} \partial x_\gamma^{(1)}} \sum_i c_{i\alpha} c_{i\gamma} f^{(0)}}_{(*)}$$

$$(f) \rightarrow \underbrace{\frac{\delta t}{2} \frac{\partial^2}{\partial x_\beta^{(1)} \partial x_\gamma^{(1)}} \sum_i c_{i\beta} c_{i\gamma} c_{i\alpha} f^{(0)}}_{(**)}$$

$$(g) \rightarrow \underbrace{\frac{w}{\delta t} \sum_i f^{(2)} c_{i\alpha}}_{Eq.A.2a} = 0$$

$$(h) \rightarrow \underbrace{\frac{\delta t}{2} \frac{1}{RT} \frac{\partial}{\partial t^{(1)}} \sum_i c_{i\alpha} c_{i\beta} f^{(0)} F_\alpha}_{\rho RT F_\alpha} = \frac{\delta t}{2} \frac{\partial \rho F_\alpha}{\partial t^{(1)}}$$

$$(i) \rightarrow \frac{\delta t}{2} \frac{1}{RT} \frac{\partial}{\partial x_\alpha^{(1)}} \sum_i c_{i\alpha} c_{i\beta} c_{i\gamma} f^{(0)} F_\alpha = 0\,.$$

In above, the second and third order moments of $f^{(0)}$ has been denoted by asterisk. Summation of (*)-terms in (c) and (e) gives

$$- \frac{\partial}{\partial x_\beta^{(1)}} \frac{\delta t}{w} \frac{\partial}{\partial t^{(1)}} \sum_i c_{i\alpha} c_{i\beta} f^{(0)} + \delta t \frac{\partial^2}{\partial t^{(1)} \partial x_\gamma^{(1)}} \sum_i c_{i\alpha} c_{i\gamma} f^{(0)}$$

$$= \left(\delta t - \frac{\delta t}{w} \right) \frac{\partial^2}{\partial t^{(1)} \partial x_\beta^{(1)}} \underbrace{\sum_i c_{i\alpha} c_{i\beta} f^{(0)}}_{P_{\alpha\beta}^0} = \left(\delta t - \frac{\delta t}{w} \right) \frac{\partial^2 P_{\alpha\beta}^0}{\partial t^{(1)} \partial x_\beta^{(1)}}\,. \qquad (A.19a)$$

Summation of the (**)-terms in (c) and (f) results in

$$- \frac{\delta t}{w} \frac{\partial}{\partial x_\beta^{(1)}} \frac{\partial}{\partial x_\gamma^{(1)}} \sum_i c_{i\alpha} c_{i\beta} c_{i\gamma} f^{(0)} + \frac{\delta t}{2} \frac{\partial^2}{\partial x_\beta^{(1)} \partial x_\gamma^{(1)}} \sum_i c_{i\beta} c_{i\gamma} c_{i\alpha} f^{(0)}$$

$$\left(\frac{\delta t}{2} - \frac{\delta t}{w} \right) \frac{\partial^2}{\partial x_\beta^{(1)} \partial x_\gamma^{(1)}} \underbrace{\sum_i c_{i\beta} c_{i\gamma} c_{i\alpha} f^{(0)}}_{P_{\alpha\beta\gamma}^0} = \left(\frac{\delta t}{2} - \frac{\delta t}{w} \right) \frac{\partial^2 P_{\alpha\beta\gamma}^0}{\partial x_\beta^{(1)} \partial x_\gamma^{(1)}}\,, \qquad (A.19b)$$

where

$$P^0_{\alpha\beta\gamma} = \sum_i c_{i\beta}c_{i\gamma}c_{i\alpha}f^{(0)} = \rho u_\alpha u_\beta u_\gamma + \rho RT\big(u_\alpha\delta_{\alpha\gamma} + u_\beta\delta_{\beta\gamma} + u_\gamma\delta_{\alpha\beta}\big)\,. \qquad (A.20)$$

Summing up (a) to (i) in Eq.(A.18) gives

$$\sum_i c_{i\alpha}G_i^1 = \frac{\partial(\rho u_\alpha)}{\partial t^{(2)}} + \left(\frac{\delta t}{2} - \frac{\delta t}{w}\right)\underbrace{\frac{\partial^2 P^0_{\alpha\beta}}{\partial t^{(1)}\partial x_\beta^{(1)}}}_{(j)} + \left(\frac{\delta t}{2} - \frac{\delta t}{w}\right)\underbrace{\frac{\partial^2 P^0_{\alpha\beta\gamma}}{\partial x_\beta^{(1)}\partial x_\gamma^{(1)}}}_{(k)}\,, \qquad (A.19c)$$

where

$$(j) \to \frac{\partial^2 P^0_{\alpha\beta}}{\partial t^{(1)}\partial x_\beta^{(1)}} = \frac{\partial^2\big(\rho u_\alpha u_\beta\big)}{\partial t^{(1)}\partial x_\beta^{(1)}} + \frac{\partial^2 \rho}{\partial t^{(1)}\partial x_\beta^{(1)}}RT\delta_{\alpha\beta}$$

$$\{Introduce\ Eq.A.13$$

$$= -\frac{\partial^2\big(\rho u_\alpha u_\beta u_\gamma\big)}{\partial x_\beta^{(1)}\partial x_\gamma^{(1)}} - \frac{\partial^2\big(\rho u_\gamma\big)}{\partial x_\beta^{(1)}\partial x_\gamma^{(1)}}RT\delta_{\alpha\beta}$$

$$(k) \to \frac{\partial^2 P^0_{\alpha\beta\gamma}{}_{[Eq.A.20}}{\partial x_\beta^{(1)}\partial x_\gamma^{(1)}} = \frac{\partial^2(\rho u_\alpha u_\beta u_\gamma)}{\partial x_\beta^{(1)}\partial x_\gamma^{(1)}} + \frac{\partial^2 \rho\big(u_\alpha\delta_{\alpha\gamma} + u_\beta\delta_{\beta\gamma} + u_\gamma\delta_{\alpha\beta}\big)}{\partial x_\beta^{(1)}\partial x_\gamma^{(1)}}RT$$

The *(j)* and *(k)* terms in Eq.(A.19c) gives

$$\sum_i c_{i\alpha}G_i^1 = \frac{\partial(\rho u_\alpha)}{\partial t^{(2)}} + \left(\frac{\delta t}{2} - \frac{\delta t}{w}\right)RT\frac{\partial^2(\rho u_\alpha)}{\partial x_\beta^{(1)}\partial x_\beta^{(1)}}\,. \qquad (A.21)$$

By adding up Eq.(A.14) and Eq.(A.16) (the zeroth order moments), one finally obtains

$$\epsilon\frac{\partial\rho}{\partial t^{(1)}} + \epsilon\frac{\partial(\rho u_\gamma)}{\partial x_\gamma^{(1)}} + \epsilon^2\frac{\partial\rho}{\partial t^{(2)}} = 0$$

and descaling by Eq.(A.7) gives the continuity equation

$$\frac{\partial\rho}{\partial t} + \frac{\partial(\rho u_\alpha)}{\partial x_\alpha} = 0\,. \qquad (A.22)$$

Besides, adding up Eq.(A.14) and Eq.(A.21) (the first order moments), one can derive

$$\epsilon\frac{\partial(\rho u_\alpha)}{\partial t^{(1)}} + \epsilon\frac{\partial\big(\rho u_\alpha u_\beta\big)}{\partial x_\beta^{(1)}} + \epsilon\frac{\partial\rho}{\partial x_\beta^{(1)}}RT\delta_{\alpha\beta} - \epsilon\rho F_\alpha + \epsilon^2\frac{\partial(\rho u_\alpha)}{\partial t^{(2)}}$$

$$+\epsilon\left(\frac{\delta t}{2} - \frac{\delta t}{w}\right)RT\frac{\partial^2(\rho u_\alpha)}{\partial x_\beta^{(1)}\partial x_\beta^{(1)}} = 0\,.$$

Introducing the scales in Eq.(A.7) yields the Navier-Stokes equation

$$\frac{\partial(\rho u_\alpha)}{\partial t} + \frac{\partial(\rho u_\alpha u_\beta)}{\partial x_\beta} = -\frac{\partial\rho}{\partial x_\alpha}RT\delta_{\alpha\beta} + \left(\frac{\delta t}{w} - \frac{\delta t}{2}\right)RT\frac{\partial^2(\rho u_\alpha)}{\partial x_\beta^2} + \rho F_\alpha\,, \qquad (A.23)$$

where the viscosity is realized as

$$\nu = \delta t\left(\frac{1}{w} - \frac{1}{2}\right)RT = \delta t\left(\tau_\nu - \frac{1}{2}\right)c_s^2. \qquad (A.24)$$

We have derived the Navier-Stokes equation from the Boltzmann equation in the way, presented above. The energy equation is also derived from the Boltzmann equation and the derivation can be found in (Viggen, 2014), (Gilberto, 2010), (Carlo, 1988), (Wolf-Gladrow, 2000) and (Alexander, 2008).

A2 Derivation of the Advection-Diffusion equation

We have briefly discussed the lattice Boltzmann method to solve diffusion and advection-diffusion problem in Section 2.3.1 and 5.3.1. Now we shall see the derivation of the macroscopic advection-diffusion equation from the lattice Boltzmann equation using the Chapman-Enskog expansion.

As the procedure elaborated in A1, the distribution function for scalar variable can be expanded into its perturbation terms using a small parameter ϵ:

$$g_i(\mathbf{x},t) = g_i^{(0)}(\mathbf{x},t) + \epsilon g_i^{(1)}(\mathbf{x},t) + \epsilon^2 g_i^{(2)}(\mathbf{x},t) + \cdots \ . \tag{$A.25$}$$

Summation of this leads

$$\sum_{i=0}^{N} g_i(\mathbf{x},t) = \sum_{i=0}^{N} \left[g_i^{(0)}(\mathbf{x},t) + \epsilon g_i^{(1)}(\mathbf{x},t) + \epsilon^2 g_i^{(2)}(\mathbf{x},t) + \cdots \right] , \tag{$A.26$}$$

where the left side is Eq.(395), the first term in right hand $g_i^{(0)}$ is the equilibrium part of the distribution function, g_i^{eq}. Therefore, other terms should be zero, i.e.

$$\sum_{i=0}^{N} g_i^{(1)}(\mathbf{x},t) = 0 \text{ and} \tag{$A.27$}$$

$$\sum_{i=0}^{N} g_i^{(2)}(\mathbf{x},t) = 0 \, . \tag{$A.28$}$$

Now, the updated distribution function in Eq.(282), the first term in left hand, is expanded using Taylor series,

$$g_i(\mathbf{x} + \delta x, t + \delta t) = g_i(\mathbf{x},t) + \frac{\partial g_i(\mathbf{x},t)}{\partial x} \mathbf{c}_i \delta t + \frac{\partial g_i(\mathbf{x},t)}{\partial t} \delta t$$
$$+ \frac{1}{2} \delta t^2 \left[\frac{\partial^2 g_i(\mathbf{x},t)}{\partial x^2} \mathbf{c}_i^2 + 2 \frac{\partial^2 g_i(\mathbf{x},t)}{\partial x \partial t} \mathbf{c}_i + \frac{\partial^2 g_i(\mathbf{x},t)}{\partial t^2} \right] + O(\delta t)^3 \ . \tag{$A.29$}$$

Introducing scales, space $\frac{\partial}{\partial x} \to \epsilon \frac{\partial}{\partial x}$ and time $\frac{\partial}{\partial t} \to \epsilon \frac{\partial}{\partial t_1} + \epsilon^2 \frac{\partial}{\partial t_2}$, into Eq.(A.29) and substituting the result into the lattice Boltzmann equation in Eq.(282) yields:

$$\frac{\delta t}{\tau_s} \left(g_i^{eq} - g_i \right) = \epsilon \frac{\partial g_i}{\partial x} \mathbf{c}_i \delta t + \epsilon \frac{\partial g_i}{\partial t_1} \delta t + \epsilon^2 \frac{\partial g_i}{\partial t_2} \delta t$$
$$+ \frac{1}{2} \delta t^2 \epsilon^2 \left[\frac{\partial^2 g_i}{\partial x^2} \mathbf{c}_i^2 + 2 \frac{\partial^2 g_i}{\partial x \partial t_1} \mathbf{c}_i + \frac{\partial^2 g_i}{\partial t_1^2} \right] + O(\delta t)^3 + O(\epsilon)^3 \, , \tag{$A.30$}$$

where we simplified $g_i(\mathbf{x}, t)$ into g_i. Substituting expansion in Eq.(A.25) into the above equation gives

$$-\frac{1}{\tau_s}\left(\epsilon g_i^{(1)} + \epsilon^2 g_i^{(2)}\right) = \epsilon \frac{\partial g_i^{(0)}}{\partial x}\mathbf{c}_i + \epsilon^2 \frac{\partial g_i^{(1)}}{\partial x}\mathbf{c}_i + \epsilon \frac{\partial g_i^{(0)}}{\partial t_1} + \epsilon^2 \frac{\partial g_i^{(1)}}{\partial t_1} + \epsilon^2 \frac{\partial g_i^{(0)}}{\partial t_2}$$

$$+\frac{\delta t}{2}\epsilon^2\left[\frac{\partial^2 g_i^{(0)}}{\partial x^2}\mathbf{c}_i^2 + 2\frac{\partial^2 g_i^{(0)}}{\partial x \partial t_1}\mathbf{c}_i + \frac{\partial^2 g_i^{(0)}}{\partial t_1^2}\right] + O(\delta t)^3 + O(\epsilon)^3 . \qquad (A.31)$$

For further simplification of Eq.(A.31), we collect the terms of order of ϵ and ϵ^2 from both sides of the equation and emphasize the results.

The terms of order of ϵ are

$$-\frac{g_i^{(1)}}{\tau_s} = \frac{\partial g_i^{(0)}}{\partial x}\mathbf{c}_i + \frac{\partial g_i^{(0)}}{\partial t_1} \; . \qquad (A.32)$$

The distribution function $g_i^{(1)}$ can be differentiated with respect to t_1 and x as follows

$$-\frac{1}{\tau_s}\frac{\partial g_i^{(1)}}{\partial t_1} = \frac{\partial^2 g_i^{(0)}}{\partial t_1 \partial x}\mathbf{c}_i + \frac{\partial^2 g_i^{(0)}}{\partial t_1^2} \text{ and} \qquad (A.33)$$

$$-\frac{1}{\tau_s}\frac{\partial g_i^{(1)}}{\partial x} = \frac{\partial^2 g_i^{(0)}}{\partial x^2}\mathbf{c}_i + \frac{\partial^2 g_i^{(0)}}{\partial t_1 \partial x} . \qquad (A.34)$$

Multiply Eq.(A.34) by $\mathbf{c}_i$, add it to Eq.(A.33) and summate the equations,

$$-\frac{1}{\tau_s}\left(\frac{\partial g_i^{(1)}}{\partial t_1} + \frac{\partial g_i^{(1)}}{\partial x}\mathbf{c}_i\right) = \frac{\partial^2 g_i^{(0)}}{\partial t_1^2} + 2\frac{\partial^2 g_i^{(0)}}{\partial t_1 \partial x}\mathbf{c}_i + \frac{\partial^2 g_i^{(0)}}{\partial x^2}\mathbf{c}_i\mathbf{c}_i \, , \qquad (A.35)$$

which will be used later.

The summation over lattice direction for Eq.(A.32) gives

$$-\frac{1}{\tau_s}\sum_{i=0}^{N} g_i^{(1)} = \frac{\partial}{\partial x}\sum_{i}^{N} g_i^{(0)}\mathbf{c}_i + \frac{\partial}{\partial t_1}\sum_{i}^{N} g_i^{(0)} \, , \qquad (A.36)$$

where the first term is zero according to Eq.(A.27), the second term is

$$\frac{\partial}{\partial x}\sum_{i}^{N} g_i^{(0)}\mathbf{c}_i = \frac{\partial}{\partial x}(\phi\mathbf{u}) \, ,$$

and the third term is

$$\frac{\partial}{\partial t_1}\sum_{i}^{N} g_i^{(0)} = \frac{\partial \phi}{\partial t_1} .$$

Thus, Eq.(A.36) reformulated as,

$$0 = \frac{\partial(\phi\mathbf{u})}{\partial x} + \frac{\partial \phi}{\partial t_1} \, , \qquad (A.37)$$

which is the advection equation.

Return to Eq.(A.31), gather the terms of order of ϵ²:

$$-\frac{g_i^{(2)}}{\tau_s}=\frac{\partial g_i^{(1)}}{\partial x}\mathbf{c}_i+\frac{\partial g_i^{(1)}}{\partial t_1}+\frac{\partial g_i^{(0)}}{\partial t_2}+\frac{\delta t}{2}\epsilon^2\left[\frac{\partial^2 g_i^{(0)}}{\partial x^2}\mathbf{c}_i^2+2\frac{\partial^2 g_i^{(0)}}{\partial x\partial t_1}\mathbf{c}_i+\frac{\partial^2 g_i^{(0)}}{\partial t_1^2}\right], \qquad (A.38)$$

where the summation in bracket is the same as Eq.(A.35). Considering Eq.(A.35), after the simple arrangements, Eq.(A.38) becomes

$$-\frac{1}{\tau_s}g_i^{(2)}=\left(1-\frac{\delta t}{2\tau_s}\right)\left(\frac{\partial g_i^{(1)}}{\partial t_1}+\frac{\partial g_i^{(1)}}{\partial x}\mathbf{c}_i\right)+\frac{\partial g_i^{(0)}}{\partial t_2}. \qquad (A.39)$$

Summing Eq.(A.39) over lattice directions gives

$$-\frac{1}{\tau_s}\underbrace{\sum_{i=0}^{N} g_i^{(2)}}_{=0}=\left(1-\frac{\delta t}{2\tau_s}\right)\left(\frac{\partial}{\partial t_1}\underbrace{\sum_{i}^{N} g_i^{(1)}}_{=0}+\underbrace{\sum_{i}^{N}\frac{\partial g_i^{(1)}}{\partial x}\mathbf{c}_i}_{Eq.A.34}\right)+\frac{\partial}{\partial t_2}\underbrace{\sum_{i}^{N} g_i^{(0)}}_{=\phi}. \qquad (A.40)$$

Introducing Eq.(A.34) to the term in Eq.(A.40) yields

$$\sum_{i}^{N}\frac{\partial g_i^{(1)}}{\partial x}\mathbf{c}_i=\underbrace{-\tau_s\sum_{i}^{N}\frac{\partial^2 g_i^{(0)}}{\partial t_1\partial x}\mathbf{c}_i}_{(a)}\underbrace{-\tau_s\sum_{i}^{N}\frac{\partial^2 g_i^{(0)}}{\partial x^2}\mathbf{c}_i\,\mathbf{c}_i}_{(b)}. \qquad (A.41)$$

The term (a) become zero due to the cancelling of discrete velocities and the term (b) gives

$$-\tau_s\sum_{i}^{N}\frac{\partial^2 g_i^{(0)}}{\partial x^2}\mathbf{c}_i\,\mathbf{c}_i=-\tau_s RT\frac{\partial^2\phi}{\partial x^2}, \qquad (A.42)$$

where the second order moments of the distribution function for the advection-diffusion problem is used. So substituting findings into Eq.(A.40) yields

$$0=-\tau_s RT\left(1-\frac{\delta t}{2\tau_s}\right)\frac{\partial^2\phi}{\partial x^2}+\frac{\partial\phi}{\partial t_2}, \qquad (A.43)$$

or

$$0=RT\left(\frac{\delta t}{2}-\tau_s\right)\frac{\partial^2\phi}{\partial x^2}+\frac{\partial\phi}{\partial t_2}. \qquad (A.44)$$

Finally, adding Eq.(A.37) and Eq.(A.44) in Eq.(A.31) yields

$$0=\epsilon\frac{\partial\phi}{\partial t_1}+\epsilon^2\frac{\partial\phi}{\partial t_2}+\epsilon\frac{\partial(\phi\mathbf{u})}{\partial x}+\epsilon^2 RT\left(\frac{\delta t}{2}-\tau_s\right)\frac{\partial^2\phi}{\partial x^2}. \qquad (A.45)$$

Removing the space and time scales with the small parameter from Eq.(A.45) gives the advection-diffusion equation

$$\frac{\partial\phi}{\partial t}+\mathbf{u}\frac{\partial\phi}{\partial x}=RT\left(\tau_s-\frac{\delta t}{2}\right)\frac{\partial^2\phi}{\partial x^2} \qquad (A.46)$$

with the diffusion coefficient of

$$D=RT\left(\tau_s-\frac{\delta t}{2}\right). \qquad (A.47)$$

REFERENCES

A.A.Mohamad, 2011. *Lattice Boltzmann Method: fundamentals and Engineering applications with Computer Codes.* New York: Springer.

Alamyane, A. A. & Mohamad, A. A., 2010. Simulation of forced convection in a channel with extended surfaces by the lattice Boltzmann method. *Computers & Mathematics with Applications,* 59(7), pp. 2421-2430.

Albuquerque, P., Alemani, D., Chopard, B. & Leone, P., 2004. Coupling a lattice Boltzmann and a finite difference scheme. *Computational Science-ICCS 2004,* pp. pp.540-547.

Alexander, F. J., Chen, S. & Sterling, J. D., 1993. Lattice boltzmann thermohydrodynamics. *Physical Review E,* 47(4), p. R2249.

Alexander, W. J., 2008. A Practical Introduction to the Lattice Boltzmann Method. *Adt. notes for Statistical Mechanics,* 463(2008), p. 663.

Alexiades, V., 1992. *Mathematical modeling of melting and freezing processes.* s.l.:CRC Press.

Ammer, R. et al., 2014. Simulating fast electron beam melting with a parallel thermal free surface lattice Boltzmann method. *Computers & Mathematics with Applications,* 67(2), pp. 318-330.

Amsden, A. A. & Harlow, F. H., 1970. A simplified MAC technique for incompressible fluid flow calculations. *Journal of computational physics,* 6(2), pp. 322-325.

Ansumali, S., 2004. *Minimal kinetic modeling of Hydrodynamics (Ph.D thesis).* Zurich: Swiss Federal Institute of Technology in Zurich.

Ansumali, S., Chikatamarla, S. S., Frouzakis, C. E. & Boulouchos, K., 2004. Entropic lattice Boltzmann simulation of the flow past square cylinder. *International Journal of Modern Physics C,* 15(03), pp. 435-445.

Arid, A. et al., 2012. Numerical simulation of ice melting near the density inversion point under periodic thermal boundary conditions. *Fluid Dynamics & Materials Processing,* 8(3), pp. 257-275.

Arvanaghi, H. & Navid, N. O., 2013. Sharp-crested weir discharge coefficient. *Journal of Civil Engineering and Urbanism,* 3(3), pp. 87-91.

Attar, E. & Körner, C., 2011. Lattice Boltzmann model for thermal free surface flows with liquid–solid phase transition. *International Journal of Heat and Fluid Flow,* 32(1), pp. 156-163.

Ayurzana, B., 2016. ЛАТТИС БОЛЬЦМАНЫ АРГААР СҮВЭРХЭГ ОРЧИН ДОТУУРХ ЧӨЛӨӨТ ГАДАРГУУТАЙ ШИНГЭНИЙ УРСГАЛЫГ ЗАГВАРЧЛАХ. *Proceedings of the Mongolian Academy of Sciences,* 56(01), pp. 42-54.

Ayurzana, B. & Hosoyamada, T., 2016. *Phase change simulations of water near its density inversion point by Lattice Boltzmann method.* Ann Arbor, s.n.

Ayurzana, B. & Hosoyamada, T., 2017. Application of the lattice Boltzmann method to liquid-solid phase change in free surface flow: An example of Mongolian small hydropower plant. *Journal of Japan Society of Civil Engineers, Ser.B1 (Hydraulic Engineering),* 73(4), pp. I_607-I_612.

Ayurzana, B. & Hosoyamada, T., 2018. Lattice Boltzmann method for the numerical simulations of the melting and floating of ice. *Free surface flows and transport processes.*

Ayurzana, B., Hosoyamada, T. & Nasanbayar, N., 2016. *Hydraulics application of the Free-surface Lattice Boltzmann method.* Novosibirsk, Russia, s.n.

Ayurzana, B., Khenmedekh, L. & Hosoyamada, T., 2017. Parallel implementation of Entropic lattice Boltzmann method for flow past a circular cylinder at high Reynolds number. *Transactions on GIGAKU,* 4(01), pp. 04006/1-8.

Bagnold, R. A., 1941. *The physics of blown sand and desert dunes.* Mineola, New York: Courier Corporation.

Bailey, P. et al., 2009. Accelerating lattice Boltzmann fluid flow simulations using graphics processors. *Parallel Processing, 2009. ICPP'09. International Conference on,* pp. 550-557.

Baker Gregory, R., Daniel I, M. & Steven A, O., 1982. Generalized vortex methods for free-surface flow problems. *Journal of Fluid Mechanics,* Volume 123, pp. 477-501.

Bandringa, H., 2010. *Immersed boundary methods.* Groningen: Master's thesis, University of Groningen.

Bartoloni, A. et al., 1993. LBE simulations of Rayleigh-Bénard convection on the APE100 parallel processor. *International Journal of Modern Physics C,* 4(05), pp. 993-1006.

Beltaos, S., 1990. Fracture and breakup of river ice cover. *Canadian Journal of Civil Engineering,* 17(2), pp. 173-183.

Belytschko, T., 1980. Fluid-structure interaction. *Computers & Structures,* 14(4), pp. 459-469.

Belytschko, T., Yun, Y. L. & Lei, G., 1994. Element-free Galerkin methods. *International journal for numerical methods in engineering,* 37(2), pp. 229-256.

Benjamin, T. B., 1968. Gravity currents and related phenomena. *Journal of Fluid Mechanics,* 31(02), pp. 209-248.

Bhatnagar, P., Gross, E. & Krook, M., 1954. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical review,* 94(3), p. 511.

Bogner, S., 2009. *Simulation of floating objects in free surface flows (Master thesis).* Erlangen: University of Erlangen-Nuremberg.

Boldbaatar, B., 2013. *Assessment of future hydropower plant investment in Mongolia,* Seoul: (master thesis) Seoul National University.

Boussinesq, J., 1877. *Essai sur la théorie des eaux courantes.* s.l.:Imprimerie nationale.

Breuer, M., Bernsdorf, J., Zeiser, T. & Durst, F., 2000. Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume. *International Journal of Heat and Fluid Flow,* 21(2), pp. 186-196.

Brown, D. L., 1995. Performance of under-resolved two-dimensional incompressible flow simulations. *Journal of Computational Physics,* 12(1), pp. 165-183.

Bulgarelli, U., Casulli, V. & Greenspan, D., 1984. *Pressure methods for the numerical solution of free surface fluid flows.* Swansea: UK: Pineridge Press.

Carlo, C., 1988. *The Boltzmann equation and its Application.* Milano: Springer-Verlag.

Chatterjee, D. & Chakraborty, S., 2006. A hybrid lattice Boltzmann model for solid–liquid phase transition in presence of fluid flow. *Physics Letters A,* 351(4), pp. 359-367.

Chen, H. et al., 2003. Extended Boltzmann kinetic equation for turbulent flows. *Science,* 301(5633), pp. 633-636.

Chen, S. & D.Doolen, G., 1998. Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics,* Volume 30, pp. 329-364.

Chen, S., Hudong, C., Daniel, M. & William, M., 1991. Lattice Boltzmann model for simulation of magnetohydrodynamics. *Physical Review Letters,* 67(27), p. 3776.

Chikatamarla, S. S. & Karlin, I. V., 2013. Entropic lattice Boltzmann method for turbulent flow simulations: Boundary conditions. *Physica A: Statistical Mechanics and its Applications,* 392(9), pp. 1925-1930.

Chorin, A. J., 1967. The numerical solution of the Navier-Stokes equations for an incompressible fluid. *Bulletin of the American Mathematical Society,* 73(6), pp. 928-931.

Chorin, A. J., 1968. Numerical solution of the Navier-Stokes equations. *Mathematics of computation,* 22(104), pp. 745-762.

Chung, T. J., 2002. *Computational fluid dynamics.* Cambridge: Cambridge University Press.

Crank, J. & Phyllis, N., 1947. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Mathematical Proceedings of the Cambridge Philosophical Society,* 43(01), pp. 50-67.

Danaila, I., Moglan, R., Hecht, F. & Masson, S. L., 2014. A Newton method with adaptive finite elements for solving phase-change problems with natural convection. *Journal of Computational Physics,* Volume 274, pp. 826-840.

Dardis, O. & McCloskey, J., 1998. Lattice Boltzmann scheme with real numbered solid density for the simulation of flow in porous media. *Physical Review E,* 57(4), p. 4834.

Das, A. K. & Das, P. K., 2009. Bubble evolution through submerged orifice using smoothed particle hydrodynamics: basic formulation and model validation. *Chemical Engineering Science,* 64(10), pp. 2281-2290.

Deardorff, J. W., 1970. A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics,* 41(02), pp. 453-480.

Deardorff, J. W., 1970. A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics,* 41(02), pp. 453-480.

Debroux, F., Mahesh, P. & Paul, C., 2001. *Three-dimensional modelling of a tsunami interacting with real topographical coastline using smoothed particle hydrodynamics.* Adelaide, Australia, s.n., pp. 311-314.

d'Humières, D., 2002. Multiple–relaxation–time lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences,* 360(1792), pp. 437-451.

Drew, D. A., 1983. Mathematical modeling of two-phase flow. *Annual review of fluid mechanics,* 15(1), pp. 261-291.

Dupuis, A., 2002. *From a lattice Boltzmann model to a parallel and reusable implementation of a virtual river.* GENEVE: Diss. Thèse soutenue à Genève en Suisse.

Einstein, H. A. & Krone, R. B., 1961. Estuarial sediment transport patterns. *Journal of the Hydraulics Division,* 87(2), pp. 51-59.

Einstein, H. & Krone, R., 1962. Experiments to determine modes of cohesive sediment transport in salt water. *Journal of Geophysical Research,* 67(4), pp. 1451-1461.

Elton, B. H., C, D. L. & Garry H, R., 1995. Convergence of convective–diffusive lattice Boltzmann methods. *SIAM Journal on Numerical Analysis,* 32(5), pp. 1327-1354.

Esfahani, B. R., Calzavarini, E., Hirata, S. & Ouarzazi, M.-N., 2016. *Melting from below driven by thermal convection.* Seville, Spain, s.n.

Ettema, R., 1980. *Scour at bridge piers,* New Zealand: No. 216 Monograph.

Ettema, R., 2002. Review of alluvial-channel responses to river ice. *Journal of Cold Regions Engineering,* 16(4), pp. 191-217.

Ettema, R., 2008. Ice effects on sediment transport in rivers. In: *Sedimentation Engineering: Processes, Measurements, Modeling, and Practice.* Reston, Virginia: American Society of Civil Engineers, pp. 613-648.

Ettema, R., Braileanu, F. & Muste, M., 2000. Method for estimating sediment transport in ice-covered channels. *Journal of Cold Regions Engineering,* 14(3), pp. 130-144.

Ettema, R. & Daly, S. F., 2004. *Sediment transport under ice,* Hanover, New Hampshire: No. ERDC/CRREL TR-04-20. ENGINEER RESEARCH AND DEVELOPMENT CENTER HANOVER NH COLD REGIONS RESEARCH AND ENGINEERING LAB.

Ettema, R., Kennedy J, F. & F, K., 1982. *Influence of turbulence and temperature on frazil ice formation.* Edmonton, Alberta, s.n.

Ettema, R. & Zabilansky, L., 2001. *Observations of Ice-Cover and Openwater-Lead Formation along the Fort Peck Reach of the Missouri River.* Ottawa, Ontario, s.n.

Faizal, F. & Septiawan, R. R., 2014. Computational Study on Melting Process Using Smoothed Particle Hydrodynamics. *Journal of Modern Physics,* 5(3), p. 43127.

Feng, Y., Han, T. K. & Owen, D. R. J., 2007. Coupled lattice Boltzmann method and discrete element modelling of particle transport in turbulent fluid flows: Computational issues. *International Journal for Numerical Methods in Engineering,* 72(9), pp. 1111-1134.

Feng, Z.-G. & Michaelides, E. E., 2004. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *Journal of Computational Physics,* 195(2), pp. 602-628.

Fernandino, M., K, B. & T, Y., 2009. Large eddy simulation of turbulent open duct flow using a lattice Boltzmann approach. *Mathematics and Computers in Simulation,* 79(5), pp. 1520-1526.

Flemming, B. W. & Thum, A. B., 1978. The settling tube—a hydraulic method for grain size analysis of sands. *Kieler Meeresforschungen Sonderheft,* Volume 4, pp. 82-95.

forum, M. w., 2013. *Ecology and social-economy study on the Ider river basin,* Ulaanbaatar: s.n.

Frisch, U., Brosl, H. & Yves, P., 1986. Lattice-gas automata for the Navier-Stokes equation. *Physical review letters,* 56(14), p. 1505.

Frisch, U. et al., 1987. Lattice gas hydrodynamics in two and three dimensions. *Complex systems,* 1(4), pp. 647-707.

Fukumoto, Y., 2015. *Particle Based Multiphysics Simulation for Applications to Design of Soil Structures and Micromechanics of Granular Geomaterials,* Kyoto: Kyoto University.

G. K. Batchelor, F., 1967. *An Introduction to Fluid Dynamics.* Cambridge: Cambridge University press.

Gac, J., 2014. A large eddy based lattice-Boltzmann simulation of velocity distribution in an open channel flow with rigid and flexible vegetation. *Acta Geophysica,* 62(1), pp. 180-198.

Gebre, S. et al., 2013. Review of ice effects on hydropower systems. *Journal of Cold Regions Engineering,* 27(4), pp. 196-222.

Gibbs, R. J., 1974. A settling tube for sand-size analysis. *Journal of Sedimentary Research,* 44(2), pp. 583-588.

Gilberto, K. M., 2010. *An introduction to the Boltzmann equation and transport process in gases.* Curitiba: Springer Science & Business Media.

Gingold, R. A. & Monaghan, J. J., 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society,* 181(3), pp. 375-389.

Ginzburg, I., 2005. "Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation. *Advances in Water resources,* 28(11), pp. 1171-1195.

Ginzburg, I. & Konrad, S., 2003. Lattice Boltzmann model for free-surface flow and its application to filling process in casting. *Journal of Computational Physics,* 185(1), pp. 62-99.

Ginzburg, I., Verhaeghe, F. & d'Humieres, D., 2008. Two-relaxation-time lattice Boltzmann scheme: About parametrization, velocity, pressure and mixed boundary conditions. *Communications in computational physics,* 3(2), pp. 427-478.

Gu, C., Shyam S, C. & Iliya V, K., 2014. Simulation of Flow Past a Circular Cylinder Using Entropic Lattice Boltzmann Method. *International Journal of Modern Physics C,* 25(01), p. 1340024.

Gunstensen, A. K., Daniel H, R., Stéphane, Z. & Gianluigi, Z., 1991. Lattice Boltzmann model of immiscible fluids. *Physical Review A,* 43(8), p. 4320.

Guo, Z. & Shu, C., 2013. *Lattice Boltzmann Method and its Application in engineering.* Singapore: World scientific Publishing.

Guo, Z., Xu, K. & Wang, R., 2013. Discrete unified gas kinetic scheme for all Knudsen number flows: Low-speed isothermal case. *Physical Review E,* 88(3), p. 033305.

Hardy, J., Y, P. & O, D. P., 1973. Time evolution of a two-dimensional classical lattice system. *Physical Review Letters,* 31(5), p. 276.

Harlow, F. H. & Welch, J. E., 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids,* 8(12), pp. 2182-2189.

Henderson, F. M., 1996. *Open channel flow.* New York: Macmillan.

He, X., Chen, S. & Raoyang, Z., 1999. A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability. *Journal of Computational Physics,* 152(2), pp. 642-663.

He, X., Xiaowen, S. & Gary, D. D., 1998. Discrete Boltzmann equation model for nonideal gases. *Physical Review E,* 57(1), p. R13.

Hideo, S. et al., 2013. Implications of Flood Event Layers in Littoral Sedimentary Environments. *Journal of Japan Society of Civil Engineers, Ser. B2 (Coastal Engineering),* 69(2), pp. I_691-I_695.

Hirt, C. W. & Cook, J. L., 1972. Calculating three-dimensional flows around structures and over rough terrain. *Journal of Computational Physics,* 10(2), pp. 324-340.

Hirt, C. W. & Nichols, B. D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics,* 39(01), pp. 201-225.

Hirt, C. W., Nichols, B. D. & Romero, N. C., 1975. *SOLA: A numerical solution algorithm for transient fluid flows,* Los Alamos: NASA STI/Recon Technical Report N 75, 32418.

Hlushkou, D., Kandhai, D. & Tallarek, U., 2004. Coupled lattice-Boltzmann and finite-difference simulation of electroosmosis in microfluidic channels. *International journal for numerical methods in fluids,* 46(5), pp. pp.507-532.

Hou, S., Sterling, J., Chen, S. & Doolen, G., 1996. A lattice Boltzmann subgrid model for high Reynolds number flows. *Pattern Formation and Lattice Gas Automata,* Volume 6, p. 149.

Huang, H., Lu, X. & Sukop, M., 2011. Numerical study of lattice Boltzmann methods for a convection–diffusion equation coupled with Navier–Stokes equations. *Journal of Physics A: Mathematical and Theoretical,* 44(5), p. 055001.

Huang, H., Michael, S. & Xiyun, L., 2015. *Multiphase lattice Boltzmann methods: Theory and application.* s.l.:John Wiley & Sons.

Huang, R., Wu, H. & Cheng, P., 2013. A new lattice Boltzmann model for solid–liquid phase change. *International Journal of Heat and Mass Transfer,* Volume 59, pp. 295-301.

Huber, C. et al., 2008. Lattice Boltzmann model for melting with natural convection. *International Journal of Heat and Fluid Flow,* 29(5), pp. 1469-1480.

Hu, H. & Argyropoulos, S. A., 1996. Mathematical modelling of solidification and melting: a review. *Modelling and Simulation in Materials Science and Engineering,* 4(4), p. 371.

Huppert, H. E. & Simpson, J. E., 1980. The slumping of gravity currents. *Journal of Fluid Mechanics,* 99(04), pp. 785-799.

Ikehata, Y. & Honji, H., 2000. LES studies of a negatively buoyant jet discharged into a stratified fluid. *Journal of Japan Society of Fluid Mechanics,* Volume 19, pp. 332-341.

Infrastructure, T. M. o., 2002. *Mongolian sustainable energy sector development strategy plan,* Ulaanbaatar: The Ministry of Infrastructure.

Iwasaki, K., Uchida, H., Dobashi, Y. & Nishita, T., 2010. Fast Particle-based Visual Simulation of Ice Melting. *Computer Graphics Forum,* 29(7), pp. 2215-2223.

Jansen, C. & Krafczyk, M., 2011. Free surface flow simulations on GPGPUs using the LBM. *Computers and Mathematics with Applications,* 61(12), pp. 3549-3563.

Janssen, C. & Krafczyk, M., 2010. A lattice Boltzmann approach for free-surface-flow simulations on non-uniform block-structured grids. *Computers & Mathematics with Applications,* 59(7), pp. 2215-2235.

Jiaung, Wen-Shu, Ho, J.-R. & Kuo, C.-P., 2001. Lattice Boltzmann method for the heat conduction problem with phase change. *Numerical Heat Transfer: Part B: Fundamentals,* 39(2), pp. 167-187.

Junk, M., 2001. A finite difference interpretation of the lattice Boltzmann method. *Numerical Methods for Partial Differential Equations,* 14(4), pp. pp.383-402.

Karabelas, S. J., 2010. Large eddy simulation of high-Reynolds number flow past a rotating cylinder. *International journal of heat and fluid flow,* 31(4), pp. 518-527.

Karlin, I. V., Bösch, F. & Chikatamarla, S. S., 2014. Gibbs' principle for the lattice-kinetic theory of fluid dynamics. *Physical Review E,* 90(3), p. 031302.

Karpiński, M., Bialik, R. & Rowiński, P., 2013. Application of Lattice Boltzmann Method for generation of flow velocity field over river bed-forms. In: P. Rowinski, ed. *GeoPlanet: Earth and Planetary Sciences.* Warshaw: Springer Berlin Heidelberg, pp. pp. 327-335.

Kerson, H., 1987. *Statistical mechanics.* Massachusetts: John Willey and Sons.

Kirkwood, J. G., 1947. The statistical mechanical theory of transport processes II. Transport in gases. *The Journal of Chemical Physics,* 15(1), pp. 72-76.

Koelman, J. M. V. A., 1991. A simple lattice Boltzmann scheme for Navier-Stokes fluid flow. *EPL (Europhysics Letters),* 15(6), p. 603.

Körner, C. & Robert F, S., 2000. Processing of metal foams—challenges and opportunities. *Advanced Engineering Materials,* 2(4), pp. 159-165.

Körner, C. et al., 2005. Lattice Boltzmann model for free surface flow for modeling foaming. *Journal of Statistical Physics,* 121(1-2), pp. 179-196.

Koshizuka, S., Nobe, A. & Oka, Y., 1998. Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids,* 26(7), pp. 751-769.

Koshizuka, S. & Oka, Y., 1996. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear science and engineering,* 123(3), pp. 421-434.

Kruyt, H. R., Jonker, G. H. & Overbeek, J. T. G., 1952. *Colloid Science. Vol. 1. Irreversible Systems,* Amsterdam: Elsevier.

Kundu, P. & Cohen, I., 2004. *Fluid Mechanics Third Edition.* San Diego: Elsevier Academic Press.

Ladd, A. J., 1994. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *Journal of fluid mechanics,* 271(1), pp. 285-309.

Lallemand, P., Luo, L.-S. & Peng, Y., 2007. A lattice Boltzmann front-tracking method for interface dynamics with surface tension in two dimensions. *Journal of Computational Physics,* 226(2), pp. 1367-1384.

Latt, J., 2007. *How to implement your DdQq dynamics with only q variables per node (instead of 2q),* Medford: Tufts University.

Lau, Lam, Y. & Krishnappan, B. G., 1985. Sediment transport under ice cover. *Journal of Hydraulic Engineering,* 111(6), pp. 934-950.

Laurendeau, N. M., 2005. *Statistical thermodynamics: fundamentals and applications.* Cambridge: Cambridge University Press.

Lin, P. & Liu, P. L., 1999. Free surface tracking methods and their applications to wave hydrodynamics. *Advances in Coastal and Ocean Engineering,* Volume 5, pp. 213-240.

Li, Q. et al., 2016. Lattice Boltzmann methods for multiphase flow and phase-change heat transfer. *Progress in Energy and Combustion Science,* Volume 52, pp. 62-105.

Li, S. & Liu, W. K., 2002. Meshfree and particle methods and their applications. *Applied Mechanics Reviews,* 55(1), pp. 1-34.

Liu, T. et al., 2008. *Extended lattice Boltzmann equation for simulation of flows around bluff bodies in high Reynolds number.* Milano, Italy, s.n., pp. 1-16.

Luo, L.-S.et al., 2011. Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations. *Physical Review E,* 83(5), p. 056710.

Masselot, A. a. C. B., 1998. A lattice Boltzmann model for particle transport and deposition. *EPL (Europhysics Letters),* 42(3), p. 259.

McNamara, G. R., Garcia, A. L. & Alder, B. J., 1995. Stabilization of thermal lattice Boltzmann models. *Journal of Statistical Physics,* 81(1), pp. 395-408.

McNamara, G. R. & Gianluigi, Z., 1988. Use of the Boltzmann equation to simulate lattice-gas automata. *Physical Review Letters,* 61(20), p. 2332.

Michael, S. C. & Daneal T, T. J., 2006. *Lattice Boltzmann modeling: An introduction for geoscientist and engineers.* Heidelberg: Springer.

Mittal, R. & Balachandar, S., 1995. Effect of three-dimensionality on the lift and drag of nominally two-dimensional cylinders. *Physics of Fluids,* 7(8), pp. 1841-1865.

Mittal, R. & Iaccarino, G., 2005. Immersed boundary methods. *Annu. Rev. Fluid Mech.,* Volume 37, pp. 239-261.

Mohamad, A. A. & Kuzmin, A., 2010. A critical evaluation of force term in lattice Boltzmann method, natural convection problem. *International Journal of Heat and Mass Transfer,* 53(5), pp. 990-996.

Monaghan, J. J., 1994. Simulating free surface flows with SPH. *Journal of computational physics,* 110(2), pp. 399-406.

Monaghan, J. J. & A, K., 1995. SPH simulation of multi-phase flow. *Computer Physics Communications,* 87(1-2), pp. 225-235.

Mooney, M., 1951. The viscosity of a concentrated suspension of spherical particles. *Journal of colloid science,* 6(2), pp. pp.162-170.

Morris, G. & Fan, J., 1998. *Reservoir sedimentation handbook: design and management of dams, reservoirs, and watersheds for sustainable use.* s.l.:McGraw Hill Professional.

Nichols, B. D., Hirt, C. W. & Hotchkiss, R. S., 1980. *SOLA-VOF: A solution algorithm for transient fluid flow with multiple free boundaries.* NM (USA): Los Alamos Scientific Lab.

Nils, T., 2007. *Physically based Animation of Free Surface (Ph.D thesis).* Nurnberg: Universtitat Erlangen-Nurnberg.

Noble, D. R. & Torczynski, J. R., 1998. A lattice-Boltzmann method for partially saturated computational cells. *International Journal of Modern Physics C,* 9(08), pp. 1189-1201.

Ochoa, J. S. & Fueyo, N., 2004. Large Eddy Simulation of the flow past a square cylinder: PHOENICS. *Journal of Computational Fluid Dynamics and its Applications,* Volume 17, pp. 1-18.

Onate, E., Idelsohn, S., Zienkiewicz, O. C. & Taylor, R. L., 1996. A finite point method in computational mechanics. Applications to convective transport and fluid flow. *International journal for numerical methods in engineering,* 39(22), pp. 3839-3866.

Ong, L. & Wallace, J., 1996. The velocity field of the turbulent very near wake of a circular cylinder. *Experiments in fluids,* 20(6), pp. 441-453.

Orszag, S. A., 1970. Analytical theories of turbulence. *Journal of Fluid Mechanics,* 41(02), pp. 363-386.

Papanicolaou, A. T. N. et al., 2008. Sediment transport modeling review—current and future developments. *Journal of Hydraulic Engineering,* 134(1), pp. 1-14.

Patankar, S. V. & Spalding, D. B., 1972. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International journal of heat and mass transfer,* 15(10), pp. 1787-1806.

Peskin, C. S., 1977. Numerical analysis of blood flow in the heart. *Journal of computational physics,* 25(3), pp. 220-252.

Pope, S. B., 2001. *Turbulent flows.* New York: Cambridge University press.

Prowse, T., 1993. Suspended sediment concentration during river ice breakup. *Canadian Journal of Civil Engineering,* 20(5), pp. pp.872-875.

Prowse, T. D. & Beltaos, S., 2002. Climatic control of river-ice hydrology: a review. *Hydrological processes ,* 16(4), pp. 805-822.

Qian, H. Y., D, d. & P, L., 1992. Lattice BGK models for Navier-Stokes equation. *EPL (Europhysics Letters),* 17(6), p. 479.

Qian, Y. H., Dominique, d. & Pierre, L., 1992. Lattice BGK models for Navier-Stokes equation. *Europhysics Letters,* 17(6), p. 479.

Rahman, M. M., Karim, M. M. & Alim, M. A., 2007. Numerical investigation of unsteady flow past a circular cylinder using 2-D finite volume method. *Journal of Naval Architecture and Marine Engineering,* 4(1), pp. 27-42.

Rajani, B. N., Kandasamy, A. & Majumdar, S., 2016. LES of Flow past Circular Cylinder at Re= 3900. *Journal of Applied Fluid Mechanics,* 9(3), pp. 1421-1435.

Rajaratnam, N., 1990. Skimming flow in stepped spillways. *Journal of Hydraulic Engineering,* 116(4), pp. 587-591.

Rapaport, D. C., 2003. *The Art of Molecular Dynamics Simulation.* Cambridge: Cambridge University press.

Raudkivi, A. J., 1976. *Loose boundary hydraulics (2nd edition).* Auckland: Pergamon press.

Rüde, U. & Thürey, N., 2004. *Free surface lattice-Boltzmann fluid simulations with and without level sets.* s.l., s.n., p. 199.

Ruetsch, G. & Fatica, M., 2011. *CUDA Fortran for scientists and engineers: Best Practices for Efficient CUDA Fortran programming.* Santa Clara: NVIDIA Corporation 2701.

Sane, M., H. Yamagishi, M., Tateishi & Izumia, T., 2005. Environmental impacts of the regulation of the Shinano River, Niigata prefecture, Japan. *WIT Transactions on Ecology and the Environment,* 81(4), pp. 419-428.

Scardovelli, R. & Stéphane, Z., 1999. Direct numerical simulation of free-surface and interfacial flow. *Annual review of fluid mechanics,* 31(1), pp. 567-603.

Schmitt, F. G., 2007. About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity. *Comptes Rendus Mécanique,* 335(9-10), pp. 617-627.

Schohl, G. A. & Ettema, R., 1986. Theory and laboratory observations of naled ice growth. *Journal of Glaciology,* 32(111), pp. 168-177.

Semma, E., Ganaoui, M. E., Bennacer, R. & Mohamad, A. A., 2008. Investigation of flows in solidification by using the lattice Boltzmann method. *International Journal of Thermal Sciences,* 47(3), pp. 201-208.

Shan, X., 1997. Simulation of Rayleigh-Bénard convection using a lattice Boltzmann method. *Physical Review E,* 55(3), p. 2780.

Shan, X. & Chen, H., 1993. Lattice Boltzmann model for simulating flows with multiple phases and components. *Physical Review E,* 47(3), p. 1815.

Shan, X. & Hudong, C., 1993. Lattice Boltzmann model for simulating flows with multiple phases and components. *Physical Review E,* 47(3), p. 1815.

Shin, J. O., Dalziel, S. B. & Linden, P. F., 2004. Gravity currents produced by lock exchange. *Journal of Fluid Mechanics,* Volume 521, pp. 1-34.

Sielecki, A. & Wurtele, M. G., 1970. The numerical integration of the nonlinear shallow-water equations with sloping boundaries. *Journal of Computational Physics,* 6(2), pp. 219-236.

Smagorinsky, J., 1963. General circulation experiments with the primitive equations: I. The basic experiment. *Monthly weather review,* 91(3), pp. 99-164.

Spreafico, M. & Bruk, S., 2004. Unesco's international sedimentation initiative Background and current status. *Journal of Soils and Sediments,* 4(4), pp. 217-218.

Sterling, J. D. & Shiyi, C., 1996. Stability analysis of lattice Boltzmann methods. *Journal of Computational Physics,* 123(1), pp. 196-206.

Stewart, H., 2004. *An Introduction to the Theory of the Boltzmann Equation.* New York: Dover Publications.

Strack, O. E. & Cook, B. K., 2007. Three-dimensional immersed boundary conditions for moving solids in the lattice-Boltzmann method. *International Journal for Numerical Methods in Fluids,* 55(2), pp. 103-125.

Sussman, M., Peter, S. & Stanley, O., 1994. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics,* 114(1), pp. 146-159.

Swift, M. R., Orlandini, E., Osborn, W. R. & Yeomans, J. M., 1996. Lattice Boltzmann simulations of liquid-gas and binary fluid systems. *Physical Review E,* 54(4), p. 5041.

Takeshi, O., 2016. *大河津分水路河口における密度差の影響を受ける土砂移動機構の把握,* Nagaoka: (Master thesis) Nagaoka University of Technology.

Takeshi, O., Ayurzana, B. & Hosoyamada, T., 2016a. NUMERICAL STUDY OF SETTLING AND TRANSPORT OF SEDIMENT DISCHARGED FROM LARGE RIVER MOUTH: AN APPLICATION TO THE OKOZU RIVER MOUTH. *Journal of Japan Society of Civil Engineers, Ser. B1 (Hydraulic Engineering),* 72(4), pp. I_355-I_360.

Takeshi, O., Ayurzana, B. & Hosoyamada, T., 2016b. A NUMERICAL STUDY OF SEDIMENT DISTRIBUTION DISCHARGED FROM A RIVER CONSIDERING ENHANCED SETTLING DUE TO FLOCCULATION CAUSED BY MIXING PROCESS OF RIVER AND SEA WATER. *Journal of Japan Society of Civil Engineers, Ser. B2 (Coastal Engineering),* 72(2), pp. I_1309-I_1314.

Tan, D. S., 2014. *Modelling Melting Ice using the Smooth Dissipative Particle Dynamics Method.* Melbourne, Australia, s.n.

Tennekes, H. & Lumley, J. L., 1972. *A first course in turbulence.* Cambridge: MIT press.

Thorne, D. T. & Sukop, M. C., 2004. Lattice Boltzmann model for the Elder problem. *Developments in Water Science,* Volume 55, pp. 1549-1557.

Thürey, N., Körner, C. & Rüde, U., 2005. *Interactive free surface fluids with the lattice Boltzmann method,* Germany: Technical Report05-4, University of Erlangen-Nuremberg.

Thürey, N. & Rüd, U., 2009. Stable free surface flows with the lattice Boltzmann method on adaptively coarsened grids. *Computing and Visualization in Science,* 12(5), pp. 247-263.

Thürey, N. et al., 2006. Optimization and stabilization of LBM free surface flow simulations using adaptive parameterization. *Computers & fluids,* 35(8), pp. 934-939.

Tokuzo, H. et al., 2001. NUMERICAL SIMULATION OF THE DENSITY DRIVEN CURRENTS AND THE SUPENDED SEDIMENT AT A RIVER ESTUARY. *PROCEEDINGS OF HYDRAULIC ENGINEERING,* Volume 45, pp. 955-960.

Tong, W. & Koster, J. N., 1993. Natural convection of water in a rectangular cavity including density inversion. *International journal of heat and fluid flow,* 14(4), pp. 366-375.

Tritton, D. J., 1988. *Physical fluid dynamics-2nd ed..* Oxford: Oxford university press.

Turcotte, B. & Morse, B., 2013. A global river ice classification model. *Journal of Hydrology,* Volume 507, pp. 134-148.

Tuthill, A., 1998. Structural ice control: A review. *Journal of cold regions engineering,* 12(2), pp. pp.33-51.

Van der Sman, R. & Ernst, M., 2000. Convection-diffusion lattice Boltzmann scheme for irregular lattices. *Journal of Computational Physics,* 160(2), pp. 766-782.

Viggen, E. M., 2014. *The lattice Boltzmann method: Fundamentals and acoustics (Ph.D thesis).* Trondheim: NTNU.

Virag, Z., Živić, M. & Galović, A., 2006. Influence of natural convection on the melting of ice block surrounded by water on all sides. *International journal of heat and mass transfer,* 49(21), pp. 4106-4115.

Voller, V. R. & Prakash, C., 1987. A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems. *International Journal of Heat and Mass Transfer,* 30(8), pp. 1709-1719.

Vreugdenhil, C. B., 2012. *Computational hydraulics: an introduction.* Petten: Springer Science & Business Media.

Walsh, S. D., Burwinkle, H. & Saar, M. O., 2009. A new partial-bounceback lattice-Boltzmann method for fluid flow through heterogeneous media. *Computers & Geosciences,* 35(6), pp. 1186-1193.

Walter, H. G., 1984. *Hydraulics of Sediment Transport.* Chelsea: Water Resources Publication.

Wang, X. et al., 2014. Direct numerical simulation and large eddy simulation on a turbulent wall-bounded flow using lattice Boltzmann method and multiple GPUs. *Mathematical Problems in Engineering,* Volume 1, p. 742432.

Welander, P., 1954. On the temperature jump in a rarefied gas. *Arkiv for Fysik,* Volume 7, pp. 507-553.

Wilson, T. L., Nichols, B. D., Hirt, C. W. & Stein, L. R., 1988. *SOLA-DM: A numerical solution algorithm for transient three-dimensional flows,* Alamos: No. LA-11161-MS. Los Alamos National Lab.

Winterwerp, J. C., 1998. A simple model for turbulence induced flocculation of cohesive sediment. *Journal of Hydraulic Research,* 36(3), pp. 309-326.

Wolf-Gladrow, D., 1995. A lattice Boltzmann equation for diffusion. *Journal of statistical physics,* 79(5), pp. 1023-1032.

Wolf-Gladrow, D. A., 2000. *Lattice-gas cellular automata and lattice Boltzmann models: An introduction.* Bremerhaven: Springer Science & Business Media.

Wolfram, S., 1986. Cellular automaton fluids 1: Basic theory. *Journal of statistical physics,* 45(3-4), pp. 471-526.

Yanagi, T., 1989. *Coastal oceanography. Vol. 1.* Tokyo: Boom Koninklijke Uitgevers.

Zhang, T. et al., 2012. General bounce-back scheme for concentration boundary condition in the lattice-Boltzmann method. *Physical Review E,* 85(1), p. 016701.

Zhou, J. G., 2009. A lattice Boltzmann method for solute transport. *International journal for numerical methods in fluids,* 61(8), pp. 848-863.

Баасандорж, Ж. et al., 1999. *Инженерийн математик 1.* Улаанбаатар: Техникийн их сургууль.

Болд, П., Жамбалдорж, Б., Даваажамц, Ш. & Пүрэвдорж, Ч., 2013. *Гидравлик: Аэродинамик.* Улаанбаатар: Артсофт.

伊勢屋ふじこ, 1985. *沈降式粒度分析の手引き,* 筑波: 筑波大学水理実験センター報告 9 .

細山田得三, et al., 2001. NUMERICAL SIMULATION OF THE DENSITY DRIVEN CURRENTS AND THE SUPENDED SEDIMENT AT A RIVER ESTUARY. *PROCEEDINGS OF HYDRAULIC ENGINEERING,* Volume 45, pp. pp.955-960.

鶴谷広一, 村上和男 & 入江功, 1989. 多層レベルモデルによる航路埋没の予測計算. *海岸工学論文集,* Volume 36, pp. 379-383.