Solid-Particles Deposition Through a Turbulent Impinging Jet Using Lattice Boltzmann Method

by

Ali Adulkadhim

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2018

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Dr. Edgar Akio Matida
Supervisor: Professor Fue-Sang Lien
Internal Member: Dr. Jean Pierre Hickey
Internal Member: Professor Zhongchao Tan
Internal-external Member: Professor Francis Poulin

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

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

Abstract

Solid particle distribution on an impingement surface has been simulated utilizing a graphical processing unit (GPU). An in-house computational fluid dynamics (CFD) code has been developed to investigate a 3D turbulent impinging jet using the lattice Boltzmann method (LBM) in conjunction with large eddy simulation (LES) and the multiple relaxation time (MRT) models. This work proposed an improvement in the LBM-cellular automata (LBM-CA) probabilistic method. In the current model, the fluid flow utilizes the D3Q19 LBM lattice model, while the particles movement employs the D3Q27 one.

The particle numbers are defined at the same regular LBM (fluid) nodes, and the transport of particles from one node to its neighbouring nodes are determined in accordance with the particle bulk density and velocity by considering all the external forces. The previous CA models distribute particles at each time step without considering the local particles number and velocity at each node. The present model overcomes the deficiencies of the previous LBM-CA models and, therefore, can better capture the dynamic interaction between particles and the surrounding turbulent flow field.

Despite increasing popularity of the LBM-MRT model in simulating complex multiphase fluid flows, this approach is still expensive in term of memory size and computational time required to perform 3D simulations. To improve the throughput of simulations, a single GeForce GTX TITAN X GPU is used in the present work. The CUDA parallel programming platform and the CuRAND library are utilized to form an efficient LBM-MRT-CA algorithm. The LBM-MRT fluid (i.e. no particles) model results were compared with two benchmark test cases ones. The first case is a turbulent free square jet, and the second one is a circular turbulent impinging jet for L/D=2 at Reynolds number equals to 25,000, where L is the nozzle-to-surface distance and D is the jet diameter.

The LBM-CA simulation methodology was first validated against a benchmark test case involving particle deposition on a square cylinder confined in a duct. The flow was unsteady and laminar at Re=200 (Re is the Reynolds number), and simulations were conducted for different Stokes numbers. The GPU code was then used to simulate the particle transport and deposition in a turbulent impinging jet at Re=10,000. The effect of changing Stokes number on the particle deposition profile was studied at different L/D ratios, i.e. L/D=2, 4, and 6. The current model was finally used to simulate the particle impaction pattern from a circular jet for L/D=0.5, where the effect of changing Stokes and Reynolds numbers on the particle transport and deposition was examined. The present LBM-CA solutions agree well with other results available in the open literature.

For comparative studies, another in-house serial CPU code was also developed, coupling LBM with the classical Lagrangian particle dispersion model. Agreement between results

obtained with LBM-CA and LBM-Lagrangian models and the experimental data for the impinging jet case of L/D=0.5 is generally good, and the present LBM-CA approach on GPU achieves a speedup ratio of about 150 against the serial code running on a single CPU.

Another new model was proposed to incorporate the solid particle phase effect (i.e. two-way coupling) on the fluid flow. The LMB-Lagrangian approach was used in this model to track solid particles in the computational domain. The solid particle phase was considered as a porous medium moving in the computational domain. The impact of the porous medium (i.e. the solid particle phase) on the fluid flow characteristics (e.g. fluid velocity) is a function of the particle phase volume fraction and velocity in the LBM. Particle-particle collision (i.e. four-way coupling) was also considered in this model by utilizing the discrete element method (DEM). This approach can numerically capture the multi-particle collision behaviours in dense particle suspension problems. This model data were compared with the numerical study ones for a single bubble injected in a fluidized bed, and the results of the bubble diameters at different injection velocity were in good agreement.

Acknowledgements

I thank all who in one way or another contributed in the completion of this thesis. First and foremost , I would like to express my sincere gratitude to my supervisor Prof. Fue-Sang Lien for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would also like to thanks the Iraqi Ministry of Higher Education and Scientific Research and the University of Technology (Baghdad-Iraq) for the financial support.

Getting through my dissertation required more than academic support, and I have many, many people to thank for listening to and, at times, having to tolerate me over the last years. I cannot begin to express my gratitude and appreciation for their friendship.

Last but not the least, I would like to thank my family: my mother and my brothers and sister for supporting me spiritually throughout my study.

Table of Contents

List of Tables

List of Figures

Nomenclature

Δt	Time step
Δx	Grid size
ν	Fluid kinematic viscosity ($\frac{m^2}{s}$)
ν_t	Eddy viscosity ($\frac{m^2}{s}$)
ν_{total}	Fluid total viscosity ($\frac{m^2}{s}$)
Ω	Collision operator
ω	Fluid relaxation rate ($\frac{1}{s}$)
Φ_p	Volume fraction of particle phase
ρ_f	Fluid density ($\frac{kg}{m^3}$)
ρ_p	Solid particle density ($\frac{kg}{m^3}$)
τ	Fluid relaxation time (s)
τ_e	Turnover time of large eddy (s)
τ_k	Kolmogrove time scale (s)
τ_p	Particle relaxation time (s)
$\mathbf{c}_i$	Discrete velocity

$\mathbf{F}$	External force (N)
$\mathbf{r}$	Spatial node location
$\mathbf{u}$	Fluid macroscopic velocity ($\frac{m}{s}$)
θ	Thermal velocity
C_L	Length conversion factor (m)
C_s	Smagorinsky constant
c_s	Speed of sound ($\frac{m}{s}$)
C_t	Time conversion factor (s)
C_U	Velocity conversion factor ($\frac{m}{s}$)
d_p	Particle diameter (m)
f	Fluid distribution function
f^{eq}	Equilibrium distribution function
k_b	Boltzmann constant (J/K)
k_n	Knudsen number
m_p	Mass of single solid particle (kg)
T	Temperature (K)
t	Time (s)
u'	Streamwise fluctuating velocity ($\frac{m}{s}$)
U_j	Jet inlet velocity (m/s)
U_m	Jet centerline velocity (m/s)

u_{rms}	Streamwise root mean square velocity ($\frac{m}{s}$)
v'	Spanwise fluctuating velocity ($\frac{m}{s}$)
V_p	Volume of single solid particle (m^3)
v_{rms}	Spanwise root mean square velocity ($\frac{m}{s}$)
w_i	Discrete weight function
M	Transformation matrix

List of Abbreviations

Re	Reynolds number
BGK	BhatnagarGrossKrook
CA	Cellular automata
CFD	Computational fluid dynamics
D3Q19	Three dimensions-nineteen velocity directions LBM model
DEM	Discrete element method
DNS	Direct numerical simulation
EIM	Eddu interaction model
GPUs	Graphics processing units
LBB	Link bounce-back boundary
LBM	Lattice Boltzmann method
LES	Large eddy simulation
MD	molecular dynamics

MRT	Multiple relaxation time
RANS	Reynolds average Navier-Stokes
SGS	subgrid scale
SRT	Single relaxation time

Chapter 1

Introduction

Single-phase flows have been studied for many years; problems related to these kinds of flows have been traditionally modelled using Navier-Stokes equations, which describe the mass, momentum, and energy of the single-phase fluid flows. One difficulty in modelling fluid flows appears when turbulence take place in the flow. Turbulence has large effects on the transport of mass, momentum, and energy which can be difficult to model. The method of direct numerical simulation (DNS) can be the most accurate way to calculate the turbulence effects on the problem under study. To obtain the most accurate results for turbulent flows using DNS, all time and space scales must be captured which requires large and often unacceptable amounts of computational resources. In addition to single-phase flow problems, multiphase flow problems have more recently been gaining attention and the formulas describing the equations of mass, momentum, and energy are still ambiguous. Multiphase flows can be classified in to four main groups: liquid-gas, liquid-solid, gas-solid and three phase flows. Fluid-particle or particle laden flows are two phase flows; the first phase is the fluid (gas or liquid) which occupies the majority of the flow region and the second phase is the particles. The complex hydrodynamic interactions between particles and fluid makes the analytical solution for such flow very complicated or impossible to be obtained [61]. Instead, numerical modelling is more suitable to understand the complex behaviour of such systems. In simple cases, particles suspended in fluid can be described as rigid bodies moving in a Newtonian fluid. Complex fluid flows such as turbulent flows may add additional dissipative motion which needs more accurate modelling to describe. Increasing the solid fraction raises the particle-fluid interaction and leads to two-way coupled flows. Brownian motion and particle-particle interaction force further complicate the hydrodynamics of the system. With microscopic particle suspension, increasing the particle fraction may change the fluid properties significantly and produce a non-Newtonian fluid

[61]. A wide range of applications related to the gas-particles flows has led to different formulations to model such flows. In general, there are three kinds of gas-solid models based on LBM: Lagrangian point-like particle tracking approach [19], Lagrangian fully-resolved particle tracking approach [53], and cellular automata probabilistic approach [64][65].

1.1 Impinging Jets

Turbulent vertical impinging jets have wide applications in engineering and industrial fields. Due to the high levels of momentum and temperature diffusion near the impinging surface, impinging jets can be used in the removal of heat from industrial and electronic devices. To improve the heat and mass transfer within the wall jet zone, solid or liquid particles (second phase) can be added to the jet flow (first phase). The new generation of the vertical takeoff and landing aircraft utilize vertical impinging jets to generate the necessary lift force to push up the aircraft with zero forward speed. According to Jambunathan et al. [44], vertical impinging jets have five main zones as shown in Figure 1.1, which are:

1. Potential core zone: In this region the nozzle centerline velocity U_m is nearly equal the jet exit velocity U_j. The end of this region is defined when $U_m = 0.95\ U_j$.
2. Initial mixing zone: This region lies at the outer edge of the flow region. Because of the velocity difference in this region (surrounding velocity = zero), the mixing region would be generated causing the surrounding fluid to be entrained in to the flow zone.
3. Established zone: For the free jet flow or when the impinging wall is placed in distance larger than the core zone, this zone will be generated. In this zone the fall of the centerline velocity and the jet half width (jet width where $U = U_m/2$) will be directly proportional to the axial distance after the core zone.
4. Deflection zone: Close to the impinging wall, the axial velocity is decreased rapidly to the wall velocity (zero velocity). As a result, static pressure is raised corresponding to the velocity decreasing.
5. Wall jet zone: Close to the wall, the transverse velocity rises rapidly and its value is high near the wall and start to fall far from it. Due to the high turbulence levels, this zone is very important when the high rate of momentum and heat transfer are needed.

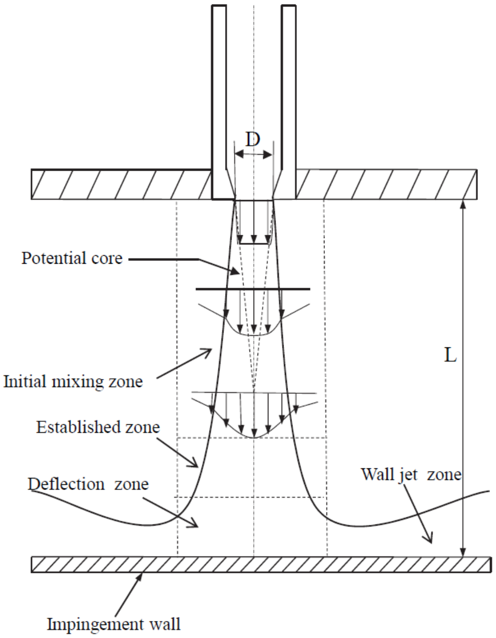


Figure 1.1: Flow zones in an impinging jet.

1.2 Fluid-Particle Impinging Jet Flows

To improve the heat and mass transfer within the wall jet zone, solid or liquid particles (a second phase) can be added to the jet flow (the first phase). Fluid-particle laden flows appear in numerous applications starting from home devices like ink-jet printers, air freshener, and car engines and extend to many industrial and environmental applications such as sand conveying tubes, gas turbines, sand grinding, surface erosion, pollutants transport in atmosphere, and paint and paper manufacturing. As well as, there are many applications in the pharmaceutical field such as aerosols which are pressurized packages containing very small drug particles (e.g. liquid or solid) dispersed in a propellant (e.g. gas). The aerosol can deliver drugs to the lungs utilizing the high pressure, high speed, and the turbulent characteristics of the impinging jets to provide better distribution of the drugs inside the lungs which work as impinging surfaces.

1.2.1 Fluid-Particle Interaction

Fluid-particle or particle laden flows are two phase flows; the first phase is the fluid phase (e.g. gas or liquid) and it occupies the major size of the flow region, and the second phase is the particle phase (e.g. solid or liquid). As the second phase occupies small fraction of the flow, it is named the dispersed phase. The two phases are separated by interface and they are interacted and correlated dynamically through it. The main flow is generated by the first phase due to pressure or gravity forces, while the second phase or the particle phase acquires its motion from the first phase due to the drag force between the two phases.

According to Tsuji [95] and Elghobashi [22], turbulent particle laden flows can be classified as three main categories according to the particle volume fraction or concentration in the flow which may affect the nature of interaction between the particles and fluid. Figure 1.2 depicts these categories, and the characteristic quantities in this figure are defined as:

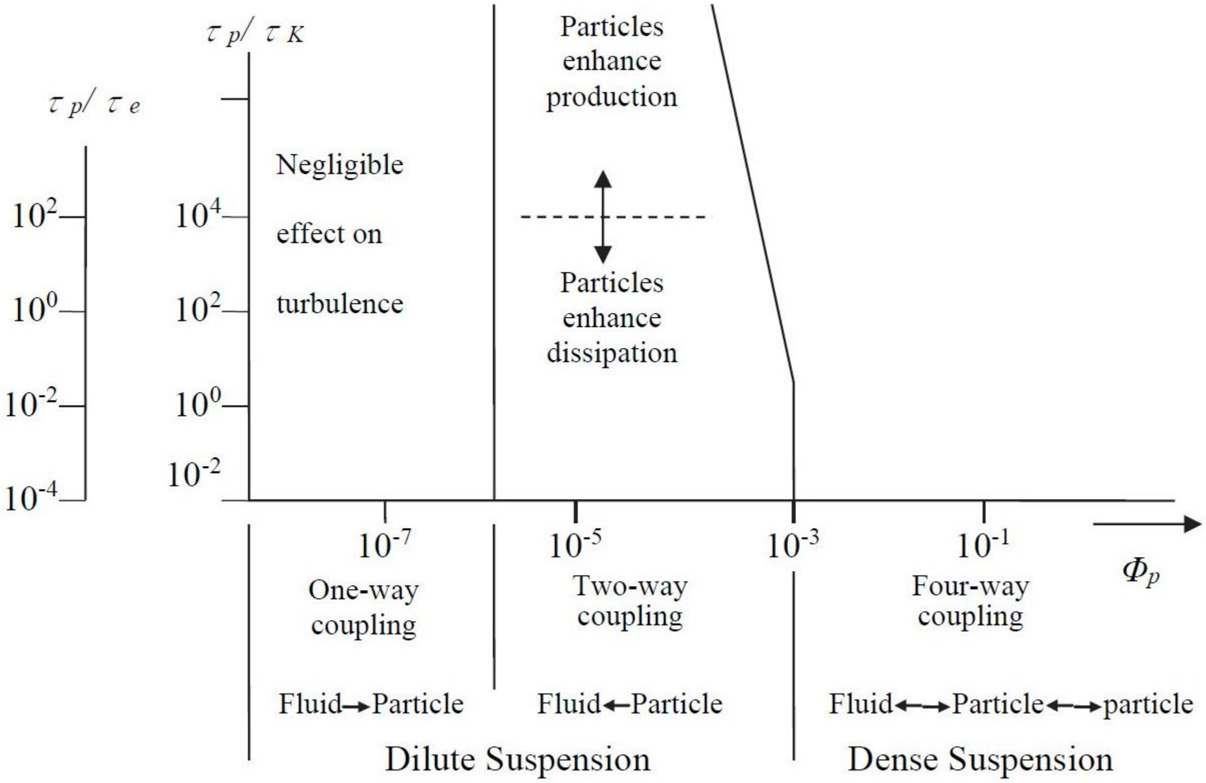


Figure 1.2: Map of flow regimes in particle-laden flows [24].

- V_p: volume of single particle,
- M: number of particles,
- V: volume occupied by both particles and fluid,
- Φ_p: volume fraction of particles defined as, $\Phi_p = MV_p/V$,
- d: diameter of particle,
- ρ_f, ρ_p, and ν: fluid density, particle density, and fluid kinematic viscosity, respectively,
- τ_p: particle response time (the time required for a particle to acquire the fluid phase motion) defined as: $\tau_p = \rho_p d^2/(18\rho_f \nu)$,
- τ_k: Kolmogorov time scale $= (\nu/\epsilon)^{1/2}$,
- τ_e: turnover time of large eddy $= \ell/u$,

where u is the *rms* fluid velocity, ℓ is the length scale of energy containing eddies, and ϵ represents the dissipation rate of turbulence kinetic energy. The three categories in Figure 1.2 are:

1. Collision free flow (dilute phase flow): in this region, the volume fraction of particles is very low ($\Phi_p \leq 10^{-6}$). The particles have a negligible effect on turbulence, and the interaction between the turbulence and the particles is called one-way coupling. The term one-way means that the momentum transfer from the main turbulent flow to the particles while the opposite transfer of momentum from particles to flow is insignificant due to the low particle concentration. The dispersion of particles in this flow highly depends on the turbulence of the flow field.

2. Collision dominated flow (medium concentration flow): in this region, the volume fraction of particles is considered as a medium fraction ($10^{-6} < \Phi_p \leq 10^{-3}$). The momentum transfer from the particle to the fluid phase causing a significant effect on the fluid turbulence structure, and the interaction between the fluid and the particles in this case is called two-way coupling. Figure 1.2 shows that within this region at constant Φ_p, the turbulent energy dissipation is increased with decreasing τ_p. The reason for this phenomenon is when the particle response time τ_p is decreased (i.e. decreasing the particles diameter) and for the same volume fraction, the contact area

between the fluid and the particles will be increased. On the other hand, increasing τ_p at the same Φ_p causing the particle Reynolds number Re_p to be increased. As stated in [24], when the particle Reynolds number reach values in the range of $Re_p \geq 400$, vortex shedding (around particles) is generated resulting enhancement in the turbulent energy production.

3. Contact dominated flow (dense phase flow): in this region, the volume fraction of particles becomes very high ($\Phi_p \geq 10^{-3}$), and the flow in this case is referred as dense suspension flow. In addition to two-way coupling interaction between particles and turbulence in fluid, each particle can collide with others causing additional momentum transfer between particles. This particle-particle-fluid interaction is known as four-way coupling. When Φ_p approaches 1, the granular flow is produced and in this case the solid phase will occupy the major part of the flow regime.

1.2.2 Particles Deposition Mechanisms

The most important mechanisms for the deposition of solid particles suspended in turbulent flows onto surfaces are: diffusion, gravity, inertia, and electrostatic effects, as show in Figure 1.3. Diffusion can be divided into two mechanisms: Brownian diffusion and turbulent diffusion. In Brownian diffusion, particles collide randomly with surrounding fluid molecules, and this mechanism is significant for very small particles ($d_p \ll 1\mu m$) [3]. The turbulent diffusion (turbulent dispersion) mechanism is governed by the fluid flow turbulence and it depends on the particle Stokes number. In gravitational mechanism, which is

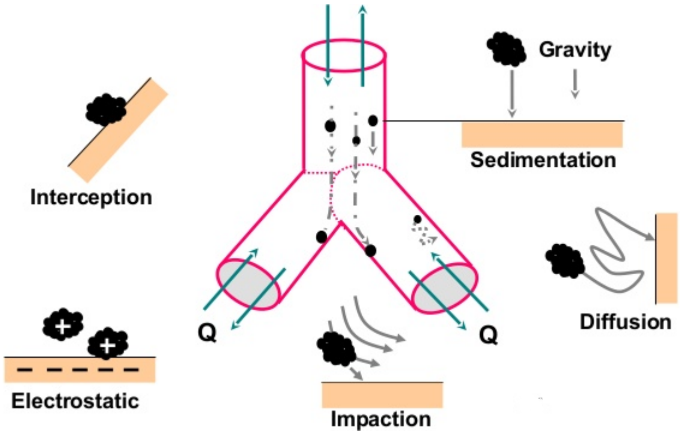


Figure 1.3: Particles deposition mechanisms [43].

also called sedimentation, the particles in a fluid moves under the effect of gravity force and reaches a velocity called settling velocity. The inertial mechanism includes the effects of impaction and interception. In inertial impaction, the fluid velocity experiences a rapid change in magnitude and direction which causes a high relative velocity between the fluid and particles especially for large particle size (i.e. high Stokes number). The inertial interception mechanism is dominant for curved surfaces or in complex flow systems where strong curvature exists. Finally, charged particles near the solid wall may experience an electrostatic force due to induced charges on the wall surface [3].

1.3 Levels of Fluid Dynamics Modelling

Modelling fluid dynamics systems mathematically is interesting and challenging work. Such models can be achieved with Hamilton's equations (Hamiltonian mechanics), Boltzmann equation, and Navier-Stokes equations at the micro-, meso-, and macro-scale, respectively [66]. These three methods of modelling (or description) of fluid flow dynamics are illustrated in Figure 1.4. The LBM is considered to be at the meso-scale level as it describes the molecular movement of particles in a statistical way that connects the fundamental

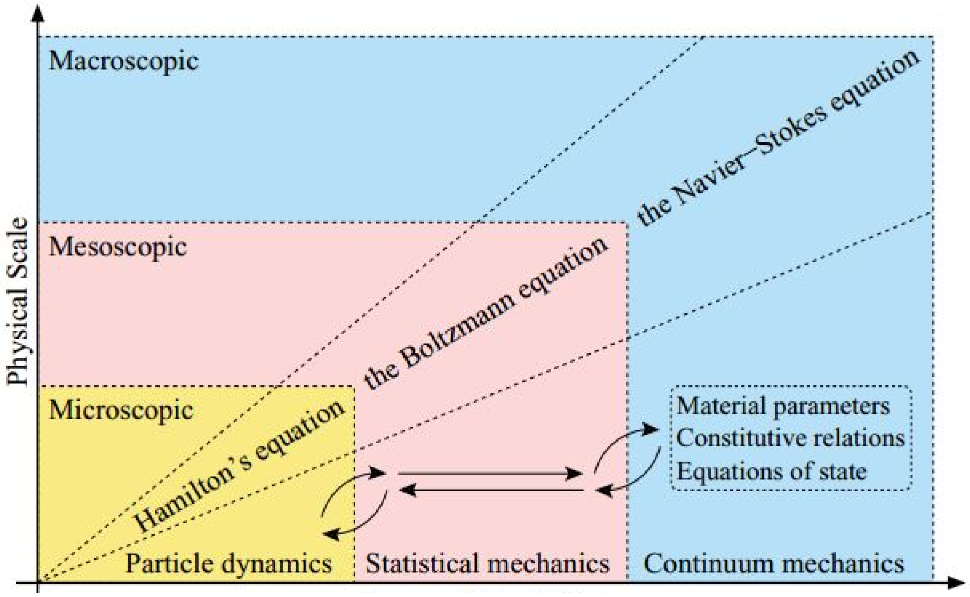


Figure 1.4: Fluid dynamics description levels [66].

roles of kinetic theory (micro-scale) with the continuous macro-scale level. Theoretically, all transport parameters such as viscosity and thermal diffusivity can be deduced from the fundamental knowledge of the fluid particle state. In the microscopic scale, the state of particles (atoms or molecules) is described by the local momentum and location for each particle. In addition, the comprehensive definition of the particles state in a finite region leads to define the hydrodynamics variables like pressure, velocity, temperature, and density.

Chapter 2

Literature Survey

2.1 Introduction

A wide range of applications related to the gas-particles flows has led to extended formulations to model such flows. The numerical approaches which deal with these formulations expand from direct numerical simulation-discrete particle system (DNS/DPS) models to two-fluid models considering different levels of description [77]. Some literature that this study is based on are summarized in Table 2.1. This chapter is organized such that each section review some of the literature approaches in a specific area (e.g. single phase impinging jet and Lagrangian particle tracking). After these reviews, the pros and cons of each approach are presented showing the motivation of using the present model.

Table 2.1: Summary of previous studies on single and two-phase flow

Researcher and year	Type of study	Flow type	Fluid model	Particle model	Focus of study
Cooper et al., 1993 [15]	Experimental	Single	-	-	impinging jet flow for $L/D = 2 - 10$
Craft et al., 1993 [16]	Simulation	Single	RANS	-	impinging jet flow for $L/D = 2$
Geers et al., 2004 [33]	Experimental	Single	-	-	impinging jet arrays
Alatawi and Matida 2012 [4]	Simulation	Two	RANS LES	Lagrangian EIM	Particle deposition in impinging jet flow
Elghobashi, 1993 [24]	Simulation (Review paper)	Two	DNS LES	Two-fluid Lagrangian	Particle-Laden turbulent flows
Xiong et al., 2014 [104]	Simulation	Two	LBM	Lagrangian DEM	Four-way coupling bubbling fluidized bed
Chopard et al., 2000 [13]	Simulation	Two	LBM	CA	Dynamics of solid particles erosion, transport, and deposition
Zhang et al., 2014 [109]	Simulation	Two	LBM	Lagrangian DEM	Particle-Particle collision. Particle Sedimentation
Salmanzadeh et al., 2007 [85]	Simulation	Two	FVM	Lagrangian	Particle deposition on a rectangular Obstruction in a duct flow
Burwash et al., 2006 [7]	Experimental Simulation	Two	CFD (Ansys)	Lagrangian	Particle deposition on an impingement surface
Sethi and John, 1993 [86]	Experimental	Two	-	-	Particle deposition pattern on an impingement surface
Marple, 1972 [62]	Experimental Simulation	Two	FDM	Lagrangian	Round and rectangular particle impactor

2.2 Single-phase Impinging Jet Flows

Cooper et al. [15] accomplished a comprehensive experimental work focused on turbulent fluid flow and heat transfer of the orthogonally impinging jet. In this work, different L/D values ranging from 2 to 10 were studied and the distances 2 and 6 were the main focus of this experiments, and two values of Reynolds number had been considered (i.e. 2.3×10^4 and 7×10^4). The results reported the mean and *rms* velocity profiles in the vicinity of the impingement surface. The mean velocity profiles showed small variation when L/D was changed, while the *rms* velocities were increased with increasing L/D. In general, the mean velocities are not affected by the wall within a distance between the jet exit and $y/D \simeq 0.7$ from the wall. The wall starts to show more pronounced influence on the mean velocity in the near wall region ($y/D < 0.7$) due to action of the turbulent shear layer. Contrarily, the turbulence intensity increases smoothly within a distance $y/D < 2$, and this growing was attributed to the turbulent diffusion within the mixing region of the jet.

The turbulence intensity reaches its maximum values at distance $y/D \simeq 0.3$ because of the near wall shear layer, and the intensity falls abruptly close to the wall due to the wall damping effect.

The numerical study of Hällqvist [38] utilized the LES turbulence model to simulate a circular impinging jet at Reynolds number $Re = 20000$. The LES results were compared with experimental data and RANS based model results. The author stated that the RANS model, which is computationally cheaper than LES, provides plausible results for free jet flow simulation; however, the more complex hydrodynamics (e.g. dynamics of the coherent structures) need more accurate turbulence modelling such as LES to resolve. For that reason, it was believed that the comparison of mean and *rms* velocity values between LES and RANS was reasonable. Primary vortices were formed after the jet outlet due to large radial velocity gradient as a result of the low momentum surrounding fluid. These vortices starts to interact with each other in the downstream fluid flow, and they are deflected by the wall which produces secondary vortices due to the radial convection of the primary vortices.

2.3 Eulerian-Lagrangian Solid Particle Simulation

In this method, the fluid flow simulation is based on the Eulerian fixed node approach (e.g. CFD and LBM) and the solid particles movement is estimated based on Newton's second law of motion.

Derksen [19] incorporated point-like solid particles in LBM-LES flow fields in a mixing tank and swirling tube. In this study, the underlying fluid (water) simulation is based on LBM- Eulerian (fixed grid) approach, while the motion of each single particle is determined by integrating the equations of motion (Lagrangian simulation). The particle diameter is $d_p = 0.3\ mm$ and the Stokes number is 1. Despite that the flow in this simulation is turbulent, the particle-turbulence interaction is not considered (i.e. one-way coupling). The Lagrangian particles tracking approach can determine the exact spatial location of each individual particle at each time step. The LBM-Lagrangian approach may lose the parallelism inherent to LBM because the two phases (i.e. fluid and particle) are represented in different (Eulerian and Lagrangian) frameworks [99].

2.3.1 RANS-Eddy Interaction Model

In the Eddy Interaction Model (EIM), one particle is allowed to interact successively with various eddies. Each eddy has a characteristic lifetime, length, and velocity scales that are obtained from the primary flow calculation results. The end of the interaction between the particle and one eddy occurs when the lifetime of the eddy is over or when the particle crosses the eddy. At this instant, a new interaction for the particle with a new eddy is started, and the particle will have another trajectory according to its equation of motion [47] [67]. When the particle enters the eddy, the local fluctuating velocity at the particle location is added to the local mean velocity to obtain the instantaneous fluid velocity used in the particle equation of motion.

Alatawi [3] predicted in his work the aerosol particle transport and deposition in impinging jet flow. This work utilized Reynolds Average Navier Stokes (RANS) model for the fluid flow simulation and the modified EIM method for the fluid-particle interaction. The near wall correction technique through the modified EIM was used to model the near-wall region to accurately predict the particle deposition in this region especially for small particles where the turbulence fluctuation is the main driving mechanism. The deposition of a particle is considered when the particle centre reaching a distance from the wall equal to the particle radius where the particle stick to the impingement wall and no additional bounce-back movement is allowed.

The RANS approach used in traditional CFD method is unable to predict the turbulent fluctuating velocities around the solid particles, which are crucial in the turbulent fluid-particle interaction problems. Contrarily, the LES approach is highly desirable for providing the turbulence (instantaneous) hydrodynamics surrounding the suspended solid particles. In general, the eddy viscosity (subgrid viscosity) in the LES models (e.g. Smagorinsky model) is time and space dependent and it is based on the local deformation rate. Consequently, one of the advantage of applying LBM over the traditional CFD methods is that the deformation rate can be calculated locally from the non-equilibrium part of the distribution function, while the CFD based methods (e.g. finite difference method) need to calculate the spatial derivatives of the velocity field [19].

2.4 Lagrangian Fully Resolved Particle Simulation

The fluid-particle interface in this method can be defined according to the distance between the LBM grid links. As shown in Figure 2.1, the particle volume in this method should be large enough to include many LBM nodes inside it for better describing the fluid-particle interface (i.e. the particle should include at least ten lattice nodes [53]). In other words, for small particle size (e.g. order of micrometer), the grid resolution should be very high to resolve the particle size. In LBM, the distribution function at each lattice link provides momentum according to the velocity and direction of the link. The net force exerted on the solid particle can be calculated from the sum of the rate of momentum exchanged between the nodes inside the particle and the surrounded fluid nodes through the boundary links.

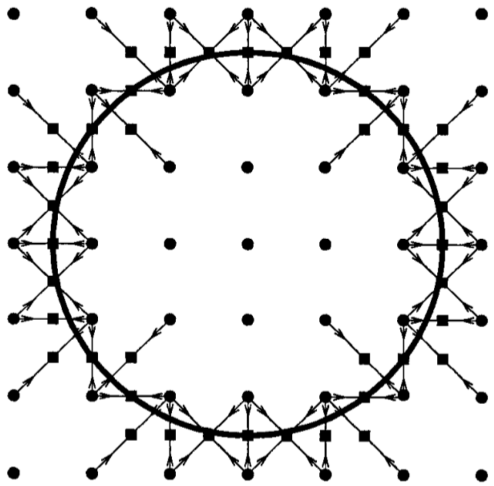


Figure 2.1: Location of the boundary nodes for a circular particle [52].

From this idea, Ladd [53] [54] introduced the main concepts of the force calculation for sphere particles driven by fluid flows. In these studies, series of stair-like boundary links were used to describe the particle location in the flow (Figure 2.1). At each link, the idea of bounce back boundary condition was applied to calculate the net momentum exchange between both sides of the boundary (i.e. the fluid-solid interface). As shown in Figure 2.2, a boundary node $\mathbf{x}_w$ is located at halfway distance between the fluid node $\mathbf{x}_f$ and the solid node $\mathbf{x}_b$. The momentum exchange between two opposite sides neighbouring the wall node $\mathbf{x}_w$ is represented as:

$$\mathbf{F} = [\Big(\mathbf{e}_{\bar{\alpha}}\tilde{f}_{\bar{\alpha}}(\mathbf{x}_b, t) - \mathbf{e}_{\alpha}\tilde{f}_{\alpha}(\mathbf{x}_f, t)\Big)\,\Delta x/\Delta t], \tag{2.1}$$

where $\mathbf{e}_{\bar{\alpha}}$ represents the lattice velocity direction opposite to $\mathbf{e}_{\alpha}$, $\tilde{f}$ represents the post collision distribution function, and $\mathbf{x}_f = \mathbf{x}_b + \mathbf{e}_{\bar{\alpha}}\Delta t$. By calculating the momentum exchanges for the lattice links (N_w) and sum for all the boundary nodes ($\mathbf{x}_w$), the total force on the solid particle can be calculated as follows:

$$\mathbf{F} = \sum_{\mathbf{x}_w} \sum_{\alpha=1}^{N_w} \mathbf{e}_{\bar{\alpha}}[\tilde{f}_{\alpha}(\mathbf{x}_b, t) + \tilde{f}_{\bar{\alpha}}(\mathbf{x}_f, t)] \times [1 - w(\mathbf{x}_b + \mathbf{e}_{\bar{\alpha}}\Delta t)]\Delta x/\Delta t. \qquad (2.2)$$

The outer summation in Equation 2.2 calculates the force for the interface boundary nodes $\mathbf{x}_w$, and the inner summation calculates the force contribution from all lattice links which intersect the boundary. The parameter w represents an indicator for the strength of each neighbouring nodes.

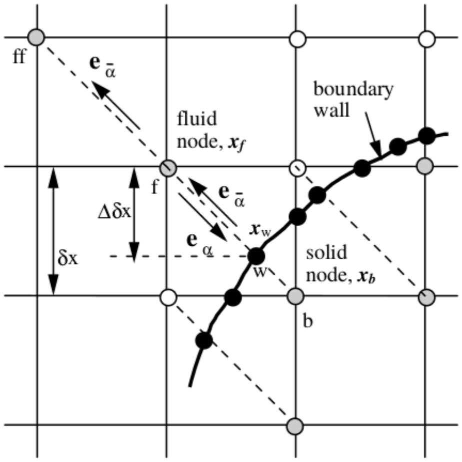


Figure 2.2: Inter-phase boundary nodes [52].

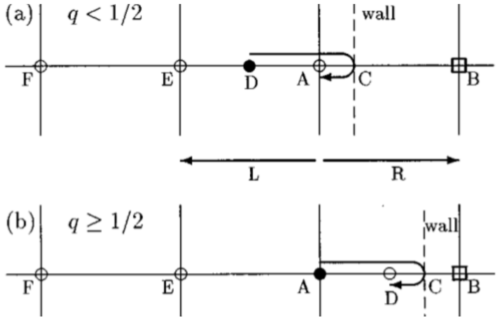


Figure 2.3: Wall boundary location [52].

Bouzidi et al. [6] proposed a new boundary condition for a moving curved boundary. This boundary condition is based on the bounce-back scheme and spatial interpolation of the distribution function. The nodes A and B in Figure 2.3 are the fluid and solid boundaries, respectively, and C represents the wall boundary location. The location of the wall is characterized by $q = |AC|/|AB|$. In the case of $q = 1/2$, the distribution function leaving A to the wall needs one time step to complete one cycle and returns to A in the opposite direction. Consequently, any change in q leads to change in the time period needed to complete one spatial step. The remedy for this issue was proposed according to the q value as following:

1. $q < 1/2$: In this case a new location point (D) is constructed and a fictitious fluid population is calculated at this point. As seen in Figure 2.3 (a), the distribution

function leaving point D reaches point A after one time step which is consistent with the LBM scheme.

2. $q \geq 1/2$: In this case point D is located between point A and C as shown in Figure 2.3 (b). The post-advection distribution function at A can be calculated using the distribution values at E and F.

In both two cases, interpolation is needed to construct the new distribution functions at the fluid nodes after collision. To do this, suppose $\mathbf{r}_l$ is the fluid node such that $\mathbf{r}_l + \mathbf{c}_i$ is a solid node. Let $\mathbf{c}'_i$ the opposite direction velocity of $\mathbf{c}_i$ ($\mathbf{c}'_i = -\mathbf{c}_i$). Using linear interpolation leads to:

$$f'_i(\mathbf{r}_l, t+1) = 2q\tilde{f}_i(\mathbf{r}_l, t) + (1-2q)\tilde{f}_i(\mathbf{r}_l - \mathbf{c}_i, t) \qquad for \qquad q < 1/2, \tag{2.3}$$

$$f'_i(\mathbf{r}_l, t+1) = \frac{1}{2q}\tilde{f}_i(\mathbf{r}_l, t) + \frac{2q-1}{2q}\tilde{f}_i(\mathbf{r}_l, t) \qquad for \qquad q \geq 1/2, \tag{2.4}$$

where $\tilde{f}_i$ represents the after collision (before streaming) distribution function.

The link bounce-back (LBB) boundary condition was proposed by Ladd [53] [54] for the fluid-moving particle interaction simulation. In this method the bounce back boundary condition was imposed at the fluid-particle boundary where the boundary is located at a half link distance between a pair of fluid-solid nodes. The no-slip condition was applied for the fluid nodes adjacent to the particle wall, which means that both solid and fluid nodes have the same linear and rotational velocities of the particle, and the interface velocity is calculated as:

$$\mathbf{V}_b = \mathbf{U}_P + \Omega_P \times [(\mathbf{x} + \frac{1}{2}\mathbf{c}_i \Delta t - \mathbf{x}_p)], \tag{2.5}$$

where $\mathbf{U}_P$, Ω_P and $\mathbf{x}_P$ are the linear velocity, angular velocity, and the solid particle centre of mass location, respectively. Applying the BB boundary condition for fluid boundary nodes and adding the effect of the moving particle momentum to the distribution function produces the following modified distribution function:

$$f'_i(\mathbf{x}, t+1) = \tilde{f}_i(\mathbf{x}, t) - 2w_i \rho \mathbf{V}_b \cdot \mathbf{c}_i. \tag{2.6}$$

Recall that w_i is the weighting factor for the lattice direction i and f'_i is the distribution function in the direction opposite to f_i. Obviously, when the particle velocity is zero, the bounce back boundary condition (Section 3.8.1) will be recovered from Equation 2.6. At this point, two opposite distribution functions were derived at each link which connects the solid and fluid nodes, and the net momentum exchange rate for these distribution functions produce the force acting on that link as follows:

$$\mathbf{F}'_i(\mathbf{x} + \frac{1}{2}\mathbf{c}_i\Delta t, t + \frac{1}{2}\Delta t) = 2[\tilde{f}_i(\mathbf{x}, t) - \tilde{f}'_i(\mathbf{x} + \mathbf{c}_i\Delta t, t) - 2\omega_i\rho\mathbf{V}_b \cdot \mathbf{c}_i]\mathbf{c}_i. \tag{2.7}$$

The half time step in Equation 2.7 comes from the fact that the distribution function and its momentum reach the boundary nodes at time equal to half time step when the interface boundary nodes are located at half link distance. Subsequently, the force calculation is computed after two time steps in order to update the velocity and location of the particle. Another technique for calculating the force is to take the average value over two time steps and use this value for the updating process.

2.5 Lagrangian Particle-Particle Interaction

Mao [61] employed a hybrid model that used the lattice spring model (LSM) and LBM. LBM with MRT were used to compute the fluid hydrodynamics, while the LSM calculate the particle-particle interaction force. This model was able to model the spherical and spheroidal deformable particle in dilute suspension.

Han et al. [39] and Feng et al. [26] utilized LBM with large eddy simulation and discrete element method (DEM) for fluid-particle simulation. The DEM is used to determine the interaction force between solid particles. In this work, the traditional bounce back boundary condition at the particle surface was modified to overcome the high oscillation force at the boundary nodes. The modification is based on an immersed moving boundary scheme proposed by Noble and Torczynski [71]. In addition to reducing the hydrodynamic fluctuations at the nodes by smoothing the lattice representation of particles, the fluid density distribution is modified in this scheme to accurately apply the no-slip condition at the fluid-particle interface.

Owen et al. [76] presented a computational framework which couples LBM for the fluid flow and DEM for the particles suspended in flow interaction. The link bounce-back method (LBB) presented in the previous section [53] is one of the most popular fluid-particle interaction calculation method; however, this method suffers from inaccurate representation of the particle boundary and statistical noise during the process of force and torque calculations [76]. In this model (Owen et al. [76]), the immersed moving boundary method proposed by Nobel and Torczynski [71] was adopted for the hydrodynamic coupling between LBM and DEM. The advantage of this method over the others is the locality of information required for force and torque calculations, while the other methods need information from neighbouring nodes. This locality makes this method consistent with the parallel nature of the LBM and very valuable for parallel implementation code.

The numerical modelling approach proposed by Zhang et al. [109] combined LBM for fluid flow simulation, immersed boundary method (IBM) for fluid-particle interaction, and DEM for particle-particle interaction. This model was used to simulate the sedimentation of 2D circular particles in incompressible flow. The BGK LBM scheme with force term (F_{b_i}) were used to modify the fluid distribution functions (f_i) according to the force exerted from the solid particle phase as:

$$f_i(\mathbf{r} + \Delta t\mathbf{c}_i, t + \Delta t) = f_i(\mathbf{r}, t) - \omega\left(f_i(\mathbf{r}, t) - f_i^{eq}(\mathbf{r}, t)\right) + F_{b_i}\Delta t. \tag{2.8}$$

The fluid force acting on each solid particle ($\mathbf{F}_p$) is calculated based on the momentum exchange at the particle location. The total force ($\mathbf{F}(r, t)$) acting on a group of particles inside a control volume at time t (i.e. a volume of an LBM cell at location r) is the sum of all forces acting on the particles. The force term in Equation 2.8 is computed from $\mathbf{F}(r, t)$ as follows:

$$F_{b_i} = \left(1 - \frac{\omega}{2}\right) w_i \left(3(\mathbf{c}_i - \mathbf{u}) + 9(\mathbf{c}_i \cdot \mathbf{u})\mathbf{c}_i\right) \cdot \mathbf{F}(r, t). \tag{2.9}$$

The fully resolved particle simulation is accurate in defining the hydrodynamics and the drag force on the fluid-particle interface. However, for small solid particles (order of micron), the computational resources required for this method is huge and not available even with the most developed computers today. Because of the limited computational resources, the fully resolved particle method is limited for small scale problems with a limited number of particles.

2.6 Lattice Boltzmann and Cellular Automata Method

The works of Masselot and Chopard [65] [64] [12] merged the cellular automata (CA) method [31] [83] [102] [84] and LBM to produce a new model (conveniently named LBM-CA). In these works, fluid and solid particles are both described in the same Eulerian framework (i.e. LBM nodes). This model utilized 2D and 3D LBM schemes to simulate the fluid flow (wind model), and CA to predict the solid particles movement (snow model) considering the transport, deposition, and erosion of the solid particles. The snow model suggests a probability distribution parameter p_i for each solid particle to judge which link the particle will follow according to its local velocity. Moreover, the model defined the particle deposition rate by considering a freezing process for each particle follow the link to the solid boundary. The erosion is also considered in this model by allowing the deposited particle to be ejected upward with a specific probability depending on the wind speed.

Wang et al. [100] developed the LBM-CA model by considering the two-way coupling (i.e. the effect of solid phase presence in the fluid flow) according to the particle phase volume fraction (Section 1.2.1). The feedback forcing of particles on the flow field is represented by adding an external force term in the evolution equation of LBM as follows:

$$f_i(\mathbf{r}+\Delta t\mathbf{c}_i, t+\Delta t) - f_i(\mathbf{r},t) = \frac{1}{\tau}(f_i^{eq}(\mathbf{r},t) - f_i(\mathbf{r},t)) + F_i \cdot \Delta t, \tag{2.10}$$

where F_i is the force term at each lattice link i (i.e. $F_i = 3w_i\mathbf{c}_i \cdot \mathbf{F}/c_r$), and $\mathbf{F}$ is the total force exerting on fluid by particles in a control volume (lattice volume). The force term is derived from the momentum transfer between the two phases and can be written as:

$$F = -\frac{\rho_p}{\rho} V_r \sum_{p=1}^{M} F_{f\to p}, \tag{2.11}$$

where M is the total number of solid particles within the control volume, V_r is the ratio of one particle volume to the lattice control volume, and $F_{f\to p}$ is the external fluid (drag) force acting on the particle p and it can be found from Newton's equation of motion:

$$F_{f\to p} = \frac{d\mathbf{u}_{f,p}}{dt} = \frac{\mathbf{u}_f - \mathbf{u}_p}{\tau_p}. \tag{2.12}$$

The relative velocity $\mathbf{u}_{f,p}$ is the difference between the particle velocity $\mathbf{u}_p$ and the fluid velocity $\mathbf{u}_f$ at the particle location, and τ_p is the particle relaxation time.

$$\tau_p = \frac{\rho_p d_p^2}{8\mu_f} \tag{2.13}$$

More details about the LBM-CA method will be presented in Section 3.11.

2.7 Summary and Motivation

In summary, the fully resolved particle model can determine (exactly) the interaction between the fluid and the solid particles when the particle size contain lattice nodes enough to accurately represent the fluid-solid interface. This method requires large computational resources to resolve the flow domain with a large number of small particles. The LBM-Lagrangian approach is able to overcome the limited computational resources issue by viewing the solid particles as point-like particles without volumes. There are two main obstacles with using LBM-Lagrangian approach: first, many calculations (e.g. fluid drag force, particle velocity, and particle location calculations) are required for each single particle; second, the fluid and solid particles are represented in two different frameworks (i.e. Eulerian and Lagrangian) which may violate the parallelism inherent in LBM.

The previous LBM-CA models distribute particles through the LBM nodes at each simulation time step without considering the effect of the local fluid velocity on the particle transport algorithm (e.g. the number of moving particles at each simulation time step). In addition, the implementation of a 3D LBM-MRT-CA algorithm on serial coding is very computationally expensive because of the massive number of calculations required to execute such algorithm. A new parallel LBM-CA probabilistic model is proposed in this thesis to overcome the deficiencies of the previous models and can better capture the dynamic interaction between particles and the surrounding turbulent flow.

Chapter 3

Lattice Boltzmann Method

3.1 Introduction

Many observable macroscopic phenomena can be described according to the microscopic behaviours of the system: surface tension, intermolecular forces (attractive and repulsive), and phase separation are examples of macroscopic phenomena where the microscopic world play major roles on their formations. The major difficulty in modelling macroscopic problems is the scale separation between the observable effects and the underlying molecular effects on the problems. One of the remedies to capture all scales in simulation is using the molecular dynamics method (MD)[81]. However, dealing with a tremendous number of particles (i.e. on the order of Avogadro number) is very cumbersome and computationally expensive. Instead, statistical methods (e.g. LBM) implementation are used as computationally feasible alternatives to MD. The fact that a large amount of molecular level details such as atom speed and location are not important in the macroscopic description of flow support the use of statistical methods.

3.2 Microscopic Description of Gases

The original derivation of the meso-scopic statistical description of fluid dynamics is based on the microscopic molecular perspective. To start with a microscopic description, say there are N particles in a three dimensional system; in this case the system state is a function of N states for each particle in that system. The suitable state variables for each particle are the instantaneous position $\mathbf{r}_i$ and momentum $\mathbf{p}_i$, where $i = 1, ..., N$. The instantaneous particle velocity could be used as a state variable instead of the momentum, i.e. $\mathbf{c}_i = \mathbf{p}_i/m_i$, where m is the particle mass which is considered constant for all particles. In such system, there are $6N$ degrees of freedoms and because of the huge value of N, following each particle in the system needs enormous computational power which is unfeasible even with most developed supercomputer today. The Hamiltonian equation used to describe the particles motion in the system is as follows:

$$\frac{\partial \mathbf{r}_i}{\partial t} = \frac{\partial H}{\partial \mathbf{p}_i}, \quad \frac{\partial \mathbf{p}_i}{\partial t} = -\frac{\partial H}{\partial \mathbf{r}_i}, \quad i = 1, ..., N, \tag{3.1}$$

where H is the total energy of the system including the particle kinetic energy and any potential energy involved such as external electrical or gravitational field and interaction forces between the particles. The phase space $\mathbb{P}$ can be constructed as a coordinate frame work for the particles system. It has $6N$ mutual orthogonal axes, and each axis is associated with a degree of freedom, i.e. position or momentum of a particle. The instantaneous particles system could be specified as a single point $\mathbf{q} \equiv (\mathbf{r}_1, \mathbf{p}_1, ..., \mathbf{r}_N, \mathbf{p}_N)$ in the phase space $\mathbb{P}$. Theoretically, if the initial particles state is specified, the trajectories of the particles can be tracked using Equation 3.1, and the instantaneous system states could be specified. Practically, following this procedure is impossible because of the large number of particles and their related degrees of freedoms. Moreover, defining the initial state condition for each particle in the system is not affordable.

3.2.1 Kinetic Theory of Gases

Gases can be imagined as massive number of particles (molecules or atoms) moved in nearly constant velocity in random directions. The study of these particles and their motion (streaming and collision) inside enclosures can provide a satisfactory description of macroscopic properties of gases such as pressure, temperature, and molecular and thermal diffusion coefficients. The size of a fluid atom is very small which is on the order of one *Angstrom* ($1\overset{\circ}{A} = 0.1\ nm = 1 \cdot 10^{-10}\ m$). The concept of mean free path (l_m) (the distance traveled by a single particle before it collides another particle) was first defined by Clausius [9]. Consequently, the concept of mean free time (τ_m) was introduced as the time needed by a particle to travel between two successive collisions. Another term related to the kinetic theory of gases is the mean collision time (τ_μ) which is the time needed by two particles to complete one collision. The main idea of the kinetic theory is based on the principle of the ideal gas. This idea presumes that each particle in a gas spends most of travel time without any interruption by a collision. In addition, the gas is dilute enough in order to consider only binary collisions between particles, and any collision between more than two particles is rare enough to be considered. Knudsen number is defined as the ratio between the mean free length to the characteristic length-scale of the macroscopic flow configuration ($K_n = l_m/L$). Knudsen number is used as a limit indicator for the flow problem which can be solved by the continuum assumption, e.g. continuum condition is applied at $K_n \leq 10^{-2}$.

3.3 Statistical Approach

The kinetic theory of gases has developed our understanding about the connection between the molecular state and the macroscopic properties of gases in closed systems is in equilibrium. Properties of gases under non-equilibrium conditions topic has been developing for many years, and researchers are still working to build a better insight for the non-equilibrium constituent relations. The general basis for non-equilibrium kinetic theory of gases was provided by Liouville equation and hierarchy of equations [50]. Fortunately, in the practical macroscopic applications, information about each particle in the system is rarely required, rather it is important to know the statistical behavior for a collection of particles. The Hamilton equation defines fluid particles as points in the phase space where the fluid system can be envisaged as a group of small distinct systems ($\mathcal{N}$) which has many particles, enough to produce hydrodynamic properties equal to the macroscopic system properties. The collection of replicate systems is referred to as Gibbs ensemble [66]. Each replicate represents a point ($\mathbf{q}$) in the phase space $\mathbb{P}$; if the system contain a large number of replicates, they can be dense enough to define their distribution as a continuous function called density function. The phase space element $d\mathbf{q}$ should be large enough to contain appropriate points and small enough to keep the distribution function continuous. The density function can be normalized to define a probability distribution function (F_N) in the phase space. The probability of finding $\mathcal{N}$ points in an element $d\mathbf{q}$ in the phase space is $F_N d\mathbf{q}$. The probability distribution function evolution is governed by Liouville's theorem:

$$\frac{\partial F_N}{\partial t} + \sum_{i=1}^{N} \mathbf{c}_i \cdot \frac{\partial F_N}{\partial \mathbf{r}_i} + \sum_{i=1}^{N} \mathbf{F}_i \cdot \frac{\partial F_N}{\partial \mathbf{p}_i} = 0, \tag{3.2}$$

where $\mathbf{F}_i$ is the force acting on particle i. Equation 3.2 is a conservation equation for the probability distribution function F_N.

3.4 Boltzmann Equation

In 1872, Ludwig Boltzmann described statistical behaviours of systems under non-equilibrium thermodynamic conditions. Boltzmann defined the hydrodynamics and thermal gradients as the effects of biased random distribution of the particles in the systems, and was able to describe this random distribution by a transport equation. The transport equation could be solved to define the macroscopic variables as moments of this equation. There are some limits for using Boltzmann equation based on the following assumption [98] [66]:

1. The gas is sufficiently dilute to allow only two particles to involve in a collision. This assumption assumes the interaction potential between particles is in a short range and the potential only accounts for two particles which are close to colliding. Otherwise, long range potentials could affect the two particles collision which violates the dilute assumption.

2. The collisions process starts and finishes in a very short range of space and time. This assumption ensures the locality of the collision process such that the collision time is very small compared to the macroscopic time scale. Consequently, the collision process will not transport, i.e. will not affect transport equation time scale.

3. During the collision process, total mass, momentum, and kinetic energy are conserved quantities assuming that the particle collision is elastic and no energy transformation exists in this process.

4. The collision process is microscopically reversible in time. This assumption is based on the random motion of particles which means that there is no preference for the particles movement before and after collision. For example, suppose there are two particles have initial velocities $\mathbf{c}$ and $\mathbf{c}_1$, respectively. The probability of these particles to acquire velocities $\widetilde{\mathbf{c}}$ and $\widetilde{\mathbf{c}_1}$ after collision is as same as the probability if they have initial velocities $\widetilde{\mathbf{c}}$ and $\widetilde{\mathbf{c}_1}$ to acquire final velocities $\mathbf{c}$ and $\mathbf{c}_1$.

5. The state of molecular chaos is assumed before collision; this means it is assumed that velocities of two particles close to a collision are uncorrelated

According to the above assumptions, Boltzmann was able to construct a formula based on binary collision in the following form:

$$\frac{\partial f}{\partial t} + \mathbf{c} \cdot \frac{\partial f}{\partial \mathbf{r}} + \mathbf{a} \cdot \frac{\partial f}{\partial \mathbf{c}} = \frac{1}{m} \int V_R(\bar{f}\bar{f}_R - f f_R) d\eta d\theta d\mathbf{c}_R, \tag{3.3}$$

where f is the mass density distribution function which is a function of time t, coordinate $\mathbf{r}$, and velocity $\mathbf{c}$. The $\mathbf{V}_R$ ($\mathbf{V}_R = \mathbf{c} - \mathbf{c}_1$) is the relative velocity between two particles close to collision, and V_R is the relative speed. The distance between two particles after collision is measured by a directional vector $\mathbf{b}(s, \eta, \theta)$ where the parameters s, η, and θ are the coordinates in a spherical system and $\mathbf{a}$ is the acceleration due to any external forces. The left hand side of Equation 3.3 represents the linear transport operator, while the right hand side is the collision non-linear operator. The collision operator term will be abbreviated by Ω, and the final Boltzmann equation can be re-written in the following index vector notation form:

$$\partial_t f + \mathbf{c}_\alpha \partial_\alpha f + \frac{F_\alpha}{m} \partial_{\mathbf{c}_\alpha} f = \Omega(\mathbf{r}, t). \tag{3.4}$$

Mass, momentum, and kinetic energy (ϵ) are conserved during the collision process and the conservation equation can be written as:

$$\int \Omega \begin{pmatrix} 1 \\ c_\alpha \\ \epsilon \end{pmatrix} = 0. \tag{3.5}$$

3.4.1 BGK Collision Operator

The collision operator term (Ω) in Equation 3.4 is non-linear and very complicated to solve. Higuera et al. [42] derived a linearized collision operator term considering it as a relaxation process to the equilibrium state. Qian et al. [80] proposed a simple LBM discretized scheme named (LBGK), as follows:

$$\underbrace{f_i(\mathbf{r} + \Delta t \mathbf{c}_i, t + \Delta t) = f_i(\mathbf{r}, t)}_{\text{streaming}} - \underbrace{\omega \left(f_i(\mathbf{r}, t) - f_i^{eq}(\mathbf{r}, t) \right)}_{\text{collision}}, \tag{3.6}$$

where $\omega = 1/\tau$, and τ is the relaxation time of the distribution function. The parameter f_i^{eq} is the equilibrium distribution function and it can be represented by the form of the Maxwellian equation:

$$f^M(\mathbf{r}, \mathbf{c}, t; \rho, \mathbf{u}, T) = \rho \left(\frac{m}{2\pi k_b T} \right)^{3/2} e^{-mv^2/2k_b T}, \tag{3.7}$$

where $v = \mathbf{c} - \mathbf{u}$ is the peculiar or thermal velocity which is the difference between the molecular and the macroscopic fluid velocities, and k_b is the Boltzmann constant ($k_b \approx 1.38 \times 10^{-23}\ J/K$). As shown in Equation 3.6, there are two main steps evolves at each time step in the LBM:

- streaming: fluid particles (i.e. distribution functions) move to the nearest node in the direction of i link velocity,
- collision: fluid particles arrive at a node from the neighbouring nodes and interact by changing their velocity directions.

3.4.2 Non-Dimensional Boltzmann Equation

Equations 3.4 and 3.7 represent the dimensional form of the density distribution function in unit of $(kg.s^3/m^6)$. To derive a non-dimensionalized form of the Boltzmann equation, we need to find appropriate dimensional parameter scales for time, spatial coordinate, and velocity. The characteristic length scale L_o could be used for spatial coordinate and the speed of sound ($c_s = \sqrt{\frac{k_b T}{m}}$) of the ideal gas can be used as a characteristic velocity. For simplicity, isothermal condition will be considered ($T = const.$), and the non-dimensionalized Boltzmann equation can be written as [51]:

$$\frac{\partial \frac{f}{\rho\left(\frac{k_b T}{m}\right)^{3/2}}}{\partial t \frac{c_s}{L_O}} + \frac{c_\alpha}{c_s} \frac{\partial \frac{f}{\rho\left(\frac{k_b T}{m}\right)^{3/2}}}{\partial \frac{x_\alpha}{L_O}} + \frac{F_\alpha L_o}{m c_s^2} \frac{\partial \frac{f}{\rho\left(\frac{k_b T}{m}\right)^{3/2}}}{\partial \frac{c_\alpha}{c_s}} = -\frac{f - f^{eq}}{\rho\left(\frac{k_b T}{m}\right)^{3/2}} \frac{1}{\tau \frac{c_s}{L_o}}, \tag{3.8}$$

after some algebra, the final non-dimensional form of the Boltzmann equation is as follows:

$$\partial_{\hat{t}} \hat{f} + \hat{c}_\alpha \partial_{\hat{\alpha}} \hat{f} + \hat{F}_\alpha \partial_{\hat{c}_\alpha} \hat{f} = -\frac{\hat{f} - \hat{f}^{eq}}{\hat{\tau}}. \tag{3.9}$$

The hat symbol is used for non-dimensional variables and it will be omitted for simplicity. The discretized form (as shown later in Section 3.5) of the equilibrium distribution function (Equation 3.7) is:

$$f_i^{eq}(\rho(\mathbf{r}, t), \mathbf{u}(\mathbf{r}, t)) = w_i \rho \left(1 + \frac{c_{i_\alpha} u_\alpha}{\theta} + \frac{c_{i_\alpha} u_\alpha c_{i_\beta} u_\beta}{2\theta^2} - \frac{u_\alpha u_\alpha}{2\theta}\right), \tag{3.10}$$

where w_i is the weight factor for each discrete particle velocity c_i, and the parameter θ is the squared sound speed. The equilibrium distribution function arguments, ρ and $\mathbf{u}$, are the local macroscopic fluid density and velocity. The local density and velocity (momentum) are defined as zeroth and first moments of the discretized density distribution function, and they can be calculated by summing the distribution functions over all discrete links:

$$\rho(\mathbf{r}, t) = \sum_i f_i(\mathbf{r}, t), \tag{3.11}$$

$$\rho(\mathbf{r},t)u_\alpha(\mathbf{r},t) = \sum_i c_{i_\alpha} f_i(\mathbf{r},t). \tag{3.12}$$

The weight coefficients w_i and the parameter θ are specified when the set of discrete velocities is fixed according to the problem under study, e.g. two or three dimensions problem. The convention for LBM model enumeration is abbreviated as DdQq, where d and q refer to the number of spatial dimension and the number of discrete velocities, respectively [80]. The discrete velocity set models D2Q9 and D3Q19 involve three speeds, 0, c_r and $\sqrt{2}c_r$ and the number of discrete velocities having these speeds for D2Q9 are 1, 4, and 4 (e.g. four links have the speed value of c_r, which are c_1, c_2, c_3, and c_4 as shown in Figure 3.1), while for D3Q19, they are 1, 6, and 12 (Figure 3.2), respectively. The speed c_r in Figure 3.1 is a reference speed and it has non-dimensional unit of value equal to one for square and cubic lattice schemes. The lattice spatial difference Δr is connected to the reference speed c_r by the discrete time step Δt ($\Delta r = \Delta t\ c_r$).

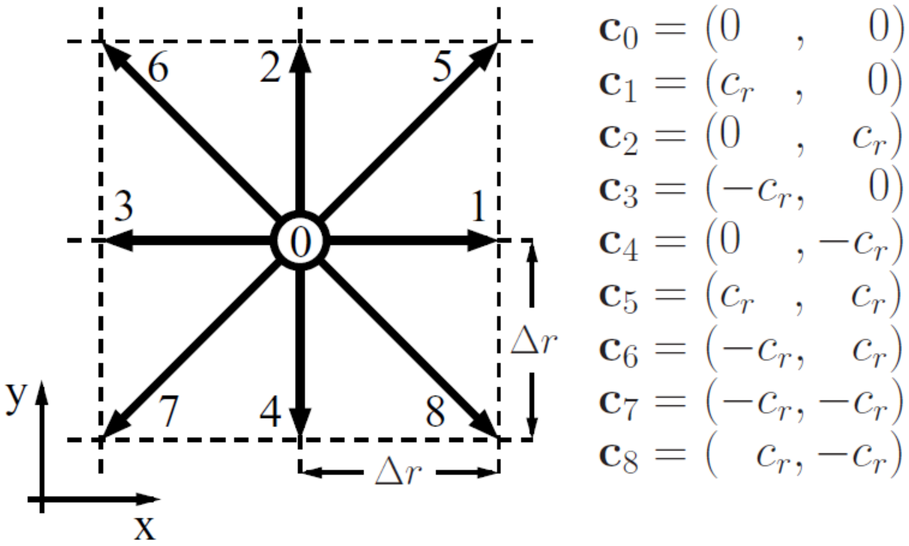


Figure 3.1: The set of nine discrete velocities D2Q9.

The values of the weight factors w_i are obtained from Hermite expansion (See Appendix A) of the distribution function f by considering a specific discrete velocity number. These values are independent on the velocity directions, rather they are speed magnitude dependent as shown in Table 3.1 [80]. The values of $w_0, w_1, w_2,$ and w_3 are for the discrete velocity speeds of 0, 1, $\sqrt{2}$, and $\sqrt{3}$, respectively.

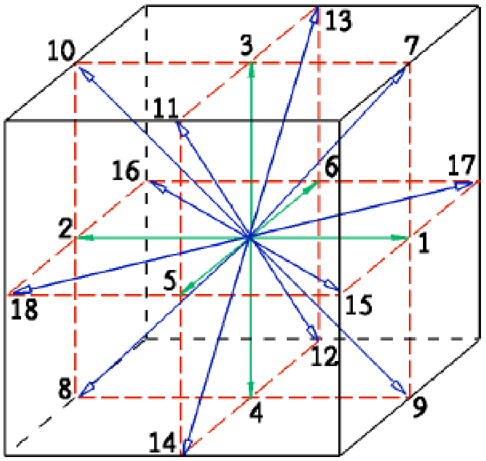


Figure 3.2: The set of nineteen discrete velocities D3Q19.

Model	w_0	w_1	w_2	w_3
$D1Q3$	2/3	1/6	0	0
$D2Q9$	4/9	1/9	1/36	0
$D3Q15$	2/9	1/9	0	1/78
$D3Q19$	1/3	1/18	1/36	0
$D3Q25$	1/3	1/36	0	0

Table 3.1: Weight coefficients for different lattice models.

3.5 Discrete Equilibrium Function

In numerical computation perspective, any continuous function should be convert to a discrete form to allow handling it by a computer algorithm. Accordingly, the continuous equilibrium distribution function (Equation 3.7) must be written in a discretized form. Some assumptions are needed to accomplish this task; the first assumption is that the flow is isothermal, which means that the temperature of the flow stays constant over space and time. Although the temperature actually fluctuates in the lattice Boltzmann scheme, these fluctuations can be neglected when the temperature distributions are not considered as main objective. A constant reference temperature T_0 is considered for the entire flow domain. The other assumptions is that the flow is incompressible and the fluid is Newtonian. The incompressibility in LBM is not clear because of local density fluctuations. It is not possible to neglect the density fluctuations in similar way to the temperature fluctuations because of the importance of these density difference to derive the pressure gradients in the flow domain from the ideal gas law. To simplify the equilibrium distribution function (Equation 3.7), a constant $\theta = k_b T_0$ is defined and the equation is rewritten as:

$$\begin{aligned} f^{eq}(\mathbf{c}, \rho, \mathbf{u}) &= f^M(\mathbf{c}, \rho(\mathbf{r}, t), \mathbf{u}(\mathbf{r}, t), T_0) = \rho (2\theta\pi)^{-d/2} e(-v_\alpha v_\alpha / 2\theta) \\ &= \underbrace{(2\theta\pi)^{-d/2} e(-c_\alpha c_\alpha / 2\theta)}_{=w_b(c_\alpha)} . \underbrace{e(c_\alpha u_\alpha / \theta)}_{=B(c_\alpha, u_\alpha)} . \underbrace{e(-u_\alpha u_\alpha / 2\theta)}_{=C(u_\alpha)}, \end{aligned} \tag{3.13}$$

where d, $e()$, and $v_\alpha = c_\alpha - u_\alpha$ are the spatial dimension, the exponential function, and the peculiar or relative velocity, respectively. The variable $\sqrt{\theta}$ has velocity unit and it is defined for the thermal speed of the fluid particles or the model speed of sound. Recalling Taylor series expansion for the exponential function and applying this expansion for the two exponential functions B and C in Equation 3.13:

$$\begin{aligned} B(c_\alpha, u_\alpha) &= 1 + \left(\frac{c_\alpha u_\alpha}{\theta} \right) + \frac{1}{2} \left(\frac{c_\alpha u_\alpha}{\theta} \right)^2 + ..., \\ C(u_\alpha) \quad &= 1 - \left(\frac{u_\alpha u_\alpha}{2\theta} \right) + \frac{1}{2} \left(\frac{u_\alpha u_\alpha}{2\theta} \right)^2 + ... \end{aligned}$$

The Mach number in the above expansions is $Ma \sim u_\alpha / \sqrt{\theta}$ and $c_\alpha \sim \sqrt{\theta}$. By considering the low Mach number flow, the terms involving higher-order Mach number (i.e. third order and higher) will be neglected. Only terms up to the second order are maintained and substitution of the above expansions in Equation 3.13 leads to:

$$f^{eq}(\mathbf{c}, \rho, \mathbf{u}) \simeq w_b(c_\alpha) \rho \left(1 + \frac{c_\alpha u_\alpha}{\theta} + \frac{c_\alpha u_\alpha c_\beta u_\beta}{2\theta^2} - \frac{u_\alpha u_\alpha}{2\theta} \right). \tag{3.14}$$

The microscopic velocity set (e.g., $\{c_{0_\alpha}, c_{1_\alpha}, c_{2_\alpha}, ..., c_{(q-1)_\alpha}\}$) is defined according to the spatial dimension and the lattice configuration, where the subscripts (0 to $q-1$) in the velocity define the velocity direction according to the lattice configuration. While the Greek letters α and β are the spatial components of the velocity vectors.

3.6 Chapman-Enskog Analysis

The macroscopic hydrodynamic equations of fluid and their relevant properties are functions of the microscopic original world. To verify this fact, the macroscopic equation (Navier-Stokes Equations) must be derived from Boltzmann equation. To accomplish this derivation, multiple-scale analysis is utilized for the discrete Boltzmann equation (Equation 3.6) [66]. First, a Taylor series expansion is used to find the finite-difference approximation for Equation 3.6. The derivative parameters for space ($\Delta r = \Delta t c_r$) and time (Δt) are small in the macroscopic scale. Therefore, only second order derivatives are retained:

$$\Delta t \partial_t f_i + \frac{\Delta t^2}{2}\partial_t^2 f_i + \Delta t c_{i_\alpha}\partial_\alpha f_i + \Delta t^2 c_{i_\alpha}\partial_\alpha \partial_t f_i + \frac{\Delta t^2 c_{i_\alpha} c_{i_\beta}}{2}\partial_\alpha \partial_\beta f_i = -\omega f_i^{neq}, \tag{3.15}$$

where $f_i^{neq} = f_i - f_i^{eq}$ is the non-equilibrium distribution function and $f_i = f_i(\mathbf{r}, t)$. Equation 3.15 contains two time scales, the convective τ_1 (*fast*) and the diffusive τ_2 (*slow*). These scales are connected to the discrete time step Δt by a small parameter ϵ, where $\tau_1 = \epsilon^{-1}\Delta t$ and $\tau_2 = \epsilon^{-2}\Delta t$. According to these time scales, two time variables are defined for the Boltzmann equation evolution, $t_1 = t_1^* \tau_1$ and $t_2 = t_2^* \tau_2$, where the star represents the non-dimensional time variables. The distribution function is now function of the two time variables $f_i(\mathbf{r}, t_1, t_2)$ and the time derivative is redefined as follows:

$$\partial_t \equiv \frac{\partial}{\partial t_1} + \frac{\partial}{\partial t_2} \equiv \partial_t^{(1)} + \partial_t^{(2)}.$$

The dimensionless spatial parameter can be defined as: $r^* = r/L$, where L is the characteristic length scale $L = c_r \tau_1$. Then, the temporal and spatial derivatives in Equation 3.15 can be written in the following form:

$$\Delta t \partial t = \Delta t \left(\frac{\partial}{\partial t_1^* \tau_1} + \frac{\partial}{\partial t_2^* \tau_2} \right) = \epsilon \frac{\partial}{\partial t_1^*} + \epsilon^2 \frac{\partial}{\partial t_2^*}, \tag{3.16}$$

$$\Delta t c_{i_\alpha} \partial t = \Delta t c_{i_\alpha}^* c_r \frac{\partial}{\partial r_\alpha^* L} = \Delta t c_{i_\alpha}^* c_r \frac{\partial}{\partial r_\alpha^* c_r \tau_1} = \epsilon c_{i_\alpha}^* \frac{\partial}{\partial r_\alpha^*}. \tag{3.17}$$

Based on the perturbation theory, the distribution function f_i can be represented using a power series with the expansion parameter ϵ (i.e. $\epsilon = \Delta t/\tau_1$):

$$f_i = f_i^{(0)} + \epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)} + ..., \tag{3.18}$$

where $f_i^{(0)}$ is the equilibrium distribution function which has an exact solution form (Maxwellian). The next step is substituting the series 3.18 in Equation 3.15, applying the multiple-scale derivatives and retaining only up to the second order term of ϵ, giving:

$$\begin{aligned}\Delta t\left(\partial_t^{(1)} f_i^{(0)} + \partial_t^{(2)} f_i^{(0)} + \epsilon\partial_t^{(1)} f_i^{(1)}\right) + \frac{\Delta t^2}{2}\partial_t^{(1)}\partial_t^{(1)} f_i^{(0)} + \Delta t c_{i_\alpha} \\ \left(\partial_\alpha^{(1)} f_i^{(0)} + \epsilon\partial_\alpha^{(1)} f_i^{(1)}\right) + \Delta t^2 c_{i_\alpha}\partial_\alpha^{(1)}\partial_t^{(1)} f_i^{(0)} + \frac{\Delta t^2 c_{i_\alpha} c_{i_\beta}}{2} \\ \partial_\alpha^{(1)}\partial_\beta^{(1)} f_i^{(0)} = -\omega\left(f_i^{(0)} + \epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)} - f_i^{(eq)}\right).\end{aligned} \tag{3.19}$$

The coefficients of ϵ with order of $0, 1$ and 2 should be matched on the both sides of Equation 3.19, and after some algebra the following is found:

$$\mathcal{O}(\epsilon^0): \qquad f_i^{(0)} = f_i^{eq}, \tag{3.20}$$

$$\mathcal{O}(\epsilon^1): \qquad \partial_t^{(1)} f_i^{(0)} + c_{i_\alpha}\partial_\alpha^{(1)} f_i^{(0)} = -\frac{\omega\epsilon}{\Delta t} f_i^{(1)}, \tag{3.21}$$

$$\mathcal{O}(\epsilon^2): \qquad \partial_t^{(2)} f_i^{(0)} + \epsilon\left(1 - \frac{\omega}{2}\right)\left(\partial_t^{(1)} f_i^{(1)} + c_{i_\alpha}\partial_\alpha^{(1)} f_i^{(1)}\right) = -\frac{\omega\epsilon^2}{\Delta t} f_i^{(2)}. \tag{3.22}$$

From the above formula, the zeroth order of the expansion $f_i^{(0)}$ in Equation 3.18 can be related to the equilibrium distribution function $f_i^{(eq)}$ and the higher terms (i.e $\epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)}$) can be related to the non-equilibrium distribution. It was stated before that the zeroth and the first moments of the equilibrium distribution function are conserved moments. This means that the equivalent moments for the higher order distribution vanish, yielding:

$$\sum_{i=0}^{q-1} f_i^{neq} \equiv \sum_{i=0}^{q-1}\left(\epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)}\right) = 0, \tag{3.23}$$

$$\sum_{i=0}^{q-1} c_{i_\alpha} f_i^{neq} \equiv \sum_{i=0}^{q-1} c_{i_\alpha}\left(\epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)}\right) = 0. \tag{3.24}$$

The next step is taking the zeroth moment for Equation 3.21,

$$\partial_t\rho + \partial_\alpha\rho u_\alpha = 0. \tag{3.25}$$

Equation 3.25 is the mass conservation equation. Next, taking the first moments of Equations 3.21 and 3.22 and combining them will recover the momentum conservation equation as:

$$\partial_t\rho u_\alpha + \partial_\beta\left(\Pi_{\alpha\beta}^{(0)} + \rho\theta\left(\frac{1}{\omega} - \frac{1}{2}\right)\Delta t\ \Pi_{\alpha\beta}^{(1)}\right) = 0, \tag{3.26}$$

where $\Pi^{(0)}_{\alpha\beta}$ and $\Pi^{(1)}_{\alpha\beta}$ are the convective and the diffusive momentum flux tensors, respectively, defined as follows:

$$\Pi^{(0)}_{\alpha\beta} = \sum_{i=0}^{q-1} c_{i_\alpha} c_{i_\beta} f_i^{eq} = \rho\theta\delta_{\alpha\beta} + \rho u_\alpha u_\beta = P\delta_{\alpha\beta} + \rho u_\alpha u_\beta, \tag{3.27}$$

$$\Pi^{(1)}_{\alpha\beta} = \sum_{i=0}^{q-1} c_{i_\alpha} c_{i_\beta} f_i^{(1)}. \tag{3.28}$$

It can be deduced from Equation 3.26 that the kinematic viscosity in LBM is:

$$\nu = \theta\left(\frac{1}{\omega} - \frac{1}{2}\right)\Delta t.$$

The final result can be written for the incompressible limit as follows:

$$\rho\partial_t u_\alpha + \rho u_\beta \partial_\beta u_\alpha = -\partial_\alpha P + \nu\rho\partial_\beta\partial_\beta u_\alpha. \tag{3.29}$$

3.7 Multiple Relaxation Time (MRT) Model

In this model, the hydrodynamics moments of the velocity distribution function are relaxed at different rates, or at different relaxation times. Each moment's relaxation rate could be tuned separately from different physical behaviours of the flow which leads to an increase in the simulation stability by choosing suitable relaxation times [21]. This model was provided by D'Humieres [20] and it demonstrated a superior numerical stability over the BGK single relaxation time. The distribution functions in this model should be mapped to their moments space to allow applying a different relaxation time for each moment. The mapping process is linear by the transformation matrix M such that:

$$|m\rangle = M|f\rangle, \tag{3.30}$$

where $|m\rangle$ and $|f\rangle$ are the moments and the distribution functions column vectors, respectively, such that $|m\rangle \equiv (m_0, m_1, m_2, ..., m_{q-1})^T$ and $|f\rangle \equiv (f_0, f_1, f_2, ..., f_{q-1})^T$. The transformation matrix is constructed to provide appropriate moments spanning the moment space and it is orthogonalized by the Gram-Schmidt orthogonalization procedure. When the transformation matrix M becomes an orthogonal matrix, the collision matrix S is then chosen to be diagonal matrix of eigen-value elements of M, yielding:

$$\hat{S} = MSM^{-1}, \tag{3.31}$$

where $\hat{S} \equiv diag(s_0, s_1, s_2, ..., s_N)$, and the transformation matrix M for the D3Q19 model is:

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-30 & -11 & -11 & -11 & -11 & -11 & -11 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
12 & -4 & -4 & -4 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & -4 & 4 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & -4 & 4 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & -4 & 4 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
0 & 2 & 2 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\
0 & -4 & -4 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -2 & -2 & -2 & -2 \\
0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -2 & -2 & 2 & 2 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1
\end{pmatrix} \tag{3.32}$$

The evolution equation for the MRT model will take the following form:

$$|f(\mathbf{r}_i + c_i\Delta t, t + \Delta t)\rangle - |f(\mathbf{r}_i, t)\rangle = -M^{-1}\hat{S}\ [|m(\mathbf{r}_i, t)\rangle - |m^{eq}(\mathbf{r}_i, t)\rangle], \tag{3.33}$$

where the $|m^{eq}(\mathbf{r}_i, t)\rangle$ represents the equilibrium moments vector. The D3Q19 MRT model will be presented here to give a better insight to this model. The 19 discrete velocities are:

$$c_{i_\alpha} = \begin{cases} (0,0,0) & i = 0, \\ (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0 \pm 1) & i = 1, 2, ..., 6, \\ (\pm 1, \pm 1, 0), (\pm 1, 0, \pm 1), (0, \pm 1, \pm 1) & i = 7, 8, ..., 18, \end{cases} \tag{3.34}$$

and the corresponding 19 moments are:

$$|m\rangle = (\delta\rho, e, \epsilon.j_x, q_x, j_y, q_y, j_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{xz}, m_x, m_y, m_z)^T.$$

The 19 moments are defined as the following: $\delta\rho$ is the density fluctuation, e is the energy, ϵ is the energy square, $j_{x,y,z}$ are the momentum components of x, y and z directions, $q_{x,y,z}$ are the heat flux components, $3p_{xx}$ and p_{ww} are the trace of the strain rate tensor, p_{xy}, p_{yz}, and p_{zx} are the symmetric and traceless strain rate tensor, $3\pi_{xx}$ and π_{ww} are fourth order moments, and $m_{x,y,z}$ are third order moments. The diagonal collision matrix $\hat{S}$ is:

$$\hat{S} \equiv diag(0, s_1, s_2, 0, s_4, 0, s_4, 0, s_4, s_9, s_{10}, s_9, s_{10}, s_{13}, s_{13}, s_{13}, s_{16}, s_{16}, s_{16}, s_{16}).$$

The equilibrium moments of the non-conserved moments are functions of the local flow density and momentum as follows:

$$e^{eq} = -11\delta\rho + \frac{19}{\rho_0}\boldsymbol{j}.\boldsymbol{j} = -11\delta\rho + \frac{19}{\rho_0}(j_x^2 + j_y^2 + j_z^2), \tag{3.35}$$

$$\epsilon^{eq} = w_\epsilon \delta\rho + \frac{w_{\epsilon j}}{\rho_0}\boldsymbol{j}.\boldsymbol{j}, \tag{3.36}$$

$$q_x^{eq} = -\frac{2}{3}j_x, \qquad q_y^{eq} = -\frac{2}{3}j_y, \qquad q_z^{eq} = -\frac{2}{3}j_z, \tag{3.37}$$

$$p_{xx}^{eq} = \frac{1}{3\rho_0}[2j_x^2 - (j_y^2 + j_z^2)], \qquad p_{ww}^{eq} = \frac{1}{\rho_0}[j_y^2 - j_z^2], \tag{3.38}$$

$$p_{xy}^{eq} = \frac{1}{\rho_0}j_x j_y, \qquad p_{yz}^{eq} = \frac{1}{\rho_0}j_y j_z, \qquad p_{xz}^{eq} = \frac{1}{\rho_0}j_x j_z, \tag{3.39}$$

$$\pi_{xx}^{eq} = w_{xx}p_{xx}^{eq}, \qquad \pi_{ww}^{eq} = w_{xx}p_{ww}^{eq}, \tag{3.40}$$

$$m_x^{eq} = m_y^{eq} = m_z^{eq} = 0. \tag{3.41}$$

The parameters in the above equilibrium equation are chosen to optimize the linear stability of the model as follows: $w_\epsilon = w_{xx} = 0$ and $w_{\epsilon j} = -\frac{475}{63}$. It has been proven that the MRT model has superior advantages over the single relaxation time (SRT) method in the following perspective [107]:

- MRT has better stability characteristics because of the more freedom to tune the relaxation rate for different hydrodynamic and non-hydrodynamic moments.
- MRT has better accuracy characteristics. The small fluctuations in the spurious non-hydrodynamic moments may add non-physical contribution to the hydrodynamic moments.
- MRT model can simulate problems at a Reynolds number four times higher than with the SRT model .

3.8 Boundary Conditions

In LBM, the macroscopic hydrodynamic characteristics of the flow field are not directly computed; rather, the dynamics of the underlying microscopic world are simulated. In traditional CFD methods, the implementation of the hydrodynamic boundary condition is a straightforward task as these methods adopt the macroscopic variables for flow field dependence. The boundary conditions for the macroscopic world are normally defined by specifying the hydrodynamic properties (e.g. pressure, velocity, density, and temperature) at the flow boundaries. Defining the microscopic characteristics (e.g. number of particles and particles speed) at the flow boundaries is a very difficult or impossible task. For that reason, deriving the unknown microscopic elements from the known macroscopic properties is an essential process to achieve the accurate results. Many LBM boundary conditions have been developed and only boundary conditions used in this work will be presented.

3.8.1 Bounce Back Boundary Condition.

The bounce back (BB) boundary is used to define the unknown distribution functions at the solid (stationary or moving) boundaries. The implementation of this method is simple by considering an intuitive process for objects bouncing back from a solid wall when they collide with it. Figure 3.3 depicts this method in D2Q9 model where the unknown distributions (f_2, f_5, f_6) are replaced with the known distributions (f_4, f_7, f_8) at the solid wall as follows:

$$
\begin{aligned}
f_2 &= f_4, \\
f_5 &= f_7, \\
f_6 &= f_8.
\end{aligned}
$$

There are two ways to implement this boundary condition. First, by locating the solid wall at one lattice length far from the inner node (full-way BB). In this case the collision process for LBM will take place before bounce back. The second way is to locate the solid wall at half lattice length from the inner node (half-way BB). The collision process in this way is not performed for the boundary node which allows the bounce back process to be more natural and consistent with the LBM spatial discretization [102].

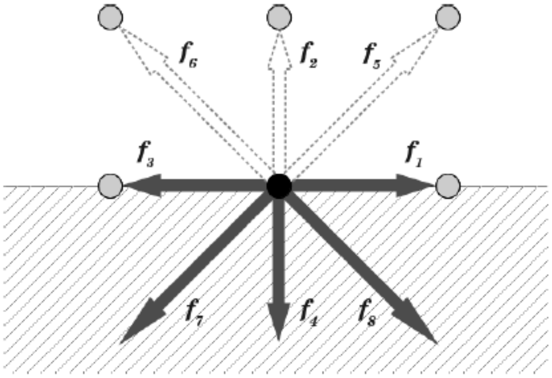


Figure 3.3: Full-way bounce back boundary condition.

3.8.2 Velocity Boundary Condition

The velocity, pressure and density have usually known values at the flow domain boundary. Zou and He [110] applied the idea of bounce-back boundary condition for the non-equilibrium part of the distribution function:

$$f_i^{(1)} = f_{opp(i)}^{(1)}, \tag{3.42}$$

where $f_i^{(1)}$ is the non-equilibrium distribution at c_i direction and $f_{opp(i)}^{(1)}$ is the non-equilibrium distribution at the opposite direction $c_{opp(i)} \equiv -c_i$. The equilibrium part of the distribution function is known at the inlet boundary (Equation 3.14). Then, substituting its value in Equation 3.42 leads to the inlet boundary condition in the following form [108]:

$$f_{\overline{\alpha}}(\mathbf{x}_b, t + \Delta t) = \tilde{f}_\alpha(\mathbf{x}_b, t) - 6 w_\alpha \rho_b \mathbf{u}_b \cdot \mathbf{c}_\alpha, \tag{3.43}$$

where the subscript b is symbol for the boundary values, $\tilde{f}_\alpha$ is the known after collision distribution, and $f_{\overline{\alpha}}$ is the unknown distribution entering the the flow domain.

3.8.3 Periodic Boundary Condition

The periodic boundary condition is used when the system boundary is at an infinite point or very large distance away where it becomes meaningful or impossible to specify in the computer code implementation. To illustrate this boundary condition, Figure 3.4 shows the a schematic for a flow domain where the periodic boundary can be applied at the north and south boundaries as follows:

$$f_7^N = f_7^S, \qquad f_4^N = f_4^S, \qquad f_8^N = f_8^S,$$

$$f_5^S = f_5^N, \qquad f_2^S = f_2^N, \qquad f_6^S = f_6^N,$$

where the subscripts N and S represent the north and south walls, respectively.

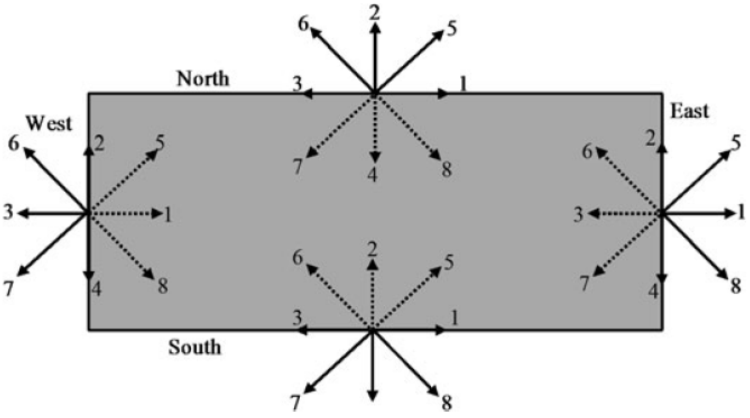


Figure 3.4: Periodic boundary condition.

3.8.4 Fully Developed Boundary Condition

The fully developed boundary condition is applied at the boundary where there is no spatial derivatives for the flow variables. The procedure for this boundary condition is to copy all distribution function values from the nodes preceding the final nodes and to past them to the last nodes.

3.8.5 Curved Boundary Condition

In order to accurately describe the bounce back distribution functions on curved boundaries, the traditional BB boundary condition (Section 3.8.1) is modified to exactly find the location of the wall nodes [36]. The wall nodes distance (q) from the adjacent fluid nodes in the BB boundary condition equals half the LBM link distance ($q = 0.5$). However, this distance may be larger or less than 0.5 when the curved boundary exists. The interpolation method for the bounce back (IBB) distributions proposed by Bouzidi et al. [6] was adopted in this study. As shown in Figure 3.5, the distance from the wall to the adjacent fluid node is $q\delta$. The distance between two fluid nodes is δ, and the fluid particles traverse this distance when they move from fluid to fluid nodes and from fluid to wall to fluid (BB boundary condition) nodes within one time step. The main idea of IBB is to keep the distance travelled by the fluid particles at the curved boundary as same as δ as follows. Assume $\delta = 1$ (i.e. dimensionless LBM link length), for the case of $q < 0.5$, a fictitious node r_B is assumed (Figure 3.5 (a)) such that:

$$\| r_B - r_A \| = 1 - 2q. \tag{3.44}$$

The post collision distribution functions (f_i^*) at r_B can be evaluated by applying a linear interpolation between r_A and $r_A - c_i$ as:

$$f_{i'}(r_A, t+1) = f_i^*(r_B, t) = 2q f_i^*(r_A, t) + (1 - 2q) f_i^*(r_A - c_i, t), \tag{3.45}$$

where i' represents the opposite velocity direction of i (i.e. $c_{i'} = -c_i$). For the case of $q \geq 0.5$, node r_B is located between the wall node and the adjacent fluid node (Figure 3.5 (b)) such that the fluid particles leaving r_A will collide the will and bounce back to r_B within one time step, and the distance between r_A and r_B becomes:

$$\| r_B - r_A \| = 2q - 1. \tag{3.46}$$

Fluid particles leaving r_A in the opposite direction ($f_{i'}$) will reach node $r_A - c_i$ at the next time step, and can be estimated as follows:

$$\begin{aligned} f_{i'}(r_A, t+1) &= \frac{(2q-1)}{2q} f_{i'}(r_A - c_i, t+1) + \frac{1}{2q} f_{i'}(r_B, t+1) \\ &= \frac{(2q-1)}{2q} f_{i'}^*(r_A, t) + \frac{1}{2q} f_i^*(r_A, t). \end{aligned} \tag{3.47}$$

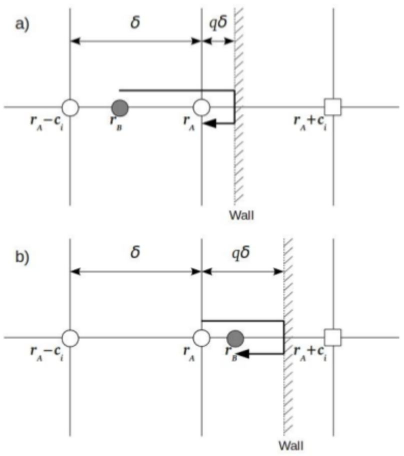


Figure 3.5: Curved boundary condition; (a): $q < 0.5$. (b): $q > 0.5$.

3.9 LBM-MRT Local Grid Refinement

The lattice Boltzmann method is based on uniform grid discretization which makes this method inadequate for simulations with different resolution requirements (e.g. turbulent wall bounded flow). One option to overcome this problem is applying hierarchical grid refinement techniques, which preserve the uniformity of each grid level and allow the different levels to exchange their information (e.g. distribution functions) by applying a suitable transformation and interpolation techniques. The grid refinement method proposed by Filippova and Hänel [28] is adopted in this work. The lattice speed (c) in this method is kept constant, while the grid size (δ_x) and the time step (δ_t) are modified by a refinement factor (n) which depends on the required grid refinement level. The most important feature in the grid refinement techniques is retaining the same flow characteristics (e.g. Reynolds number and viscosity) for the different grids (i.e. coarse and fine grid). In LBM-MRT, the viscosity is a function of the relaxation parameter s_9 as:

$$\nu = \left(\frac{1}{s_{9,c}} - \frac{1}{2}\right)\frac{\delta_x c}{3}. \tag{3.48}$$

The relaxation parameter $s_{9,c}$ in Equation 3.48 is for the coarse uniform original LBM grid, where the subscript c refers to the coarse grid and the subscript f will be used for the fine grid. The modified relaxation parameter for the fine grid ($s_{9,f}$) will be a function of the refinement factor as follows:

$$s_{9,f} = \frac{2}{1 + n\left(2/s_{9,c} - 1\right)}. \tag{3.49}$$

In addition, the hydrodynamic variables and their derivatives should be continuous between the coarse and fine grids. To ensure continuity, the non-equilibrium part of the post-collision distribution functions (f^*) moving from the fine to coarse and from coarse to fine grids must be scaled as follows:

$$f_c^* = f_f^{eq} + \left(f_f^* - f_f^{eq}\right) \frac{(1 - s_{9,c})\, s_{9,f}^n}{s_{9,c}\left(1 - s_{9,f}\right)}, \tag{3.50}$$

$$f_f^* = f_c^{eq} + \left(f_c^* - f_c^{eq}\right) \frac{s_{9,c}\left(1 - s_{9,f}\right)}{(1 - s_{9,c})\, s_{9,f}^n}. \tag{3.51}$$

Equations 3.50 and 3.51 scaling the distribution functions in the velocity space by using s_9. In order to keep a consistent MRT algorithm in the simulation, the latter equations can be used to scale moments in the moment space, and s_9 should be replaced by different relaxation parameters (i.e. the elements in $\hat{S}$) according to their corresponding moments (Section 3.7).

The refinement factor is set to 2 in this work which means that the time needed to pass each refinement level will be half the time in the coarse grid. Consequently, the LBM algorithm in the fine grids should be marched by two time steps for each time step in the coarse grid. As shown in Figure 3.6, the internal nodes (white nodes) in the fine grid receive the required information (e.g. distribution functions) from the external nodes (red nodes) of the fine grid. For the external nodes, spatial bilinear and trilinear interpolations can be used for the 2D and 3D problems, respectively, to find the distribution function values in the external nodes (fine grid) from the adjacent nodes in the coarse grid. Linear interpolation can also be used for the temporal interpolation [30]. The spatial interpolation is utilized to find the distribution functions at node a (Figure 3.6) in the fine grid from the nodes in the coarse grid (A, B, C, and D) at time $t = t_0$ as follows:

$$f_{t_0}^a = \frac{1}{16}\left(9 f_{t_0}^D + 3 f_{t_0}^B + 3 f_{t_0}^C + f_{t_0}^A\right). \tag{3.52}$$

At time t_0+1, the spatial interpolation (Equation 3.52) is used to find the distribution functions at a ($f_{t_0+1}^a$), and the temporal linear interpolation can be used to find the distribution functions at time $t = t_0 + \frac{1}{2}$ at node a:

$$f_{t_0+\frac{1}{2}}^a = \frac{1}{2}\left(f_{t_0}^a + f_{t_0+1}^a\right) \tag{3.53}$$

At the end of time $t_0 + 1$, the LBM algorithm (streaming and collision) was applied two times in the fine grid, and the distribution functions on the fine grid nodes are averaged spatially to find the distributions in the corresponding coarse grid nodes.

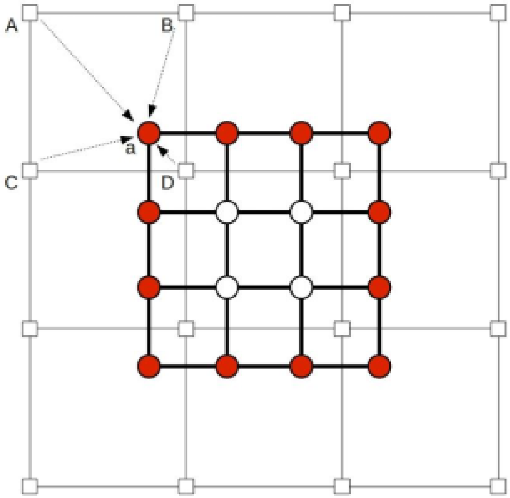


Figure 3.6: Schematic of the grid refinement by a factor of 2.

3.10 Large Eddy Simulation Subgrid Model

To model high Reynolds number flows with the regular LBM, some actions should be taken [64]:

- increasing the characteristic inlet velocity. This action may increase the Mach number and the error related to it in the discrete equilibrium distribution function.
- decreasing the relaxation time τ value (viscosity). This action causes the numerical scheme to be unstable.
- increasing the lattice number. This action increases the computational resources required for the calculation.

To overcome the obstacles related to the aforementioned actions, Large Eddy Simulation (LES) can be adopted for the (unresolved) subgrid scale (SGS) eddy modelling, while eddies which are resolved by the grid scale will be simulated directly by LBM. The LES method utilizes filtering operation (with spatial filter G) for a physical variable (w). The new filtered variable can be defined as:

$$\overline{w(r)} = \int w(r')G(r - r')dr'. \tag{3.54}$$

The SGS turbulent flow effects can be incorporated in the filtered grid scale flow by defining the SGS viscosity (eddy viscosity) ν_t and add its value to the fluid kinematic viscosity ν_0 to produce a total kinematic viscosity ν_{total} as follows:

$$\nu_{total} = \nu_0 + \nu_t. \tag{3.55}$$

Smogorinsky eddy viscosity model defines the eddy viscosity as a function of the local fluid deformation:

$$\nu_t = (C_s\Delta_x)^2|\overline{S}|, \tag{3.56}$$

where $|\overline{S}| = \sqrt{2S_{\alpha\beta}S_{\alpha\beta}}$ is the strain rate tensor magnitude, $S_{\alpha\beta}$ is the strain rate tensor, C_s is Smagorinski constant, and Δ_x is the filter length scale and in LBM-LES can be taken as the same lattice length ($\Delta_x = \Delta x_{LBM} = 1$). In LBM, the deformation rate tensor can be obtained by local calculations from the non-equilibrium part (i.e. $f_i - f_i^{eq}$) of the distribution function:

$$S_{\alpha\beta} = \sum_{i=1}^{q-1}(f_i - f_i^{eq})c_{i_\alpha}c_{i_\beta}. \tag{3.57}$$

3.11 LBM-CA Model

The traditional LBM-CA model was studied and validated by a comprehensive work of Masselot [64]. This work studied obstacles on near-mountains roads and railways which are generated by snow falls or (and) snowdrifts. The author stated that the effect of snowdrifts is more hazardous than snow falls when they are deposited on the roads. Snow particles are transported by wind following three different modes:

- Creeping: particles are rolling on the ground.
- Saltation: particles are ejected and fly vertically along a parabolic trajectory.
- Suspension: during the saltation process, particles can be taken in large scale eddies and fly for longer distance.

This work (Masselot [64]) simulates the natural phenomena that affect snow particles movement by using the BGK (single relaxation time) LBM to models the fluid (wind) flow, and a probabilistic algorithm to describe the solid (snow) particles movement. Classical approaches that model solid particles behaviour (e.g. position and velocity) under the action of a fluid flow can be split into two methods which are:

- The Eulerian method: On each cell of the computational domain, all parameters (e.g. density and velocity) of the solid phase are spatially averaged. The transport of the particle phase is then governed by state equations, describing the phase exchange between adjacent cells.

- The Lagrangian method: This approach tracks each particle in a set of particles individually by considering the interaction between these particles and the surrounding fluid.

The cellular automata approach for the solid phase lies between these methods, where the flow domain is decomposed in cells and the solid particles are distributed within these cells. There is no information of exactly where a particle is located inside a cell. On the other hand, particles can be tracked individually inside the domain in the Lagrangian way, but the tracking will not recognize the identity of each particle.

Solid particles at a site are represented as quantities evolving synchronously and discretely over the same lattice. At time t, on site $\mathbf{r}$ (Figure 3.8), the number of particles travelling with velocity $\mathbf{c}_i$ is noted $p_i(\mathbf{r}, t)$, and the number of particles remaining in the site is $p_0(\mathbf{r}, t)$. Contrary to the LBM approach, p_i are integer values, and the particles are represented as discrete variables. The particle phase local density is defined as:

$$\rho_p = \sum_{q}^{i=0} p_i(\mathbf{r}, t), \tag{3.58}$$

and the particle flux:

$$\mathbf{J}_p(\mathbf{r}, t) = \sum_{q}^{i=0} p_i(\mathbf{r}, t)\mathbf{c}_i, \tag{3.59}$$

where q is the number of lattice cell neighbours. This number can be different than the number of lattice links in LBM.

3.11.1 Particle Motion

For a one-dimensional lattice case (Figure 3.7), a particle position at time $t + 1$ will be $\mathbf{r}+(\mathbf{u}_f+\mathbf{u}_{fall})$, where $\mathbf{u}_f$ is the fluid local velocity, and the parameter $\mathbf{u}_{fall}$ takes into account the gravity effect on the particle. Both $\mathbf{u}_f$ and $\mathbf{u}_{fall}$ are normalized (i.e. non dimensional) by the fluid characteristic velocity. Then the following probabilistic algorithm is proposed:

- The direction of motion depends on the sign of $\mathbf{u}_f + \mathbf{u}_{fall}$.
- The particle moves with probability ξ_x, and rests (does not move) with probability $(1 - \xi_x)$, where $\xi_x =\mid \mathbf{u}_f + \mathbf{u}_{fall} \mid$.

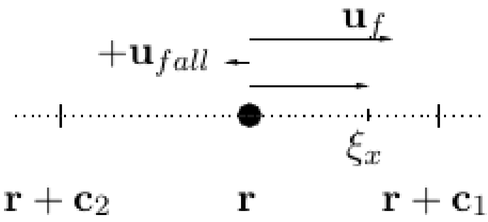


Figure 3.7: Particle movement over a one-dimensional lattice.

The algorithm is straightforward and the next position for the particle will be the right one when $\mathbf{u}_f \gg \mathbf{u}_{fall}$ as shown in Figure 3.7.

For a two-dimensional lattice (Figure 3.8), the previous technique (i.e. the one-dimensional algorithm) can be extended. First, the quadrant where the a particle is leading towards should be defined (e.g. north, north-east, south, etc.). This decision is based on the sign of the two components (x and y) of $\mathbf{u}_f + \mathbf{u}_{fall}$. The probabilities of the vertical and horizontal motions of the particle can be defined as: $\xi_x =\mid \mathbf{u}_{f_x} + \mathbf{u}_{fall_x} \mid$ and $\xi_y =\mid \mathbf{u}_{f_y} + \mathbf{u}_{fall_y} \mid$, where the subscripts x and y are for the horizontal and vertical components.

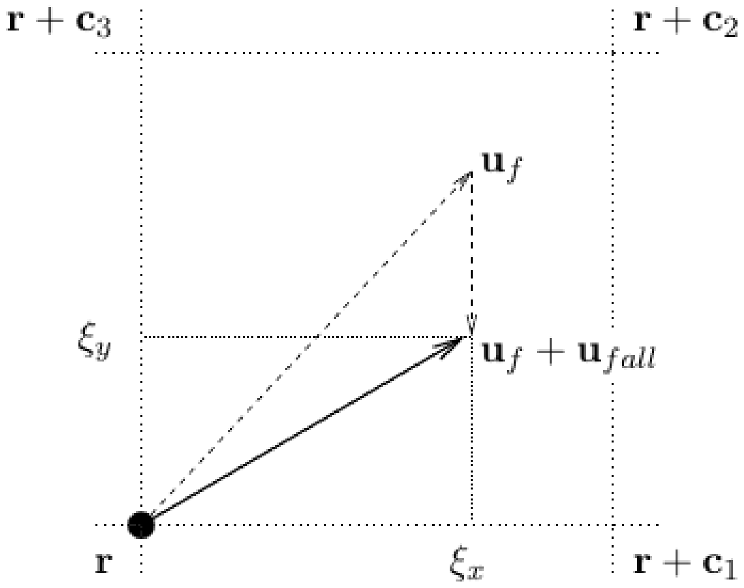


Figure 3.8: Particle movement on a two-dimensional square lattice.

Therefore, the probabilities of resting or reaching each of the three neighbours in a specific quadrant (Figure 3.8) are:

- moving along the diagonal $\xi_x\xi_y$
- moving to the x-direction (east) $\xi_x(1-\xi_y)$
- moving to the y-direction (north) $\xi_y(1-\xi_x)$
- resting $(1-\xi_x)(1-\xi_y)$

For a three-dimensional lattice (Figure 3.9), the two-dimensional algorithm is directly extended. The three components of $\mathbf{u}_f + \mathbf{u}_{fall}$ give three probabilities of moving in $x-$, $y-$, or $z-$direction, which are ξ_x, ξ_y, and ξ_z, respectively. The moving probabilities are:

- $P_{xyz} = \xi_x\xi_y\xi_z$
- $P_{yz} = \xi_y\xi_z - P_{xyz}$
- $P_{xz} = \xi_x\xi_z - P_{xyz}$

- $P_{xy} = \xi_x \xi_y - P_{xyz}$
- $P_x = \xi_x - P_{xy} - P_{xz} - P_{xyz}$
- $P_y = \xi_y - P_{xy} - P_{yz} - P_{xyz}$
- $P_z = \xi_y - P_{xz} - P_{yz} - P_{xyz}$
- $P_0 = (1 - \xi_x)(1 - \xi_y)(1 - \xi_z)$

In the above algorithm, for example, P_0 is the probability of resting, and P_{xy} is the probability of moving along the x and y directions (i.e. $\mathbf{r} + \mathbf{c}_7$ in Figure 3.9).

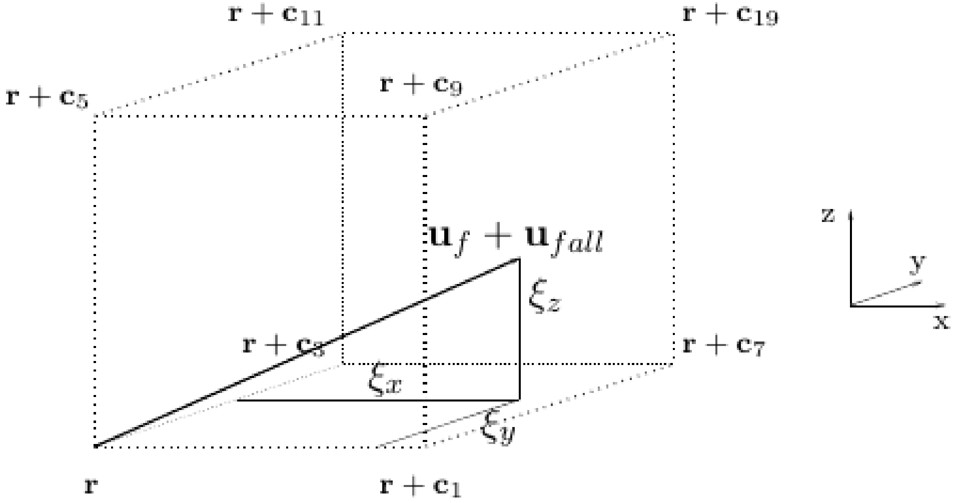


Figure 3.9: Particle movement on a three-dimensional lattice.

Chapter 4

Fluid Flow LBM Code Validation

The fluid flow simulation was carried out by employing a three-dimensional LBM in-house code. This code utilized the D3Q19 LBM scheme with the LES model (i.e. Smagorinsky subgrid-scale model) for the turbulence in the flow. To validate the code, different numerical simulations were performed for two benchmark test cases. Flow in a free turbulent square jet at Re=25000 was first examined, and the results were compared with the experimental work results by Ghasemi et al. [35]. The second test case is the round turbulent impinging jet where the simulation results of this case were compared with the LES numerical simulation ones by Hällqvist [38], and with the experimental work results by Cooper et al. [15] and Geers et al. [33].

4.1 Numerical Aspect (Non-dimensional Units in LBM)

Before going through the simulation of test cases and their results, the unit conversion in LBM will be reviewed with a practical test case. The LBM quantities have non-dimensional units (LBM units), while the physical problems have dimensional units, e.g. length (m), mass (kg), and time (s). Unit conversion factors are needed to provide a good understanding of the non-dimensional LBM units and their equivalent physical units.

To give a better insight for these conversion factors, a simulation case will be examined to find the unit conversion factors. The case study is the experimental work of Burwash et al. [7], where this work has the following dimensional characteristics:

- Air density, $\rho_a = 1.14\ kg/m^3$.

- Air dynamic viscosity, $\mu_a = 1.824 \cdot 10^{-5}\ kg/m \cdot s$.
- Nozzle diameter, $D = 15\ mm$.
- Nozzle exit velocity, $u = 10.5\ m/s$.
- Solid particle diameter, $d_p = 5\ \mu m$.
- Solid particle density, $\rho_p = 1050\ kg/m^3$.

In the LBM simulation, some of the dimensionless variables should be selected first, and the others can be calculated; the selected simulation variables are:

- Nozzle diameter, $D_{lbm} = 70$ LBM nodes.
- Nozzle exit velocity, $u_{lbm} = 0.1$ (dimensionless).
- Air density, $\rho_{lbm} = 1$ (dimensionless).

The non-dimensional parameters (e.g. Reynolds and Stokes numbers) should be invariant for both physical and numerical (non-dimensional) units, e.g. $Re = Re_{lbm} = 10000$, where the subscript *lbm* is used for the LBM dimensionless quantities. The following conversion equation can be used to calculate the conversion factors:

$$Q = Q_{lbm} \times C_Q, \tag{4.1}$$

where Q, Q_{lbm} and C_Q are the physical quantity, dimensionless LBM quantity, and the conversion factor for the quantity Q, respectively.

4.1.1 Length Conversion

The nozzle diameter is considered as a characteristic length scale, therefore:

$$D = D_{lbm} \times C_L \qquad \Longrightarrow C_L = D/D_{lbm},$$

$$C_L = \frac{0.015(m)}{70} = 0.000214\ (m) = 0.214\ (mm). \tag{4.2}$$

Equation 4.2 states that each LBM lattice length is equivalent to 0.000214 m (0.214 mm).

4.1.2 Velocity Conversion

$$u = u_{lbm} \times C_U \qquad \Longrightarrow C_U = u/u_{lbm},$$

$$C_U = \frac{10.5\ (m/s)}{0.1} = 105\ (m/s). \tag{4.3}$$

4.1.3 Density Conversion

$$\rho = \rho_{lbm} \times C_\rho \qquad \Longrightarrow C_\rho = \rho/\rho_{lbm},$$

$$C_\rho = \frac{1.14(kg/m^3)}{1} = 1.14\ (kg/m^3). \tag{4.4}$$

4.1.4 Kinematic Viscosity Conversion

The LBM kinematic viscosity can be calculated from the LBM Reynolds number as follows:

$$\nu_{lbm} = \frac{u_{lbm} \cdot D_{lbm}}{Re_{lbm}}. \tag{4.5}$$

Equation 4.5 can be used to calculate the LBM viscosity, or the invariant Reynolds number can be utilized to derive the viscosity conversion, as follows:

$$Re = Re_{lbm} \qquad \Longrightarrow \frac{u \cdot D}{\nu_a} = \frac{u_{lbm} \cdot D_{lbm}}{\nu_{lbm}}, \tag{4.6}$$

yielding:

$$\frac{\nu_a}{\nu_{lbm}} = \frac{u}{u_{lbm}} \frac{D}{D_{lbm}} \Longrightarrow C_\nu = C_U C_L = 105\ (m/s) \cdot 0.000214\ (m) = 0.02247\ (m^2/s). \tag{4.7}$$

4.1.5 Time Conversion

In the LBM, lattice length and time step have values of 1 for convenient calculation, thus:

$$\Delta x = C_L, \qquad \Delta t = C_t.$$

The physical kinematic viscosity in the Navier-Stokes equation is connected to the LBM relaxation time by the following formula (Chapman-Enskog analysis):

$$\nu = \left(\tau - \frac{1}{2}\right) c_s^2 \Delta t, \tag{4.8}$$

where the squared speed of sound c_s^2 in LBM is:

$$c_s^2 = \frac{1}{3}\frac{\Delta x^2}{\Delta t^2}. \tag{4.9}$$

Substituting Equation 4.9 in Equation 4.8 leads to:

$$\nu = \frac{\tau - \frac{1}{2}}{3} \cdot \frac{\Delta x^2}{\Delta t^2} \cdot \Delta t, \quad \Longrightarrow \nu = \underbrace{\frac{\tau - \frac{1}{2}}{3}}_{\nu_{lbm}} \cdot \frac{C_L^2}{C_t}, \tag{4.10}$$

and finally:

$$C_t = \nu_{lbm} \cdot \frac{C_L^2}{\nu} = \frac{1}{C_\nu} \cdot C_L^2 = \frac{0.000214^2}{0.02247} = 2 \cdot 10^{-6}\ (s). \tag{4.11}$$

4.2 Results and Discussions

4.2.1 Inlet Jet Condition

For the jet flow simulations in this study, a top-hat velocity profile was used for the inlet jet condition. Turbulent fluctuations in three components of velocity are added at each time step by generating a random number with a rang of $(0-10\%)$ from the mean velocity value for the x-direction (axial direction), and with a range of $(0 - 5\%)$ for the y and z-direction (radial and spanwise directions), respectively.

4.2.2 Free Square Jet

Numerical simulations are performed with four different combinations of inflow and side boundary conditions. The computational domain for the free square jet is depicted in Figure 4.1. Table 4.1 shows these combinations for each simulation case. The main difference between the first two cases (1 and 2) and the second two cases (3 and 4) is in the side wall boundary condition, where the no-slip solid wall (BB boundary condition) and the periodic boundary condition were applied for these cases. The inlet jet perturbation was also changed (i.e. with and without) in the simulation.

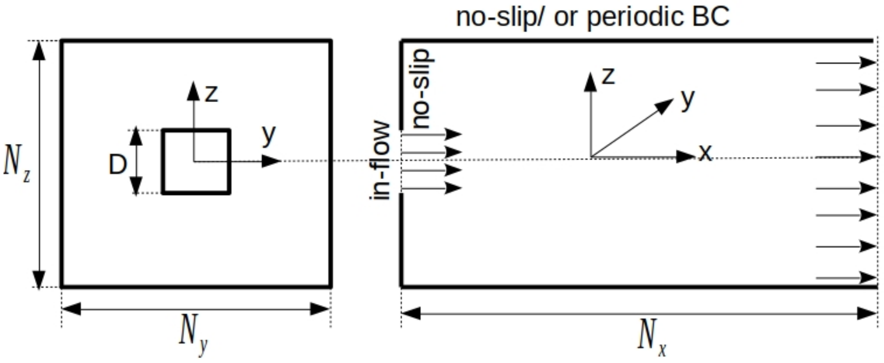


Figure 4.1: Computational domain of the square jet.

Table 4.1: Square free jet cases.

Case Characteristics	Case1	Case2	Case3	Case4
Jet width D (N_{lbm})	20	20	20	20
N_x	$25D$	$25D$	$25D$	$25D$
N_y	$7D$	$7D$	$7D$	$7D$
N_z	$7D$	$7D$	$7D$	$7D$
Wall BC	NoSlip	NoSlip	Periodic	Periodic
Re	25000	25000	25000	25000
Jet speed U_j	0.1	0.1	0.1	0.1
Jet perturbation	0	1	0	1
Smagorinsky constant C_s	0.1	0.1	0.1	0.1
LBM time steps	300000	300000	300000	300000

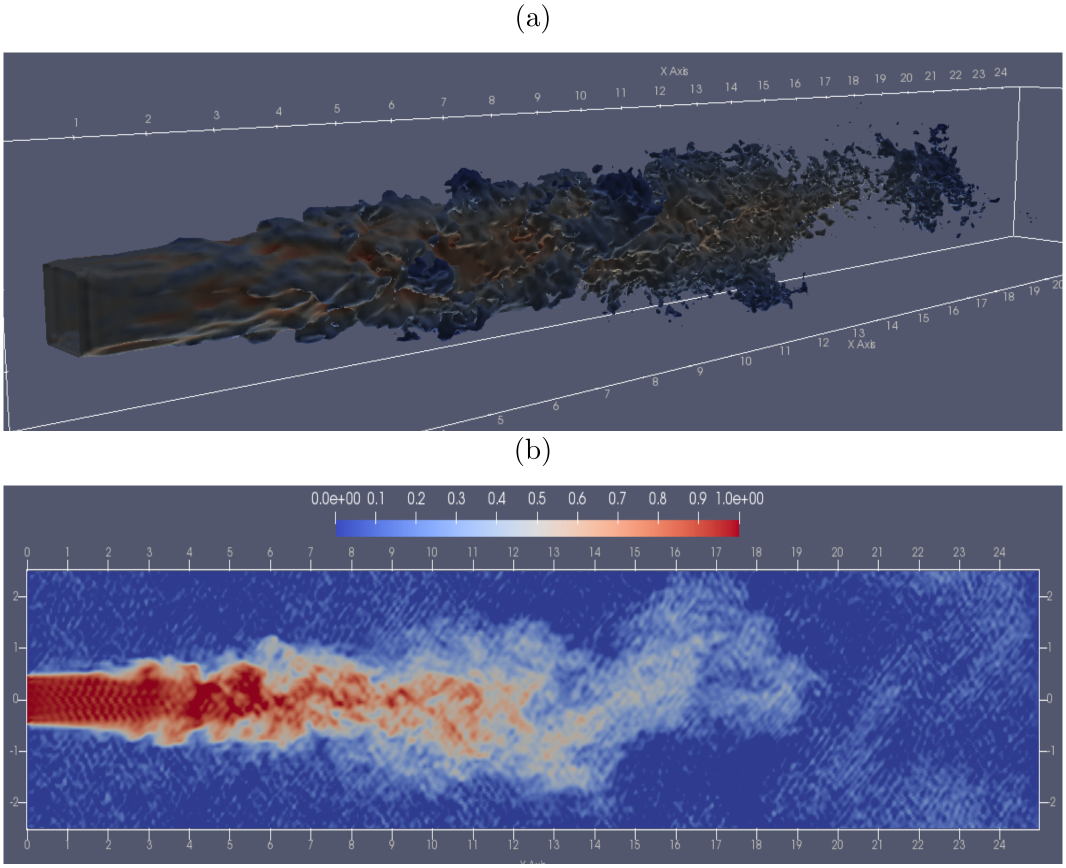


Figure 4.2: (a)Instantaneous streamwise velocity contour, (b) instantaneous streamwise velocity.

Figure 4.3 demonstrates the spanwise distribution of the normalized mean streamwise velocity profiles (U/U_j) at different streamwise distances (x/D) from the jet exit. The central portion in this figure ($y/D = 0$) illustrates the potential core of the jet where the velocity is at its maximum value or equal to the jet exit velocity. The jet core velocity decreases with increasing x/D and this behaviour is expected for this portion. The potential core is defined in the near field of the jet (i.e. $0 < x/D < 7$) where the streamwise centerline velocity is almost constant. This region is more obvious for contraction nozzles where the shear layer affect the the central velocity after the core region. In computational perspective, it is not easy to perform smooth jet contraction and that leads to the velocity decreasing in this portion in this simulations. The shear layer causes the the core velocity decreasing because of the development in turbulence. The streamwise velocity decreases when it moves in the radial direction toward the shear layer. The top hat velocity profile is retained in the region close to the jet exit (Figure 4.3 $x/D = 1$), and the Gaussian distribution is dominated in the region far from the jet exit as shown in Figure 4.3 at $x/D = 4, 5$. Comparison with the experimental results (Ghasemi et al. [35]) shows good agreement; however, the numerical result profiles show non-smooth curves with slight deviation compared with the experiment ones. The reasons for these two differences may be attributed to the spanwise distance from the shear layer in the solution domain, where this distance may be not sufficiently large to vanish the effects of the side boundary on the flow and probably the averaging time is not large enough to produce smooth profile. Comparison between the four different simulation cases shows that there are no effects of the inlet and the wall boundary conditions on the obtained results from these cases.

Figure 4.4 shows the normalized streamwise turbulence intensity distribution (u_{rms}/U_j) at different streamwise distances from the jet exit. At the streamwise distance $x/D = 1$ and for $y/D = 0$ (i.e the core region), the maximum values of u_{rms}/U_j for cases 1 and 2 are less than 0.03 because this region is far from the effect of the shear layer of the jet. The maximum values of u_{rms}/U_j are reached at distance $y/D \simeq 0.5$ in the shear layer region. The results in the downstream direction show increases in the turbulence intensity because of the the ambient fluid entrained into the core zone. At $x/D = 1$, the u_{rms}/U_j values at the core zone ($y/D = 0$) for cases 3 and 4 are more than the values for cases 1 and 2. The periodic boundary condition for the former cases causes spreading in the u_{rms}/U_j profiles at the side boundaries and that push the profiles up in the core zone.

Figure 4.5 depicts the spanwise turbulence intensity profiles. The v_{rms}/U_j is strongly affected by the shear layer and the side boundary. At $x/D = 1$, in the core zone, the values of v_{rms}/U_j are very low for cases 1 and 2 because of the weak influence of the shear layer on the core zone. However, cases 3 and 4 exhibit higher values of spanwise turbulence intensity in the core zone resulting from the periodic boundary condition at the sides. The no-slip boundary condition used in cases 1 and 2 damps the spanwise fluctuations at the side wall. In contrast, in cases 3 and 4 the periodic boundary allows for more fluctuations at the side wall and consequently these fluctuation are penetrated into the jet zones.

Reynolds shear stress profiles ($\overline{u'v'}/U_j^2$) are shown in Figure 4.6. Close to the jet exit (i.e. $x/D = 1$), the Reynolds stress at $y/D = -0.5$ (shear layer) exhibits sharp peak with minus value. This minus value is attributed to the opposite direction for both u' and v' at the jet shear layer. The jet core is less affected by the vortical eddies at the shear layer and has a zero Reynolds stress value. As the flow move in the downstream direction, the vortices penetrate to the core zone and spread the Reynolds stress profile, and the zero Reynolds stress value is retained in the centerline zone.

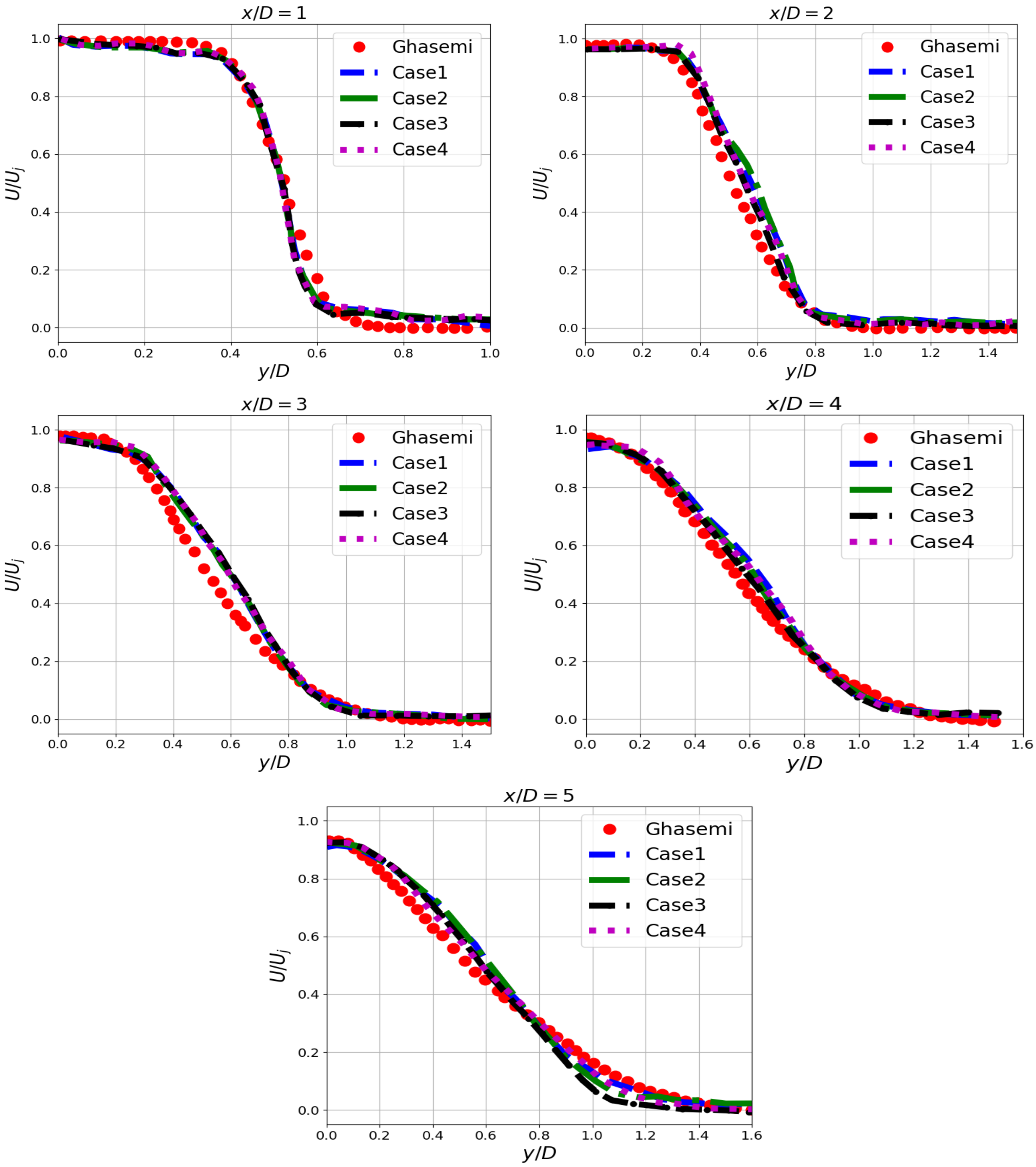


Figure 4.3: Spanwise distribution of U/U_j at different streamwise distances.

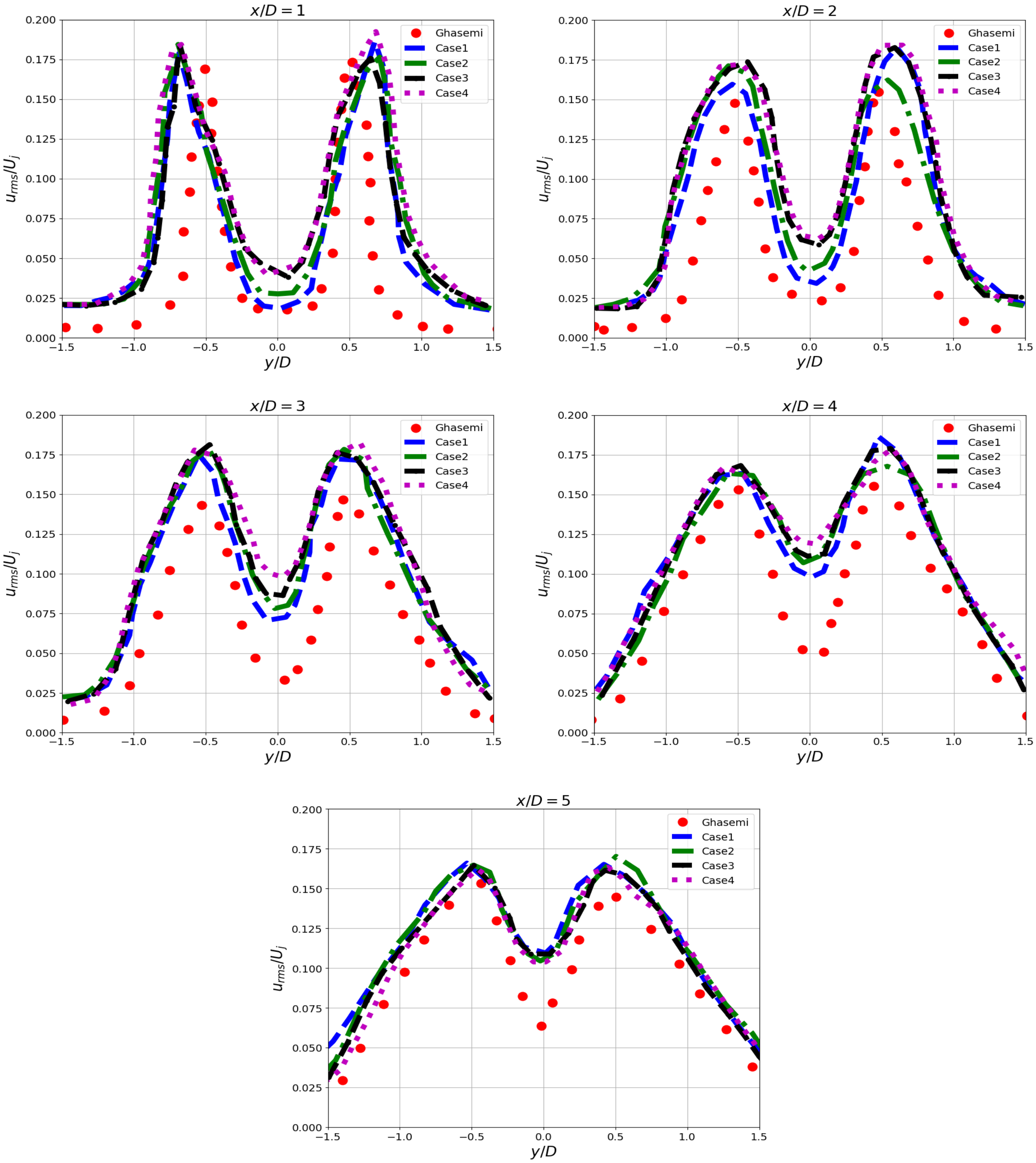


Figure 4.4: Spanwise distribution of u_{rms}/U_j at different streamwise distances.

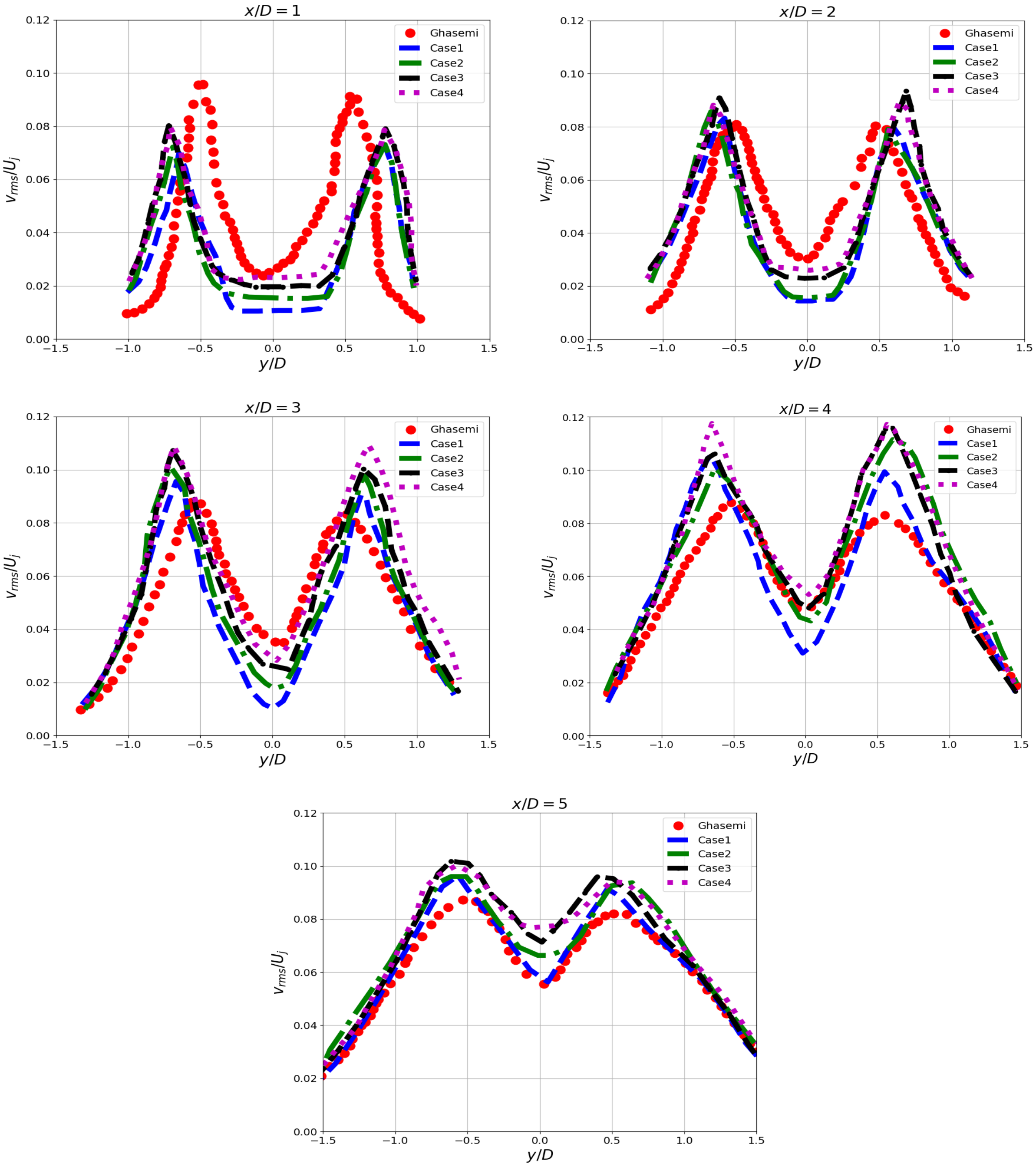


Figure 4.5: Spanwise distribution of v_{rms}/U_j at different streamwise distances.

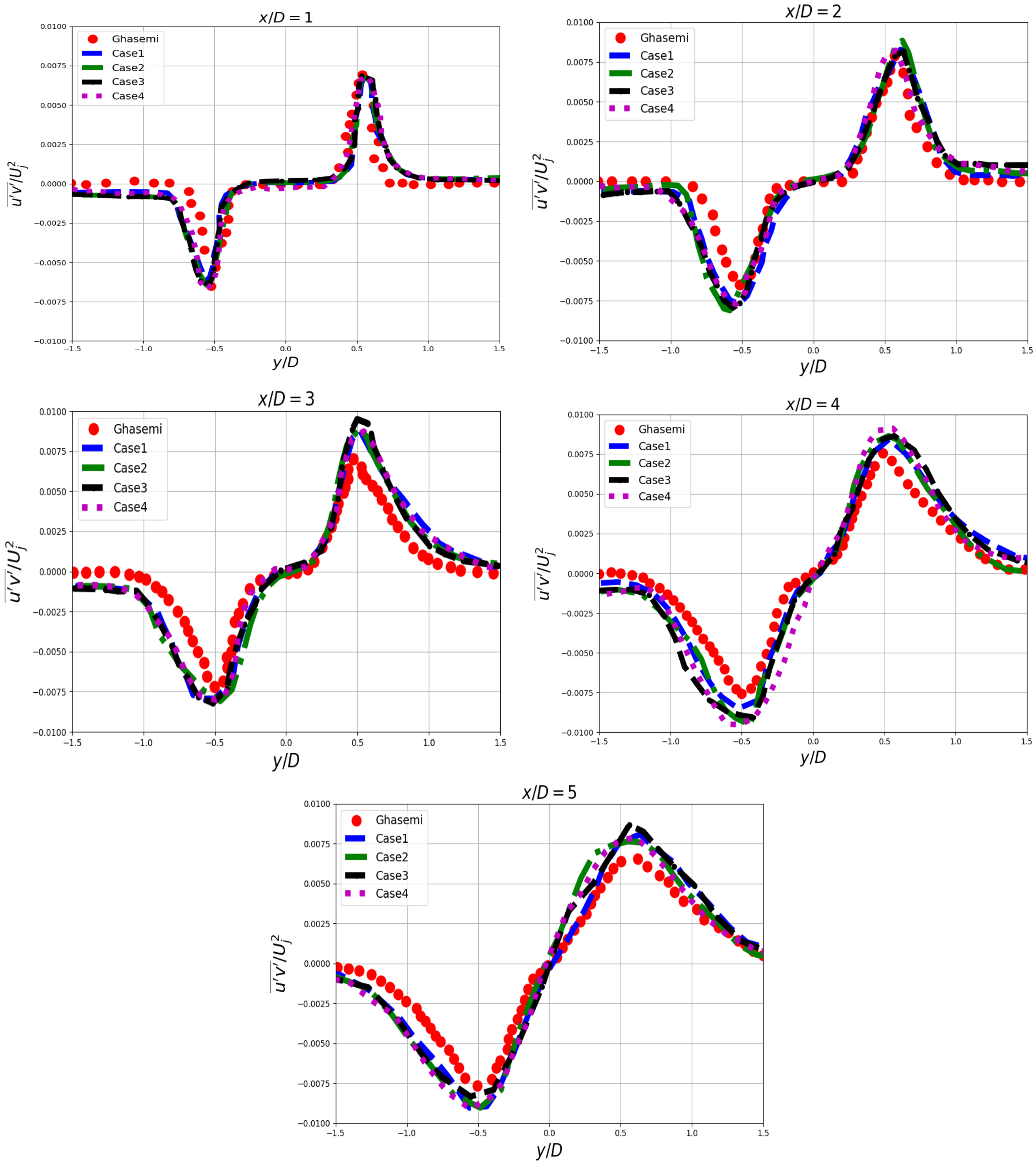


Figure 4.6: Lateral distribution of $\overline{u'v'}/U_j^2$ at a: $x/D = 1$; b: $x/D = 2$; c: $x/D = 3$; d: $x/D = 4$; e: $x/D = 5$.

4.2.3 Circular Impinging Jet

The impinging jet results in this simulation were compared with the numerical ones by Hällqvist [38] and with the experimental works results by Cooper et al. [15] and Geers et al. [33]. Figure 4.7 shows a schematic view of the round impinging jet. The impingement plate to the jet exit distance was chosen to be constant at $L/D = 2$.

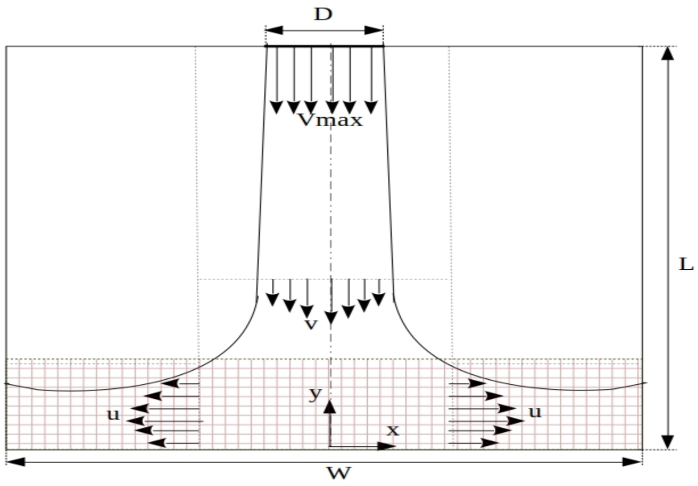


Figure 4.7: Computational domain of the circular impinging jet.

Computational grid

The computational scheme and the grid resolution effects have been evaluated for different test cases where the grid resolution was changed. For this case of the impinging jet (i.e. $L/D = 2$), four different (coarse) grids were used: $500 * 100 * 500$ (case1), $600 * 120 * 600$ (case2), $700 * 140 * 700$ (case3), and $800 * 160 * 800$ (case4). The previous grid resolutions are the coarse grids and all these grids were refined in the near wall region (i.e. high velocity gradient). Table 4.2 shows the impinging jet characteristics for the different grids. The viscous length-scale (y^+) was used in this study to estimate the grid spatial resolution adequacy [38]. It should be noted that this estimation is not accurate enough because the near wall flow is not fully developed, however the value of y^+ is determined in a range for the first grid line adjacent to the wall. For the different test cases, the ranges of y^+ are: $y^+ \leq 7.2$ (Case1), $y^+ \leq 5$ (Case2), $y^+ \leq 3.8$ (Case3), and $y^+ \leq 1.6$ (Case4). Coarser grid for the $L/D = 2$ case can reduce the range of y^+ and increase the computational cost (e.g. $y^+ \leq 1.2$ for $D = 120$). The characteristics of Case4 grid will be considered in this section results to keep a reasonable computational cost (e.g. available GPU memory), and to provide a dimensional consistency for the different L/D cases (e.g. $L/D = 2, 4, 6$) in the next results (Section 5.3.3). Table 4.3 shows the higher resolution test case (i.e. Case4 in Table 4.3) characteristics in the LBM non-dimensional units.

Table 4.2: Impinging jet characteristics for different cases.

Case Characteristics	Case1	Case2	Case3	Case4
Jet diameter D (Nodes)	50	60	70	80
N_y	$2N$	$2N$	$2N$	$2N$
N_x	$10N$	$10N$	$10N$	$10N$
N_z	$10N$	$10N$	$10N$	$10N$
Re	20000	20000	20000	20000
Uj	0.04	0.04	0.04	0.04
Jet perturbation	1	1	1	1
C_s	0.1	0.1	0.1	0.1
Time steps	400000	400000	400000	400000
Grid refinement level	2	2	2	2
Refinement height (y/D)	0.15	0.15	0.15	0.15

Errors in LBM scheme

The LBM parameters for a specific problem should be chosen carefully to ensure stable and accurate solution. In general, there are several errors related to the LBM simulation [51]:

- The spatial discretization error which is of order Δx^2.
- The temporal discretization error which is of order Δt^2.
- The compressibility error which is $\propto Ma^2 \propto u_{lbm}^2$. The LBM velocity (u_{lbm}) decreases with increasing the velocity conversion factor ($C_U = \frac{C_L}{C_t}$), then this error is $\propto \frac{\Delta t^2}{\Delta x^2}$.

Increasing the lattice resolution (i.e. decreasing the spatial discretization error) will grow the compressibility error, and decreasing the time step does not decrease the spatial error.

Table 4.3: Impinging jet (Case4) characteristics.

Case Characteristics	Case4
Jet diameter	80
Axial dimension (coarse grid)	160
Spanwise dimension (coarse grid)	800
Grid size Δx (coarse grid)	0.0125
Time step Δt (coarse grid)	0.0005
Grid size Δx (fine grid)	0.00625
Time step Δt (fine grid)	0.00025
Inlet boundary condition	Velocity boundary condition (Section 3.8.2)
Outlet boundary condition	Fully developed boundary condition (Section 3.8.4)
Wall boundary condition	Bounce back boundary condition(Section 3.8.1)

Thus, it is important to compromise a relation between Δx and Δt to control the errors. To decrease the LBM problem errors in this work, the relation between the time step and grid size was chosen to be close to $\Delta t \sim \Delta x^2$ [51].

Results

The simulations in this section were carried out for different grids (Table 4.2). The results for these grids are shown in Figures 4.8 and 4.9, however, the results discussion in this section is only considered for the high resolution grid (i.e. Case4 in Table 4.2).

Figure 4.8 (a) shows the normalized mean axial velocity V/U_j as a function of normalized wall-normal distances y/D at the normalized radial distance $x/D = 0$. The results show good agreement with the numerical and experimental results. The axial velocity profile at $x/D = 0.5$ is shown in Figure 4.8 (b) and the results are also shown a good agreement with the literature result. Figure 4.8 (c) depicts the results for U/U_j profile at the radial distance $x/D = 1$. The LBM simulation result is in good agreement with the simulation and experimental results in the region far from the wall (i.e. $y/D > 0.05$); however, the result slightly deviates from the literature values in the near wall region ($y/D < 0.05$). The main reason for the discrepancy in the near wall results could be attributed to insufficient LBM grid resolution in this region which makes the current resolution unable to resolve and capture the radial velocity parallel to the wall.

Figure 4.9 (a) illustrates the normalized axial turbulence intensity v_{rms}/U_j at $x/D = 0.5$ (shear layer). In general, the v_{rms} profile follows the numerical result profile ([38])

within the region $y/D < 0.4$. Figure 4.9 (b) demonstrates the normalized axial turbulence intensity (v_{rms}/U_j) at $x/D = 1$. The LBM profile (this work) exhibits good agreement with the numerical simulation result at $y/D < 0.14$, while it shows some variation for $y/D > 0.14$, The reason for this disagreement might be related to the jet inlet boundary condition.

Figure 4.9 (c) shows the normalized radial turbulence intensity u_{rms}/U_j at the radial distance $x/D = 0.5$ (shear layer). Down to axial distance $y/D \approx 0.15$, the LBM result shows good agreement with the numerical simulation and Cooper's experimental results. Close to the wall (i.e. $y/D < 0.02$), the u_{rms} value is gradually decreased as a result of the damping from solid wall. Figure 4.9 (d) depicts the normalized radial fluctuating velocity at radial distance $x/D = 1$. The LBM result shows acceptable agreement with the numerical and experimental results except for the distance $y/D < 0.04$, where the effect of the grid resolution becomes very crucial in determining near-boundary accurate results.

The fully developed boundary condition, used in this simulation, enforces vanishing spatial derivatives along the radial (x) direction, and providing an unrealistic non-physical condition in the region directly preceding the outflow boundary. To insulate the physically accurate results from the outflow boundary condition, a buffer zone could be placed before the flow exit boundary and the results can be calculated before this zone. This buffer zone should be about the last 1/3 of the entire computational domain [108].

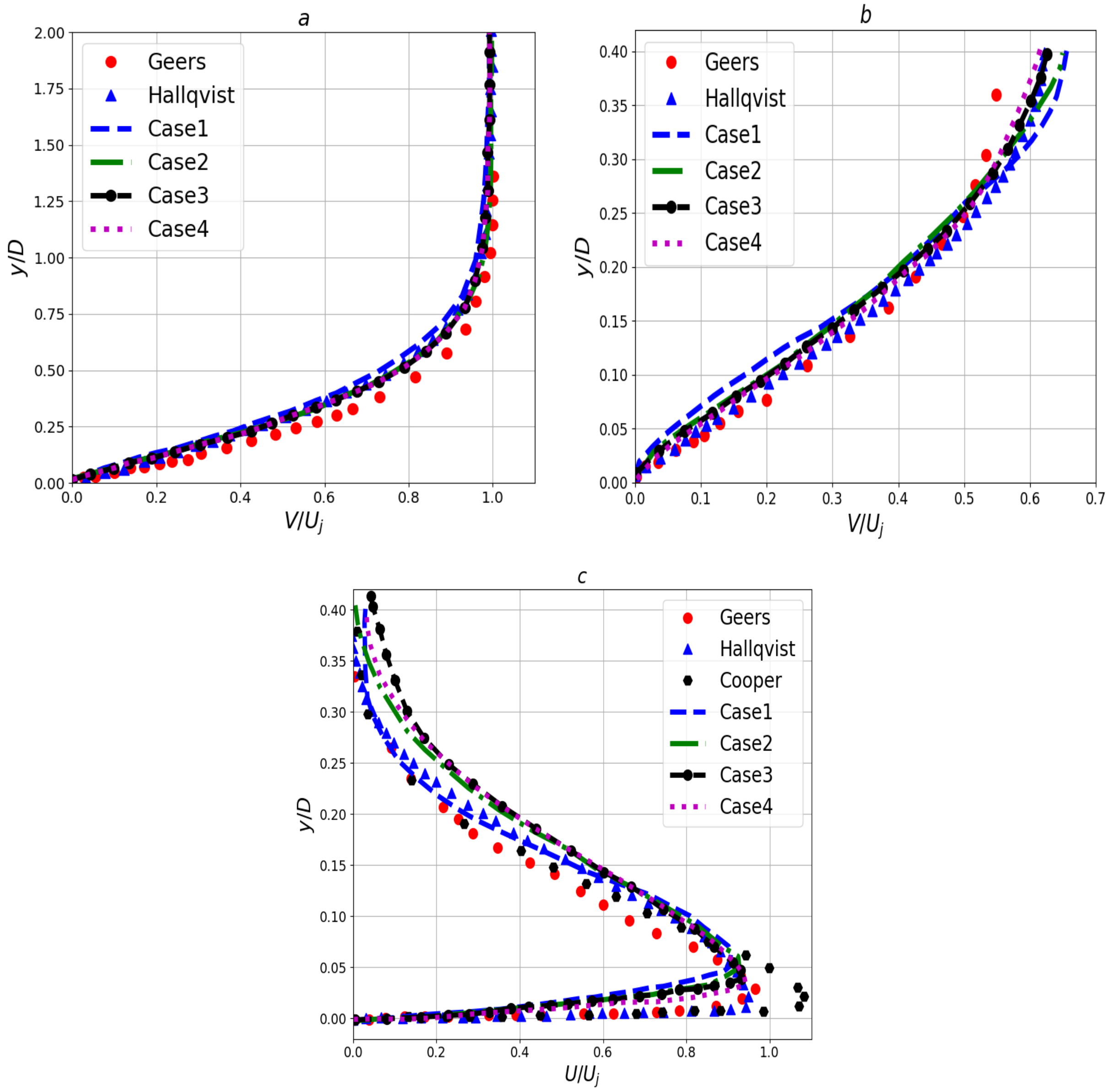


Figure 4.8: a: Mean axial velocity V/U_j at $x/D = 0$; b: Mean axial velocity V/U_j at $x/D = 0.5$; c: Mean radial velocity U/U_j at $x/D = 1$.

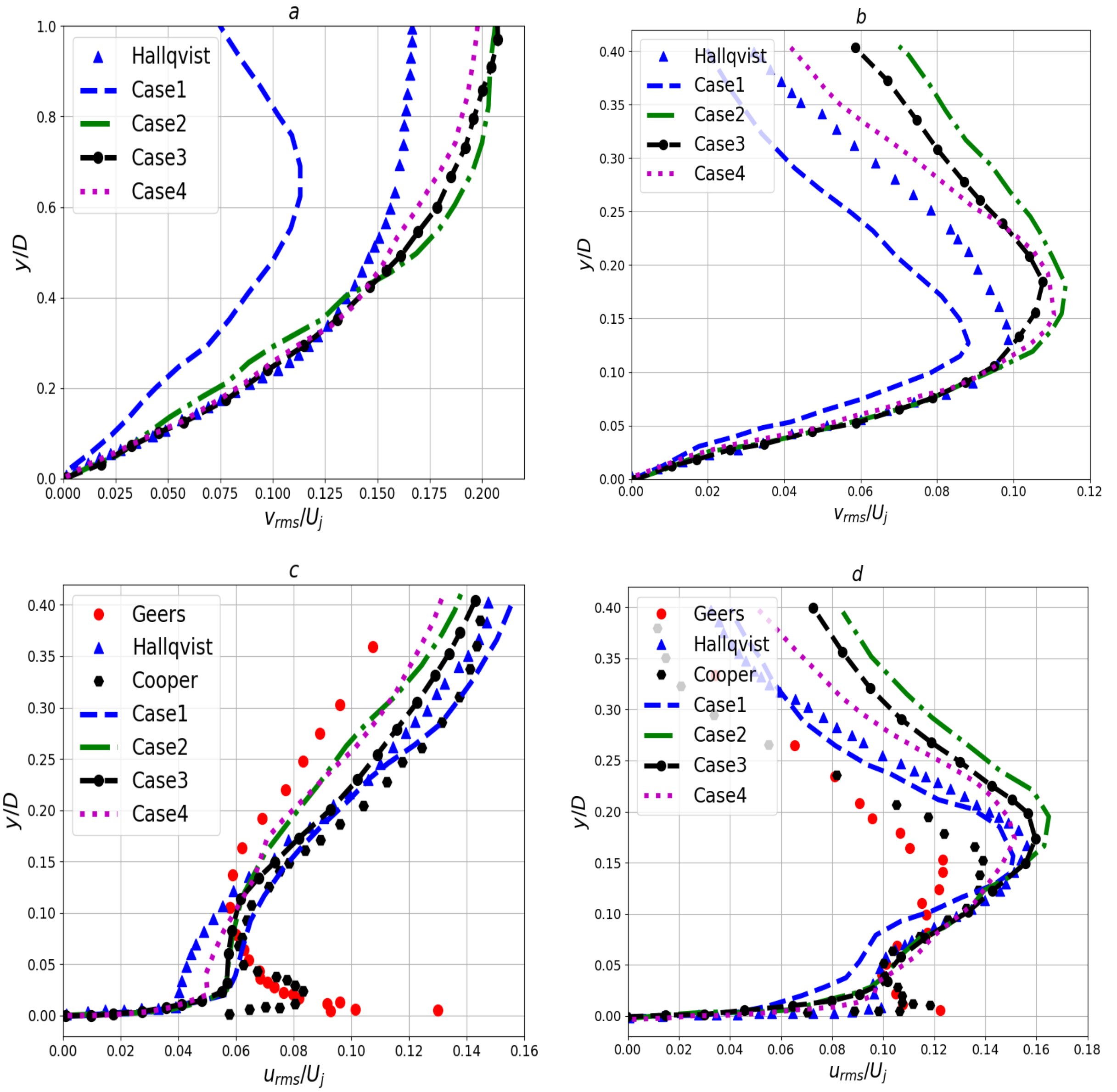


Figure 4.9: a: Axial v_{rms}/U_j at $x/D = 0.5$; b: Axial v_{rms}/U_j at $x/D = 1$; c: Radial u_{rms}/U_j at $x/D = 0.5$; d: Radial u_{rms}/U_j at $x/D = 1$.

Figures 4.10 and 4.11 depict the energy spectral content of the flow analyzed by the power spectrum. The figures show the power spectrum of u_{rms} and v_{rms} at two different stations within the flow domain. The first station is located within the shear layer close to the impingement wall at $y/D = 0.05$ and $x/D = 0.5$, and the second station is positioned at $y/D = 1$ and $x/D = 0.5$.

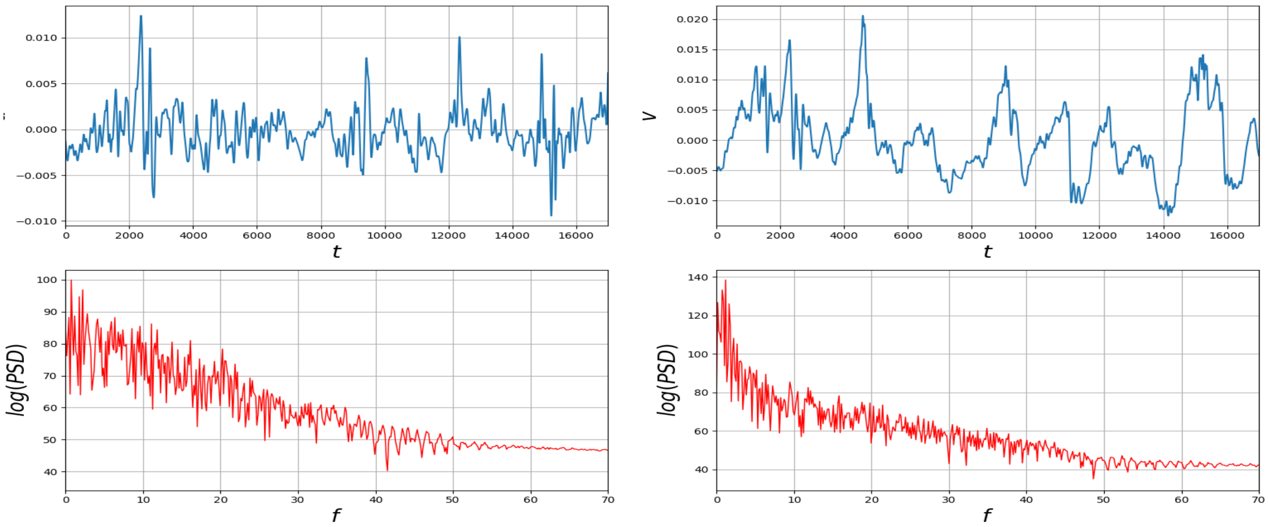


Figure 4.10: Power Spectral Density *(PSD)* for a: v'; b: u' at station 1 ($y/D = 0.05$, $x/D = 0.5$).

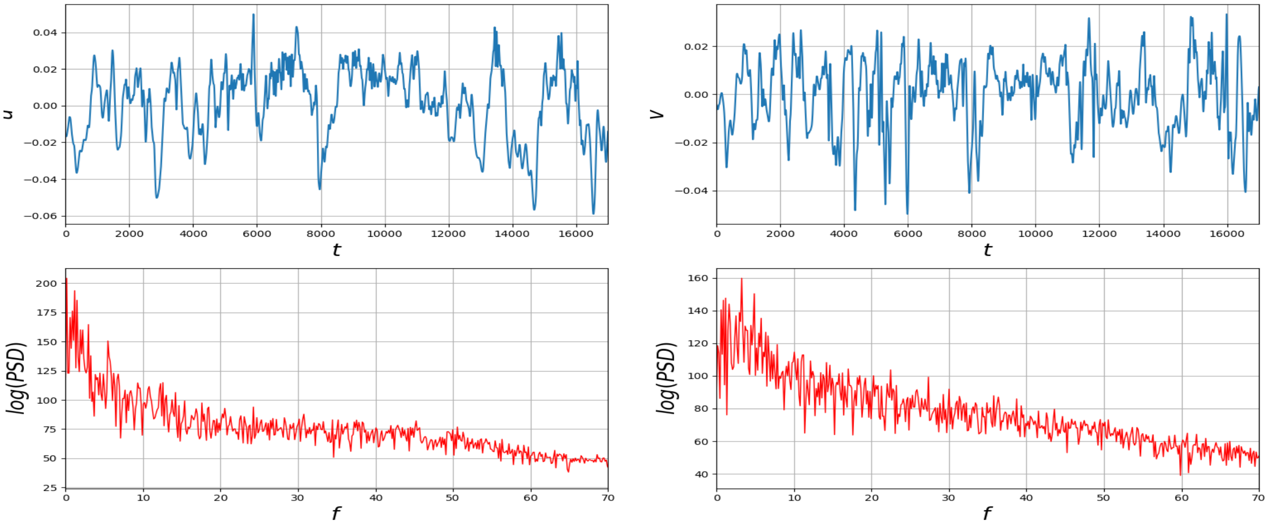


Figure 4.11: Power Spectral Density *(PSD)* for a: v'; b: u' at station 2 ($y/D = 1$, $x/D = 0.5$).

Chapter 5

Modified LBM-CA Model Implementation on CUDA GPUs

Due to the massive number of calculations, the implementation of the MRT LBM-CA algorithm in a serial CPU code increases the computational load in a substantial manner. However, this deficiency in the serial CPU code can be alleviated by utilizing GPU CUDA programming to provide a parallel multi-threaded code. Inspite of the fact that GPUs are designed originally to process large graphic data in a fast way, tremendous strides have been made in order to reduce the limitation in the GPUs programming and facilitate them for general purpose calculations. In this study, a single NVIDIA GeForce GTX TITAN X was used to accomplish the numerical simulation. The technical specifications of this GPU are given in Table 5.1.

Table 5.1: Technical specifications for the NVIDIA GeForce GTX TITAN X

Compute Capability	6.1
CUDA Cores	3072
Memory Size	12(GB)
Memory Bandwidth	336.5(GB/sec)
Single Precision Floating Point Performance	7(TFLOPS)

5.1 LBM-MRT Fluid Flow Implementation on GPUs

The data locality in LBM makes this method very suitable to be implemented on GPUs. In a very basic LBM CUDA code, each thread, which is the smallest processing unit in GPU, is assigned to an LBM node in the problem domain. The major problem in the LBM code implementation is the memory transaction related to the streaming step. To involve the distribution functions from the neighbouring nodes in the streaming process, each thread needs to access the contiguous memory locations associated with diffent LBM nodes. To reduce the time required for the memory accessing process, Structure of Arrays (SoA) memory arrangement is usually used instead of Array of Structures (AoS) one; the former arrangement ensures optimal memory access by providing a coalesced memory locations. Moreover, a multi-dimensional LBM nodes array can be stored in a linear one-dimensional block array. Algorithm 1 shows the pseudocode of conversion three-dimensional block array in to one-dimensional linear array. The value of J represents the location of an LBM node in the linear memory, while x, y, and z are the positions of the LBM node in x, y, and z direction, respectively. More details about the LBM fluid code implementation on GPUs can be found in [48].

Algorithm 1 Conversion of three-dimensional block array in to one-dimensional linear array

```
int $x$ = threadIdx.x+blockIdx.x*blockDim.x;
int $y$ = threadIdx.y+blockIdx.y*blockDim.y;
int $z$ = threadIdx.z+blockIdx.z*blockDim.z;
int $J$ = x+y*pitch+z*YDIM*pitch; //linear memory
//pitch = pitch size of array
```

5.2 The LBM-CA Dispersed Particles Model on GPUs

A new model is developed to simulate the solid particles transport and deposition. In this model, the number of solid particles is defined in the LBM fluid nodes. This method can reduce the number of memory locations required when the number of particles in the simulation is larger than the LBM lattice number. In addition, updating particle velocities only requires the local information of the LBM nodes. The particles in this model are forced to move in the same fluid LBM lattices by considering all external forces and applying Newton's second law. In three-dimensional LBM-CA simulation, the fluid flow utilizes the D3Q19 model, while the particles movements employs the D3Q27 one. Figure 5.1 shows an LBM lattice; the blue links represents the fluid links, and the extra red colour links are defined for the particles movements. In addition to the fluid simulation parameters, the implementation of the LBM-CA model in GPU involves new parameters in the GPU global memory. A particle number array is used to define the number of particles at each node. To avoid the data interaction conflict, two sets of this array, N_A and N_B (source and destination), are defined in the global memory throughout the simulation, and they are swapped at each time step. Another array for the particles local average velocity is allocated in the GPU memory. This array is a three-dimensional array, and each dimension represents the velocity component in the Cartesian three-dimensional LBM model. The particle velocity is not defined for each single particle; instead, it is defined at each LBM node representing the average velocity for a group of particles residing inside the LBM lattice boundaries.

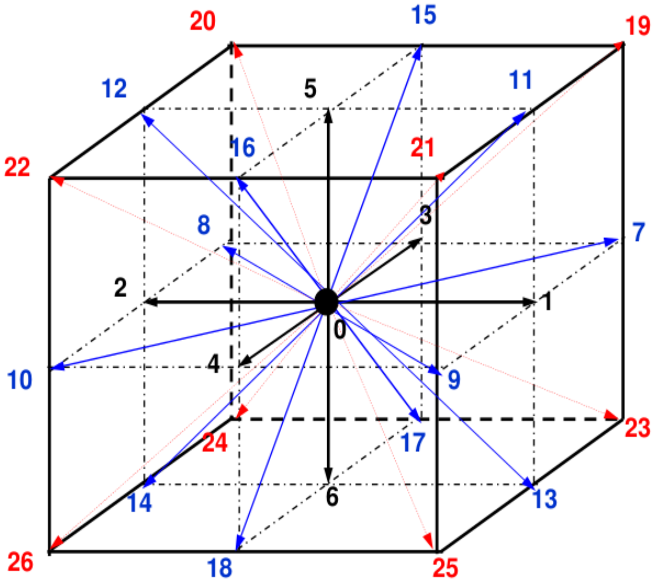


Figure 5.1: LBM lattice; blue: fluid model (D3Q19); blue and red: solid particles model (D3Q27)

5.2.1 The Particle Transport Algorithm of LBM-CA

The movements of the solid particles in this model are constrained in the same regular LBM nodes. At each node, the number of particles (N_p) and the particles average velocity ($\mathbf{U}_p$)

are defined initially according to the initial particles distribution condition. At the next simulation time step, the number of particles moved to the neighbouring nodes (N_{move}) is determined according to the local particle number and speed:

$$N_{move} = integer\left(\frac{N_p}{t_{move}}\right), \tag{5.1}$$

where t_{move} is the number of time steps required to move all the particles in a specific node to the next neighbouring nodes, and it is defined as follows:

$$t_{move} = integer\left(\frac{1}{Up_{abs}}\right), \tag{5.2}$$

where Up_{abs} is the local absolute particles velocity magnitude.

$$Up_{abs} = \sqrt{(Up_x)^2 + (Up_y)^2 + (Up_z)^2}\ , \tag{5.3}$$

where $Up_x, Up_y,$ and Up_z are the local particle velocity components in $x, y,$ and z direction, respectively. In Equations 5.1 and 5.2, the term *integer* is used in the GPU code to round the real values to the nearest integer values. However, the N_{move} can take a real number form which represents the exact fraction of the N_p distributed at each time step. Note: integer values of N_{move} are used in this work because using real values leads to losing some particles during the simulations. The value N_{move} in Equation 5.1 may be decreased when the local number of particles (N_p) is decreased during the simulation; then applying Equation 5.1 at each time step will give unrealistic number of the moving particles (i.e. less than the real moving particles). To avoid the decrease in N_{move} during the particles movement, an array of N_{move} can be built and saved in the GPU memory. For each LBM node, N_{move} value can be raised with increasing N_p through Equation 5.1 and it is unchanged when N_p is decreased.

Figure 5.2 depicts an example of movement of particles in a two-dimensional lattice. In this figure, 100 particles (N_p) are moving downward (i.e. one-dimensional movement) at a uniform non-dimensional velocity [$Up_{abs} = 0.2$ lattice/LBM time step (t)]. The red colour particles represent the particles which are initially ($t = 0$) in the lattice, while the blue colour is for particles moving into the lattice from the upper boundary. According to these condition, the blue particles need 5 time steps to occupy the lattice space ($t_{move} = \frac{1}{Up_{abs}} = 5$), and the red particles require the same time steps to leave the lattice. The number of red particles leaving and blue particles entering at each time step is: $N_{move} = \frac{N_p}{t_{move}} = 20$.

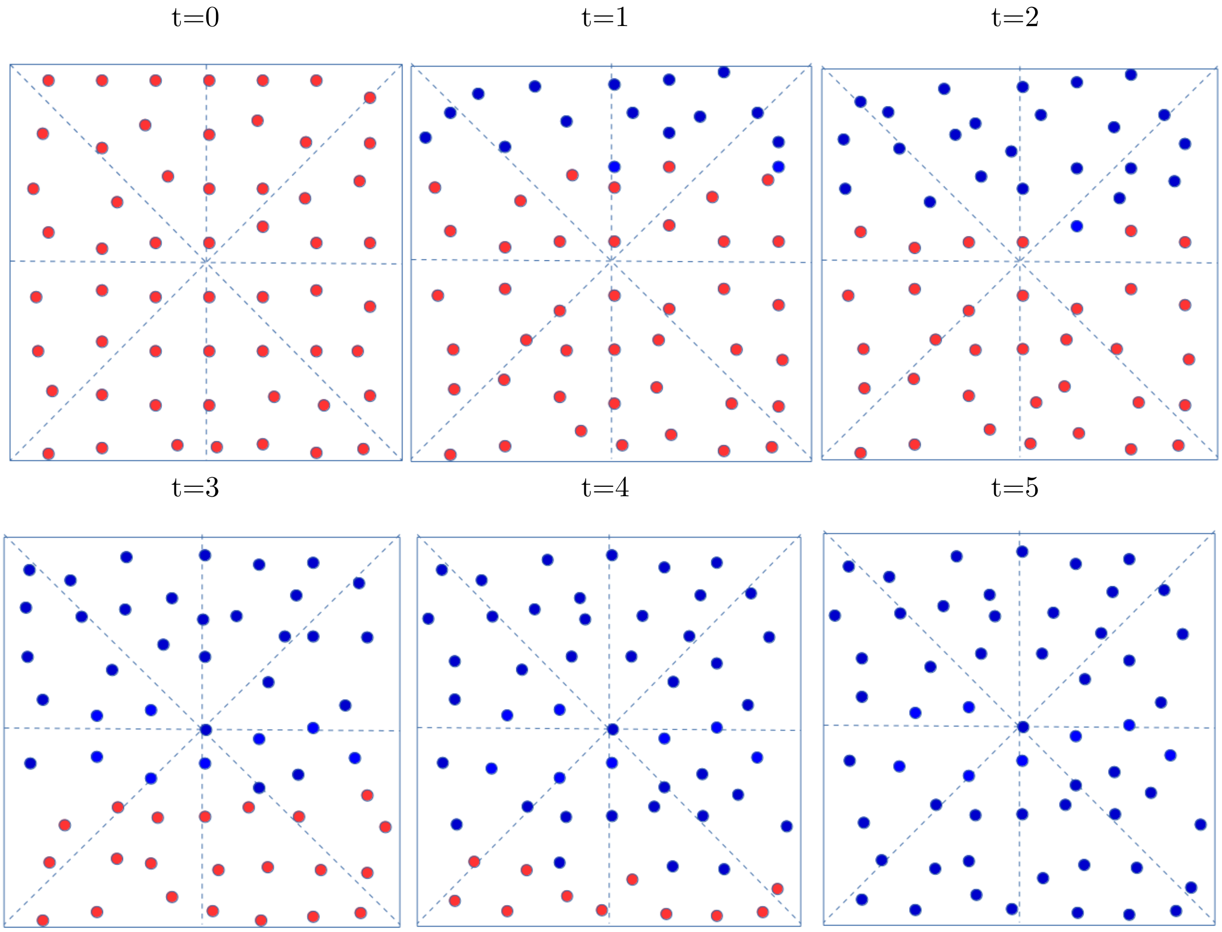


Figure 5.2: Schematic view of time (t) evolution of particles movement in a two-dimensional LBM lattice.

In three-dimensional D3Q27 LBM model, particles in a node can move along 26 links (i.e. 26 probabilities) to new nodes as shown in Figure 5.1. To reduce the number of probabilities where the particles can move, a cubic lattice is equally divided into eight sub-cubes. According to the particle velocity components in three dimensions, the sub-cube which particles move through can be determined as shown in Figure 5.3. Then, the distribution of the particles from each node to the next nodes is accomplished on the GPU by a program looping through N_{move}. This loop allows a GPU thread to deal with each single particle separately and move it to the next node according to a probabilistic algorithm.

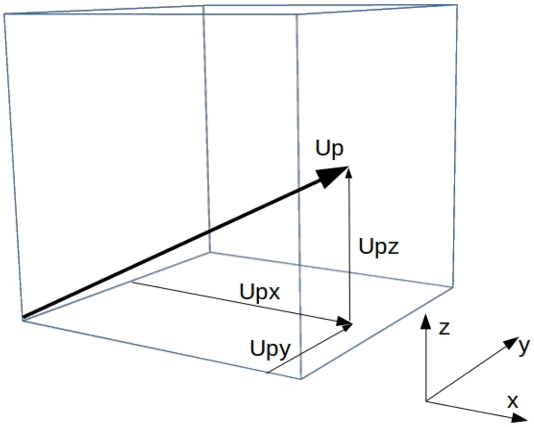


Figure 5.3: Distribution of particles inside a sub-cube.

The first step inside the loop is generating three real random numbers (r_1,r_2, and r_3) in an interval $[0, 1]$. The cuRAND library in CUDA, which used in this work, provides a simple and efficient generation of high-quality pseudorandom and quasirandom numbers [73] [93]. These three numbers are compared with the absolute normalized particle velocity components Upx_n, Upy_n, and Upz_n (e.g. $Upx_n = \mid Up_x \mid / Up_{abs}$), and the new position is determined according to the conditions in Algorithm 2. Note: this algorithm is for the sub-cube shown in Figure 5.3 and different algorithms are applied for different sub-cubes.

Algorithm 2 Particle movement conditions

```
if $(r_1 < Upx_n \ \& \ r_2 > Upy_n \ \& \ r_3 > Upz_n)$, then, $(X_{add} = 1; Y_{add} = 0; Z_{add} = 0)$,
if $(r_1 > Upx_n \ \& \ r_2 < Upy_n \ \& \ r_3 > Upz_n)$, then, $(X_{add} = 0; Y_{add} = 1; Z_{add} = 0)$,
if $(r_1 > Upx_n \ \& \ r_2 > Upy_n \ \& \ r_3 < Upz_n)$, then, $(X_{add} = 0; Y_{add} = 0; Z_{add} = 1)$,
if $(r_1 < Upx_n \ \& \ r_2 < Upy_n \ \& \ r_3 > Upz_n)$, then, $(X_{add} = 1; Y_{add} = 1; Z_{add} = 0)$,
if $(r_1 < Upx_n \ \& \ r_2 > Upy_n \ \& \ r_3 < Upz_n)$, then, $(X_{add} = 1; Y_{add} = 0; Z_{add} = 1)$,
if $(r_1 > Upx_n \ \& \ r_2 < Upy_n \ \& \ r_3 < Upz_n)$, then, $(X_{add} = 0; Y_{add} = 1; Z_{add} = 1)$,
if $(r_1 < Upx_n \ \& \ r_2 < Upy_n \ \& \ r_3 < Upz_n)$, then, $(X_{add} = 1; Y_{add} = 1; Z_{add} = 1)$,
```

The values of X_{add}, Y_{add}, and Z_{add} are the spatial increments (i.e. in three-dimensions) of the particle located at (x, y, z) which coincides with the fluid node (e.g. node 0 in Figure 5.1). The new linear memory location for a moving particle is defined in Algorithm 3.

Algorithm 3 The new linear memory location (J_{new})

```
int $X_{new}$ = x + $X_{add}$;
int $Y_{new}$ = y + $Y_{add}$;
int $Z_{new}$ = z + $Z_{add}$;
int $J_{new} = X_{new} + Y_{new} * pitch + Z_{new} * YDIM * pitch$; // new linear memory location
```

The two copies of the particle number array (N_A and N_B) are passed to the particle update kernel. The array N_A contains the initial particle number at each node, while the array N_B has zero values for all nodes. The two arrays are updated at the end of the particle distribution loop (through N_{move}) as shown in Algorithm 4.

Algorithm 4 Updating particle number in N_A and N_B

```
$N_A[J] = N_A[J] - 1$;
$N_B[J_{new}] = N_B[J_{new}] + 1$;
```

After finishing the particle updating loop, each particle number element in array N_A is added to its equivalent element (i.e. same linear memory location) in array N_B. After the addition process, all the particle number values in N_A are cleared (zeroed). Finally, the memory addresses of A and B for the two arrays are swapped to make N_B the source array and N_A the destination array, as illustrated in Algorithm 5.

Algorithm 5 Finalizing the GPU algorithm

```
//End of particles distribution loop
$N_B[J] = N_B[J] + N_A[J]$;
$N_A[J] = 0$;
//End of the GPU kernel
// Return to the CPU time loop
swap(A,B);
```

5.2.2 Particle Velocity Update

The local particles motion can be described by applying Newton's second law considering external forces:

$$\frac{d\mathbf{U}_p}{dt} = \mathbf{F}_{drag} + \mathbf{F}_{buo} + \mathbf{F}_g + \mathbf{F}_{other} \tag{5.4}$$

where $\mathbf{F}_{drag}$, $\mathbf{F}_{buo}$ and $\mathbf{F}_g$ are the drag, buoyancy, and gravity forces, respectively. The $\mathbf{F}_{other}$ represents other external forces such as Basset force, Magnus force, Saffman force, and Brownian force. Only drag, buoyancy, and gravity forces are considered in this work because of the high particle to fluid density ratio. The non-dimensional (LBM units) equation of motion is given as [85]:

$$\frac{dU_{p_i}}{dt} = \frac{U_{f_i} - U_{p_i}}{Stk} + \frac{1}{Fr^2}\left(1 - \frac{1}{S}\right) \tag{5.5}$$

where Stk and Fr are Stokes and Froud numbers, respectively, and they are defined as:

$$Stk = \frac{\rho_p d_p^2 u_0}{18\mu_g l_0}. \tag{5.6}$$

$$Fr = \frac{u_0}{\sqrt{g_0 l_0}}. \tag{5.7}$$

where ρ_p and d_p are the particle density and diameter, respectively; μ_g is the fluid dynamic viscosity; u_0 is the fluid characteristic velocity; l_0 is the characteristic dimension of the obstacle (typically the particle diameter); S is the solid particle to fluid density ratio. Equation 5.5 is solved using the fourth order Runge-Kutta method. The particle velocity update process is performed for each node containing particles in N_A, and the a non-dimensional time step (Δt_{RK}) used in Equation 5.5 is:

$$\Delta t_{RK} = \frac{\Delta t_{lbm}}{t_{move}} = \frac{1}{t_{move}}, \tag{5.8}$$

where Δt_{lbm} is the non-dimensional LBM time step, which is equal to 1. The velocity values of all particles moving to N_B are summed to produce total velocity ($\mathbf{U}_B$), and the new average particle velocity is calculated by the pseudocode shown in Algorithm 6.

Algorithm 6 Particle velocity update

```
if $N_A[j] = 0$ // No particles in array A
$\mathbf{U}p[j] = \mathbf{U}_B[j]/N_B[j]$ ;
else if $N_A[j] \neq 0$
$\mathbf{U}p[j] = (\mathbf{U}p[j] * N_A[j] + \mathbf{U}_B[j] * N_B[j])/(N_A[j] + N_B[j])$ ;
```

5.2.3 Particle Sedimentation on the Wall

Particles deposited on the wall may occupy significant volumes of the LBM wall nodes. In this case, the particles will build a collection of heaps with different heights based on the particles number. Consequently, the wall nodes will be deformed and the curved boundary condition (Section 3.8.5) can be applied instead of the traditional BB boundary condition. Figures 5.4 shows the effect of the particle deposition on the wall, where the distance ($dx/2$) between the fluid and the wall nodes (a and b) is reduced by the factor ζ which is estimated as follows:

$$\zeta = \frac{N_p V_p}{V_{lbm}/2}, \tag{5.9}$$

where N_p, V_p, and V_{lbm} are the number of deposited particles, particle volume, and the LBM cell volume, respectively. When a large number of particles(e.g. $N_p \times V_p = V_{lbm}/2$) are deposited next to a wall node, the neighbouring fluid node (node c in Figure 5.4) is solidified and considered as a wall node. The latter wall boundary process is only considered when the deposited particle volume exceeds a given threshold value (P_{thr}). In this study, the threshold value is equal to 20% of the LBM wall cell volume (Note: less than 20% may not affect the fluid flow and particle deposition significantly).

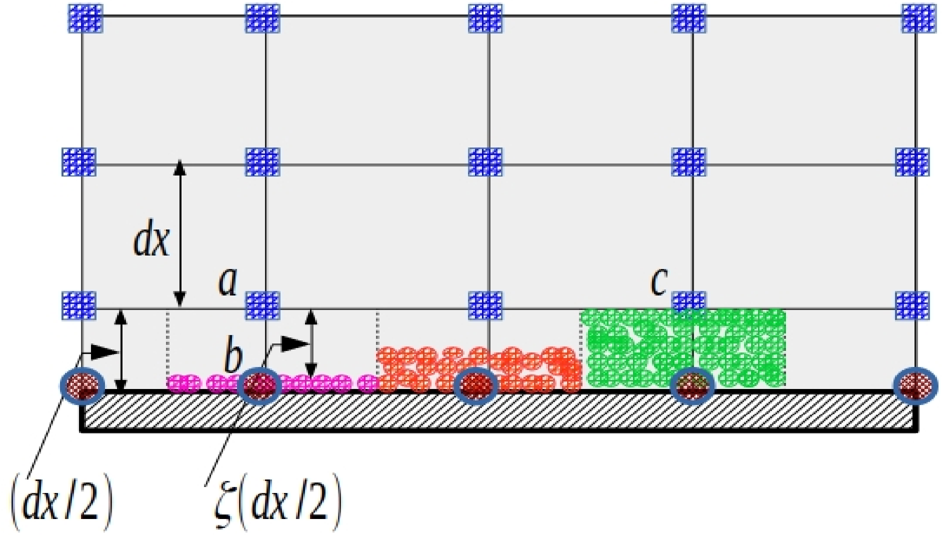


Figure 5.4: Schematic view of the particle sedimentation on the wall.

5.2.4 LBM-CA Particle Transport Grid Refinement Model on GPUs

In this study, a new LBM-CA local grid refinement technique for the particle transport is developed. The movement of particles in the fluid fine grid should be adjusted to build a compatible robust model for the fluid-particle interaction in the refined region of the flow. Figure 5.5 depicts a two-dimensional schematic view of the coarse (blue) and fine (red) grids in an LBM flow domain for a refinement factor $n = 2$.

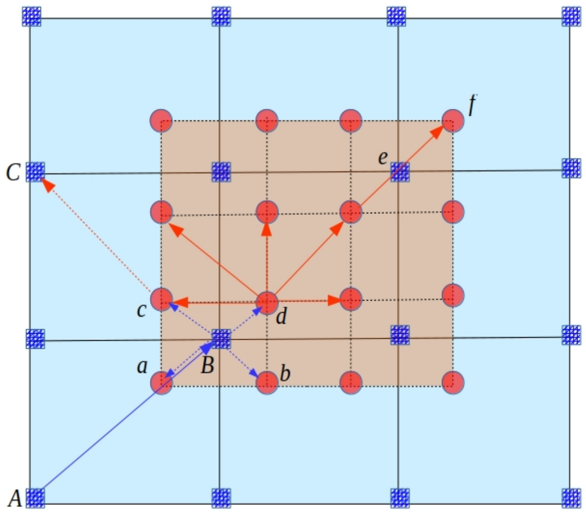


Figure 5.5: Particle transport between the coarse grid (blue) and the fine grid (red).

The model of the particle transport between the coarse and fine grid and inside the fine grid is constructed as follows. The fluid distribution and all other variables (e.g velocities) are defined in both (coarse and fine) grids (Section 3.9). The particles numbers is known in the coarse nodes which are contiguous to the fine grid. Particle in the coarse node (i.e. node A in Figure 5.5) can be transport spatially to the neighbouring nodes in the coarse grid by applying the transport algorithm (Section 5.2.1). The line between nodes A and B shows that some particles are moved from node A to node B which is located inside the fine grid. The particles in node B are distributed spatially to the adjacent fine grid nodes (i.e. a, b, c, and d). This distribution can be accomplished by applying the linear extrapolation technique, such that:

$$N_a = N_b = N_c = N_d = \frac{N_B}{4}, \tag{5.10}$$

where N_a, N_b, N_c, N_d, and N_B are the number of particles at nodes a, b, c, d and B, respectively. The distribution in Equation 5.10 can be applied when the number of particles is defined as a real number in the LBM nodes. For the case of integer number values of the particles, the distribution is completed by applying successive particle movement from node B to the fine grid nodes (i.e. the successive distribution starting from node d), as in Algorithm 7.

Algorithm 7 Distribute particles in a refined grid

```
for (int i=0; i < NB ;i++)
(
Nb=Nb+1;
NB=NB-1;
if(NB > 0)
(
Nc=Nc+1;
NB=NB-1;
)
if(NB>0)
(
Nd=Nd+1;
NB=NB-1;
)
if(NB>0)
(
Na=Na+1;
NB=NB-1;
)
)
```

5.2.5 LBM-CA Near-Wall Turbulent Dispersion Model

The LBM-CA probabilistic model is considered in the flow region where the turbulence intensity (e.g., u_{rms}) is much lower than the mean fluid flow velocity U such that ($u_{rms} << U$). Near the wall, the turbulence is of the same order as the mean flow ($u_{rms} \approx U$). As shown in Figure 5.6, the solid areas are constructed on the wall due to the particle deposition. Velocities at points a and b in the fine grid may exhibit high fluctuations because of the non-uniform wall boundaries (Section 5.2.3). In this situation, it is necessary to find a model that can control the particle movements at these points based on the turbulence intensity. The near wall particle distribution can be defined in term of the particle distribution function (PDF) or the particle dispersion function as follows:

$$f_p(\mathbf{c}_p, \mathbf{x}, t) = \langle \delta_p(\mathbf{x}, t)\delta(\mathbf{c}_p) \rangle, \tag{5.11}$$

where δ is the Kronecker delta symbol and $\delta_p(\mathbf{x}, t)$ is the dispersed phase function such that:

$$\begin{aligned} &\delta_p(\mathbf{x}, t) = 1, \quad \text{if the particle center is located at } \mathbf{x} \text{ at time t,} \\ &\delta_p(\mathbf{x}, t) = 0, \quad \text{otherwise,} \end{aligned} \tag{5.12}$$

where the $\langle . \rangle$ symbol represents the ensemble average over a very large number of particles. The probable number of particles located in the volume $[\mathbf{x}, \mathbf{x} + d\mathbf{x}]$ at time t with velocities in the range $[\mathbf{c}_p, \mathbf{c}_p + d\mathbf{c}_p]$ is $f_p(\mathbf{c}_p, \mathbf{x}, t)d\mathbf{x}d\mathbf{c}_p$. The total number of particles $n_p(\mathbf{x}, t)$ (zero moment) at a volume centre $\mathbf{x}$ and a time t is calculated by integrating the distribution functions over all possible particle velocities. In addition, higher order moments can also be calculated in similar way to LBM, as follows:

$$\begin{aligned} n_p(\mathbf{x}, t) &= \int_{\mathbf{c}_p=-\infty}^{\infty} f_p(\mathbf{c}_p, \mathbf{x}, t) dc_p, && \text{(zero moment)} \\ U_{p_i} &= \frac{1}{n_p} \int_{\mathbf{c}_p=-\infty}^{\infty} c_{p_i} f_p(\mathbf{c}_p, \mathbf{x}, t) d\mathbf{c}_p, && \text{(first moment)} \\ \langle u'_{p_i} u'_{p_j} \rangle &= \frac{1}{n_p} \int_{\mathbf{c}_p=-\infty}^{\infty} (c_{p_i} - U_{p_i})(c_{p_j} - U_{p_j}) f_p(\mathbf{c}_p \mathbf{x}, t) d\mathbf{c}_p. && \text{(second moment)} \end{aligned} \tag{5.13}$$

The single particle velocity PDF following the Boltzmann kinetic equation is [87] [25]:

$$\frac{\partial f_p}{\partial t} + \frac{\partial}{\partial x_i}(c_{p_i} f_p) + \frac{\partial}{\partial c_{p_i}} \left(\langle \frac{du_{p_i}}{dt} \rangle f_p \right) = \left(\frac{\partial f_p}{\partial t} \right)_{collision}. \tag{5.14}$$

The third term in Equation 5.14 represents the force acting on particles from the flow turbulence. The term on the right $\left(\frac{\partial f_p}{\partial t}\right)_{collision}$ is the modification of PDF as a result of the inter particle collision and it can be neglected for dilute particle dispersion. The fluid-particle interaction can be separated in to two terms: first, fluid-particle interaction through the mean flow velocities; second, turbulent interaction through the fluid turbulence motion.

The particles equilibrium state can be defined to take the Gaussian distribution form [58]:

$$f_p^{eq} = \frac{n_p}{\left(2\pi \langle u'_f u'_p \rangle\right)^{1/2}} exp \left(-\frac{(c_p - U_p)^2}{2\langle u'_f u'_p \rangle} \right), \tag{5.15}$$

where $\langle u'_f u'_p \rangle$ represents the fluid-particle correlation and can be defined according to Tchen theory [8]:

$$\langle u'_f u'_p \rangle = \langle u'_f u'_f \rangle \frac{1}{1 + Stk} = \langle u'_p u'_p \rangle, \tag{5.16}$$

where the term $\langle u'_p u'_p \rangle$ is the particle-particle correlation. The Stk is Stokes number (i.e. $Stk = \tau_p/\tau_f$), where τ_f is the fluid relaxation time defined as the Lagrangian integral time scale measured along a fluid particle trajectory [25]. It was noticed that the near-wall velocity fluctuations in this study cases (e.g. turbulent impinging jet) are very small compared with the fluid mean velocities. The axial velocity fluctuation (towards the wall) is important when the particle deposition is the main focus of simulations. Consequently, the near-wall turbulence will account in the nodes which are adjacent to the wall nodes by adding the particle-particle correlation (Equation 5.16) to the particle mean axial velocity.

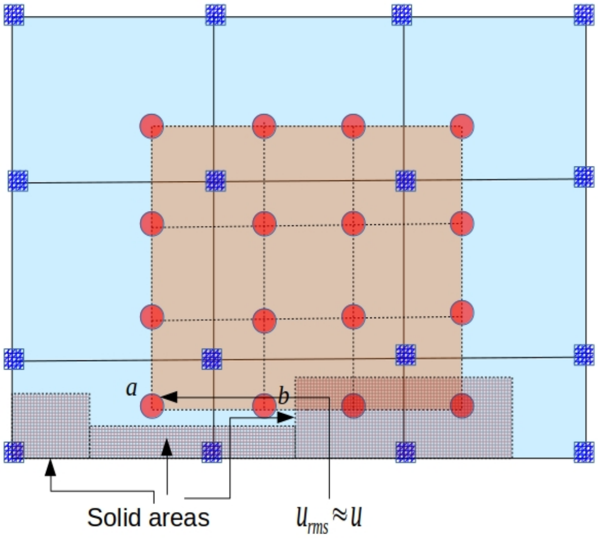


Figure 5.6: Turbulent dispersion model.

5.3 Results and Discussion

5.3.1 Flow Over a Rectangular Cylinder Confined in a Duct

The schematic view of the computational domain for this test case is shown in Figure 5.7. A rectangular cylinder with width B and length W is located at the centre of the duct. The duct length (L) is equal to $24B$, and H is the height of the duct. The dimensions in this simulation were set to facilitate a comparison to the numerical work of Salmanzadeh et al. [85]. For the case of obstruction aspect ratio (W/B) equal to 2, simulation was conducted using a uniform mesh with $B = 30$, $W = 60$, $H = 120$, $L = 720$ nodes, and the cylinder was located at distance $10B$ from the duct entrance. The inflow was determined according to the prescribed velocity profile as:

$$u = U_{max}\left(1 - \frac{r^2}{R^2}\right), \tag{5.17}$$

where $R = H/2$, and U_{max} is the maximum velocity in the axial (x) direction. The simulation was conducted in three-dimensional flow domain and the z dimension was set to 10 nodes; however, the current results were compared with the simulation ones in two-dimensional domain. For that, particle movements were restricted to be in two dimensions only (i.e. x and y directions).

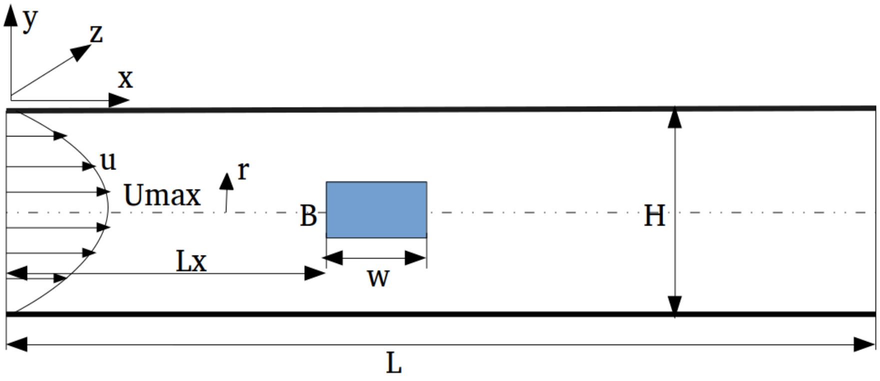


Figure 5.7: Schematic view of the duct with an obstacle.

Fluid Flow Simulation on GPUs

A comprehensive study for this problem was accomplished by Koda and Lien [49]. An LBM code was developed utilizing the GPU CUDA architecture, and the results were validated against results of a test case involving the flow in a duct over a square cylinder. Good agreement was obtained, and the GPU code was faster than the traditional CPU code by a factor of 150. More information about the fluid flow LBM code implementation on GPUs can be found in literature, [49][48][74][90][91].

LBM-CA Particle Transport and Deposition Code Validation

The results in this section were obtained for particle trajectories and deposition rates with different Stokes (Stk) numbers and blockage ratios ($Br = B/H$). In this simulation, the fluid flow simulation was first started, and after sufficient simulation time steps (e.g. 30000 LBM time steps), 50 particles were injected in each of the duct entrance nodes. The time when the particle injection started is consider a zero (i.e. t=0). The particle diameter can be controlled by the Stokes number which is the square of the non-dimensional particle diameter [86].

Figure 5.8 shows the particle trajectories for different Stokes numbers at different time steps in the presence of span-wise gravity force (i.e. $Fr = 3$). The blockage ratio is 0.25, cylinder aspect ratio is 2, and the Reynolds number is 200. For a small Stokes number (i.e. $Stk = 0.001$), the drag force is very high compared to the gravity force. Consequently, the particles follow the streamlines of the flow in the duct and no particles are deposited on the duct walls. The small particles (i.e. small Stokes number) have low inertia forces; so few of these particles reach the cylinder surface and are deposited on it, and most of them follow the curved streamlines around the cylinder. It can be seen from Figure 5.8 (i.e. $Stk = 0.001$) that particles past the cylinder are entrained in to the area behind it by the effect of the vortex shedding which takes place downstream the cylinder. The small stokes number problem does not necessarily include nano-particles because Stokes number is a function of many parameters (Equation 5.6) in addition to the particle size; hence the Cunningham slip correction factor is not considered in this study.

When the Stokes number is increased (e.g. $Stk = 0.1$), the effect of the gravity force becomes more obvious and the particle deposition on the lower wall is increased significantly. Particles with higher Stokes numbers also have greater inertia force which enables them to move out of the curved streamlines before the cylinder and reach its front face. As shown in Figure 5.8 (i.e. $Stk = 0.1$), the effect of the vortex shedding becomes less dominant and few particles are entrained into the projection area behind the cylinder. For high Stokes numbers (e.g. $Stk = 1$ and 5), the gravity effect dominates the movement of the particles and most of the particles are deposited on the cylinder and the lower wall of the duct before the duct outlet. Table 5.2 shows this problem configuration in LBM non-dimensional units.

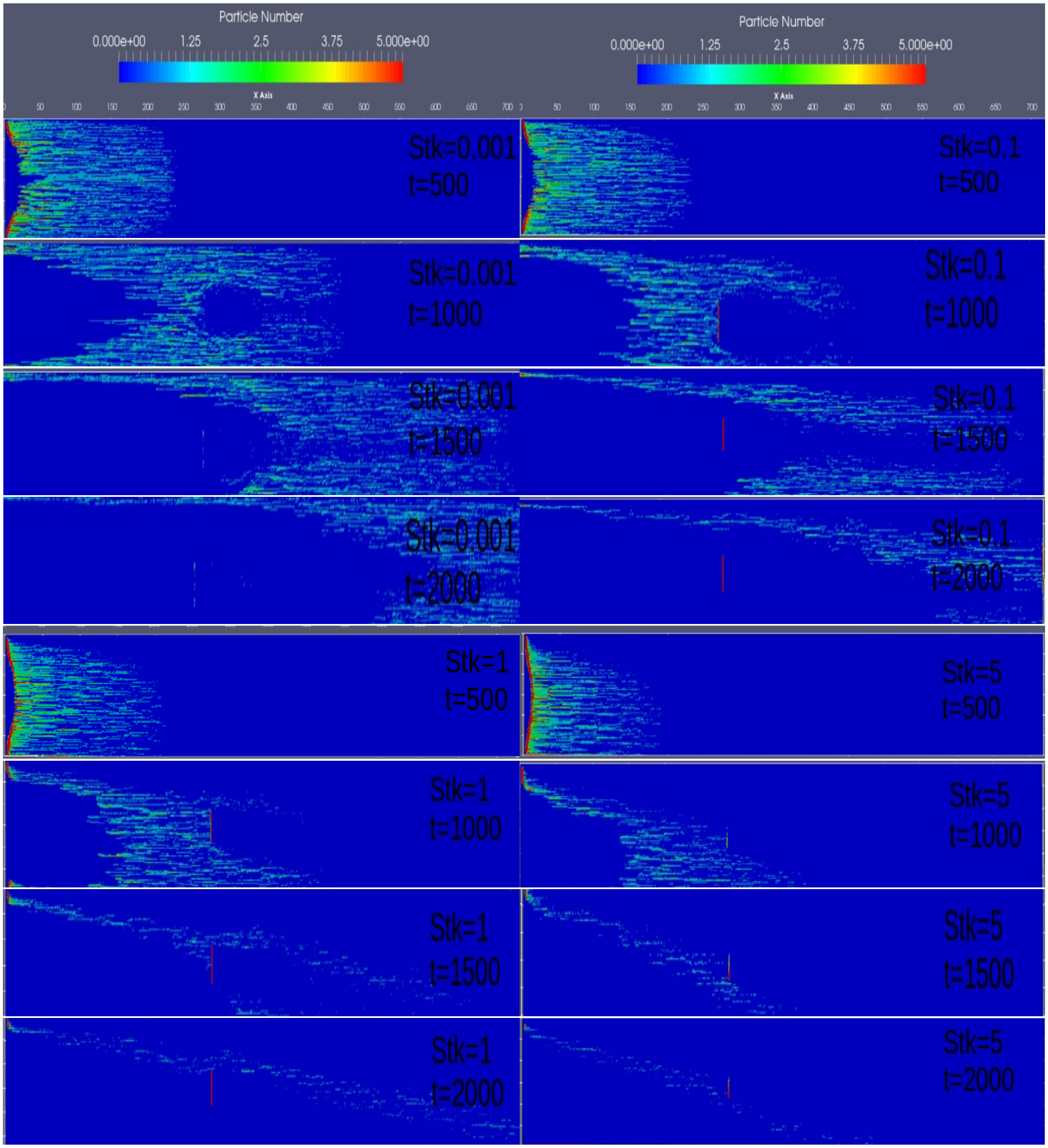


Figure 5.8: Particle trajectories at different time steps (t) for $Re = 200$, $AR = 2$, $Br = 0.25$, $Fr = 3$, and different Stokes numbers ($Stk = 0.001, 0.1, 1$, and 5).

Table 5.2: Flow over a rectangular cylinder problem configuration.

Duct height H	120
Duct Length L	720
Cylinder height B	30
Cylinder width W	60
Characteristic length	Cylinder height
Characteristic velocity (inlet velocity)	0.04
Grid size Δx (1 / Characteristic length)	0.0333
Time step Δt	0.0013
Inlet boundary condition	Velocity boundary condition (Section 3.8.2)
Outlet boundary condition	Fully developed boundary condition (Section 3.8.4)
Wall and cylinder boundary condition	Bounce back boundary condition(Section 3.8.1)

Figure 5.9 (a) shows the cumulative deposition efficiencies along the longitudinal direction of the duct for $Fr = 3$, $Br = 0.25$, and for different Stokes number. The deposition on both of the cylinder and the duct walls were considered, and the cumulative efficiency was defined as:

$$\beta = \frac{\text{Number of particles deposited at a specific distance from the duct inlet}}{\text{Total number of particles injected in the duct inlet}}. \tag{5.18}$$

In this numerical experiment, particles were injected in the duct inlet with a number of 50 particles per node within the cylinder projection area. Figure 5.9 (b) demonstrates the effect of changing the blockage ratio on the deposition efficiency for different Stokes numbers in the absence of the gravity force. The particle deposition efficiency on the front side of the cylinder is defined as:

$$\eta = \frac{\text{Number of particles deposited on the front side of the cylinder}}{\text{Number of particles injected in the cylinder projected area}}. \tag{5.19}$$

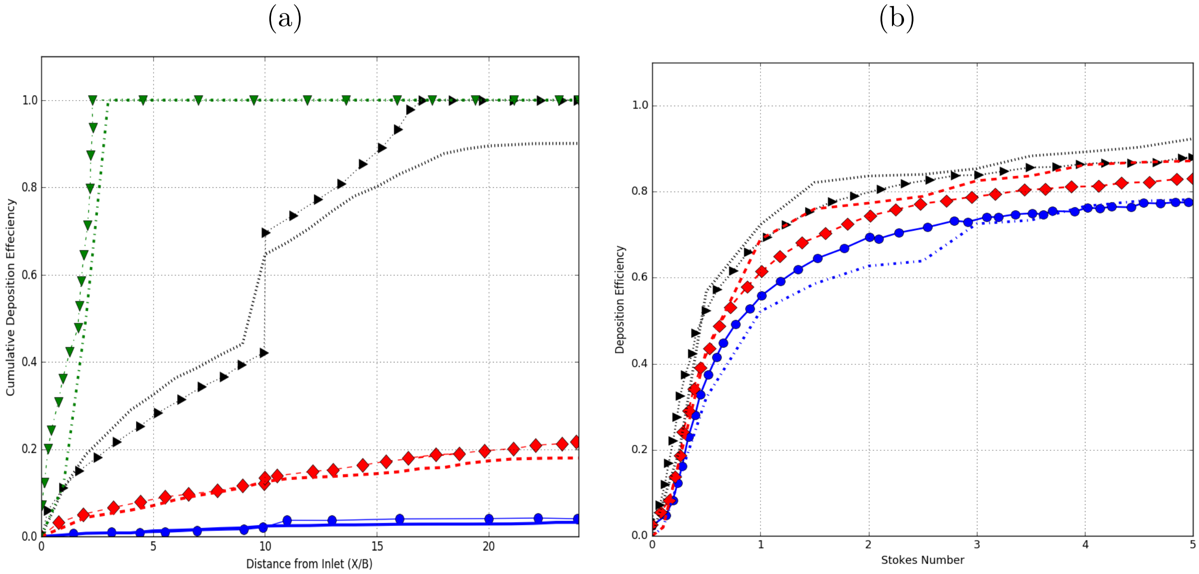


Figure 5.9: (a) Cumulative depositon efficiencies along the x direction for $Br = 0.25$ and different Stokes numbers . Green: $St = 10$; black: $St = 1$; red: $St = 0.1$; blue: $St = 0.001$. (b) Particles deposition efficiencies for different Stokes numbers and blockage ratios. Black: $Br = 0.5$; red: $Br = 0.25$; blue: $Br = 0.1$. Symbols: simulation [85]; lines: present work.

5.3.2 Fluid-particle interaction in the modified LBM-CA model

The simulations in this section are carried out to compare the modified LBM-CA model proposed in this work with the previous LBM-CA model in term of the fluid-particle interaction in vortex shedding. In this test case, 20 particles are injected in each node at the duct entrance area within 5000 time steps, and the simulations are carried out for another 20000 time steps. The simulations are performed for $Re = 300$ and the characteristic length is taken to be the height of the obstacle (B in Figure 5.7). This problem configuration is utilized to allow unsteady vortices shaped to appear downstream of the obstacle. The problem characteristics for this test case are shown in Table 5.3. Figures 5.10 and 5.11 depict particle trajectories over an obstacle in a duct for small particles (i.e. $Stk = 0.1$) at different time steps (i.e. non-dimensional LBM unit) for the modified LBM-CA model used in this study and the classical LBM-CA model, respectively. The particles are injected initially in the LBM nodes possess the same fluid velocity at each node. After few time steps (i.e. t=1200), the particles distribution in the duct of the modified LBM-CA model start to disperse taking the fluid velocity profile in the entrance area (i.e. parabolic velocity

Table 5.3: Characteristics of fluid-particle interaction problem.

L	H	B	W	U_{max}	Re	Stk	Fr (-y)	S	Lx	Time steps
510	88	20	20	0.04	300	0.1	300	1000	90	20000

distribution). The axial (x-direction) location of particles at the centerline, where the maximum fluid axial velocity equals to 0.04, is 50 which is a realistic location as the particle Stokes number is small ($Stk = 0.1$) and the particles movement is nearly as same as fluid velocity (particle axial location = particle velocity × time steps $\simeq 0.04 \times 1200 \simeq 50$). On the other hand, at time 300 for the classical LBM-CA model (Figure 5.11), many particles are moved in the duct axial direction far from the real physical location (i.e. $300 \times 0.04 = 12$). The reasons for this discrepancy between the previous and current model is that the total number of moving particles (N_{move}) at each time step in the current model is a function of the total number of particles at each LBM node and the particle average velocity (i.e. Equations 5.1 and 5.2) which provides a better particle-fluid interaction description, and the particles will be transported and distributed in a realistic manner. In contrast, the total number of moving particles in the previous LBM-CA models is considered to be the same as total number of particles in each LBM node (i.e. $N_{move} = N_p$). As can be seen in Figure 5.10 for later time steps (e.g. t=3600 → t=13200), the particles are interacted with vortex shedding behind the obstacle and entrained in the region where the vortex shedding takes place. Contrarily, the fluid-particle flow simulation of the classical LBM-CA model (5.11) can not capture the appropriate fluid-particle interaction in the vortex shedding region where most of the particles flow across the area behind the obstacle in the axial direction towards the duct outlet. The current particle transport algorithm (i.e. Algorithm 2) through the N_{move} loop provides more probabilities to distribute particles within each LBM time step. As a result, applying this algorithm for each individual particle allows this model (modified LBM-CA model) to better describe the fluid-particle interaction in complex fluid flow problems.

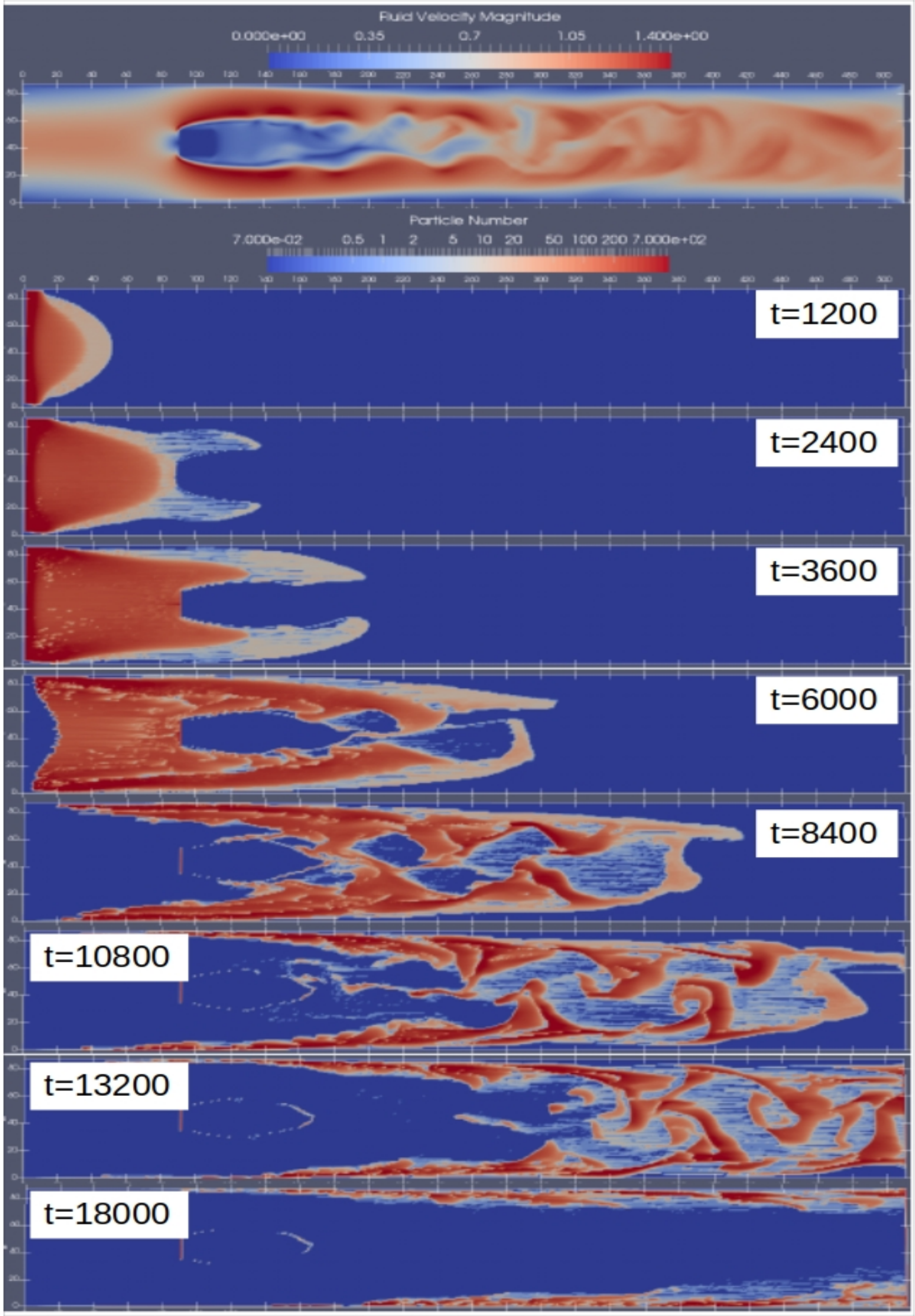


Figure 5.10: Modified LBM-CA method fluid velocity magnitude and particle number for different LBM time steps for $Stk = 0.1$ and $Re = 300$.

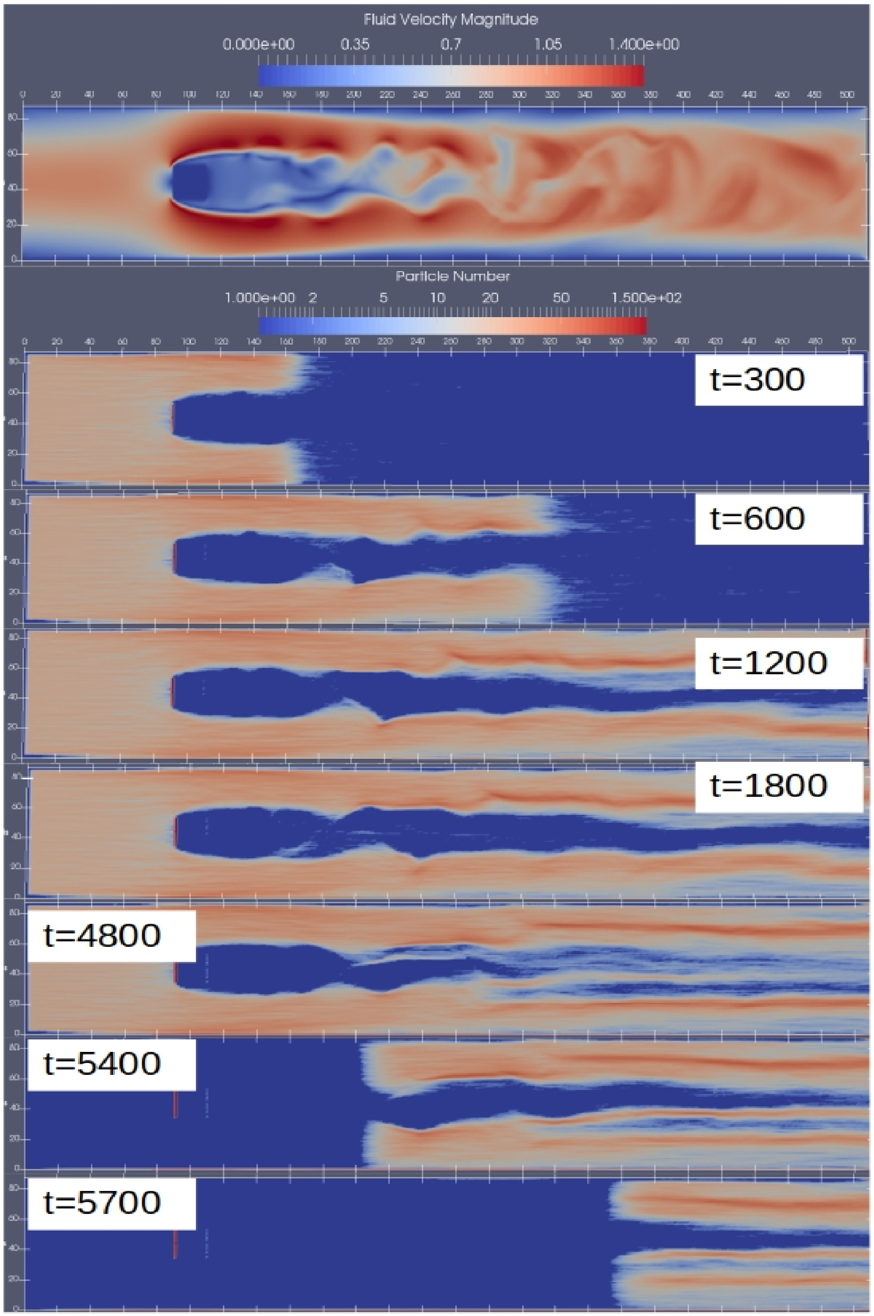


Figure 5.11: LBM-CA method fluid velocity magnitude and particle number for different LBM time steps for $Stk = 0.1$ and $Re = 300$.

5.3.3 Fluid-Particles Flow Through a Turbulent Impinging Jet

Fluid flow, particle transport and deposition through a turbulent impinging jet were examined in this simulation. Two test cases were run to validate this simulation. The first case ($Case1$) provided a qualitative study for the effect of jet-to-wall distance to jet diameter ratio (L/D) and Stokes number on the particle deposition pattern on the wall (impingement surface). The second test case ($Case2$) provided a quantitative comparison of the particle deposition density on the wall with an experimental work [86]. Figure 5.12 (a) and (b) show the impinging jet schematic for $Case1$ and $Case2$, respectively. The main difference between $Case1$ and $Case2$ is that the jet configuration in $Case1$ was set without convergence area which reduced the required computational resources for the case of large L/D ratio (e.g. 2, 4, and 6). While $Case2$ was used for $L/D = 0.5$.

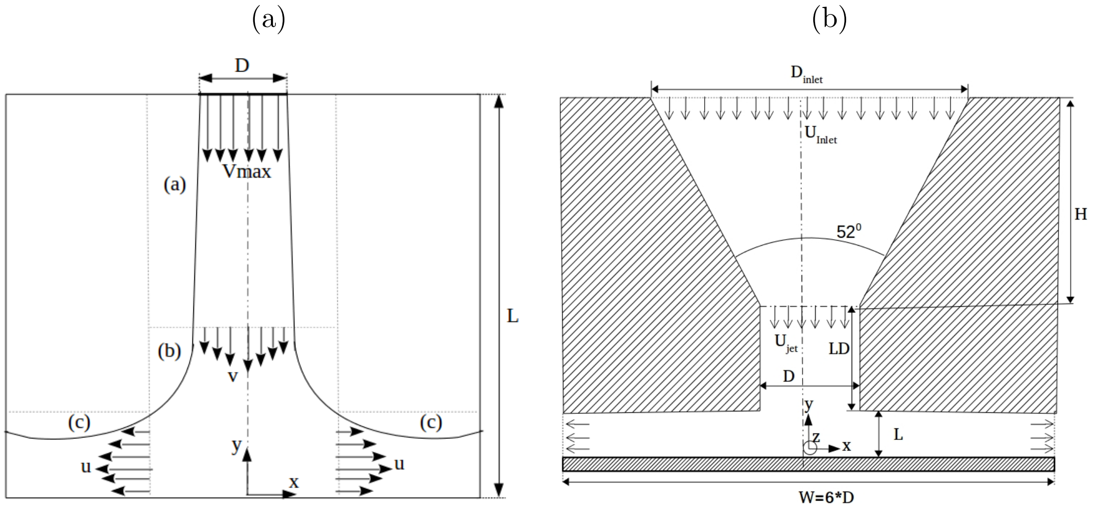


Figure 5.12: Impinging jet schematic; (a): $Case1$ and (b): $Case2$.

Simulation Validation for the Particle Transport and Deposition ($Case1$)

Three different cases were utilized to validate the current GPU particle laden flow code. The main difference in these cases is the ratio (L/D) between the jet-to-wall distance and the jet diameter (i.e. $L/D = 2, 4, \text{and } 6$). For the current simulation, to provide a qualitative comparison with the experimental work by Burwash et al. [7] and the numerical simulation by Alatawi [3] the following simulation settings were made: the LBM fluid simulation was run first for 50000 time steps before the particle injection to reach the steady state. After time step 50000, 20 particles were injected in each jet inlet node at each time step within 1000 time steps. In addition to the different L/D ratios, different Stokes numbers were used in the numerical simulations to examine their effects on the particle deposition pattern. Table 5.4 shows the simulation characteristics for different L/D values.

Table 5.4: Impinging jet (Case1) characteristics for $L/D = 2, 4,$ and 6.

Characteristics	$L/D = 2$	$L/D = 4$	$L/D = 6$
Jet diameter	80	80	80
Axial dimension (coarse grid)	160	320	480
Spanwise dimension (coarse grid)	800	800	800
Re	10000	10000	10000
Stk	0.11	0.11	0.11
Inlet jet velocity	0.04	0.04	0.04
Inlet jet perturbation	1	1	1
Grid refinement level	2	2	2
Refinement height (y/D)	0.15	0.15	0.15
Grid size Δx (coarse grid)	0.0125	0.0125	0.0125
Time step Δt (coarse grid)	0.0005	0.0005	0.0005
Grid size Δx (fine grid)	0.00625	0.00625	0.00625
Time step Δt (fine grid)	0.00025	0.00025	0.00025
Inlet boundary condition	Velocity BC (Section 3.8.2)	Velocity BC	Velocity BC
Outlet boundary condition	Fully developed BC (Section 3.8.4)	Fully developed BC	Fully developed BC
Wall boundary condition	Bounce back BC (Section 3.8.1)	Bounce back BC	Bounce back BC

Figure 5.13 depicts the particles trajectories for the case of $L/D = 6$ at different time steps (t). The most essential advantage of this model over the previous LBM-CA models (e.g. [64] and [99]) is the real physical time simulation that the current model can provide. This advantage can obtain better dynamic interaction between the particles and the instantaneous turbulence in the flow. To explain this advantage in more quantitative details, Figure 5.13 will be used. The initial particle velocity in the axial direction is the same as the fluid velocity at the jet inlet which is 0.04 (LBM dimensionless unit). In the downstream direction, inertia forces enable the particles to be less affected by the fluid velocity close to the wall. Consequently, it is expected that particles reach the wall with a velocity larger than the fluid velocity and slightly less than the particle initial velocity. For axial distance equal to 240 nodes (i.e. $D = 40$ nodes and $L/D = 6$), the expected number of time steps for all particles to reach the lower wall is ≥ 6000 time steps (i.e. $240/(\leq 0.04)$). As shown in Figure 5.13, the particles are deposited and leave the computational domain at a time step slightly more than 6000.

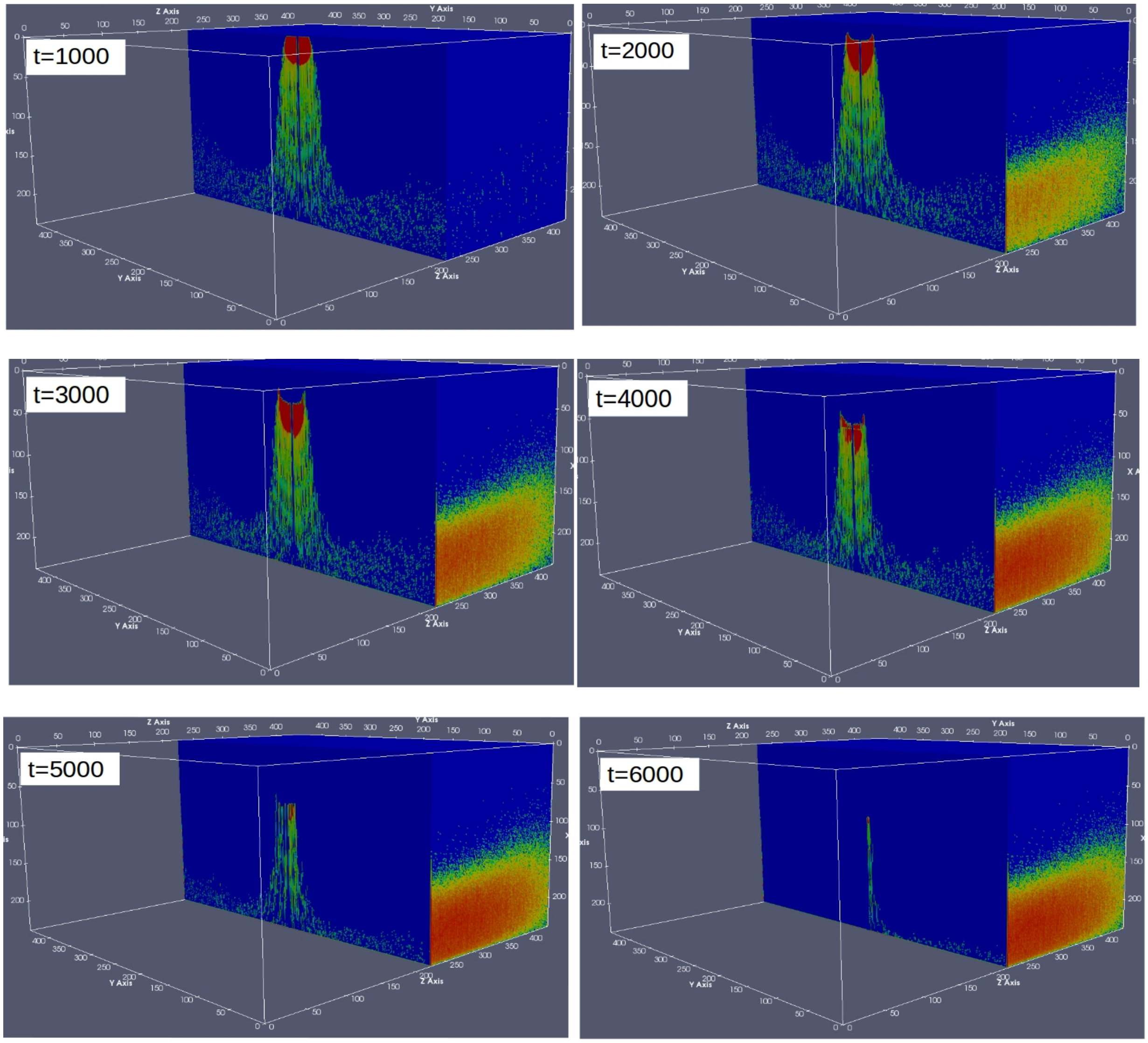


Figure 5.13: Particles trajectories for $L/D = 6$, $St = 0.11$, $Re = 10000$ and $Fr = 300$ at different time steps.

Figure 5.14 depicts the particle deposition profiles for different Stokes numbers at $L/D = 2$. To compare with previous works results ([7][3]), the total number of particles (N_{total}) deposited on the wall for the case of $St = 0.11$ was normalized by the number of 1.7×10^6 particles by multiplying the particle number at each wall node by a ratio factor α ($\alpha =$

$\frac{1.7\times10^6}{N_{total}}$) . For the other cases with different Stokes numbers, the same ratio factor is used to normalize the particles numbers.

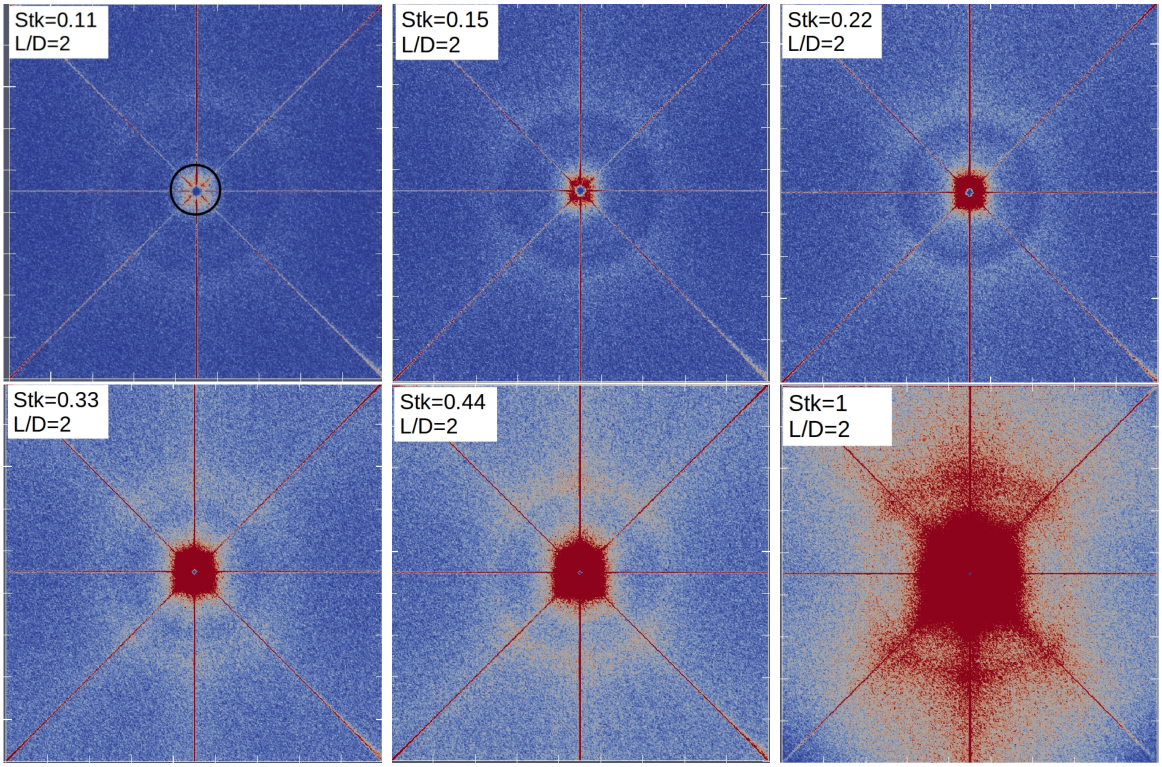


Figure 5.14: Particle deposition pattern on the impingement wall for different Stokes numbers at $L/D = 2$, and $D = 80$ nodes.

As shown in Figure 5.14 (St=0.11), there are two obvious rings where the particles are concentrated: the first ring has a diameter approximately the same as the jet diameter, and the second one is located at distance $\simeq 2.5D$ from the stagnation point. Particles which are initially injected within the jet inlet area obtain specified inertia because they have the inlet jet fluid velocity. In the downstream direction, the dispersion effect of the shear layer region starts to spread particles to the region outside the projection area of the jet. These particles obtain less ineria as a result of the drag force in the outer jet flow region which reduces the particles velocity. Subsequently, the particles with high inertia (i.e. within the jet projection area) can reach the wall and deposit on it producing the small ring particle profile. In contrast, the low axial momentum particles in the exterior area of the jet flow are carried by the radial flow close to the wall. At a distance $\simeq 2.5D$

from the stagnation point, the fluid radial velocity is decreased and the particles axial momentum and the gravity force dominate the particle movements which increase the particle deposition rate making the large ring shape. As shown in Figure 5.14 with Stokes number higher than 0.11, the particle deposition profiles retain the same trend as in case of $St = 0.11$. Increasing the Stokes number decreases the drag effect and raises the effect of gravity force on the particles, which leads to a higher deposition rate on both of the ring regions.

Figures 5.15 and 5.16 demonstrate the particles distribution on the impingement surface for cases $L/D = 4$ and $L/D = 6$, respectively. It can be seen from these figures that for the same Stokes number the total number of deposited particles on the wall are decreasing with increasing the jet to impingement surface distance. Increasing this distance provides more space for the jet flow to be developed in the axial direction. As a consequence, the established and deflection zones (Figure 1.1) become wider close the the wall which spreads the particles in the fluid flow region close to the wall. The particles dispersion affects the deposition on the wall within the small ring, making it thicker with a higher jet to wall distance. The spreading of both the axial fluid flow and the particle distribution within this flow increases the distance from the stagnation point where the large ring profile takes place. It can be seen in Figures 5.14, 5.15, and 5.16 that the particle deposition patterns within the small rings are not taken the round shape because the solid particles are not injected in a circular shape at the jet inlet (Cartesian coordinate system is used to represent the inlet jet nodes).

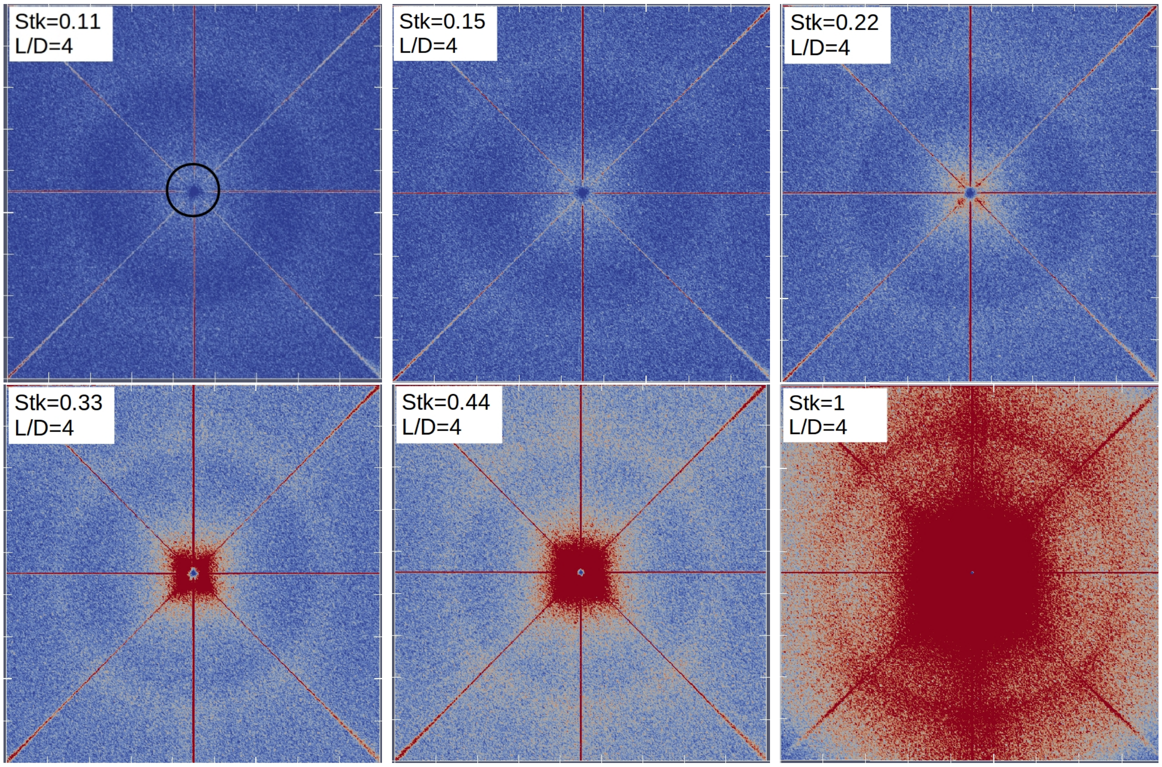


Figure 5.15: Particle deposition pattern on the impingement wall for different Stokes numbers at $L/D = 4$, and $D = 80$ nodes.

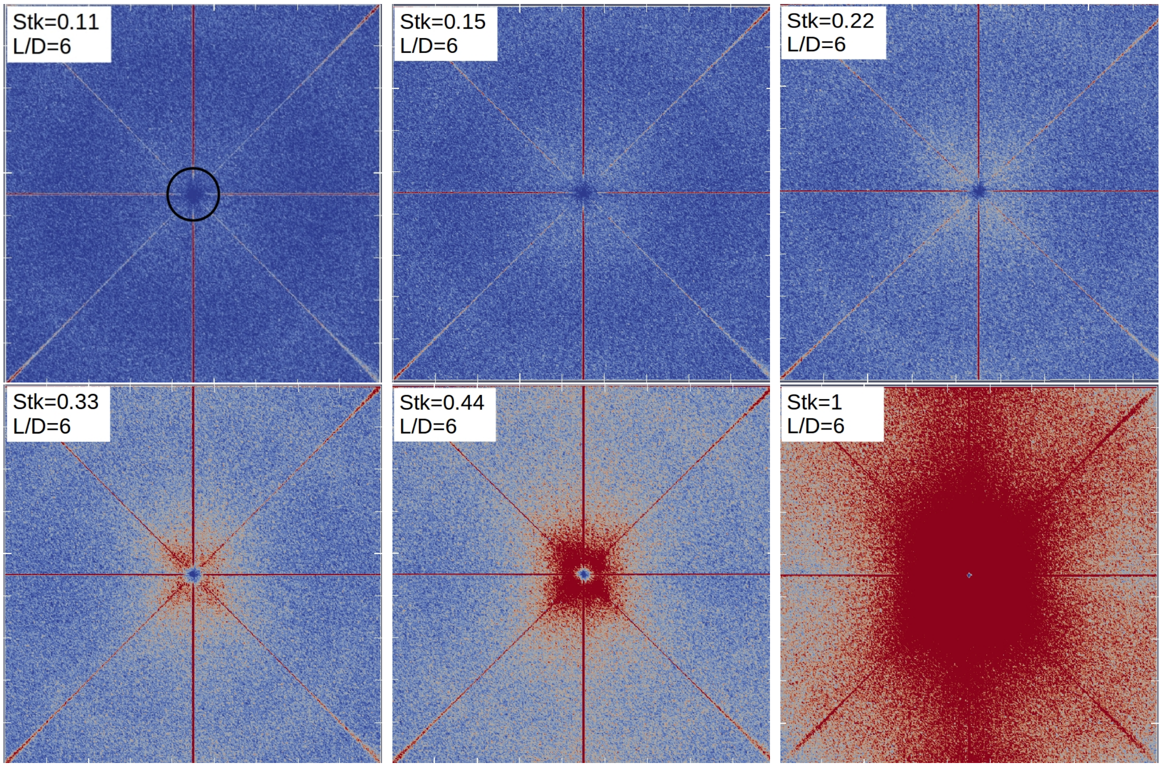


Figure 5.16: Particle deposition pattern on the impingement wall for different Stokes numbers at $L/D = 6$, and $D = 80$ nodes.

Simulation Validation for the Particle Transport and Deposition ($Case2$)

The 3D circular impinging jet used in this simulation is depicted in Figure 5.12 (b). The impinging jet configuration in this case ($Case2$) was used to provide a comparison with the experimental work by Sethi and John [86]. The jet physical diameter (D) is 1.484 mm which was set to 40 LBM nodes, and its length (LD) is the same as the jet diameter. The impingement surface was positioned at distance equal half the nozzle diameter from the jet exit (i.e. $L/D = 0.5$). The converging zone height (H) is set to be 100 LBM nodes; subsequently, the flow inlet diameter (D_{inlet}) is a function of the converging height and angle:

$$D_{inlet} = D + \frac{2H}{\tan 64^0}. \tag{5.20}$$

Particle diameter (p_d) in the experimental work was kept constant (i.e. $p_d = 3.0\ \mu m$), and the Stokes number was changed by modifying the flow rate (i.e. fluid characteristic velocity) through the jet. In this simulation, the dimensionless inlet velocity (U_{inlet}) was constant, and the simulation characteristics (e.g. Stokes number) were changed by choosing different Reynolds number (i.e. LBM fluid viscosity ν_{lbm}) to be compatible with the experimental work conditions.

Table 5.5 shows the Stokes number effect on some of the simulation parameters in the physical and LBM units. The parameters C_U, C_L, and C_T in this table are the velocity, length, and time conversion factors between the LBM units and their equivalent physical units. For example, each LBM time step equals to 1.61672×10^{-8} seconds in the physical time unit for the case of $Stk = 2.56$. Ui_{real} and Uj_{real} are the dimensional fluid velocities at the inlet and jet regions (i.e. D_{inlet} and D in Figure 5.12 (b)).

Table 5.5: Impinging jet simulation parameters for different Stokes numbers.

Stk	2.56	1.96	1.21	0.49	0.23
Ui_{real} (m/s)	4.82798	3.69642	2.28197	0.924105	0.434518
Uj_{real} (m/s)	53.1078	40.6606	25.1017	10.1652	4.7797
Re	4926	3771	2328	945	443
ν_{lbm}	1.786×10^{-4}	2.333×10^{-4}	3.779×10^{-4}	9.333×10^{-4}	1.985×10^{-3}
ω	1.99786	1.9972	1.99547	1.98886	1.97646
$C_U(m/s)$	2414	1848.21	1141	462.053	217
$C_L(m)$	3.805×10^{-5}	3.805×10^{-5}	3.805×10^{-5}	3.805×10^{-5}	3.805×10^{-5}
C_T (s)	1.6167×10^{-8}	2.1115×10^{-8}	3.4204×10^{-8}	8.44641×10^{-8}	1.79633×10^{-8}

Figure 5.17 depicts the particle normalized velocity magnitude (normalized by the fluid average velocity in the jet U_{jet}) on the impingement surface at different Stokes numbers. The horizontal axis in this figure represents the normalized radial direction (x/D). As was explained in Section 5.3.3, the effect of the inertia force of the high Stokes number particles (e.g. $Stk = 2.56$) produces high particle velocity in a spot diameter about 0.8, and particles with low velocity are deposited on the wall in the diameters between 0.8 and 1. The radial fluid velocity (momentum) drags particles with lower Stokes number (e.g. $Stk = 1.96 - 0.23$) into the region of larger diameters and reduces the particle velocity close to the stagnation point.

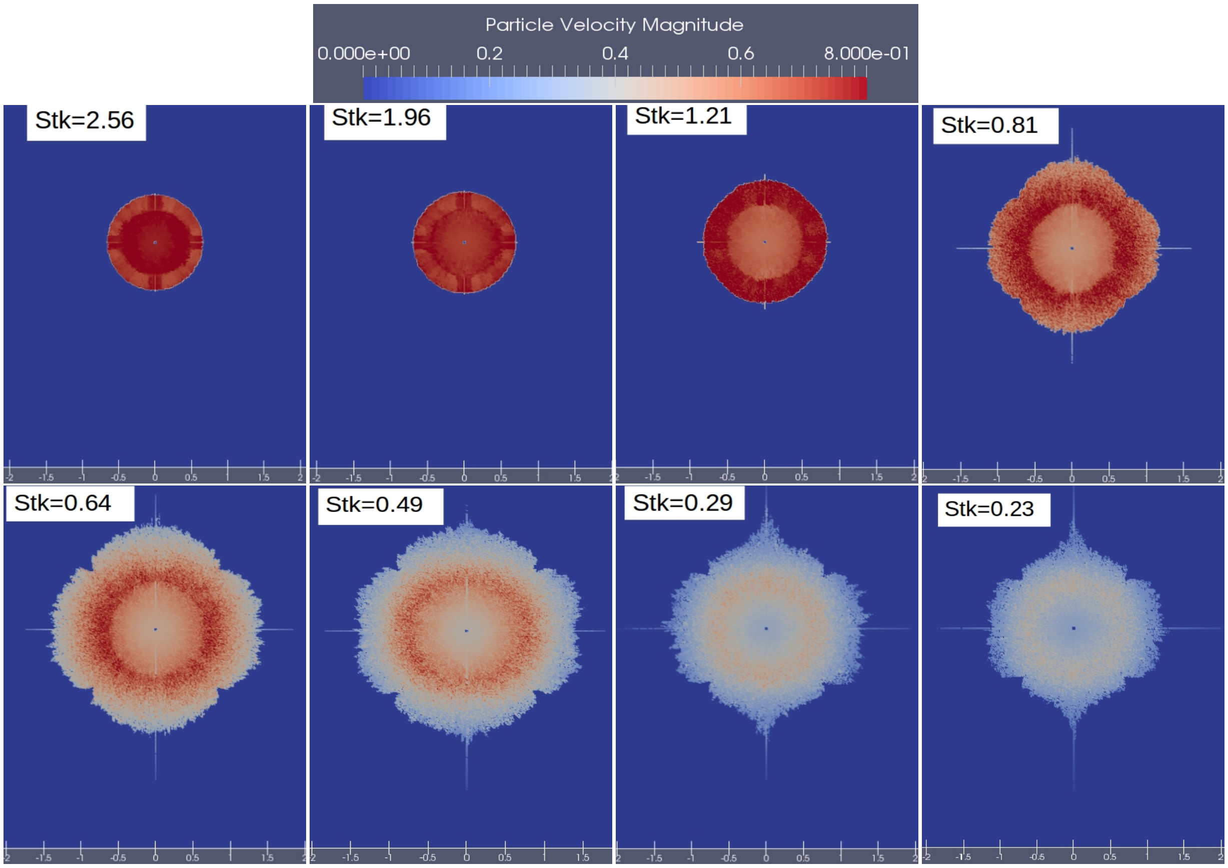


Figure 5.17: Particle deposition velocity magnitude

Figure 5.18 shows the particle deposition density on the wall with the radial distance from the stagnation point for different Stokes numbers. As can be seen in this figure that the deposition diameters obtained from experiments are less than those of the simulations. For Stokes number equal to 2.56, the experimental result shows that the deposition diameter is 1.2 mm while the jet diameter is 1.484 mm. This means that the particle locations in the nozzle during the flow are far from the jet cylindrical surface. The experimental configuration is probably the major cause of this phenomenon where the particles flow through a long pipe before the jet. The numerical simulation is restricted by the available computational resources (e.g GPUs memory size) which reduce the ability to build a simulation completely compatible with this real world phenomenon.

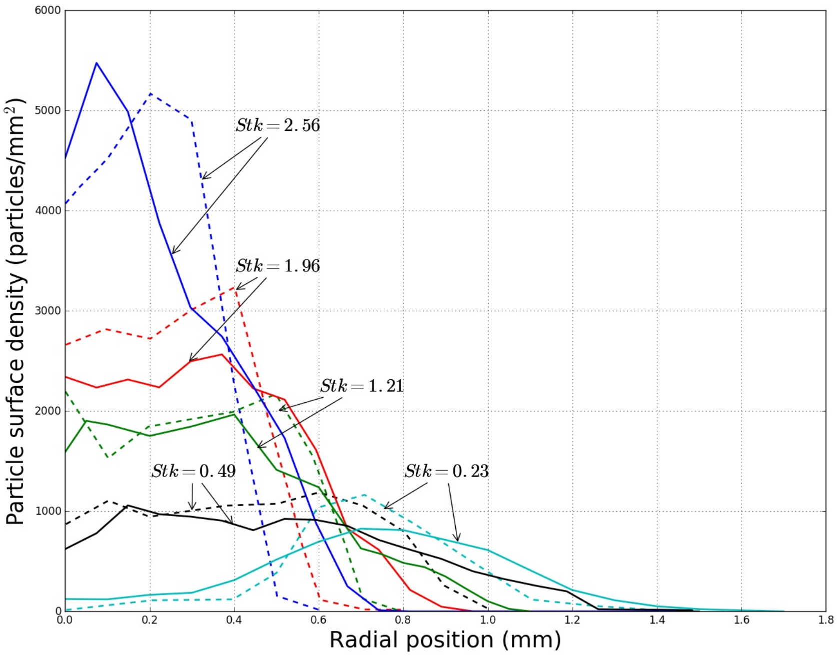
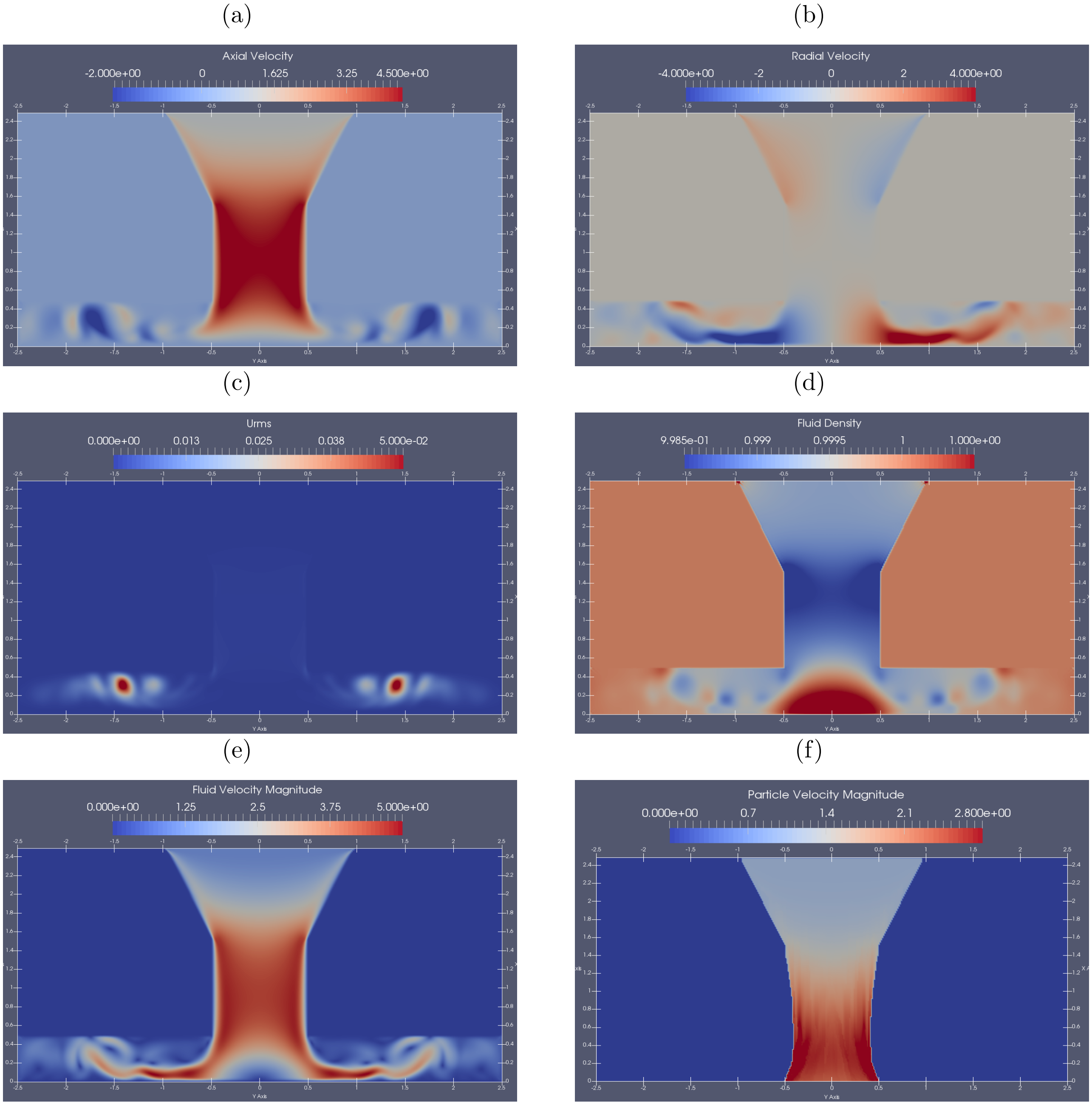


Figure 5.18: Particles deposition density for different Stokes numbers; dotted line: experimental results [86]; solid lines: simulation results (this work).

Figures 5.19, 5.20, and 5.21 show some of the LBM simulation results for different Stokes and Reynolds numbers. The sub-figures in these figures are: (a): instantaneous fluid axial velocity; (b): instantaneous fluid radial velocity; (c) fluid axial turbulence intensity; (d): instantaneous fluid density (pressure); (e): instantaneous fluid velocity magnitude; (f): solid particle mean velocity magnitude. The dimensions in these figures are normalized by the jet diameter, and the velocities are normalized by the inlet flow velocity.

Figure 5.19 is for the case of Re = 4926 and Stk = 2.56. As shown in this figure (a), the fluid axial velocity diminishes close to the wall. Particles with high stokes number (i.e. $Stk = 2.56$) are driven by their high inertia and will not follow the flow streamlines close to the wall. Consequently, these particles have velocity lower than the fluid velocity, and they are deposited on the wall within a circle with diameter almost the same as jet diameter, as shown in Figure 5.19 (f). It was seen that the axial velocity fluctuations (Figure 5.19 (c)) are very low near the wall surface, and there is no considerable effect of these fluctuations on the particle deposition rate.

At low Reynolds and Stokes numbers (Figures 5.20), the particles are dragged by the fluid flow due to the low stokes number. In the flow domain, the particles have almost the same fluid velocity and they follow the flow streamlines. As shown in Figures 5.21, increasing the Reynolds number and decreasing the Stokes number (i.e. $Re = 10000$ and $Stk = 0.1$) affects significantly the particle transport and deposition behaviours. Flow with high Reynolds number will generate vortices close to the wall, where these vortices can be seen in Figures 5.21 (b). Particles in this case are lifted by the vortices and some of them are deposited on the upper wall of the flow domain (i.e. $y/D = 0.5$).



(a) (b) (c) (d) (e) (f)

Figure 5.19: Fluid-particle impinging jet flow simulation results for Stokes number $Stk = 2.56$ and Reynolds number $Re = 4926$.

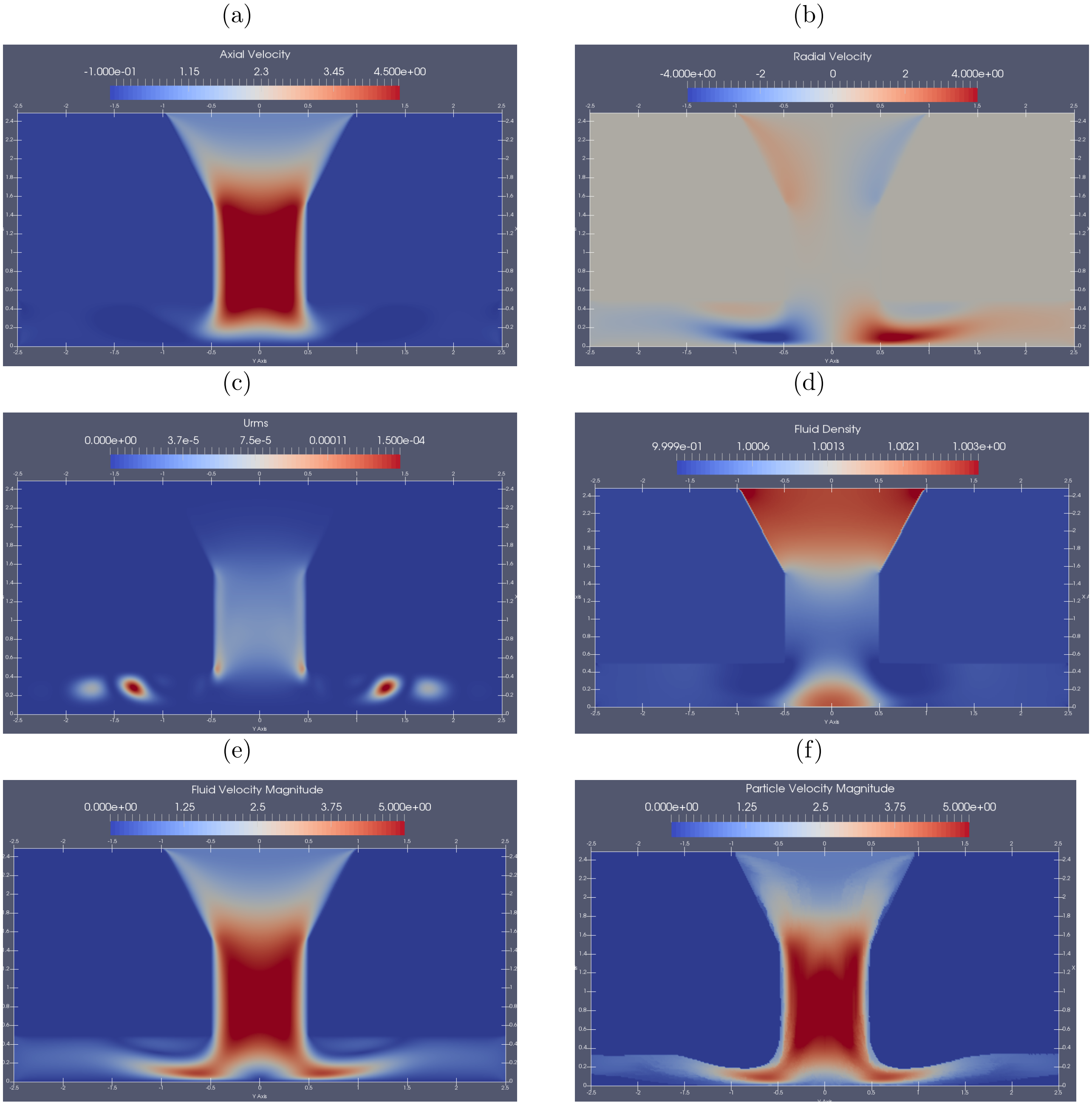


Figure 5.20: Fluid-particle impinging jet flow simulation results for Stokes number $Stk = 0.23$ and Reynolds number $Re = 443$.

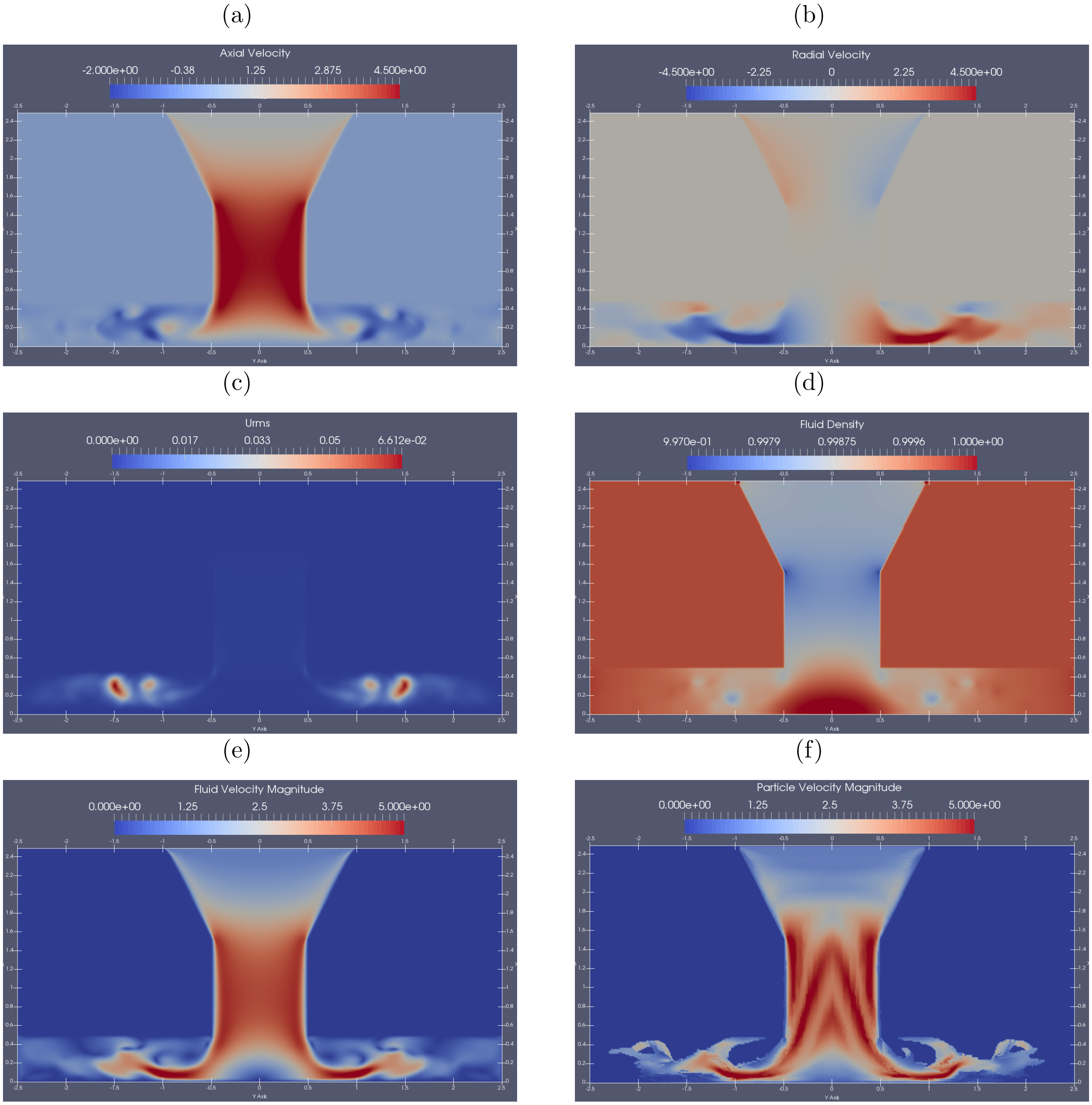


Figure 5.21: Fluid-particle impinging jet flow simulation results for Stokes number $Stk = 0.1$ and Reynolds number $Re = 10000$.

Chapter 6

Lagrangian-DEM Four-way Coupling Particle Tracking Model

Another model is proposed in this section to incorporate the feedback effect of the (Lagrangian) dispersed solid particles phase on the fluid flow (two-way coupling). A discrete Boltzmann equation with fluid-solid particles collision term is proposed as:

$$f_i(x + c_i\Delta t, t + \Delta t) = f_i(x,t) - \underbrace{\frac{1}{\tau}(f_i(x,t) - f_i^{eq}(x,t))}_{\text{fluid-fluid collision}} - \underbrace{f_{\bar{i}}^{\text{P-collision}}(x,t)}_{\text{fluid-solid particle collision}}. \tag{6.1}$$

The right-hand side term of Equation 6.5.1 (i.e. $f_{\bar{i}}^{\text{P-collision}}(x,t)$) represents the bounced-back part of the distribution function after colliding moving solid wall with direction opposite to the incoming distribution function $f_i(x,t)$ (Section 3.8.2). Recall that for the regular bounce back boundary condition (Section 3.8.1), the solid wall is located at the half link distance between the lattice nodes where the bounce back process is accomplished within one time step. As shown in Figure 6.1, the solid particles in the flow are distributed along the lattice links randomly and the bounce back process is accomplished in different time steps. The bounced back part can be written as follows:

$$f_{\bar{i}}^{\text{P-collision}}(x,t) = \int_{x}^{x+\Delta x} \int_{t}^{t+\Delta t} f_{\bar{i}}^{\text{P-collision}}(x',t')dt'dx', \tag{6.2}$$

where x' and t' are the spatial and temporal subgrid scales within the LMB spatial and temporal grid scales, respectively. Obviously, the integration in Equation 6.3 is very complicated and needs to be simplified. To simplify the collision term, the following assumptions are proposed:

- solid particles are distributed uniformly around the lattice nodes,
- the bounced back term is proportional to the solid particles volume fraction and velocity,
- the collision process is completed in one time step.

The bounce back boundary condition with moving wall is presented in Section 2.4 and it can be re-written in a new form to incorporate the moving solid particles effect instead of the wall:

$$\underbrace{f_i(x + c_i\Delta t, t + \Delta t)}_{\text{net streaming}} = \underbrace{f_i(x,t)}_{\text{before collision}} \underbrace{- \frac{1}{\tau}(f_i(x,t) - f_i^{eq}(x,t))}_{\text{fluid-fluid collision}} - \underbrace{\Phi_v 6 w_i \rho_p \langle \mathbf{U}_p \rangle \cdot \mathbf{c}_i.}_{\text{P-collision}} \tag{6.3}$$

The local solid particles volume fraction Φ_v can be defined for a uniform lattice with length Δx containing n_p particles as follows:

$$\Phi_v = n_p \frac{\pi}{6} \left(\frac{d_p}{\Delta x} \right)^3. \tag{6.4}$$

The $\langle \mathbf{U}_P \rangle$ is the solid particle phase velocity and it is defined as the ensemble average for all particles within the lattice volume:

$$\langle \mathbf{U}_p \rangle = \left(\sum_{i=1}^{n_p} \mathbf{u}_{p_i} \right) / n_p. \tag{6.5}$$

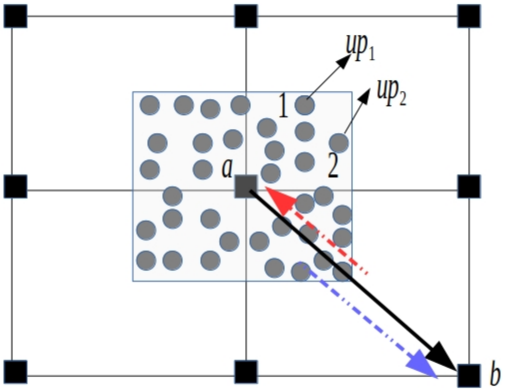


Figure 6.1: Particle distribution in an LBM cell.

6.1 Particle Motion

The linear motion of a discrete particle p with mass m_p and velocity $\mathbf{U}_p$ can be described by applying Newton's second law of motion:

$$m_p \frac{d\mathbf{U}_p}{dt} = \mathbf{F}_{drag} + \mathbf{F}_{buo} + \mathbf{F}_g + \mathbf{F}_{coll} + \mathbf{F}_{other}, \tag{6.6}$$

where $\mathbf{F}_{drag}$, $\mathbf{F}_{buo}$, $\mathbf{F}_g$, and $\mathbf{F}_{coll}$ are the drag, buoyancy, gravity, and collision force, respectively. The $\mathbf{F}_{other}$ represents other external forces such as Basset force, Magnus force, Saffman force, and Brownian force. Because of the high particle to fluid density ratio, only drag, buoyancy, gravity, and collision forces are considered in the present work. The force $\mathbf{F}_{coll}$ is the sum of the collision forces exerted by all particles, which are included in a contact list ($contact - list$), in contact with particle p, and this force is divided in to two main components, i.e. normal and tangential forces:

$$\mathbf{F}_{coll} = \sum_{l=1}^{contact-list} \left(\mathbf{F}_{p\leftrightarrow l,n} + \mathbf{F}_{p\leftrightarrow l,t}\right). \tag{6.7}$$

In addition to the linear motion, the rotational motion of particle p should also be considered as a result of the tangential contact force:

$$I_p \frac{d\omega_p}{dt} = \mathbf{T}_p, \tag{6.8}$$

where I_p is the particle moment of inertia, the angular velocity is ω_p, and the torque $\mathbf{T}_p$ is defined when the tangential contact force is present:

$$\mathbf{T}_p = \sum_{l=1}^{contact-list} (R_p \mathbf{n}_{pl} \times \mathbf{F}_{p\leftrightarrow l,t}), \tag{6.9}$$

whith R_p the radius of particle p, and $\mathbf{n}_{pl}$ is the normal unit vector between the colliding particle centers (i.e particle p and l centers).

6.1.1 Contact Force Model

In this study, the spring model proposed by Cundall and Strack [17] is employed to calculate the contact force between the colliding particles as shown in Figure 6.2 (c). The normal force exerted on two particles in contact (e.g. p_i and p_j) is computed as:

$$\mathbf{F}^c_{p_i\leftrightarrow p_j,n} = -k_n \eta \mathbf{n}_{p_i p_j} - \xi_n \mathbf{U}_{p_i p_j,n}, \tag{6.10}$$

$$\mathbf{U}_{p_i p_j,n} = \left[(\mathbf{U}_{pi} - \mathbf{U}_{pj}) \cdot \mathbf{n}_{p_i p_j}\right] \mathbf{n}_{p_i p_j}, \tag{6.11}$$

$$\mathbf{n}_{p_i p_j} = \frac{\mathbf{X}_{pi} - \mathbf{X}_{pj}}{\mid \mathbf{X}_{pi} - \mathbf{X}_{pj} \mid}, \tag{6.12}$$

$$\xi_n = \frac{-2 \ln e \sqrt{\frac{m_{pi} m_{pj}}{m_{pi} + m_{pj}} k_n}}{\pi^2 + (\ln e)^2}, \tag{6.13}$$

where k_n, η, ξ_n, and e are the normal spring stiffness, inter-particle overlap, normal damping coefficient, and the restitution coefficient for the normal collision, respectively. In this study, a collision between two particles is considered if the distance (Distance) between these particle centers is less than the particle diameter (D_p) by a specified factor (i.e. Distance $< 0.99 D_p$). The tangential contact force between the two particles is calculated according to the normal force as follows:

$$\mathbf{F}^c_{p_i\leftrightarrow p_j,t} = -\mu \mid \mathbf{F}^c_{p_i\leftrightarrow p_j,n} \mid \mathbf{t}_{p_i p_j}, \tag{6.14}$$

$$\mathbf{t}_{p_ip_j} = \frac{\mathbf{U}_{p_ip_j,t}}{\mid \mathbf{U}_{p_ip_j,t} \mid}, \tag{6.15}$$

$$\mathbf{U}_{p_ip_j,t} = \mathbf{U}_{p_ip_j} - \mathbf{U}_{p_ip_j,n}, \tag{6.16}$$

where μ is the friction coefficient, and $\mathbf{t}_{p_ip_j}$ is the unit vector in the tangential direction.

6.2 Inter-Phase Momentum Transfer

According to Yang et al. [105], the fluid force (drag force) acting on a particle is:

$$\mathbf{F}_{f\to p} = \frac{V_p\beta}{1-\epsilon_f}(\mathbf{U}_f - \mathbf{U}_p), \tag{6.17}$$

where V_p, ϵ_f, $\mathbf{U}_f$, and $\mathbf{U}_p$ are the particle volume, fluid voidage, fluid velocity, and the particle velocity, respectively. The correlated drag coefficient factor (β) is calculated based on the heterogeneous particle distribution inside the fluid cell [105], and the subscript f in Equation 6.17 refers to the fluid properties calculated at the solid particle location. The standard drag coefficient for individual particle is:

$$C_d = \begin{cases} \frac{24}{Re_p}\left(1 + 0.15 Re_p^{0.687}\right), & Re_p < 1000, \\ 0.44, & Re_p \geq 1000, \end{cases} \tag{6.18}$$

and the particle Reynolds number (Re_p) is defined as:

$$Re_p = \frac{\epsilon_f \rho_f \mid \mathbf{U}_f - \mathbf{U}_p \mid d_p}{\mu_f}. \tag{6.19}$$

The corrected drag coefficient factor is calculated from the following formula [105]:

$$\beta = \frac{3}{4}\frac{(1-\epsilon_f)\rho_f |\mathbf{U}_f - \mathbf{U}_p|}{d_p} \cdot C_d \cdot \psi, \tag{6.20}$$

and the correction factor (ψ) for the standard drag coefficient is obtained as:

$$\psi = \frac{0.0214}{4(\epsilon_f - 0.7463)^2 + 0.0044} - 0.5760, \qquad (0.74 \leqslant \epsilon_f \leqslant 0.82) \tag{6.21}$$

$$\psi = \frac{0.0038}{4(\epsilon_{f_p} - 0.7789)^2 + 0.0044} - 0.5760, \qquad (0.82 \leqslant \epsilon_{f_p} \leqslant 97) \tag{6.22}$$

$$\psi = 32.8295\epsilon_f - 31.8295. \qquad (\epsilon_f > 0.97) \tag{6.23}$$

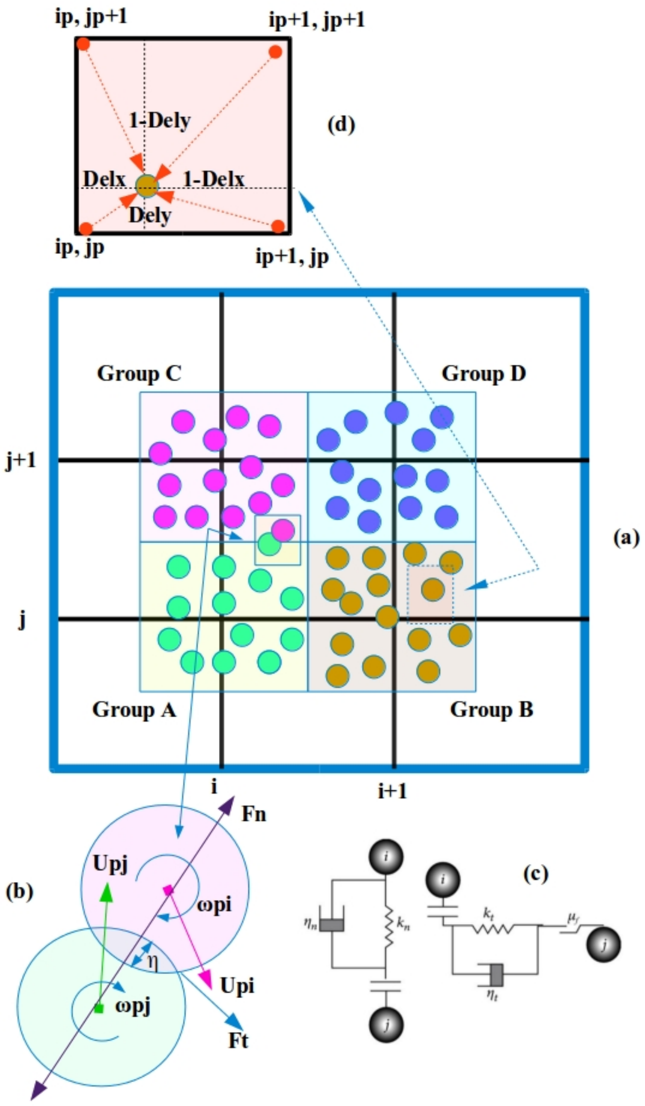


Figure 6.2: Schematic of contact force model; (a): particle grouping in LBM lattices; (b): two contacting particles; (c) spring model; (d): interpolation for the particle location.

6.3 Particle Searching Algorithm

In the DEM, searching for colliding particles is very crucial process, and it is simple and straightforward to search for all particles in the computational domain. However, for a large number of particles in the domain, this technique will be very expensive computationally (i.e. very large computational time is needed). The neighbor list method was first developed in the molecular dynamics simulation field to speedup the simulation [5]. The neighbor list contains a list of particles which are within a cut-off distance ($S_{cut-off}$) determined based on the required computational accuracy, as shown in Figure 6.3. The cut-off distance should be defined carefully; choosing very small cut-off value may lead to excluding some particles (outside $S_{cut-off}$) which are involved in collisions with particles inside the cut-off distance. On the other hand, too large cut-off value will reduce the computational efficiency (e.g. increases the computational time).

In this work, a new searching technique is proposed based on the LBM scheme. As shown in Figure 6.2 (a), particles in a two-dimensional LBM computational domain are grouped according to their center locations (e.g. Group A, B, C, and D). These particle are arranged in a cell list (i.e. belongs to the LBM node i, j), and the searching process is consider for particles in this list. The Grouping procedure is accomplished in a consecutive way starting from the left to the right side of the 2D computational domain (e.g. the next cell list is for the LBM node $i + 1, j$ in Figure 6.2 (a)). This searching technique is very efficient because there are no particles will be excluded from the searching algorithm, and it is very fast compared with the all particles searching method, e.g. about 50 times faster than all particles searching method for the fluidized bed test case (Section 6.5.1).

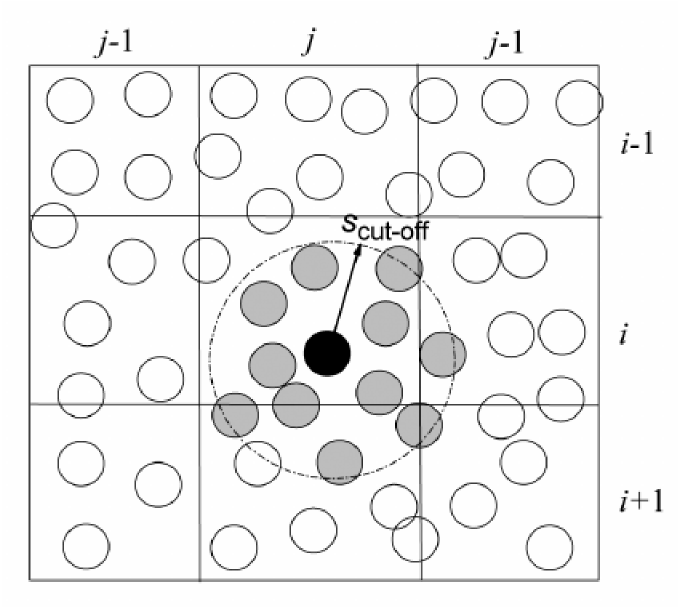


Figure 6.3: Neighbor list searching method.

6.4 Simulation Procedure

In this model, solid particles motion is described in the fluid flow domain utilizing the Lagrangian tracking approach; however, the fluid flow is defined in Eulerian nodes (i.e. LBM nodes). Consequently, the center coordinates of the solid particles rarely coincide with the LBM node locations during the simulation time. For that, each particle is linked to an LBM node according to its nearest node. Subsequently, solid particles are grouped and related to the LBM fluid nodes, e.g., Group A, B, C, and D in Figure 6.2 (a). The first step in our simulation is setting up the initial fluid and particle configuration, and then the simulation is carried out at each time step according to the following steps (Figure 6.4):

1. Calculate the nearest LBM node for each particle (e.g. node i_p, j_p in Figure 6.2 (d)), and insert this particle in an array related to this LBM node.
2. According to the total number of particles inside the LBM lattice, find the volume fraction of the fluid and particle phases.

3. Find the fluid local properties (e.g, velocity and volume fraction) at the particle location utilizing the bilinear or trilinear interpolation for the two and three dimensional flows, respectively.
4. Calculate the fluid drag force on the particles from Equation 6.17.
5. Applying the searching algorithm to find particles which are within the contact distance (less than the particle diameter) and calculate the normal and tangential forces (Equations 6.10-6.16).
6. Adding all the external forces exerting on the particles (e.g., drag, contact, gravity, and buoyancy force) and update the particle velocities (linear and rotational) and locations (Equations 6.6 and 6.8).
7. According to the solid particles velocity and volume fraction, calculate the solid phase effect term (the right-hand side term of Equation 6.3) on the fluid flow.
8. Applying the LBM equation (Equation 6.3) after adding the particle phase effect.

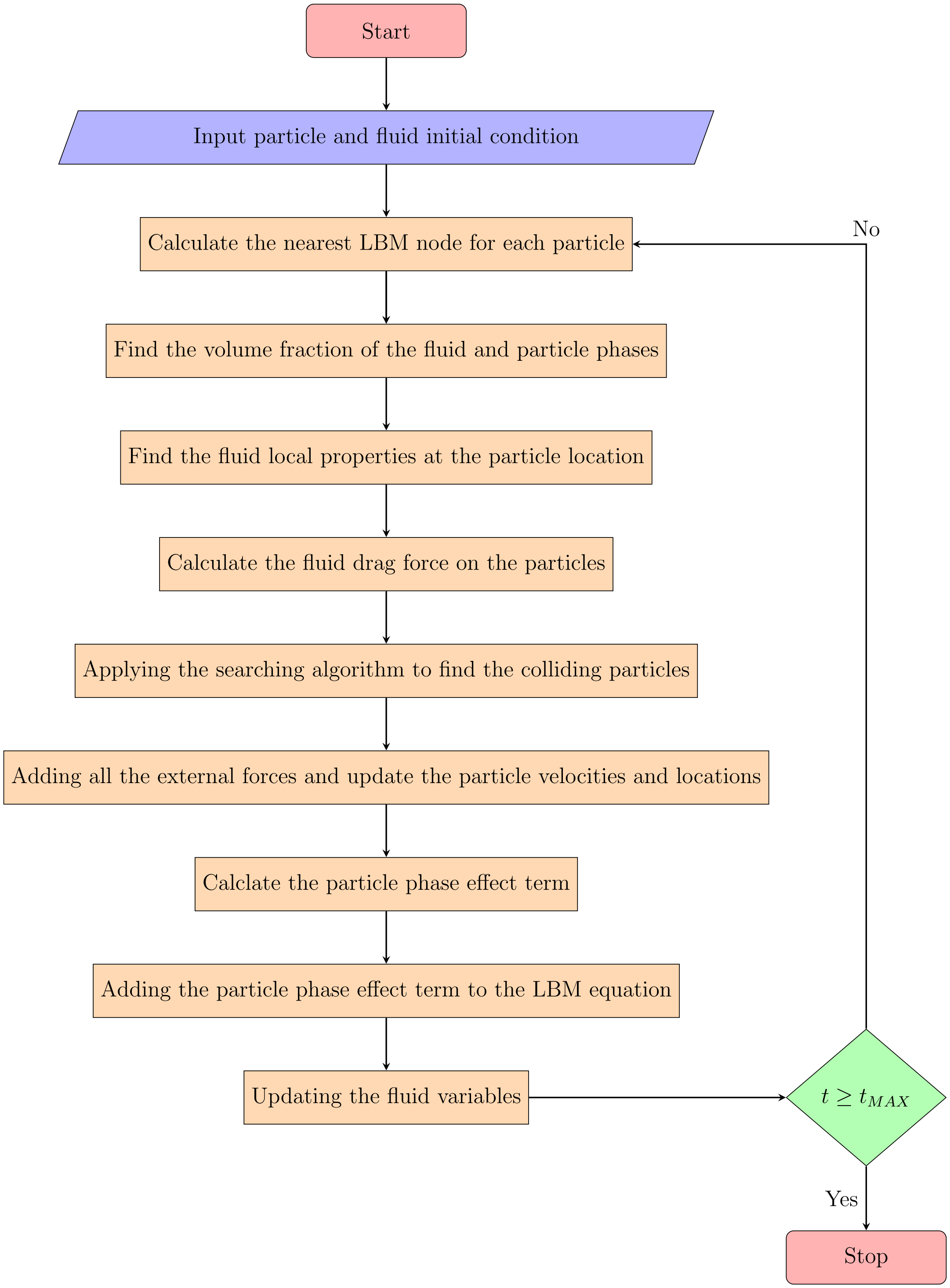


Figure 6.4: Simulation steps for the LBM-DEM model.

6.5 Results and Discussion

Two benchmark test cases were used to validate the present simulations: a bubble raised in a fluidized bed [101], and particle deposition in a turbulent impinging jet [86] .

6.5.1 A Single Bubble Injected in a Fluidized Bed

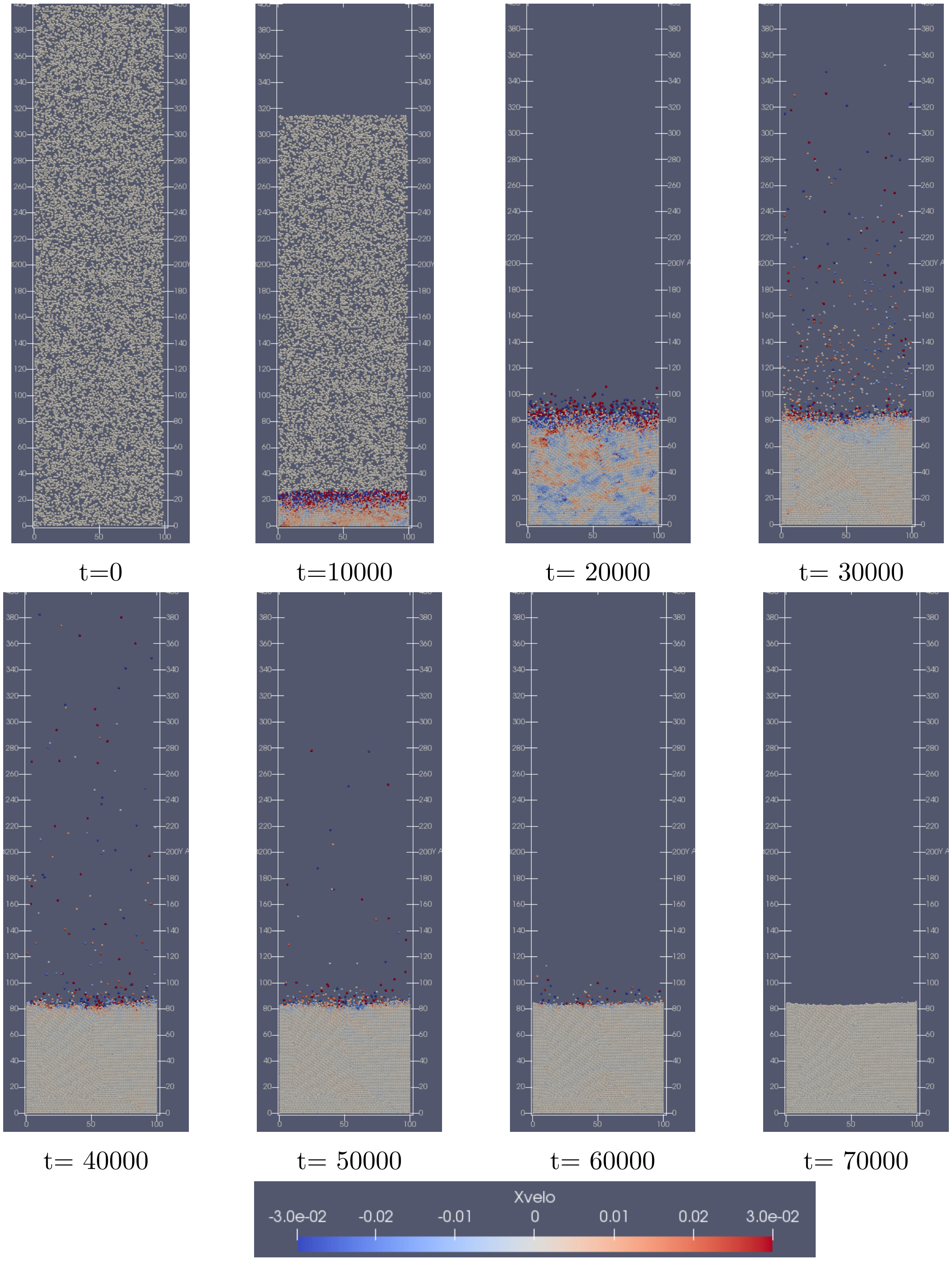
The simulation parameters used in this test case are summarized in Table 6.1.

Table 6.1: Simulation parameters for the fluidized bed at injection velocity $u_i = 0.8\ m/s$

Parameter	Physical unit	LBM unit
Bed width	0.00675 (m)	100
Bed height	0.027 (m)	400
Inlet flow width	0.00081 (m)	12
LBM cell size	6.75×10^{-5} (m)	1
Time step	3.375×10^{-6} (s)	1
Inlet velocity	0.8 (m/s)	0.04
Fluid density	$1.1795\ kg/m^3$	1
Particle density	$930\ kg/m^3$	788.46
Particle diameter	5.4×10^{-5}	0.8
Coefficient of restitution	0.9	0.9
Particle number	15000	15000
Spring stiffness	7 N/m	0.0176621

Initial Condition

In this test case, 15000 solid particles were initially positioned randomly in a fluidized bed container. The motion of the particles was generated due to the gravity and the inter-particle forces. A final packing configuration is achieved after sufficient time of simulation. Figure 6.5 depicts the particle configurations at different time steps starting from the initial particle distribution time ($t = 0$). The time in this figure is measured based on the dimensionless LBM time unit.


t=0 t=10000 t= 20000 t= 30000

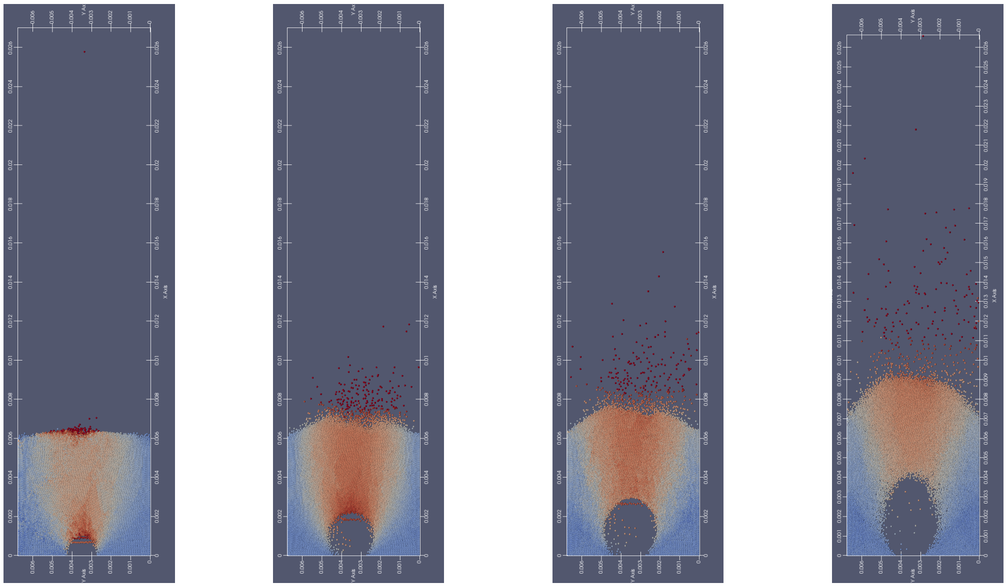
t= 40000 t= 50000 t= 60000 t= 70000

Figure 6.5: Particle axial velocities and configurations at different LBM time steps.

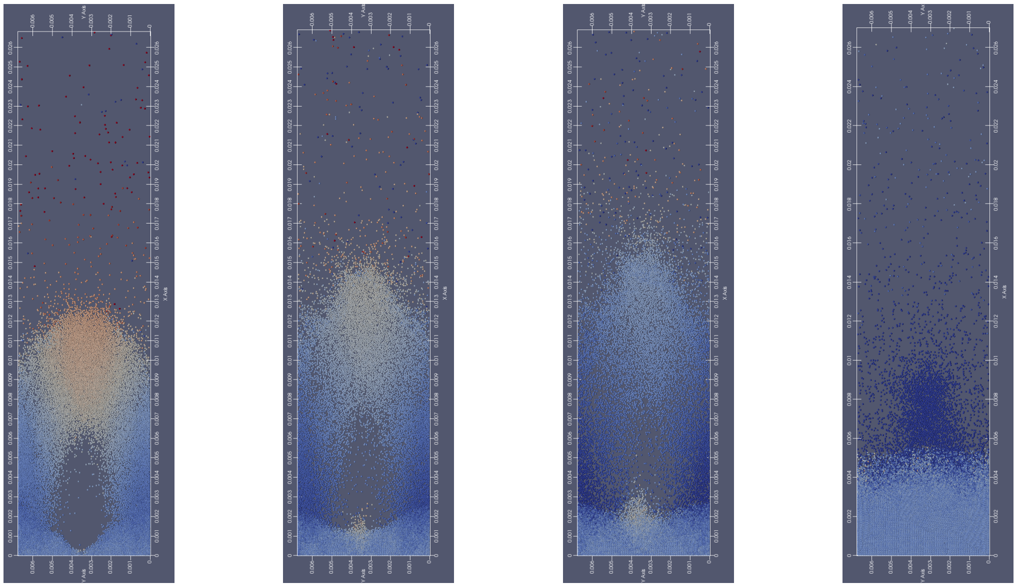
Starting the Flow

After reaching a suitable time step (e.g. $t = 7000$ in Figure 6.5), all information (e.g. particle locations, forces, and torques) is saved in the computer memory and to be used as an initial state. A bubble was injected at the bottom of the fluidized bed at different injection velocities based on the same initial condition, and the injection was continued for 0.006 seconds. The simulation results were utilized to provide a comparison with those of Wang et al. [101]. Figure 6.6 and Figure 6.7 show particle axial and spanwise velocity distributions in the fluidized bed at different time steps where the coordinate origin point was chosen to be at the center of the lower wall. The growth of the spherical bubble can be observed starting from an initially small perturbation (Figure 6.6 (a)) at the beginning of the fluid flow. Even though the fluid flow lasts 0.006 seconds, the growth is continued after this time causing increments in the horizontal and vertical bubble size. It can be also seen that the shape of the bubble is developed from a spherical shape (Figure 6.6 (b)) to an elliptical bubble (Figures 6.6 (c)-(e)). A kidney bubble detached from the bed bottom is constructed as shown in Figures 6.6 (f)-(g). Finally, the bubble shape is vanishing as seen in Figure 6.6 (h). These phenomena are in good qualitative agreement with the previous simulation results of Wang et al. [101].

The effect of the particle-particle interaction can be seen in Figure 6.6 (a) where the top surface of the particle phase exhibits maximum particle axial velocity. Particles at this surface start to rise due to the contact force exerted from the lower particle layer where the lower particles are driven by the injected fluid at the lower wall. Gravity force on the solid particles is also considered in this simulation, and its effect on the particles movement can be observed in Figure 6.7. The fluid flow drags particles far from the bubble causing a spanwise particles movement (Figure 6.7 (b)). Consequently, the gravity force exerting from the upper on the lower particles (beside the bubble) pushes the lower particles in to the bubble area. The particle entrained from the both sides (right and left) of the bubble eventually collide, and this collision forces the particles to rise and the kidney-shape bubble to be built, as shown in Figures 6.7 (f)-(g).



(a) t=0.0027 (b) t=0.0060 (c) t= 0.0081 (d) t= 0.0136



(e) t= 0272 (f) t= 0.0409 (g) t= 0.0545 (h) t= 0.1091

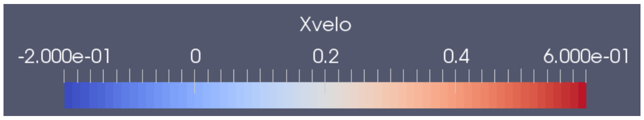
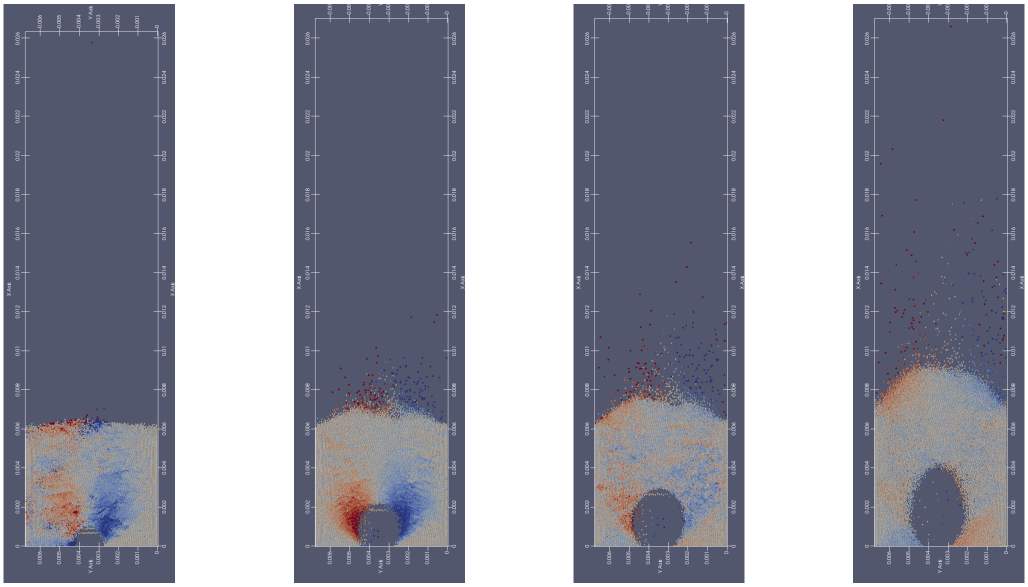
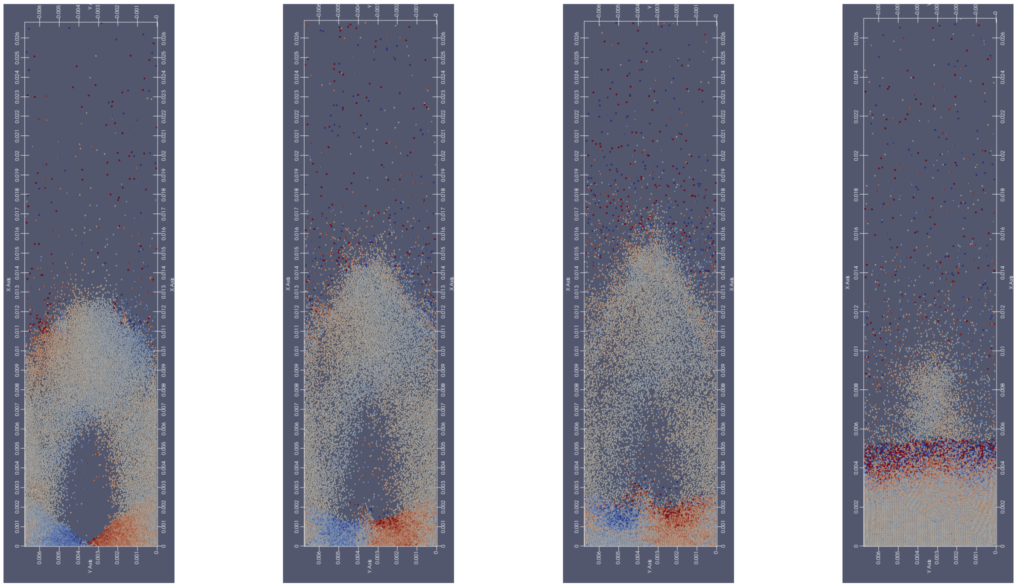


Figure 6.6: Particle axial velocity at an injection velocity $u_i = 0.8\ \frac{m}{s}$ at different time steps.



(a) t=0.0027 (b) t=0.0060 (c) t= 0.0081 (d) t= 0.0136



(e) t= 0272 (f) t= 0.0409 (g) t= 0.0545 (h) t= 0.1091

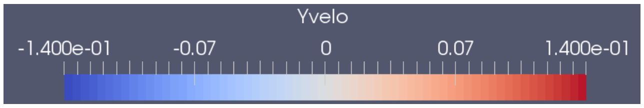


Figure 6.7: Particle spanwise velocity at an injection velocity $u_i = 0.8\ \frac{m}{s}$ at different time steps.

Bubble formation in this study was slightly different from that of the numerical simulation of Wang et al. [101]. The disagreement between the two results can be attributed to the different particle-particle interaction and inter-phase momentum transfer models used in these studies. In this study, the proposed model was able to capture the multi-particle collision mechanism and the tangential component of the contact force, while the previous simulation utilized the time-driven hard-sphere model, which only considered the binary-particle collision mechanism and the normal component of the contact force. Finally, in spite of the aforementioned differences in models, the maximum bubble diameters at different injection velocities in this study were in good agreement with those of Wang et al. [101], as shown in Figure 6.8. The bubble diameter (i.e. volume-averaged equivalent bubble diameter) is the diameter of a circle with area equals the area of region where the gas voidage is higher than 0.85 [101].

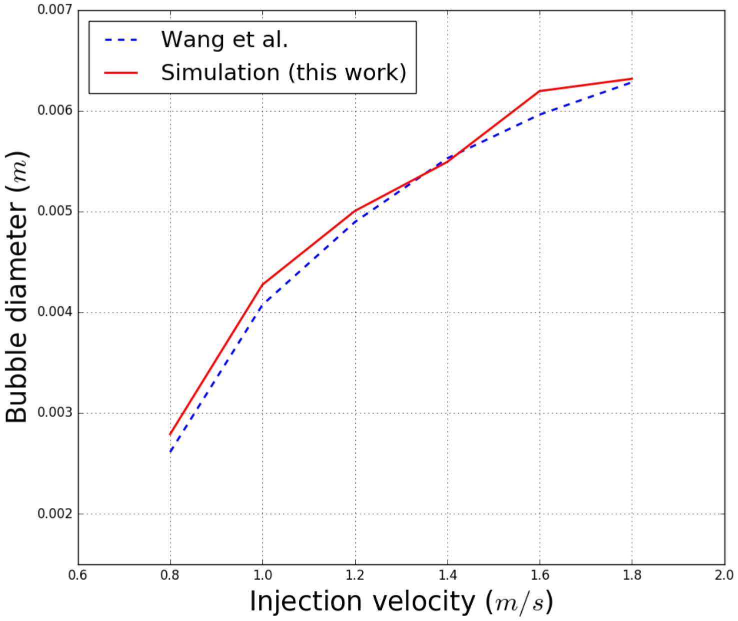


Figure 6.8: Bubble diameter (m) with different injection velocity (m/s)

6.5.2 Particle Transport and Deposition in an Impinging Jet

The circular impinging jet used in this simulation is depicted in Figure 5.12 (b), where the jet physical diameter (D) is 1.484 mm which was set to be 40 LBM units, and its length (L) is the same as the diameter length. The impingement surface was positioned at distance equal half the nozzle diameter from the jet exit. The converging zone hight (H) is set to be 100 LBM units; subsequently, the inlet diameter (D_{inlet}) is a function of the converging height and angle:

$$D_{inlet} = D + \frac{2H}{\tan 64^0}. \tag{6.24}$$

The LBM fluid flow simulation was run first for 50000 LBM time steps before the particle injection, and after this time step, 10000 particles were randomly released in the inlet (D_{inlet}) area with velocity equal to the fluid inlet velocity. The particle diameter was chosen to be constant for all simulation cases (i.e. $D_p = 3.0\ \mu m$). The particles were deposited on the impingement wall when the distances between the particle centers and the wall were equal to the particle radius. The number of the deposited particles was normalized by 2500 in order to make a comparison with the experimental work of Sethi and John [86]. The simulations were carried out for a wide range of Stokes numbers, and the square root of the Stokes number is considered as a dimensionless particle diameter. The Stokes number was changed by modifying the flow rate (i.e. fluid velocity) through the jet. The current simulation includes two units: the LBM unit which was adopted for the fluid flow simulation, and the physical unit which was used for the particle transport simulation. The LBM unit for the velocity was kept constant where the inlet velocity was $U_{inlet} = 0.002$, and this low value of velocity at the inlet produced an average velocity in the jet region equals to:

$$U_{jet} = \left(R_{inlet}^2 / R_{jet}^2\right) U_{inlet} = 0.022, \tag{6.25}$$

where R_{inlet} and R_{jet} are the radii of the inlet and jet in LBM unit, respectively. This low jet maximum velocity value ensures reducing the compressibility error and increasing the stability in the lattice Boltzmann simulation [51]. The average physical fluid velocity in the jet (U_{real}) is calculated based on Stokes number (Stk) as follows:

$$U_{real} = \frac{Stk D_{real} \nu_{real}}{\rho_p D_p^2} \tag{6.26}$$

where ν_{real}, ρ_p, and D_{real} are the fluid kinematic viscosity, solid particle density, and the physical jet diameter, respectively. The subscript *real* is used to define the physical quantities in the simulation, while the subscript *lbm* is utilized for parameters in the LBM unit. The non-dimensional variables (e.g. Stokes number and Reynolds number) are invariant in both the LBM and the physical units; accordingly, the Reynolds number can be calculated based on the physical quantities as follows:

$$Re = \frac{U_{real} D_{real}}{\nu_{real}}. \tag{6.27}$$

The LBM kinematic viscosity (ν_{lbm}) and the relaxation frequency (ω) were calculated based on the Reynolds number:

$$\nu_{lbm} = \frac{U_{jet} D_{jet}}{Re}, \qquad \omega = \frac{1}{3\nu_{lbm} + 0.5}. \tag{6.28}$$

Finally, the Van Driest correction function was implemented in the near-wall region to correct (damp) the eddy viscosity, which is calculated from Equation 3.56:

$$\begin{aligned} \nu_{t_{correc}} &= \nu_t \left(1 - e^{\frac{-y^+}{A}}\right), \\ y^+ &= \frac{y u_\tau}{\nu}, \\ u_\tau &= \sqrt{\frac{\tau_w}{\rho}}, \end{aligned} \tag{6.29}$$

where y^+, u_τ and τ_w are the non-dimensional wall distance, friction velocity and the wall shear stress, respectively. The LBM simulation results for the particle deposition density on the impingement surface were validated against the experimental work of Sethi and John [86], as shown in Figure 6.9.

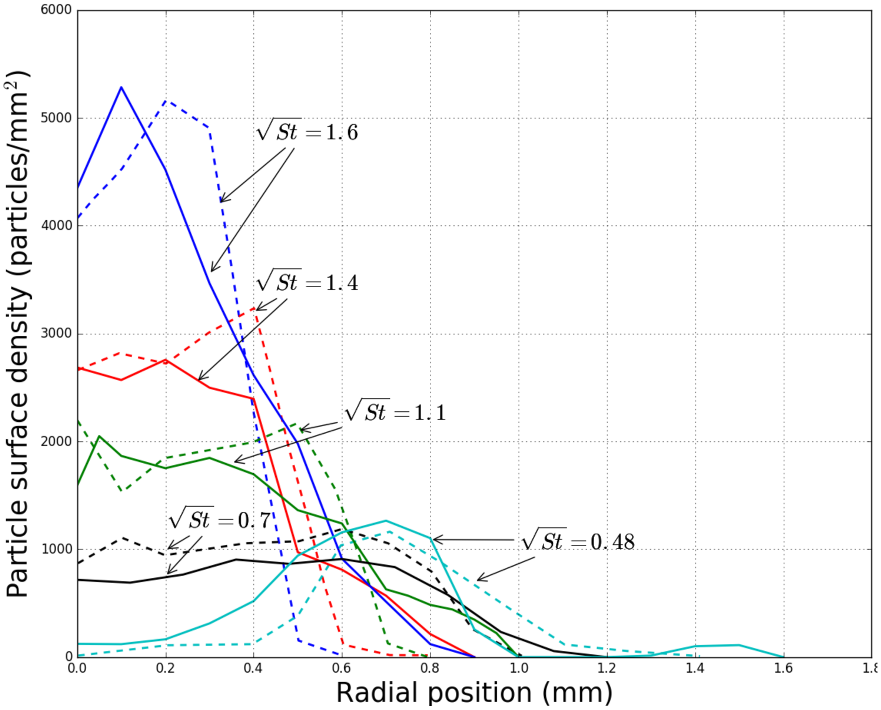


Figure 6.9: Particle surface density for different Stokes numbers; dash lines: experimental results [86]; solid lines: simulation results (this work).

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this study, the LBM-MRT was applied to simulate laminar and turbulent flows in various test cases. The LES turbulence model was implemented in the LBM-MRT scheme and to resolve the sub-grid eddies. The implementation of the solid-particle phase movement on the GPUs CUDA framework and the LBM-Lagrangian particle tracking approach are the main features of this study. The primary contributions from this work are summarized as follows:

- A new three-dimensional LBM-MRT-CA model was developed to simulate the particle transport and deposition through laminar and turbulent flows with different hydrodynamic characteristics (e.g. vortex shedding, impingement, and turbulent boundary layer).
- The particle phase information was described in the LBM nodes which allows for a massively parallel execution of this method using GPUs.
- The particle transport algorithm was based on the local (bulk) particle density and velocity which provides more realistic results of the particle movement and deposition.
- The parallel scalability of the probabilistic particle dispersion algorithm (i.e. LBM-CA) is compatible with the LBM one which allows this method and the computational system (e.g. a server machine) to handle and manage a growing amount of work.

- The representation of the particle density (i.e. number of particle) in the LBM nodes makes this method suitable to implement the effect of the particle phase on the fluid flow (i.e. two-way coupling).

- This model can be tuned to coincide with the LBM grid refinement technique which can better capture the the hydrodynamic interaction in the fine grid close to the wall surface nodes (i.e. the refined region).

- A new LBM-Lagrangian particle tracking model was developed by considering the particles influence (two-way coupling) on fluid flows. This model represents the particle phase as a porous medium moving within the LBM nodes, and the porosity of this medium is a function of the particle number in the LBM node control volumes. The traditional DEM was implemented for dense particle suspension to incorporate the particle collision (four-way coupling) in the flow domain. This model (i.e. LBM-Lagrangian) can capture the multi-particle collisions which can better describe particle-laded turbulent (high speed) flows.

7.2 Future Work

The following future work related to the topic of this thesis is recommended:

- Extending the LBM-MRT-CA CUDA code to be run in parallel on many GPUs on the same node (e.g. same computer), or on multiple GPUs on different compute nodes utilizing MPI parallelization, which can allow access to multiple compute node. The current model simulation results were shown for a single GPU implementation, while the workstation (computer), where the simulations were run, includes two GPUs (i.e. 2 × GeForce GTX TITAN X). The code was extended and executed on both GPUs successfully; however, some particles were lost during the simulations. It was noticed that the number of the missing particles was proportional to the number of the simulation time steps. This phenomenon might be attributed to GPUs synchronization issues and data transactions between the two GPUs.

- Particle deposition on a wall in the current (LBM-MRT-CA and LBM-Lagrangian) models is highly related to the solutions (e.g. fluid velocity) of fluid nodes adjacent to the wall, e.g. no more information is needed when the particles leave the fluid nodes toward the wall. For that, it is crucial to accurately simulate turbulent flows near the wall. In the current study, LES static Smagorinsky model with Van Driest damping function was used because of the simple implementation of this model in the LBM-MRT framework, and because of the simple boundary configurations used in this study, e.g. flat surfaces. However, for complex flow involving transition, it is recommended to investigate more advanced turbulence models (e.g. dynamic subgrid [79] and sigma models [78]) in the future work.

References

[1] Sylvain Aguinaga, Olivier Simonin, Jacques Boree, and Vincent Herbert. A simplified particle-turbulence interaction pdf model: application to deposition modelling in turbulent boundary layer. In *ASME 2009 Fluids Engineering Division Summer Meeting*, pages 1899–1910. American Society of Mechanical Engineers, 2009.

[2] Cyrus K Aidun, Yannan Lu, and E-JIANG DING. Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation. *Journal of Fluid Mechanics*, 373:287–311, 1998.

[3] Eid Alatawi. *Particulate Deposition Prediction of Diluted Two-Phase Impinging Jet.* PhD thesis, Carleton University Ottawa, 2013.

[4] E Alatawia and E Matidaa. Effect of anisotropy on particle deposition in impinging jet flow. *Journal ISSN*, 2724:2012, 1929.

[5] Michael P Allen and Dominic J Tildesley. *Computer simulation of liquids.* Oxford university press, 2017.

[6] Mhamed Bouzidi, Mouaouia Firdaouss, and Pierre Lallemand. Momentum transfer of a boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13(11):3452–3459, 2001.

[7] Wes Burwash, Warren Finlay, and Edgar Matida. Deposition of particles by a confined impinging jet onto a flat surface at Re=10^4. *Aerosol Science and Technology*, 40(3):147–156, 2006.

[8] Tchen Chan-Mou. *Mean value and correlation problems connected with the motion of small particles suspended in a turbulent fluid.* Springer, 2013.

[9] Sydney Chapman, Thomas George Cowling, and David Burnett. *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press, 1990.

[10] Hudong Chen, Olga Filippova, J Hoch, K Molvig, Rick Shock, Chris Teixeira, and Raoyang Zhang. Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Physica A: Statistical Mechanics and its Applications*, 362(1):158–167, 2006.

[11] Bastien Chopard. Cellular automata modeling of physical systems. In *Computational Complexity*, pages 407–433. Springer, 2012.

[12] Bastien Chopard and Alexandre Masselot. Cellular automata and lattice Boltzmann methods: a new approach to computational fluid dynamics and particle transport. *Future Generation Computer Systems*, 16(2):249–257, 1999.

[13] Bastien Chopard, Alexandre Masselot, and Alexandre Dupuis. A lattice gas model for erosion and particles transport in a fluid. *Computer Physics Communications*, 129(1-3):167–176, 2000.

[14] B Chun and AJC Ladd. Interpolated boundary condition for lattice Boltzmann simulations of flows in narrow gaps. *Physical Review E*, 75(6):066705, 2007.

[15] D Cooper, DC Jackson, Brian Edward Launder, and GX Liao. Impinging jet studies for turbulence model assessmenti. flow-field experiments. *International Journal of Heat and Mass Transfer*, 36(10):2675–2684, 1993.

[16] TJ Craft, LJW Graham, and Brian Edward Launder. Impinging jet studies for turbulence model assessmentii. an examination of the performance of four turbulence models. *International Journal of Heat and Mass Transfer*, 36(10):2685–2697, 1993.

[17] Peter A Cundall and Otto DL Strack. A discrete numerical model for granular assemblies. *Geotechnique*, 29(1):47–65, 1979.

[18] Paul J Dellar. Bulk and shear viscosities in lattice Boltzmann equations. *Physical Review E*, 64(3):031203, 2001.

[19] Jos J Derksen. *The lattice-Boltzmann method for multiphase fluid flow simulations and Euler-Lagrange large-eddy simulations*. Springer, 2007.

[20] Dominique d'Humieres. Generalized lattice-boltzmann equations. *Rarefied Gas Dynamics*, 1992.

[21] Dominique d'Humières. 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):437–451, 2002.

[22] S Elghobashi. Particle-laden turbulent flows: direct simulation and closure models. *Applied Scientific Research*, 48(3-4):301–314, 1991.

[23] S Elghobashi. Particle-laden turbulent flows: direct simulation and closure models. *Applied Scientific Research*, 48(3-4):301–314, 1991.

[24] S Elghobashi. On predicting particle-laden turbulent flows. *Applied Scientific Research*, 52(4):309–329, 1994.

[25] Pascal Fede, Victor Sofonea, Richard Fournier, Stephane Blanco, Olivier Simonin, Guillaume Lepoutère, and Victor Ambruş. Lattice Boltzmann model for predicting the deposition of inertial particles transported by a turbulent flow. *International Journal of Multiphase Flow*, 76:187–197, 2015.

[26] YT Feng, K Han, and DRJ Owen. 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):1111, 2007.

[27] Zhi-Gang Feng and Efstathios E Michaelides. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *Journal of Computational Physics*, 195(2):602–628, 2004.

[28] Olga Filippova and Dieter Hänel. Grid refinement for lattice-bgk models. *Journal of Computational physics*, 147(1):219–228, 1998.

[29] Warren H Finlay. *The mechanics of inhaled pharmaceutical aerosols: an introduction.* Academic Press, 2001.

[30] RK Freitas, M Meinke, and W Schröder. Turbulence simulation via the lattice-boltzmann method on hierarchically refined meshes. *European Conference on Computational Fluid Dynamics*, 2006.

[31] Uriel Frisch, Brosl Hasslacher, and Yves Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical Review Letters*, 56(14):1505, 1986.

[32] SA Galindo-Torres. A coupled discrete element lattice Boltzmann method for the simulation of fluid–solid interaction with particles of general shapes. *Computer Methods in Applied Mechanics and Engineering*, 265:107–119, 2013.

[33] Leon FG Geers, Mark J Tummers, and Kemal Hanjalić. Experimental investigation of impinging jet arrays. *Experiments in Fluids*, 36(6):946–958, 2004.

[34] M Geier, A Greiner, and JG Korvink. Bubble functions for the lattice boltzmann method and their application to grid refinement. *The European Physical Journal Special Topics*, 171(1):173–179, 2009.

[35] A Ghasemi, V Roussinova, Ram Balachandar, and RM Barron. Reynolds number effects in the near-field of a turbulent square jet. *Experimental Thermal and Fluid Science*, 61:249–258, 2015.

[36] Irina Ginzbourg and D d'Humières. Local second-order boundary methods for lattice boltzmann models. *Journal of Statistical Physics*, 84(5-6):927–971, 1996.

[37] SMJ Guzik, Xinfeng Gao, Todd Weisgraber, Berni Alder, and Phillip Colella. An adaptive mesh refinement strategy with conservative space-time coupling for the lattice-Boltzmann method. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 866, 2013.

[38] Thomas Hällqvist. *Large eddy simulation of impinging jets with heat transfer*. PhD thesis, Royal Institute of Technology, 2006.

[39] K Han, YT Feng, and DRJ Owen. Coupled lattice Boltzmann and discrete element modelling of fluid–particle interaction problems. *Computers & Structures*, 85(11):1080–1088, 2007.

[40] Xiaoyi He and Gary Doolen. Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder. *Journal of Computational Physics*, 134(2):306–315, 1997.

[41] Xiaoyi He and Gary D Doolen. Lattice Boltzmann method on a curvilinear coordinate system: Vortex shedding behind a circular cylinder. *Physical Review E*, 56(1):434, 1997.

[42] FJ Higuera, S Succi, and R Benzi. Lattice gas dynamics with enhanced collisions. *EPL (Europhysics Letters)*, 9(4):345, 1989.

[43] Nan-Hung Hsieh. http://www.slideshare.net/nanhomsie/deposition-modeling-of-respiratory-airborne.

[44] K Jambunathan, E Lai, MA Moss, and BL Button. A review of heat transfer data for single circular jet impingement. *International Journal of Heat and Fluid Flow*, 13(2):106–115, 1992.

[45] Michal Januszewski and Marcin Kostur. Sailfish: a flexible multi-GPU implementation of the lattice Boltzmann method. *Computer Physics Communications*, 185(9):2350–2368, 2014.

[46] Shin K Kang and Yassin A Hassan. The effect of lattice models within the lattice Boltzmann method in the simulation of wall-bounded turbulent flows. *Journal of Computational Physics*, 232(1):100–117, 2013.

[47] Inchul Kim, Said Elghobashi, and William A Sirignano. On the equation for spherical-particle motion: effect of reynolds and acceleration numbers. *Journal of Fluid Mechanics*, 367:221–253, 1998.

[48] Yusuke Koda. Lattice boltzmann method for simulating turbulent flows. Master's thesis, University of Waterloo, 2013.

[49] Yusuke Koda and Fue-Sang Lien. The lattice boltzmann method implemented on the gpu to simulate the turbulent flow over a square cylinder confined in a channel. *Flow, Turbulence and Combustion*, 3(94):495–512, 2015.

[50] Gilberto M Kremer. *An introduction to the Boltzmann equation and transport processes in gases*. Springer Science & Business Media, 2010.

[51] T Krüger et al. *The Lattice Boltzmann Method: Principles and Practice*. Springer International Publishing, 2017.

[52] AJC Ladd and R Verberg. Lattice-boltzmann simulations of particle-fluid suspensions. *Journal of Statistical Physics*, 104(5-6):1191–1251, 2001.

[53] Anthony JC Ladd. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. part 1. theoretical foundation. *Journal of Fluid Mechanics*, 271:285–309, 1994.

[54] Anthony JC Ladd. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. part 2. numerical results. *Journal of Fluid Mechanics*, 271:311–339, 1994.

[55] Pierre Lallemand and Li-Shi Luo. Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, galilean invariance, and stability. *Physical Review E*, 61(6):6546, 2000.

[56] Pierre Lallemand and Li-Shi Luo. Lattice Boltzmann method for moving boundaries. *Journal of Computational Physics*, 184(2):406–421, 2003.

[57] Jonas Latt, Bastien Chopard, Orestis Malaspinas, Michel Deville, and Andreas Michler. Straight velocity boundaries in the lattice Boltzmann method. *Physical Review E*, 77(5):056703, 2008.

[58] J Laviéville, E Deutsch, and O Simonin. Large eddy simulation of interactions between colliding particles and a homogeneous isotropic turbulence field. *ASME*, 228:347–358, 1995.

[59] Robert M MacMeccan, JR Clausen, GP Neitzel, and CK Aidun. Simulating deformable particle suspensions using a coupled lattice Boltzmann and finite-element method. *Journal of Fluid Mechanics*, 618:13–39, 2009.

[60] Orestis Malaspinas and Pierre Sagaut. Wall model for large-eddy simulation based on the lattice boltzmann method. *Journal of Computational Physics*, 275:25–40, 2014.

[61] Wenbin Mao. *Modeling particle suspensions using lattice Boltzmann method.* PhD thesis, Georgia Institute of Technology, 2013.

[62] Virgil Alan Marple. *A fundamental study of inertial impactors.* PhD thesis, University of Minnesota, 1972.

[63] F Mashayek and RVR Pandya. Analytical description of particle/droplet-laden turbulent flows. *Progress in Energy and Combustion Science*, 29(4):329–378, 2003.

[64] Alexandre Masselot. *A new numerical approach to snow transport and deposition by wind: a parallel lattice gas model.* PhD thesis, University of Geneva, 2000.

[65] Alexandre Masselot and Bastien Chopard. A lattice Boltzmann model for particle transport and deposition. *EPL (Europhysics Letters)*, 42(3):259, 1998.

[66] Keijo Mattila. Implementation techniques for the lattice boltzmann method. *Jyväskylä Studies in Computing*, (117), 2010.

[67] R Mei. Velocity fidelity of flow tracer particles. *Experiments in Fluids*, 22(1):1–13, 1996.

[68] Renwei Mei, Dazhi Yu, Wei Shyy, and Li-Shi Luo. Force evaluation in the lattice Boltzmann method involving curved geometry. *Physical Review E*, 65(4):041203, 2002.

[69] Abdulmajeed A Mohamad. *Lattice Boltzmann method: fundamentals and engineering applications with computer codes*. Springer Science & Business Media, 2011.

[70] Cristina Nita, Lucian Mihai Itu, and Constantin Suciu. GPU accelerated blood flow computation using the lattice Boltzmann method. In *High Performance Extreme Computing Conference (HPEC), 2013 IEEE*, pages 1–6. IEEE, 2013.

[71] DR Noble and JR Torczynski. A lattice-Boltzmann method for partially saturated computational cells. *International Journal of Modern Physics C*, 9(08):1189–1201, 1998.

[72] CUDA NVIDIA. Programming guide, 2008.

[73] CUDA Nvidia. Curand library, 2010.

[74] Christian Obrecht, Frédéric Kuznik, Bernard Tourancheau, and Jean-Jacques Roux. A new approach to the lattice boltzmann method for graphics processing units. *Computers & Mathematics with Applications*, 61(12):3628–3638, 2011.

[75] Christian Obrecht, Frédéric Kuznik, Bernard Tourancheau, and Jean-Jacques Roux. Multi-GPU implementation of the lattice Boltzmann method. *Computers & Mathematics with Applications*, 65(2):252–261, 2013.

[76] DRJ Owen, CR Leonardi, and YT Feng. An efficient framework for fluid–structure interaction using the lattice Boltzmann method and immersed moving boundaries. *International Journal for Numerical Methods in Engineering*, 87(1-5):66–95, 2011.

[77] Xavier Pialat, Olivier Simonin, and Philippe Villedieu. A hybrid eulerian–lagrangian method to simulate the dispersed phase in turbulent gas-particle flows. *International Journal of Multiphase Flow*, 33(7):766–788, 2007.

[78] Aniruddhe Pradhan and Sumedh Yadav. Large eddy simulation using lattice boltzmann method based on sigma model. *Procedia Engineering*, 127:177–184, 2015.

[79] Kannan N Premnath, Martin J Pattison, and Sanjoy Banerjee. Dynamic subgrid scale modeling of turbulent flows using lattice-boltzmann method. *Physica A: Statistical Mechanics and its Applications*, 388(13):2640–2658, 2009.

[80] YH Qian, Dominique d'Humières, and Pierre Lallemand. Lattice BGK models for navier-stokes equation. *EPL (Europhysics Letters)*, 17(6):479, 1992.

[81] Dennis C Rapaport. *The art of molecular dynamics simulation.* Cambridge university press, 2004.

[82] Michael W Reeks. On model equations for particle dispersion in inhomogeneous turbulence. *International Journal of Multiphase Flow*, 31(1):93–114, 2005.

[83] Daniel H Rothman. Cellular-automaton fluids: A model for flow in porous media. *Geophysics*, 53(4):509–518, 1988.

[84] Daniel H Rothman and Stiphane Zaleski. *Lattice-gas cellular automata: simple models of complex hydrodynamics*, volume 5. Cambridge University Press, 2004.

[85] M Salmanzadeh, M Rahnama, and G Ahmadi. Particle transport and deposition in a duct flow with a rectangular obstruction. *Particulate Science and Technology*, 25(5):401–412, 2007.

[86] Virendra Sethi and Walter John. Particle impaction patterns from a circular jet. *Aerosol Science and Technology*, 18(1):1–10, 1993.

[87] Olivier Simonin. Statistical and continuum modelling of turbulent reactive particulate flows. *Lecture Series*, 6, 2000.

[88] K Suga, Yoshiaki Kuwata, K Takashima, and R Chikasue. A D3Q27 multiple-relaxation-time lattice Boltzmann method for turbulent flows. *Computers & Mathematics with Applications*, 69(6):518–529, 2015.

[89] DT Thorne and C Michael. Lattice Boltzmann modeling. an introduction for geoscientists and engineers. *Springer*, 2006.

[90] Jonas Tölke. Implementation of a lattice Boltzmann kernel using the compute unified device architecture developed by nVIDIA. *Computing and Visualization in Science*, 13(1):29–39, 2010.

[91] Jonas Tölke and Manfred Krafczyk. Teraflop computing on a desktop pc with gpus for 3d cfd. *International Journal of Computational Fluid Dynamics*, 22(7):443–456, 2008.

[92] Jonas Tölke and Manfred Krafczyk. Second order interpolation of the flow field in the lattice boltzmann method. *Computers & Mathematics with Applications*, 58(5):898–902, 2009.

[93] CUDA Toolkit. 4.2: Curand guide. *NVIDIA Corporation, Santa Clara*, 2012.

[94] Henry FS Tsuda A and Butler JP. Particle transport and deposition: basic physics of particle kinetics. *Comprehensive Physiology*, 3(4):1437–1471, 2013.

[95] Y Tsuji. Activities in discrete particle simulation in japan. *Powder Technology*, 113(3):278–286, 2000.

[96] R Verberg and AJC Ladd. Lattice-Boltzmann model with sub-grid-scale boundary conditions. *Physical review letters*, 84(10):2148, 2000.

[97] R Verberg and AJC Ladd. Accuracy and stability of a lattice-Boltzmann model with subgrid scale boundary conditions. *Physical Review E*, 65(1):016701, 2001.

[98] Cédric Villani. A review of mathematical topics in collisional kinetic theory. *Handbook of Mathematical Fluid Dynamics*, 1:71–305, 2002.

[99] Haoming Wang, Haibo Zhao, Zhaoli Guo, Yongxiang He, and Chuguang Zheng. Lattice Boltzmann method for simulations of gas-particle flows over a backward-facing step. *Journal of Computational Physics*, 239:57–71, 2013.

[100] Haoming Wang, Haibo Zhao, Zhaoli Guo, and Chuguang Zheng. Numerical simulation of particle capture process of fibrous filters using lattice Boltzmann two-phase flow model. *Powder Technology*, 227:111–122, 2012.

[101] Limin Wang, Bo Zhang, Xiaowei Wang, Wei Ge, and Jinghai Li. Lattice boltzmann based discrete simulation for gas–solid fluidization. *Chemical Engineering Science*, 101:228–239, 2013.

[102] Dieter A Wolf-Gladrow. *Lattice-gas cellular automata and lattice Boltzmann models: An Introduction.* Springer Science & Business Media, 2000.

[103] Jingshu Wu and Cyrus K Aidun. Simulating 3d deformable particle suspensions using lattice Boltzmann method with discrete external boundary force. *International Journal for Numerical Methods in Fluids*, 62(7):765–783, 2010.

[104] Qingang Xiong, Ehsan Madadi-Kandjani, and Giulio Lorenzini. A lbm–dem solver for fast discrete particle simulation of particle–fluid flows. *Continuum Mechanics and Thermodynamics*, 26(6):907–917, 2014.

[105] Ning Yang, Wei Wang, Wei Ge, and Jinghai Li. Cfd simulation of concurrent-up gas–solid flow in circulating fluidized beds with structure-dependent drag coefficient. *Chemical Engineering Journal*, 96(1-3):71–80, 2003.

[106] Dazhi Yu, Renwei Mei, Li-Shi Luo, and Wei Shyy. Viscous flow computations with the method of lattice Boltzmann equation. *Progress in Aerospace Sciences*, 39(5):329–367, 2003.

[107] Huidan Yu and Sharath S Girimaji. Near-field turbulent simulations of rectangular jets using lattice Boltzmann method. *Physics of Fluids (1994-present)*, 17(12):125106, 2005.

[108] Huidan Yu, Li-Shi Luo, and Sharath S Girimaji. Les of turbulent square jet flow using an mrt lattice Boltzmann model. *Computers & Fluids*, 35(8):957–965, 2006.

[109] Hao Zhang, Yuanqiang Tan, Shi Shu, Xiaodong Niu, Francesc Xavier Trias, Dongmin Yang, Hao Li, and Yong Sheng. Numerical investigation on the role of discrete element method in combined lbm–ibm–dem modeling. *Computers & Fluids*, 94:37–48, 2014.

[110] Qisu Zou and Xiaoyi He. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids (1994-present)*, 9(6):1591–1598, 1997.

APPENDICES

Appendix A

Gauss-Hermite Quadrature

$$\int_{\infty}^{\infty} \omega(x) f(x) dx \approx \sum_{i=1}^{q} w_i f(x_i). \tag{A.1}$$

The integration accuracy of Equation A.1 depends on the number q which is the number of integration points in the values of x_i. If the number of x_i is chosen as the n roots of the Hermite polynomial of order n (i.e. $H^n = 0$ and $q = n$), then any polynomial of order $N = 2n - 1$ (i.e. $P^N(x)$) can be integrated exactly as:

$$\int_{-\infty}^{\infty} \omega(x) P^N(x) dx = \sum_{i=1}^{n} w_i P^N(x_i), \tag{A.2}$$

where the weight function W_i can be found as follows:

$$w_i = \frac{n!}{\left(nH^{n-1}(x_i)\right)^2}. \tag{A.3}$$

The weights required to integrate up to fifth order polynomials ($N = 5$) are shown in Table A.1. To integrate a third-order polynomial $P^3(x)$, we need $n = 2$ and the polynomial $H^2(x)$ with two abscissae points (i.e. ±1). The weights for this case are $w_{1,2} = 1/2$, and the integration will be:

$$\int_{-\infty}^{\infty} \omega(x) P^3(x) dx = \left(\frac{1}{2}P^3(1) + \frac{1}{2}P^3(-1)\right) \tag{A.4}$$

Table A.1: Abscissae x_i and weights w_i for exact integration of polynomials up to fifth order

Number of abscissae	Polynomial degree	Abscissae	Weights
n	$N = 2n - 1$	x_i	w_i
1	1	0	1
2	3	± 1	1/2
3	5	0	2/3
		$\pm\sqrt{3}$	1/6

The extension to multiple dimension is straightforward for any real polynomial of order N in d-dimensional space, and the form of this polynomial can be written as:

$$P^N(\mathbf{x}) = \sum_{N_1+...+N_d \leq N} a_{N_1...N_d} x_1^{N_1}...x_d^{N_d}, \tag{A.5}$$

where $a_{N_1...N_d}$ are real coefficients and $N_1, ..., N_d$ are integers. An example of a second-order polynomials in two dimension vector $\mathbf{x}$ is:

$$P^2(\mathbf{x}) = \mathbf{x} \cdot \mathbf{x} = x_1^2 + x_2^2$$

The mixed coefficients (e.g. a_{12}) are all vanished and $a_{11} = a_{22} = 1$.

Integration of such polynomial multiplied by multidimensional weight function $\omega(\mathbf{x})$ can be represented as sum of integral with 1D weight function $\omega(x)$, as follows:

$$\begin{aligned} \int \omega(\mathbf{x}) P^N(\mathbf{x}) d^d x &= \int \omega(\mathbf{x}) \sum a_{N_1} ....N_d x_1^{N_1} ....x_d^{N_d} d^d x \\ &= \sum a_{N_1....N_d} \prod_{j=1}^{d} \int \omega(x_j) x_j^{N_j} dx_j, . \end{aligned} \tag{A.6}$$

where the multidimensional weight function $omega(\mathbf{x}) = \prod_{j=1}^{d}$. The 1D integrals in Equation A.6 are decomposed utilizing Gauss-Hermite quadrature (Equation A.2) as:

$$\sum a_{N_1....N_d} \prod_{j=1}^{d} \int \omega(x_j) x_j^{N_j} dx_j = \sum a_{N_1....N_d} \prod_{j=1}^{d} \sum_{i=1}^{n_j} w_{i,j} x_{i,j}^{N_j}. \tag{A.7}$$

The $x_{i,j}$ is the j-component of the i-th abscissae. If the 1d integrals are discretized using the same Hermite polynomial, e.g. $n_1 = n_2 = ... = n_d = n$, $x_{i,1} = x_{i,2} = ...x_{i,d} = x_i$, and $w_{i,1} = w_{i,2} = ...w_{i,d} = w_i$, the product of sums can be written in the following form:

$$\prod_{j=1}^{d} \sum_{i=1}^{n_j} w_{i,j} x_{i,j}^{N_j} = \sum_{i_1=1}^{n} .... \sum_{i_d=1}^{n} w_{i_1} .... w_{i_d} x_{i_1}^{N_1} .... x_{i_d}^{N_d}. \tag{A.8}$$

New multidimensional abscissae $\mathbf{x}_i = (x_{i_1}, ..., x_{i_D})$ and weights $w_i = (w_{i_1} .... w_{i_D})$ are defined to obtain a multidimensional Gauss-Hermite quadrature, as follows:

$$\int \omega(\mathbf{x}) P^N(\mathbf{x}) d^d x = \sum_{i=1}^{n^d} w_i P^N(\mathbf{x}_i). \tag{A.9}$$

Appendix B

CUDA GPUs Programming

The main advantage of using the LBM is the data locality in the problem-solving operations. Many efforts have been made during the last decade in order to construct efficient LBM implementation on GPUs. Contrary to CPUs, GPUs are designed and manufactured for highly graphic parallel computing utilizing several Streaming Multiprocessors (SMs). Each SM contains many of Scalar Processors (SPs), shared memory, an instruction unit, and it may include an L1 cache in the Fermi GPUs architecture with compute capability 2.x.

B.1 Process Hierarchy

The CUDA programming model utilizes a hierarchy of process threads to scale efficiently through GPUs with different number of processing units. The GPU functions (kernels) are run concurrently in the GPU threads, which are the smallest processing units in GPUs. Blocks, which are the processing units used to allocate jobs to the available SMs, contains many threads as a group. The threads inside a block can be synchronized and communicated through the shared memory, and they are executed in the same SM. Blocks are grouped in a grid, which is responsible for executing the GPU functions on a group of threads. For GPUs with compute capabilities 2.0 and above, grids can be arranged in three dimensions which makes dealing with 3D problems more organized.

A SM runs a kernel when it is assigned to a thread block. Multiple blocks can be allocated in a single SM when there are available resources. The thread block is decomposed by SM into thread groups of 32 (threads), which is called a warp. The SM can only operate one warp at a given time, but it can move to different warp when the current warp is unused. The process of switching one warp to another one can hide from memory transaction. To optimize a CUDA code on a GPU, the number of warps could be maximize in a SM to increase the occupancy.

B.2 Memory Hierarchy

In GPUs programming, there are different types of memory which are organized in a hierarchical fashion. The smaller memory size is the thread local memory, which is private and only accessible by its thread. Shared memory is the memory assigned to each thread block and thread in this block can only access this memory. The lifetime of the shared memory is the same as the block lifetime. In other words, this memory is deleted once the block finish the execution cycle. The largest memory size is the global memory, which is accessible by all threads. The global memory also contains constant and texture memory, and both of them are read-only .