2011

An improved volumetric LBM boundary approach and its extension for sliding mesh simulation

Yanbing Li
Iowa State University

An improved volumetric LBM boundary approach and its extension for sliding mesh simulation

by

Yanbing Li

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Engineering Mechanics

Program of Study Committee:

Tom I-P. Shih, Co-major Professor

Hudong Chen, Co-major Professor

Mark Bryden

Rodney O. Fox

Z. J. Wang

Iowa State University

Ames, Iowa

2012

DEDICATION

I would like to dedicate this thesis to my wife Xun Zhang and to my daughter Yumeng Mia Li, without whose support I would not have been able to complete this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

First I would like to express my sincerest gratitude to my advisors, Dr. Hudong Chen and Dr. Tom I-P. Shih, for bringing me into this research opportunity and providing support and guidance to me during my PhD study. It is without doubt that I owe many of my thanks to Dr. Hudong Chen, for leading me into the scientific world of LBM, for having faith in my abilities, supervising me throughout the research, keeping me encouraged and constantly suggesting areas to explore. Without him, this work would not have been possible. I would like also to give special thanks to Dr. Tom I-P Shih, for guiding me in the CFD research field since my first day in CFD starting at Michigan State University, and for giving me insights and encouragements to inspire me and renew my hopes for completing this study.

I would like to thanks Dr. Mark Bryden, Dr. Rodney O. Fox, and Dr. Z. J. Wang, for willing to be my PhD committee members, for providing valuable insights and assistance into this research and taking the time to review my thesis.

I would like to thanks NASA STTR support for providing most of funding for this study, and I must acknowledge the support from Exa Corporation, both for providing the partial tuition coverage support for me to complete my study at ISU, and for allowing me to conduct this study on a research platform of the LBM based PowerFLOW software.

I would like to thanks my colleagues and friends at Exa Corporation, Dr. Raoyang Zhang, Dr. Hongli Fan, Dr. Chenghai Sun, Dr. Peter Roberts, Jamie Hoch, Richard Shock, Dr. Rajani Satti, Dr. Xiaowen Shan for the valuable discussions and their kind help over the past years. I would like to thanks my friends at Iowa State University, Xingkai Chi, Xin Huang, Xiaoyan Zhu, Dee Pfeiffer, ... for all their support and help they granted me during my PhD study.

I am greatly thankful to my parents, and brother and sister for their love and support, which have been the source of courage throughout my whole academic life. Last but certainly not least, I would like to thank my loving wife, Xun Zhang, and my lovely daughter, Yumeng Mia Li, for their unwavering support and devotion during the long course of this research.

ABSTRACT

The lattice Boltzmann method (LBM) has been emerging as a promising alternative CFD approach for complex fluid flows. With LBM, no-slip/free-slip wall boundary conditions are implemented via straightforward particle bounce-back/specular reflections on a solid surface, thus enable the use of Cartesian grid for accurate boundary representation. For curved boundary that is commonly encountered with complex geometry, available point-wise based LBM extrapolation/interpolation boundary schemes can not guarantee the exact hydrodynamic flux conditions. To address this fundamental issue, a volumetric LBM boundary scheme was proposed in 1998, which ensures an exact treatment of hydrodynamic fluxes on solid surface and establishes a generic framework for realizing hydrodynamic boundary conditions on curved surface.

This dissertation presents the development of an improved volumetric LBM boundary scheme. The basic idea is when reflecting (scattering) back the fluid particles from solid boundary, particles should be distributed in the affected volume according to local flow information rather than uniformly as in the original volumetric LBM boundary formulation. To realize this, a scattering correction procedure is formulated and added to the originally proposed volumetric LBM boundary scheme framework. In particular, the procedure redistributes the surface scattered particles based on local velocity variation. As a result, it reduces the solution dependence on actual boundary location/orientation with respect to the computational grid, demonstrates an improved order of accuracy for flow solutions with arbitrarily located boundary. Accuracy of this approach has been demonstrated on typical flow benchmark problems that involve curved boundaries. In the second part of this dissertation, the proposed volumetric LBM boundary scheme is extended to sliding-mesh interface condition for flow simulation involving rotating geometries. A volumetric LBM sliding-mesh interface scheme couples the flow solutions on both sides of sliding interface, and conserves the mass and momentum flux across it. Accuracy of this scheme is demonstrated by performing a LBM-sliding mesh simulation of flow past a rotating propeller.

CHAPTER 1. INTRODUCTION

1.1 Introduction & background

In the last two decades, owing to the technology advancements in computer hardware, high performance computing and efficient numerical algorithms, computational fluid dynamics (CFD) has been widely applied not only in the areas of scientific investigation of complex flow phenomena (Chen and Doolen (1998), Aidun and Clausen (2010), Passalacqua and Fox (2011)), but also in many areas of engineering applications that involve fluid flow (Agarwal (1999), Schetz (2001), Katz (2006), Chen et al. (2003), Shih (2008)). It has become an integral part of simulation based product optimization and design process (Jameson (1999), Zitney and Syamlal (2002), McCorkle et al. (2003), Johnson et al. (2005)). With CFD, a fluid domain is divided into a set of discrete spatial cells constituting a grid system, the governing equations that describe the flow field are solved numerically on the grid system. For flow problems involve complex geometries, conventional body-confirming grids that are generated by either multi-block oversetting grid (Chan and Buning (1995), Chan (2009), Meakin (2001), Wang and Parthasarathy (2000)) or unstructured grid methods (Mavriplis (1997), Anderson et al. (2005), Wang (2007)) can be used for the numerical solution. Alternatively, a Cartesian grid with special treatment on the non-confirming boundaries can also be applied (Aftosmis (1997), Peskin (2002), Wang and Srinivasan (2002)).

Conventional CFD methods are based on macroscopic level description via the Navier-Stokes equation, in which the fundamental fluid variables are velocity (or vorticity), pressure, and so on. Kinetic theory based numerical methods, such as Direct Simulation Monte Carlo methods (DSMC), Quadrature Method of Moments (QMOM), and Lattice Boltzmann methods (LBM), have been emerging as promising alternative CFD approaches for simulation of complex fluid flows (Bird (1994), Fox (2008), Chen and Doolen (1998), Aidun and Clausen (2010)). Among them, LBM has matured as a viable and efficient numerical tool for many CFD problems (Chen and Doolen (1998), Aidun and Clausen (2010)). Unlike the conventional Navier-Stokes based numerical methods, LBM describes a fluid flow in terms of a discrete kinetic equation based on the particle density distribution functions, namely the so called Lattice Boltzmann equation (LBE). The macroscopic flow properties are direct results of the moments of these particle distribution functions. In the hydrodynamic limit it has been shown that the LBE recovers the Navier-Stokes equation (Chen et al. (1992), Qian et al. (1992)). It has also been demonstrated recently that through a moment expansion procedure, LBE can be extended to describe fluid dynamics beyond Navier-Stokes hydrodynamics (Shan et al. (2006)).

LBM has a number of attractive features when compared to traditional CFD methods (Chen and Doolen (1998), Aidun and Clausen (2010)). It has nearly ideal scalability on parallel computers and is very efficient for flow simulation on large scale applications (Pohl et al. (2003), Velivelli and Bryden (2004), Clausen et al. (2010)). Its straightforward boundary condition implementation also enables numerical solutions on a Cartesian grid system while maintaining exact conservations(Chen et al. (1998)). This makes LBM very convenient for accurately handling flow that involves complex geometry. Because of its kinetic nature, it is also easy and more physical to model various complex fluid flows (Shan and Chen (1993, 1994); Chen and Doolen (1998); Aidun and Clausen (2010)). Furthermore, the unsteady Very-Large-Eddy-Simulation (VLES) turbulence model has been incorporated successfully into LBM (Chen et al. (2003, 2004)). Its numerical accuracy and robustness have been widely demonstrated and validated in scientific research as well as in real engineering applications (Chen et al. (2003); Chen and Doolen (1998); Aidun and Clausen (2010)). On the other hand, LBM has its own limitations, such as the compressibility effect in an LBM simulation may reduce the solution accuracy for incompressible flows, and the use of cubic-cell based grid system can significient increase the computation cost when local grid refinement is needed, these are the on-going research topics in the LBM community.

Among the many research fields in lattice Boltzmann community, boundary condition study is one of the most active areas, especially when enforcing no-slip boundary condition on general geometry with curved boundaries, the original standard bounce back boundary scheme degrades the accuracy of LBM scheme to first order (Ginzbourg and Adler (1994); Noble et al. (1995)). Developing an accurate and robust boundary condition algorithm for complex geometry is the first key element for reliable numerical predictions with LBM. Secondly, to study the complex flows generated by a rotating machinery with LBM, the available black-box based impeller modeling approach is not suitable for true representations of the real geometry (Eggels (1996); Derksen et al. (1999); Lu et al. (2002)). To enable the LBM sliding mesh capability for this type of flow with exact geometry effect, suitable sliding-mesh interface boundary condition is needed. There is a lack of studies on developing suitable LBM boundary algorithms for such sliding-mesh interface condition. The two topics above are the main focuses of current research.

1.2 Literature review

1.2.1 Wall boundary conditions algorithms in LBM

In kinetic theory, specular reflection, diffuse reflection and bounce back (inverse) reflection are the generic kinetic boundary conditions to model the interaction between gas particles and a solid wall (Maxwell (1879); Schnute and Shinbrot (1973); Cercignani (1989)). By modeling particle dynamics at the microscopic/mesoscopic level, desired hydrodynamic wall boundary physics on the macroscopic level can be realized. For example, the particle specular reflection process reflects particle off from the surface with the the angle of reflection equals the angle of incidence, and can be used to achieve a "free-slip" velocity on a solid boundary. With the particle bounce-back reflection, a particle completely reverses its velocity direction after interacting with a wall and is scattered back opposite to its incoming direction. In the continuum flow regime, this effectively gives a zero-velocity ("no-slip") boundary condition related to the wall. The particle diffuse reflection process is generally used to describe a slip velocity at the wall for high-Knudsen number flows at micro-channels (Gad-el Hak (1999, 2006)), which assumes particle losts its memory about history and is reflected with Maximilian equilibrium distribution from wall. These models have been commonly used in lattice gas automata for simulation of the Navier-Stokes equations and is naturally adopted in LBM due to their kinetic nature and simplicity in implementation (Wolfram (1986); Frisch et al. (1986, 1987); Lavallee et al. (1991)).

In the early studies of LBM wall boundary condition, standard bounce back scheme is widely used to enforce the no-slip boundary condition on a flat surface. This is generally done in a point-wise fashion and requires the solid wall boundary to be exactly aligned with the lattice cell boundary. Particle distribution that resides on the nodes of lattice cell is advected from a fluid node to a wall node on the solid boundary, and is then reflected back along the inverse incoming velocity direction according to the bounce-back process. Such an implementation was soon found to be of only first order accuracy in space (Cornubert et al. (1991)) and has been verified by several follow up studies (Ziegler (1993); Gallivan et al. (1997); Kandhai et al. (1999)). This degrades the overall numerical accuracy of the LBM scheme since a standard LBM scheme is second order accurate in the interior of fluid domain (Chen et al. (1996)).

Several LBM boundary schemes have been proposed to improve the numerical accuracy for realizing no-slip wall boundary conditions: Ziegler (1993) noticed that by shifting the solid wall boundary into the fluid domain with one half lattice cell and putting the wall nodes used for bounce-back inside the solid wall, the modified bounce-back scheme (half-way bounce back) can recover the desired second order accuracy. Skordos (1993) introduced an extended collision operator and suggested to include the near wall velocity gradient for updating the new equilibrium distribution function at the boundaries. Although a first order difference scheme is applied for calculating the velocity gradient at the boundary nodes, the boundary scheme still shows second order accuracy in space. Inamuro et al. (1995) proposed a diffuse reflection based no-slip wall boundary condition implementation on a flat surface, where the wall reflected distributions are assumed to be equilibrium distributions with a counter slip velocity. The latter is solved by enforcing the fluid velocity in the vicinity of wall to be equal to the actual wall velocity. Noble et al. (1995) developed a hydrodynamic boundary procedure to achieve second order accuracy. This approach uses ghost wall nodes inside a solid wall for realizing the kinetic boundary condition on a flat boundary: post-collide particle distributions from a ghost node are calculated with an exact wall velocity and an approximated near wall fluid density. The accuracy of Noble's hydrodynamic boundary scheme is also evaluated by Gallivan et al. (1997) on a flow problem with array of cylinders, and a second order of accuracy is confirmed. Chen et al. (1996) treated the LBM as a specific finite difference scheme of the kinetic equation and proposed an extrapolation based bounce back method to improve the accuracy for no-slip wall boundary condition. In this scheme, additional ghost cells were introduced into the solid wall for the solution extrapolation, the bounce-backed particle distributions in the near wall fluid nodes are directly extrapolated based on the distributions from the wall nodes and ghost cell nodes. The wall nodes are treated as a part of the flow solution and their corresponding particle equilibrium distributions are enforced to satisfy the macroscopic wall boundary conditions.

This approach demonstrates a second order accuracy in space for simple flow problems with flat wall boundaries. Ginzburg and d'Humières (2003) developed a multi-reflection boundary condition scheme that can achieve third order kinetic accuracy with good numerical stability and has been extended to moving boundaries, but it requires at least three fluid nodes to implement.

The above mentioned numerical schemes has been quite successful in improving numerical accuracy for flows with flat-wall boundaries, such as Couette flow, lid-driven square cavity flow and Poiseuille flow, etc. However, they are not designed for general lattice-wall boundary configurations whose actual wall boundaries could have arbitrary locations/inclinations to the fluid cells. For problems involving curved geometry, the actual boundary is usually approximated by a stair-case-reconstructed shape for applying the boundary condition. For example, in Ladd (1994) and Ladd and Verberg (2001) stair-cased geometry was used to enforce the half-way bounce back condition for studying particulate suspensions. It should be noted that with such a geometry treatment, not only the fidelity of real geometry is lost, it may also introduce undesired numerical fluctuations in simulation along the boundary that could contaminate the result. As pointed out in Chen et al. (1996), numerical accuracy of the solution could be seriously compromised due to the stair-cased geometry approximation, this is further confirmed in the study of Kandhai et al. (1999), where the solution accuracy was also found to be dependent on the actual wall-inclination.

In order to treat the curved boundary accurately, Filippova and Ha̋nel (1997) and Filippova and Ha̋nel (1998) proposed a boundary fitting scheme for flows involving complicated geometries that can allow applying boundary condition at an exact boundary location. This scheme combines the particle bounce-back process and a linear extrapolation of particle distribution It relies on a ghost cell inside solid walls and applies a linear interpolation of particle distributions to enforce the boundary condition at exact an fluid-solid interface location. The bounce-backed particle distributions from the solid wall to the near wall fluid node is the post-collide particle distribution in the corresponding ghost node, which can be linearly interpolated from the post-collide distribution in the near wall fluid nodes and a fictitious equilibrium distribution evaluated at the corresponding ghost node. Velocity at the ghost node location is linearly interpolated based on the imposed wall velocity and corresponding near wall fluid velocity and is used to calculate the fictitious equilibrium distribution. This scheme can achieve second order accuracy on problems involving curved geometries, however it suffers from a poor numerical instability, particularly when the near wall fluid node is very close to the solid boundary, unless an approximation of ghost node velocity has to be applied to achieve robust prediction, together with a local grid refinement for flow involves complicated geometries. After a closer examination of Filippova and Ha̋nel (1997)'s scheme, Mei et al. (1999) identified that the numerical stability of FH scheme can be further improved by extending the velocity interpolation stencil to include more near wall fluid nodes. A series of systematic studies is performed to demonstrate the robustness of this improvement as well as the desired second order accuracy. In a second study, Mei et al. (1999) found that the stability of their former proposed algorithms is still poor if the the real boundary is too close to the solid nodes. The scheme is further refined by placing the boundary node used for velocity interpolation in the closest fluid nodes if the actual wall boundary is too close to a solid node. Verberg and Ladd (2000) developed a sub-grid-scale boundary approach based on the percentage of near wall cell to perform the interpolation and also combined with the half-way bounce back scheme to obtain second order accuracy.

Guo et al. (2002c) and Guo et al. (2002b) further extended the extrapolation method proposed by Chen et al. (1996) to handle curved boundaries and decomposed the distribution function at a wall node into its equilibrium and non-equilibrium parts, the post-collide distribution at the wall node includes both the equilibrium distribution and the collision modified non-equilibrium one and is then advected to the neighboring fluid node. In Guo et al. (2002c), both the equilibrium part and non-equilibrium part are approximated based on the neighborhood node information. The non-equilibrium part of the particle density distributions on the boundary can be directly approximated based on the corresponding near wall fluid nodes information. On the other hand the equilibrium part is approximated based on a modified equilibrium function evaluated at the near wall fluid nodes. In Guo et al. (2002b), the equilibrium part is directly evaluated at the wall, with the reference wall density being extrapolated based on the neighboring flow information. When wall boundary is too close to the nearest fluid nodes, the second nearest fluid nodes was used to avoid numerical instability issue.

Instead of relying on the ghost nodes to construct fictitious particle distributions for extrapolation, Bouzidi et al. (2001) proposed a curve boundary model that relies on the interpolation of particle distributions from the internal fluid nodes. This scheme combines the bounce-back procedure and a spatial interpolation of the scattered particle distributions according to the relative wall boundary location. This scheme does not require the extrapolations from the ghost nodes in solid wall. In fact, only the near wall fluid nodes are used for the post-collide particle distribution interpolation. Based on the relative location of the wall boundary to nearby fluid nodes, different algorithms are used to ensure proper interpolations. the interpolation is done with separated set of the weighting factors. Bouzidi's scheme requires at least two fluid nodes for the interpolation and it can recover the halfway bounce back when the boundary is located in the middle of fluid cell. Yu et al. (2003) unifies Bouzidi's near wall interpolation procedure and proposed a two-step sequential interpolations to avoid the discontinuity in the boundary treatment. Both scheme demonstrate second order accuracy and can be extended to quadratic interpolations that requires three fluid nodes.

Lallemand and Luo (2003) extend Bouzidi's bounce-back/interpolation scheme to handle flow involving moving boundaries and obtained satisfactory results. They pointed out that the interpolation approaches destroy the mass conservation near the boundary and noticed that the inaccuracy in evaluating of momentum transfer can leads to a net mass flux at boundary. Lallemand and Luo (2003) investigated the available point-wise based interpolation schemes and found the use of interpolations break the mass conservations, since the inaccuracy of the evaluation of the momentum transfer at boundary leads to a net mass flux. The loss of mass conservation significantly reduces the accuracy of the computed momentum transfer at the boundary. Peng (2005) investigated the mass leakage issue in both the Filippova and Ha̋nel (1997) and Mei et al. (1999) interpolation scheme, and found that the loss of mass conservation at wall boundary can cause a serious numerical stability problem for the interpolation based boundary scheme. He demonstrated that, by enforcing a zero net mass flux across wall boundary with a proper chosen wall density, a mass-conserving boundary scheme with improved accuracy and numerical stability can be obtained.

To avoid the mass conservation issue in a point-wise based interpolation scheme and to improve the numerical stability, Kao and Yang (2008) proposed an interpolation-free approach for curved boundary treatment in LB simulation. In this approach, a re-scaled post-collide particle distribution in the near wall fluid node is directly bounce-backed at the solid surface after being advected to the curved boundary, Rescaling of particle distributions is based on the concept of "coarse to fine" projection used in lattice Boltzmann grid refinement. For example the distance from a near wall fluid node to solid surface along the particle velocity direction defines a "fine" grid length scale while the regular lattice size is used as reference length scale for "coarse" grid, these two length scales are used to derive the relaxation time factors for the “coarse to fine” projection of the non-equilibrium distributions. With this non-equilibrium distribution being transferred from “coarse” grid to “fine” grid, the need for grid interpolation/extrapolation is avoided. This modification can also improve the local mass conservation since the equilibrium distribution part is not altered in the boundary process. Also as the coarse to fine transformation increases the effective viscosity in the near wall fluid node, the numerical stability for this approach is improved. However, when the solid wall is aligned with a lattice cell boundary, this approach will be reduced to a first-order accurate standard bounce back scheme. Lee and Lee (2010) also proposed a boundary scheme that does not rely on interpolation/extrapolations. In this approach, the collision process in the near wall cell is modified with an adaptive relaxation time scale. This relaxation time scale is directly related to the partial cell volume that is formed by a wall boundary cutting scheme through regular fluid cell, and is used to calculate the post-collide distributions in the near wall cell. Such a simple modification improves the numerical stability of the boundary process and can achieve an almost second-order accuracy. Chen et al. (2010) developed a heuristic scheme for curved boundary based on the idea in surface chemistry. In this approach, the fluid particles can be absorbed into the solid wall and then be re-scattered back after a time lag defined according to the time scale difference between the regular particle advection time and the actual time for the near wall particles travel from the nearest fluid node to solid wall. A local mass correction was applied to account for the accumulated mass on the wall, this was done in the evaluation of the equilibrium distributions on the solid wall with the help of ghost node inside the solid wall. In this way, both the local mass conservation as well as second order accuracy can be obtained. This boundary treatment only requires the geometry information and fluid distributions at the near wall point. A mass correction procedure was introduced to maintain the conservation of local mass. This methods has shown to be second order accuracy in both space and time.

To simulate flow in narrow gaps, Chun and Ladd (2007) developed an interpolation based boundary scheme that requires a single fluid node to be involved. It utilizes the equilibrium distribution from the solid surface as an additional interpolation point, this second-order accurate scheme is very effective for situations where only one fluid node exists, since it’s impossible to construct linear interpolation with typical interpolation schemes since they require two fluid nodes.

Instead of using interpolation to enforce the boundary condition, Feng and Michaelides (2004) proposed an immersed boundary treatment in LBM, the approach borrowed the idea from immerse boundary methods and treat the solid boundary as a Lagrangian grid. The rigid body condition is enforced by a penalty method, which assumes the solid boundary is deformable with a high stiffness constant. The constraint force is calculated through the displacement of the “tracer” point and “marker” point. The force density, due to the constrained force, is added to the lattice Boltzmann equation. The method preserves the advantage of LBM for handling multiple particles and also provides an alternative and accurate way to treat the solid-fluid interaction. The near wall flow field is solved by adding an additional force density term in the lattice Boltzmann equation. This method can avoid the problems of force fluctuations with a typical bounce back scheme, and enable LBM to simulate problems with particle deformation and fluid-structure interaction.

The numerical issues that are associated with the interpolation based schemes are mainly due to the pointwise representation of the distributions: in the vicinity of wall of arbitrary location, a particle may not necessarily reflect back to its original location within a unit time step after interacting with wall. Thus it has to rely on the use of a ghost node to construct interpolation and apply an ad-hoc numerical modification to ensure the local mass conservation and maintain numerical stability. With a point-wise interpolation scheme, evaluation of surface forces is non-trivial and the results can be contaminated by induced numerical noise (Yu et al. (2003)).

Contrary to the point-wise implementation of boundary condition, Chen et al. (1998) proposed a volumetric boundary approach that has more solid foundation: In this approach, the solid surface is represented by a set of planar surface elements and the particles are distributed uniformly in each fluid cell. The particle-surface dynamics is constructed in a volume that is uniquely defined by the surface element and discrete incoming particle velocities. Specially, a surface element will collect/gather incoming particles within this volume and reflect/scatter them back into this volume. The scattered back particles can then be evenly redistributed to the corresponding overlapped fluid cells. Such a volumetric representation of boundary condition guarantees the exact conservation of mass since the net mass flux across the surface element is zero, the resulting scheme has a very good numerical stability and has also been extended to incorporate a turbulence wall shear-stress model for high Reynolds number flow simulation. It has been successfully implemented in the commercial CFD software PowerFLOW and applied in many industry applications (Chen et al. (2003)). Rohde et al. (2002) also applied this methods for solving flow problem involved moving boundaries and obtained good agreement with experimental data.

Another notable feature of Chen's volumetric boundary scheme is that the specular reflecting boundary condition can be directly implemented with discrete velocities in the lattice model, this is done via an intermediate particle-wall collision process and an elegant re-construction of wall equilibrium, together with an adjustment of particle scattering along the outgoing directions. This is much convenient than the traditional reflection-projection method, where for an arbitrary positioned surface the reflected velocity may not be coincident with any discrete velocity in the lattice model. Thus the incoming velocity has to be reflected to its mirror pair, then the corresponding distributions are projected onto all the discrete velocities (Qu et al. (2010)).

However the volumetric boundary scheme has noticeable numerical dissipation, particularly when the wall boundary is not aligned with cell boundary (Rohde (2004); Li et al. (2004)), an systematic study to identify its order of accuracy and potential deficiency is lacking, it's one of the current research objective to identify the source of this deficiencies and propose alternative solutions to improve its solution accuracy.

1.2.2 CFD methods for sliding mesh simulation of flow with rotating geometries

The development of enabling CFD techniques for solving flow problems that involve rotating machinery was mainly driven by the needs from process industry and turbo-machinery industry to optimize the performance of the rotating flow devices as well to reduce the cost related to full-scale experimental testing (refs). The flow problem in such a device is challenging to solve, not only because the flow field involved is characterized by three-dimensional highly unsteady complex phenomena, but also because the relative motion between the rotating geometry and the stationary configuration make it difficult to compute with a single fixed grid system. Early numerical techniques for simulating the flow in a stirred tank used the so-called "black-box" methods, where the rotating impeller geometry is excluded from the computational domain, and its effect on the flow is modeled by either prescribing experimentally measured flow quantities on its boundary (Harvey and Greaves (1982); Placek and Tavlarides (1985); Gosman et al. (1992); Brucato et al. (1990)), or alternatively, by empirically defined source terms distributed throughout its volume (Pericleous and Patel (1986)). Although these methods can yield reasonable results for certain flow configurations, they are subject to the serious drawbacks in the empiricism associated with these procedures: first, the limited experimental data used to derive the empirical models cannot reproduce the real geometry effect; second, the experimental input that are based on specific impeller-tank configurations can not be generalized to fit all the possible geometry variations.

To enable a time-accurate, truly predictive simulation of rotating geometry induced flow with an exact geometry, Rai developed a patched-grid technique for Euler and Navier-Stokes equations in a series of pioneered publications (Rai (1984, 1985, 1986, 1989a)). In this method, complex geometry configurations is divided into simple sub-domains that can be meshed independently, the grids generated for the sub-domains are patched together, with a common patched boundary to separate each other. Since the grid lines abutting the patched boundary may not align (e.g. due to the relative motion of related grids), a fully conservative patched boundary scheme is applied to ensure accurate flow information transfer across the patched boundary. For flow problems that involves relative motions between solid geometries (such as helicopter rotor-fuselage combination or turbine rotor-stator configuration), the patched grid method allows the use of separate patches for both the stationary geometry as well as the moving geometries, with appropriate patched boundary scheme to treat the relative sliding motion between patched grids and exchange flow information across the boundary. Both first-order accurate explicit patched boundary scheme and second-order accurate implicit patched boundary scheme have been developed for simulating three dimensional rotor-stator interactions inside a gas turbine system and obtained good agreement with experimental data (Rai (1989b); Rai and Madavan (1990); Gundy-Burlet (1992)). Whitfield et al. (1987) proposed a clicking-zone interface approach for a time accurate three-dimensional Euler solution of flow about single-rotating and counter-rotating prop-fans. In this approach, blocked-grids are used for both the flow sub-domains involving stationary geometry and rotating geometries. Along the grid interface, the relative motion is handled by requiring grid lines to be aligned after each timestep, which results in a a local dynamic regridding (grid distortion and regeneration) to ensure a perfect match of the grids along the two sides of the interface. Blades and Marcum (2007) developed a sliding mesh interface method based on a finite volume unstructured viscous flow solver. In this method, the cell faces along the sliding mesh interface are extruded into the adjacent sub-domain to create new cells that overlap the grid for the adjacent domain. The new cells are used to update fluxes across the interface boundary, with flow solution on cell nodes being linearly interpolated from the solution of underlying grid. The method has obtained results that is comparable to the ones from clicking-zone interface approach or patched boundary method. Steijl and Barakos (2008) presented a sliding-mesh algorithm for calculating of the flow around helicopter rotorfuselage configuration with non-matching grids on both side of the sliding mesh interface.

To enable time-dependent flow prediction in mixing tanks, Perng and Murthy (1993) developed a moving-deforming mesh technique, similar to the clicking-zone interface approach applied by Whitfield et al. (1987), a single grid and a single reference frame are used for the flow regions involve both stationary or moving boundaries, with the grid system is updated at every timestep: cells associated with the flow region near the impeller rotates with the impeller, causing the cells adjacent to the sliding mesh interface to deform. When cell deformation becomes severely bad a local grid refinement is applied to improve the cell quality, and the flow information is conservatively projected into the new grid system. The grid motion is accounted for by transforming the time derivatives in Eulerian conservation equations into a Lagrangian form. The method has been verified on two-dimensional and three-dimensional mixing tanks simulations. The quality of resulting grid systems is very difficult to control and this may impact the solution accuracy, this is a major drawback of above method.

Luo et al. (1993) proposed a sliding mesh method for the time-dependent solution of flow inside impeller-stirred tank. This is to solve the time-dependent form of the governing transport equations in two sub-domains which were fixed to the respective frames of reference, the outer one being stationary and the inner one rotating with the impeller. At the interface, the numerical grids were allowed to slide to accommodate the relative motion and a time-dependent interface matching was applied to bridge the solutions in two sub-domains. Predictions were presented for a flow inside tank configuration that being stirred by a six-blade Rushton impeller, and the agreement with the experiments was better than that obtained by using a steady-state approach and a body-force impeller model. This was also confirmed in the following on validations by Bode (1994). Murthy et al. (1994) described a similar sliding-mesh technique for simulating the impeller-baffled tank interaction: two grid systems were employed, one moving with the impeller and the other fixed to the tank. The two meshes interacted along a surface of slip. The moving grid was allowed to slide relative to the stationary one, with no mesh distortion, and a conservative interpolation was used to obtain flow variables and face fluxes across the slip surface. Simulations were conducted in a time-dependent fashion and the laboratory reference frame was used for both grids; appropriate terms were included in the governing equations to account for grid rotation. Qualitative results was reported without direct comparison to experimental data. Later studies by Daskopoulos and Harris (1996) reported rather unsatisfactory results with a severe under-prediction of the turbulence energy, which is mainly due to the use of RANS type turbulence model and a poor spatial resolution adopted in the simulation.

In the process industry, the proposed sliding mesh techniques has enabled the use of actual rotating geometry in the solution procedure, thus allow the explicit simulation of the whole flow field without any recourse to empirical impeller modeling. Since then a number of numerical studies have been performed to validate their numerical accuracy and study the affecting factors.

Tabor et al. (1996) presented both a time-dependent sliding mesh predictions and a steady state mrf simulation of fluid flow in a mixing vessel stirred by 6-bladed Rushton turbine. The computed flow field data are compared with experimental data available in the literature and obtained good agreement for both methods. They also point out that the fundamental assumption used in steady-state MRF (e.g. steady-state & axisymetric) makes sliding mesh more favorable in complex configurations. Bakker et al. (1997) applied the sliding mesh method to the predictions of flow pattern created by a pitched blade turbine at various Reynolds numbers, and the results compared favorably with experimental data obtained from LDV measurement. It is concluded that the sliding mesh method is suitable for the prediction of flow patterns in stirred tanks at low Reynolds numbers. Hartmann et al. (2004) assess the effect of turbulence modeling on sliding mesh predictions by comparing the results to experimental data from LDA measurements, and concluded that although RANS type models can obtain good predictions of mean flow field, they are unable to predict accurately the turbulent kinetic energy distributions and discharge flow behavior near the impeller region, while LES based approach gives reasonable good predictions.

It should be noted here that all the above mentioned methods are based on solving the Navier-Stokes equations in the solution domain, there are very limited studies in LBM for solving flow problem involving rotating geometry in a confined baffled tank.

Eggels (1996) was the first to introduce LBM for studying flow problem inside a baffled stirred tank. In his approach, a single grid system is used and a black-box method is employed for modeling the effect of impeller. The impact of a rotating impeller on the flow field is modeled by means of a spatially/temporally varying force field. The applied force field is derived based on the difference between the local fluid velocity and the velocity of impeller, such that desired flow velocity near impeller boundary can be enforced. A large eddy simulation (LES) approach was also used for modeling the turbulence field inside the mixing tank, and the computed mean flow and turbulence intensities are in close agreement with experimental data, demonstrated the LBM-LES approach as a potential method for investigating turbulent flows in such industry applications. Derksen et al. (1999) extended Eggels's approach with a more refined forcing algorithm for the impeller modeling: the impeller was described by a set of control points, a second order interpolation scheme was used to calculate the the mismatch between the actual flow velocity and prescribed one at these control points. An adaptive force-field was then literately applied to suppress the mismatch and enforce a desired velocity. The refined algorithm shows good parallels performance for simulation of flow driven by a Rushton turbine, the simulation results for phase-resolved velocity field, turbulent kinetic energy level and the flow vertex core development near the impeller compared well with available experimental data. Derksen (2001) applied this method to predict flow field driven by a pitched blade impeller in a baffled stirred tank, with a well resolved grid system, he reported good comparison with experimental data in terms of phase averaged velocity distributions and phase resolved velocity field. Lu et al. (2002) also implemented the LBM-LES black box method on a non-uniform grid, and achieved significant CPU time cost reduction with comparable numerical accuracy with previous predictions on an uniform grid. Revstedt et al. (2000) presented LBM based LES simulation of a baffled reactor stirred by a single Rushton turbine , the motion of the blades was described as time-dependent momentum sources.

One has to point out here that although with the LBM-LES approach a time-dependent solution can be obtained, the black-box modeling of impeller as distributed forces can not truly represent the real rotating geometry effect on the flow. Development of an LBM based sliding mesh technique is needed in order to incorporate the real rotating geometry effects into the solution procedure, this requires an accurate, stable numerical scheme that can effectively transfers flow information across the sliding interface, while maintains the conservation of local mass and momentum. To develop such a LBM sliding interface scheme is one of the research focus in this study.

1.3 Objective of this study

The objective of this work is to develop boundary condition algorithm to improve the accuracy of the volumetric LBM boundary schemes originally proposed by Chen et al. (1998), and to extend the volumetric LBM boundary scheme to handle sliding mesh interface condition encountered in sliding mesh simulation, as well as to demonstrate the performance of the proposed algorithms in various flow test cases.

The primary contributions of this thesis are as follows:

1. A close investigation of the original volumetric LBM boundary scheme propose by Chen et al. (1998) for its order of accuracy on arbitrary lattice/wall boundary configuration. The deficiency and source of error for the original volumetric boundary scheme is explored.

2. Development of boundary condition scheme to to improve the solution accuracy of the volumetric LBM boundary implementation. The proposed scheme applies a surface scattering correction procedure to redistribute the surfel scattered particles based on local flow variation, it achieves a close-to-second order accuracy on arbitrary lattice/wall boundary configuration, and reduces the solution dependence on lattice/wall boundary alignment and inclination. Validation tests have been used to demonstrate its capability for accurate flow prediction with complex geometries.

3. Extending the volumetric boundary scheme to impose sliding mesh interface conditions in a sliding mesh simulation. The approach uses a modified specular reflection boundary condition to couple flow solutions across sliding interface, it exactly conserves the mass and momentum flux across the sliding mesh interface, and enables the use of Cartesian grid to perform sliding mesh simulation with LBM. This is subsequently verified and validated using well known flow benchmarks.

1.4 Outline of the thesis

The format of this dissertation is outlined as follows. In the next chapter, an introduction of lattice Boltzmann methods and it's connection with kinetic theory is given. The types of boundary conditions are discussed.

In chapter 3, a detailed description of the LBM volumetric bounce-back reflection scheme and specular reflection boundary scheme is given. This is followed by a close examination of assumptions used in the original volumetric scheme, the source of error has been identified that affects numerical accuracy. Carefully designed numerical experiments are presented to study the accuracy of the volumetric scheme for flow with arbitrarily configured boundaries.

In chapter 4, a procedure of the particle scattering correction is proposed to improve the solution accuracy with the volumetric boundary scheme. The particle scattering correction process redistribute the surfel scattered particles in its affecting volume, the algorithm takes into account of the local flow variation and does not alter the local conservation of mass. Numerical test are conducted to demonstrate the improved order of accuracy with this scheme, the scheme is also shown to be less dependent on lattice/boundary orientations. A set of classical benchmarks has been presented to prove the accuracy of this scheme for flows involving curved boundary.

In chapter 5, a volumetric approach for enforcing sliding mesh interface condition is presented. This approach extends the volumetric specular reflection boundary algorithm to couple flow solution across sliding mesh interface, enables the use of LBM for sliding mesh simulation with exact rotating geometry. Detailed procedure of implementating this LBM sliding mesh approach on a Cartesian grid is presented This is demonstrated by simulating flows past a rotating propeller.

The conclusions of this research and directions for future work are presented in Chapter 6.

CHAPTER 2. THE LATTICE BOLTZMANN METHOD

In this chapter we present the formulations of lattice Boltzmann method. The discretized equations is shown to be directly derived from the continuum Boltzmann equation, and the resulting macroscopic dynamics of the system obeys the Navier-Stokes equation when Knudsen number is small. The kinetic theory based boundary conditions are straightforward to implement on complex geometries, and finally the extension of LBM to model high Reynolds number turbulent flow is discussed, followed by a set of comparisons with N-S based method to demonstrate LBM's advantages and limitations.

2.1 Lattice Boltzmann equations

Historically, the LBE methods was originated from the lattice gas automata (LGA) (Wolfram (1986); Frisch et al. (1986)), and was proposed as a floating point version of LGA aiming at eliminating the statistical noise plaguing LGA simulations. With the introduce of linearized particle collision operator and the use of single time relaxation approximations (Bhatnagar et al. (1954)) for the particle collision, LBM has becomes a very efficient numerical algorithm for flow computations (Chen et al. (1991); Qian et al. (1992)).

The general lattice Boltzmann equation (LBE) used in literature has the following form (Chen and Doolen (1998)),

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i \Delta t, t + \Delta t) - f_i(\boldsymbol{x}, t) = \Omega_i(\boldsymbol{x}, t), \tag{2.1}$$

in which f_i is the particle velocity distribution function indicating the probability of finding a particle at location $\boldsymbol{x}$ at time t with discrete velocity $\boldsymbol{c}_i$. Here $\boldsymbol{c}_i (i = 0, ...b)$ is a finite set of constant vectors that spans the particle velocity space. $\boldsymbol{c}_i \Delta t$ and Δt are space and time increments respectively. On the right hand side of equation 2.1, $\Omega_i(\boldsymbol{x}, t)$, is the collision term that represents the changes in the particle velocity distributions due to the particle-particle interactions. The collision term could have different forms which obey certain conservation laws (such as mass, momentum conservations and so on). The simplest and also most popular one is the so called Bhatnagar-Gross-Krook (BGK) approximation (Bhatnagar et al. (1954); Chen et al. (1991); Qian et al. (1992)), which describes the effect of collision as a process to restore particle distribution to its local equilibrium:

$$\Omega_i(\boldsymbol{x},t) = -\frac{f_i(\boldsymbol{x},t) - f_i^{eq}(\boldsymbol{x},t)}{\tau}. \tag{2.2}$$

Here τ is a single relaxation time parameter that measures the average time for the distributions to relax to its equilibrium via collisions, and f_i^{eq} is the local equilibrium distribution function which depends on local hydrodynamic properties.

The basic hydrodynamic quantities, such as fluid density ρ and velocity $\mathbf{u}$, are obtained through simple moment summations,

$$\begin{cases} \rho(\mathbf{x},t) = \sum_i f_i(\mathbf{x},t); \\ \rho\mathbf{u}(\mathbf{x},t) = \sum_i \mathbf{c}_i f_i(\mathbf{x},t) \end{cases} \tag{2.3}$$

The three-dimensional D3Q19 model (Chen et al. (1991); Qian et al. (1992)) shown in figure 2.1 is used in the present study. The particle density distribution functions are cell centered and particle interacts with its neighborhood (either fluid particle or solid boundary) to generate fluid dynamics.

The local equilibrium distribution function has the following form such that the recovered macroscopic hydrodynamics satisfy the conservation laws:

$$\begin{aligned} f_i^{eq}(\boldsymbol{x},t) \quad &= \quad \rho(\boldsymbol{x},t)w_i[1 + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}(\boldsymbol{x},t)}{T} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}(\boldsymbol{x},t))^2}{2T^2} - \frac{\boldsymbol{u}(\boldsymbol{x},t)^2}{2T} \\ &\quad + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}(\boldsymbol{x},t))^3}{6T^3} - \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}(\boldsymbol{x},t)}{2T^2}\boldsymbol{u}(\boldsymbol{x},t)^2] \end{aligned} \tag{2.4}$$

where w_i are the weighting parameters (Qian et al. (1992); Shan et al. (2006)),

$$w_i = \begin{cases} 1/18, & \text{in 6 coordinate directions;} \\ 1/36, & \text{in 12 bi-diagonal directions;} \\ 1/3, & \text{rest particles} \end{cases} \tag{2.5}$$

and T is the lattice temperature which is set to $1/3$ in isothermal simulation.

The conservation of mass and momentum is guaranteed via the following equalities,

$$\begin{cases} \rho(\boldsymbol{x},t) = \sum_i f_i(\boldsymbol{x},t) = \sum_i f_i^{eq}(\boldsymbol{x},t); \\ \rho\boldsymbol{u}(\boldsymbol{x},t) = \sum_i \boldsymbol{c}_i f_i(\boldsymbol{x},t) = \sum_i \boldsymbol{c}_i f_i^{eq}(\boldsymbol{x},t) \end{cases} \tag{2.6}$$

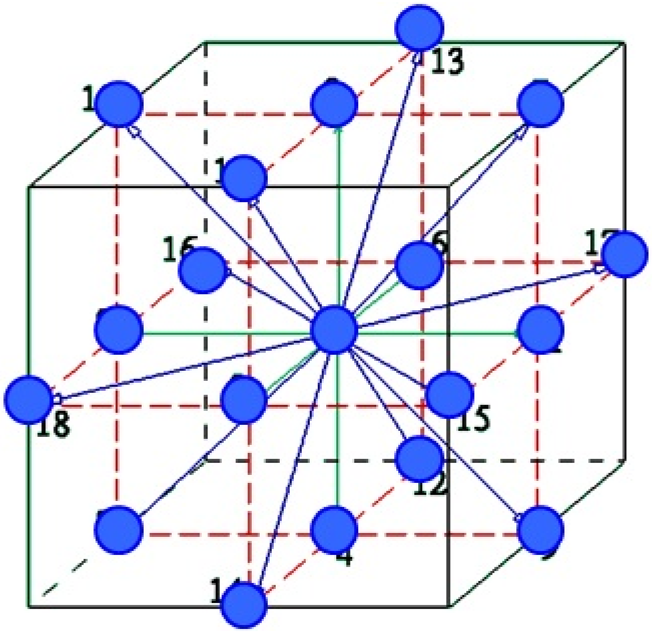


Figure 2.1   The D3Q19 LBE model used in this study

With timestep Δt being set to unity, equation 2.1 can be solved in a simple two-steps operation (figure 2.2):

1. A collision step that calculates the post-collide distribution $f_i^{'}(\boldsymbol{x}, t)$ as

$$f_i^{'}(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) - \frac{1}{\tau}[f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)] \tag{2.7}$$

2. An advection step which directly shifts particle distributions between two neighbors along the discrete velocity direction:

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i, t + 1) = f_i^{'}(\boldsymbol{x}, t) \tag{2.8}$$

Clearly, in the collision step, only local flow information such as ρ and $\rho\boldsymbol{u}$ are needed to calculate the equilibrium distribution f_i^{eq} defined in equation 2.4; while for advection step only binary exchange of data information between two neighboring lattice sites is needed. The local in space and time explicit nature of this collision-advection operations, together with the extreme simplicity of the LB equation 2.1, make the algorithm very easy to implement, and can achieve excellent scalability on parallel computers (Pohl et al. (2003), Velivelli and Bryden (2004), Clausen et al. (2010)).

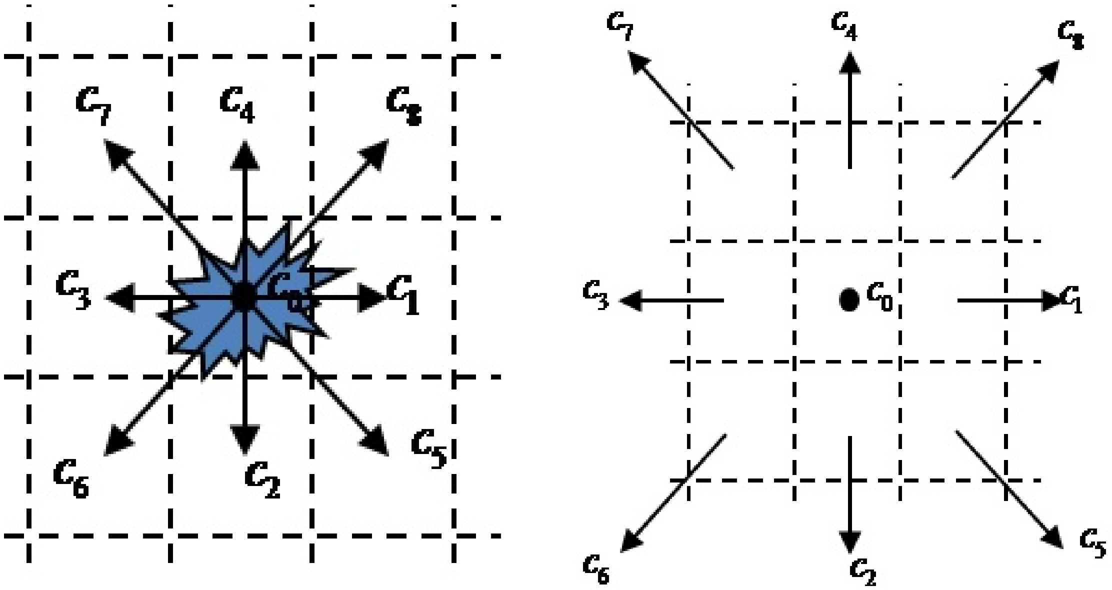


Figure 2.2 A two step implementation of LBE fluid dynamics: left, particle collision; right, particle advection

In the collision step, equation 2.7 can be re-written as:

$$\begin{aligned} f_i^{'}(\boldsymbol{x},t) &= f_i^{eq}(\boldsymbol{x},t) + (1-\frac{1}{\tau})[f_i(\boldsymbol{x},t) - f_i^{eq}(\boldsymbol{x},t)] \\ &= f_i^{eq}(\boldsymbol{x},t) + (1-\frac{1}{\tau})f_i^{neq}(\boldsymbol{x},t) \end{aligned} \tag{2.9}$$

with the non-equilibrium distribution $f_i^{neq}(\boldsymbol{x},t)$ defined as:

$$f_i^{neq}(\boldsymbol{x},t) = f_i(\boldsymbol{x},t) - f_i^{eq}(\boldsymbol{x},t) \tag{2.10}$$

This indicates that the collision process only afters the particle non-equilibrium distributions.

Equation 2.1 can be rewritten in a volumetric form (Chen (1998)), assuming a piece-wise constant particle distritution within cell volume $V(\boldsymbol{x})$:

$$N_i(\boldsymbol{x}+\boldsymbol{c}_i, t+1) - N_i(\boldsymbol{x},t) = \Omega_i(\boldsymbol{x},t)V(\boldsymbol{x}) = Q_i(\boldsymbol{x},t), \tag{2.11}$$

with $N_i(\boldsymbol{x},t) = V(\boldsymbol{x})f_i(\boldsymbol{x},t)$ is the number of particles within the cell volume with velocity $\boldsymbol{c}_i$.

In low-frequency and long-wave-length limit, one can recover the Navier-Stokes equations through a Chapman-Enskog expansion (Chen et al. (1992)). The resulting equation of state is that of an ideal gas fluid, namely the pressure p obeys a linear relation with density and temperature,

$$p = \rho T. \tag{2.12}$$

The kinematic viscosity of the fluid is related to the relaxation parameter by (Frisch et al. (1986); Chen et al. (1991); Qian et al. (1992))

$$\nu = (\tau - \frac{1}{2})T. \tag{2.13}$$

With above definition, the numerical error that was associated with the discretization of the advection operator in LBE (equation 2.1) is absorbed into the viscosity, the resulting LBGK scheme has been shown to be both second order accurate in time and space (Succi (2001)).

When τ is close to 0.5, fluid viscosity is small, in other words, particle distributions may become negative ($f_i(\boldsymbol{x}, t) < 0$) in some cells and then numerical instabilities occur quickly and contaminate the whole flow simulation. To suppress the negative distributions and improve numerical stability, a protection procedure of positivity for distributions is applied in our simulations. If a new local relaxation time parameter $\tau\prime$ in each cell is defined as the following (Li et al. (2004)):

$$\tau\prime(\boldsymbol{x}, t) = \max[\tau, 1 - \frac{f_i^{eq}(\boldsymbol{x}, t)}{f_i(\boldsymbol{x}, t)}] \qquad \text{for} \qquad \{i = 0, ...18\} \tag{2.14}$$

a positive distribution is guaranteed after each collision as long as it is positive before the collision. In other words, by effectively adding a local viscosity lower bound, which is dynamically local distribution dependent, we are able to keep distributions positive and improve the simulation stability.

2.2 Derivation of LBE from the continuum Boltzmann equation

The lattice Boltzmann equation 2.1 can be directly derived from the continuum Boltzmann kinetic equation through a Hermite-expansion based phase space discretization (Shan et al. (2006)). Let's start with the continuum Boltzmann equation that describes the evolution of the single-particle distribution function $f(\boldsymbol{x}, \boldsymbol{c}, t)$ in D-dimensional space based on the BGK collision model:

$$\frac{\partial f(\boldsymbol{x}, \boldsymbol{c}, t)}{\partial t} + \boldsymbol{c} \cdot \nabla f(\boldsymbol{x}, \boldsymbol{c}, t) = -\frac{1}{\tau_0}\left[f(\boldsymbol{x}, \boldsymbol{c}, t) - f^{eq}(\boldsymbol{x}, \boldsymbol{c}, t)\right]. \tag{2.15}$$

Here, τ_0 is the characteristic relaxation time of collisions to equilibrium and is related to fluid kinetic viscosity ν as $\tau_0 = \nu/T$. f^{eq} represents a local equilibrium distribution (*i.e.* Maxwell-Boltzmann) in the reference frame moving with the bulk flow:

$$f^{eq}(\boldsymbol{x}, \boldsymbol{c}, t) = \frac{\rho(\boldsymbol{x}, t)}{(2\pi\theta)^{D/2}} \exp\left[-\frac{(\boldsymbol{c} - \boldsymbol{u}(\boldsymbol{x}, t))^2}{2\theta}\right]. \tag{2.16}$$

The particle distribution function $f(\boldsymbol{x}, \boldsymbol{c}, t)$ can be projected onto Hermite basis with the dimensionless Hermite ortho-normal polynomials $\mathcal{H}^{(n)}(\boldsymbol{c})$ defined in velocity space $\boldsymbol{c}$:

$$f(\boldsymbol{x}, \boldsymbol{c}, t) = \omega(\boldsymbol{c}) \sum_{n=0}^{\infty} \frac{1}{n!} \boldsymbol{a}^{(n)}(\boldsymbol{x}, t) \mathcal{H}^{(n)}(\boldsymbol{c}). \tag{2.17}$$

where the dimensionless expansion coefficients, $\boldsymbol{a}^{(n)}(\boldsymbol{x}, t)$, is an integration over the entire dimensionless velocity space of $\boldsymbol{c}$ given by:

$$\boldsymbol{a}^{(n)}(\boldsymbol{x}, t) = \int f(\boldsymbol{x}, \boldsymbol{c}, t) \mathcal{H}^{(n)}(\boldsymbol{c}) d\boldsymbol{c}, \tag{2.18}$$

With Hermite polynomials as the expansion basis, all the expansion coefficients are linear combinations of the velocity moments of f, and the first few expansion coefficients can be directly connected to the five fundamental thermo-hydrodynamic variables, ρ, $\boldsymbol{u}$ and θ $(= 2\epsilon/D)$, and the momentum flux tensor $\boldsymbol{P}$ (or its traceless part, the stress tensor $\boldsymbol{\sigma}$):

$$\begin{aligned}
\boldsymbol{a}^{(0)} &= \int f d\boldsymbol{c} = \rho && \text{(2.19a)}\\
\boldsymbol{a}^{(1)} &= \int f \boldsymbol{c} d\boldsymbol{c} = \rho \boldsymbol{u} && \text{(2.19b)}\\
\boldsymbol{a}^{(2)} &= \int f(\boldsymbol{c}^2 - \boldsymbol{\delta}) d\boldsymbol{c} = \boldsymbol{P} + \rho(\boldsymbol{u}^2 - \boldsymbol{\delta}) = \boldsymbol{Q} + \boldsymbol{u}\boldsymbol{a}^{(2)} - (D-1)\rho \boldsymbol{u}^3 && \text{(2.19c)}
\end{aligned}$$

Since the leading moments of a distribution function up to N-th order are preserved by truncations of the higher order terms in its Hermite expansion, a distribution function $f(\boldsymbol{x}, \boldsymbol{c}, t)$ can be approximated by its truncated Hermite expansion $f^N(\boldsymbol{x}, \boldsymbol{c}, t)$, a projection onto a Hilbert subspace spanned by the first N Hermite polynomials without altering the first N moments, *i.e.*:

$$f(\boldsymbol{x}, \boldsymbol{c}, t) \approx f^N(\boldsymbol{x}, \boldsymbol{c}, t) = \omega(\boldsymbol{c}) \sum_{n=0}^{N} \frac{1}{n!} \boldsymbol{a}^{(n)}(\boldsymbol{x}, t) \mathcal{H}^{(n)}(\boldsymbol{c}). \tag{2.20}$$

This also guaranties that a fluid dynamic system can be constructed by a finite set of macroscopic variables (thermo-hydrodynamic moments).

The discretization of Hermite truncated distribution function $f^N(\boldsymbol{x}, \boldsymbol{c}, t)$ involves the use of the Gauss-Hermite quadrature to evaluate the expansion coefficients, $\boldsymbol{a}^{(n)}(\boldsymbol{x}, t)$, which can be expressed as a weighted sum of the functions calculated at discrete velocities $\boldsymbol{c}_i$:

$$\boldsymbol{a}^{(n)} = \sum_{i=1}^{d} \frac{w_i}{\omega(\boldsymbol{c}_i)} f^N(\boldsymbol{x}, \boldsymbol{c}_i, t) \mathcal{H}^{(n)}(\boldsymbol{c}_i). \tag{2.21}$$

where w_i and $\omega(\boldsymbol{c}_i)$, $i = 1, \cdots, d$, are respectively the weights and abscissa of a Gauss-Hermite quadrature of a degree $\geq 2N$.

By substituting equation 2.21 and 2.20 into equation 2.15 and after some straightforward algebra, the governing equations for discretized velocity distributions can then be directly evaluating at $\boldsymbol{c}_i$:

$$\frac{\partial f_a}{\partial t} + \boldsymbol{c}_i \cdot \nabla f_i = -\frac{1}{\tau}\left[f_i - f_i^{eq}\right] \quad a = 1, \cdots, d. \tag{2.22}$$

where on the right-hand-side,

$$\begin{aligned} f_i^{eq} &= w_i \rho[1 + \boldsymbol{c}_i \cdot \boldsymbol{u} + \frac{1}{2}[(\boldsymbol{c}_i \cdot \boldsymbol{u})^2 - u^2 + (\theta - 1)(\boldsymbol{c}_i^2 - D)] \\ &\quad + \frac{1}{6}\boldsymbol{c}_i \cdot \boldsymbol{u} 6[(\boldsymbol{c}_i \cdot \boldsymbol{u})^2 - 3u^2 + 3(\theta - 1)(\boldsymbol{c}_i^2 - D - 2)] + \cdots] \end{aligned} \tag{2.23}$$

Equation 2.22 is a differiential equation for the discrete velocity distributions. It can then be discretized in space and time by first integrating the equation along the velocity characteristicsin time:

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i, t+1) - f_i(\boldsymbol{x}, t) = -\frac{1}{\tau}\int_t^{t+1} dt' \left[f_i(\boldsymbol{x} + \boldsymbol{c}_i(t'-t), t') - f_i^{eq}(\boldsymbol{x} + \boldsymbol{c}_i(t'-t), t')\right] \tag{2.24}$$

The right-hand side of above equation can be approximated by a simple trapezoidal rule intergration, which leads to:

$$\begin{aligned} f_i(\boldsymbol{x} + \boldsymbol{c}_i, t+1) - f_i(\boldsymbol{x}, t) &\approx -\frac{1}{2\tau}\left[f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)\right] \\ &\quad -\frac{1}{2\tau}\left[f_i(\boldsymbol{x} + \boldsymbol{c}_i, t+1) - f_i^{eq}(\boldsymbol{x} + \boldsymbol{c}_i, t+1)\right] \end{aligned} \tag{2.25}$$

Introducing a shifted distribution function such that (He et al. (1998)):

$$\bar{f}_i(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) + \frac{1}{2\tau}\left[f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)\right] \tag{2.26}$$

Substituting above equation into equation 2.25, we arrive at the following approximation form for the intergrated BGK-Boltzmann equation:

$$\bar{f}_i(\boldsymbol{x} + \boldsymbol{c}_i, t+1) - \bar{f}_i(\boldsymbol{x}, t) = -\frac{1}{\tau_0 + 0.5}(\bar{f}_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)) \tag{2.27}$$

With $\tau = \tau_0 + 0.5 = \frac{\nu}{T} + 0.5$, and using the "shifted" distribution in the discrete equation, the standard form of the Lattice Boltzmann Equation 2.1 can be obtained.

In the above derivation, since both $\boldsymbol{x}$ and $\boldsymbol{x} + \boldsymbol{c}_i$ are lattice centroid locations, this directly implies a unity CFL (Courant-Friedrichs-Lewy) number : $|\boldsymbol{c}_i \Delta t|/\Delta x = 1$.

2.3 Recovering macroscopic equations

To derive the macroscopic equations at small Knudsen number $\mathcal{K}$ (where the Navier-Stokes description is valid), we make use of the multi-scaling Chapman-Enskog expansion (Chapman and Cowling (1970)) of discrete particle distribution f_i and time derivative ∂_t, in powers of $\mathcal{K}$:

$$\begin{cases} f_i = f_i^{(0)} + \mathcal{K} f_i^{(1)} + \mathcal{K}^2 f_i^{(2)} + \cdots; \\ \partial_t = \mathcal{K}\partial_t^{(0)} + \mathcal{K}^2 \partial_t^{(1)} + \cdots \end{cases} \tag{2.28}$$

where $f_i^{(0)} = f_i^{eq}$ corresponds to equilibrium distribution in equation 2.4

Substituting the expansion 2.28 into the discrete velocity Boltzmann equation 2.22, and matching the terms in the same order of $\mathcal{K}$, the Boltzmann-BGK equation can be turned into an infinite hierarchy of equations according to the power of $\mathcal{K}$,

$$\sum_{k=0}^{n-1} \partial_{t_k} f_i^{(n-k-1)} + \boldsymbol{c}_i \cdot \nabla f_i^{(n-1)} = -\frac{1}{\tau} f_i^{(n)}, \qquad \text{for} \qquad n = 1, 2, \cdots \tag{2.29}$$

for the first and the second order of particle distributions, we have:

$$(\partial_t^{(0)} + \boldsymbol{c}_i \cdot \nabla) f_i^{(0)} = -\frac{1}{\tau} f_i^{(1)} \tag{2.30a}$$

$$(\partial_t^{(0)} + \boldsymbol{c}_i \cdot \nabla) f_i^{(1)} + \partial_t^{(1)} f_i^{(0)} = -\frac{1}{\tau} f_i^{(2)} \tag{2.30b}$$

Taking the first two moments of above equations and combining the results together we arrive at the following conservation equations for mass and momentum:

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0 \tag{2.31}$$

$$\partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\boldsymbol{P}) = 0 \tag{2.32}$$

where $\boldsymbol{P} = \boldsymbol{P}^{(0)} + \boldsymbol{P}^{(1)}$ is the momentum flux tensor, with its equilibrium part $\boldsymbol{P}^{(0)}$ and non-equilibrium part $\boldsymbol{P}^{(0)}$ defined as:

$$\boldsymbol{P}_{ij}^{(k)} = \sum_i \boldsymbol{c}_i \boldsymbol{c}_j f_i^{(k)}, k = 0, 1 \tag{2.33}$$

For the D3Q19 model with given equilibrium distribution 2.4, the above equations yield:

$$\boldsymbol{P}_{ij}^{(0)} = p\delta_{ij} + \rho u_i u_j \tag{2.34}$$

$$\boldsymbol{P}_{ij}^{(1)} = -\nu(\nabla_i(\rho \boldsymbol{u}_j) + \nabla_j(\rho \boldsymbol{u}_i)) \tag{2.35}$$

Here p is the pressure that obeys the equation of state for idea gas $p = \rho/3$, and $\nu = (\tau - 0.5)/3$ (He et al. (1998)) is the kinematic fluid viscosity. The resulting momentum equation is the same as the Navier-Stokes equations with an error of $O(Ma^3)$ (Shan et al. (2006)).

2.4 LBM wall boundary condition

In LBM the wall boundary conditions are realized via particle reflections from solid surfaces. Two types of kinetic theory based LBE boundary conditions: the “bounce-back reflection” boundary condition and the “specular reflection” boundary condition are discussed in this section.

The bounce-back reflection boundary condition is generally used to achieve a “friction wall” effect: with bounce-back reflection, the velocity of a particle is completely reverted after the wall-particle interaction(figure2.3). This process is realized in terms of particle distribution as:

$$f_{i^*} = f_i, \qquad \text{with} \qquad \boldsymbol{c}_{i^*} = -\boldsymbol{c}_i \tag{2.36}$$

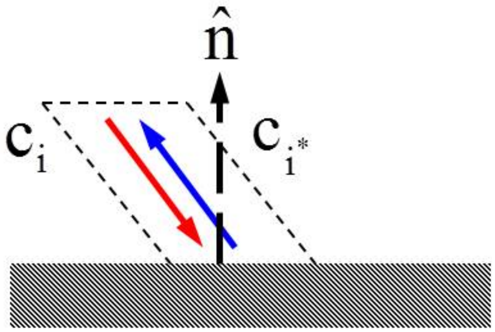


Figure 2.3 LBM bounce-back reflection boundary condition

The specular reflection boundary condition is usually applied to achieve free-slip boundary conditions, it implies that particles reflect off of the solid wall with the angles of incidence and reflection being equal (figure 2.4). The magnitude of the velocity after the collision is the same as the velocity before the collision. Even though the component of the velocity vector normal to the surface changes sign, the tangential component is preserved so that a frictionless wall property is achieved. The reflected velocity, $\boldsymbol{c}_{i^r}$, can be related to the incident particle velocity, $\boldsymbol{c}_i$, by:

$$\boldsymbol{c}_i = \boldsymbol{c}_r - 2\hat{\boldsymbol{n}}(\boldsymbol{c}_r \cdot \hat{\boldsymbol{n}}), \qquad \text{for} \qquad \boldsymbol{c}_{i^r} \cdot \hat{\boldsymbol{n}} \geq 0 \tag{2.37}$$

where $\hat{\boldsymbol{n}}$ is the surface normal vector.

During the specular reflection process, the reflected distribution is equal to the incident distribution:

$$f_{i^r} = f_i, \tag{2.38}$$

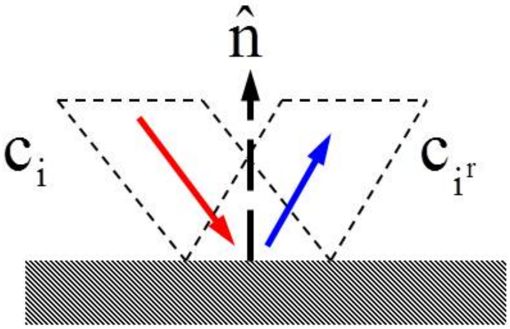


Figure 2.4 LBM specular reflection boundary condition

The effect of the boundary condition processes above on the local mass and momentum changes near wall fluids can be interpreted via a "volumetric" approach (Chen et al. (1998)). For wall surface with area A and surface normal $\hat{\boldsymbol{n}}$, within a time interval Δt and along the incoming $\boldsymbol{c}_i$ velocity direction, only the particles from the spatial volume V_i that can reach the wall boundary. This volume is defined as

$$V_i = |\hat{\boldsymbol{n}} \cdot \boldsymbol{c}_i| A \Delta t, \qquad \text{for} \qquad \boldsymbol{c}_i \cdot \hat{\boldsymbol{n}} \leq 0 \tag{2.39}$$

and the total in-coming particles within this volume are:

$$\Gamma_i^{in} = f_i V_i = f_i |\hat{\boldsymbol{n}} \cdot \boldsymbol{c}_i| A \Delta t, \qquad \text{for} \qquad \boldsymbol{c}_i \cdot \hat{\boldsymbol{n}} \leq 0 \tag{2.40}$$

For bounce-back reflection, the out-going particles are having the same volume as the in-coming ones:

$$\Gamma_{i^*}^{out} = f_{i^*} V_{i^*} = f_{i^*} |\hat{\boldsymbol{n}} \cdot \boldsymbol{c}_{i^*}| A \Delta t, \qquad \text{for} \qquad \boldsymbol{c}_{i^*} \cdot \hat{\boldsymbol{n}} > 0 \tag{2.41}$$

with equation 2.36, it's straightforward to show that the bounce-back scheme introduce zero mass into the local fluid:

$$\sum_{i^*} \Gamma_{i^*}^{out} - \sum_i \Gamma_i^{in} = 0 \tag{2.42}$$

Thus the volumetric bounce-back reflection conserves the local mass exactly. It can be shown that this is also true for the specular reflection boundary condition as well.

Since the bounce-back reflection reverts both the normal and tangential velocity components of incoming particles, the corresponding particle momentum change is directly related to the wall normal/tangential forces that exerted on the fluid:

$$\boldsymbol{F} = \frac{1}{A\Delta t}[\sum_{i^*} \boldsymbol{c}_{i^*}\Gamma_{i^*}^{out} - \sum_{i} \boldsymbol{c}_i\Gamma_i^{in}] \tag{2.43}$$

In the hydrodynamic limit, one arrives at the familiar surface pressure p and wall friction force $\tau_w = \mu\frac{\partial \boldsymbol{u}_t}{\partial \hat{\boldsymbol{n}}}$ (Chen et al. (1998)):

$$\boldsymbol{F} = p\hat{\boldsymbol{n}} + \tau_w\hat{\boldsymbol{t}} \tag{2.44}$$

The above LB boundary conditions is conceptually straightforward to implement in LBM, and its simplicity in handling arbitrary complex geometries significantly promotes the wide use of LBM in engineering applications (Chen et al. (2003)).

2.5 Turbulence modeling

In order to account for the contributions from unresolved turbulent fluctuations, the LBE is extended by replacing its molecular relaxation time scale with an effective turbulent relaxation time scale (Chen et al. (2003, 2004)), i.e., $\tau \rightarrow \tau_{\text{eff}}$, where, τ_{eff}, can be derived from a turbulence model, e.g. via systematic renormalization group (RNG) procedure as (Yakhot and Orszag (1986)):

$$\tau_{\text{eff}} = \tau + \frac{\nu_{eddy}}{T} \tag{2.45}$$

A modified two-equation, k-ε, model based on the original RNG formulation describes the sub-grid turbulence contributions ν_{eddy} (Yakhot and Orszag (1986); Yakhot and Smith (1992)), and is given by

$$\rho\frac{Dk}{Dt} = \frac{\partial}{\partial x_j}\left[\left(\frac{\rho\nu_o}{\sigma_{k_o}} + \frac{\rho\nu_T}{\sigma_{k_T}}\right)\frac{\partial k}{\partial x_j}\right] + \tau_{ij}S_{ij} - \rho\varepsilon \tag{2.46}$$

$$\rho\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial x_j}\left[\left(\frac{\rho\nu_o}{\sigma_{\varepsilon_o}} + \frac{\rho\nu_T}{\sigma_{\varepsilon_T}}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + C_{\varepsilon_1}\frac{\varepsilon}{k}\tau_{ij}S_{ij} - \left[C_{\varepsilon_2} + C_\mu\frac{{\eta_s}^3(1-\eta_s/\eta_o)}{1+\beta{\eta_s}^3}\right]\rho\frac{\varepsilon^2}{k} \tag{2.47}$$

where the parameter, $\nu_{eddy} = C_\mu k^2/\varepsilon$, is the eddy-viscosity in the RNG formulation.

The above equations can be solved on the same lattice, for example, using a modified Lax-Wendroff-like explicit time marching finite difference scheme (Teixeira (1998); Pervaiz and Teixeira (1999))

This methodology is also commonly referred to as LBM based Very Large Eddy Simulation (LBM-VLES). The LBE-VLES based description of turbulent fluctuations carries flow history and upstream information, and contains high order terms to account for the non-linearity of the Reynolds stress (Chen et al. (2004); Shan et al. (2006)). This is superior when compared with its Navier-Stokes counterpart, which uses the conventional linear eddy viscosity based Reynolds stress closure models and produces excessive dissipation when doing unsteady simulations (Chen et al. (2003, 2004)).

2.6 Advantages and drawbacks: comparison LBM with Navior-Stokes equations based numerical method

The kinetic nature of LBM introduces very unique features that distinguish it from the Navior-Stokes equation based numerical methods (Chen and Doolen (1998); Aidun and Clausen (2010)).

1. the N-S based methods solve the second order macroscopic PDEs, while a discrete velocity LBM model is a set of first-order PDEs.

2. The N-S solver has to deal with the nonlinear convection terms, while in LBM the convection terms are linear and with constant velocity values, it is handled by a simple straightforward advection scheme.

3. As an explicit method, the Courant-Friedrichs-Lewy (CFL) number in LBM is always set to 1, while for N-S based time marching schemes, CFL number genearally is set to much less than 1 for numerical stability. This results in a very low numerical dissipation with LBM.

4. Solving the pressure field in the incompressible N-S equation requires solving an elliptic Poisson equation which is numerically difficult to calculate and requires non-local flow information, while LBM approximate the incompressible N-S equations in the nearly incompressible limit and the pressure is obtained by an equation of state based on only local flow information.

5. Due to the kinetic nature of the Boltzmann equation, the physics associated with the molecular level interaction can be incorporated more easily in the LBE model

6. In LBM the boundary condition is straightforward to implement, which can allow exact boundary treatment on complex geometry with Cartesion grid.

7. The simplicity of the LBM algorithm and the pure local operations enable LBM to be an ideal numerical algorithm for large scale parallel computing (Pohl et al. (2003); Velivelli and Bryden (2004); Clausen et al. (2010)).

On the other hand, LBM also has its own limitations, which includes:

1. The Cartesian grid used in LBM is cubic-cell based, which generally results in a large grid system (and computational demanding) when high resolution grid is needed.

2. LBM is intrinsically compressible, this may introduce non-negligible density variations for incompressible flow and compromise the solution accuracy.

3. Theoretical framework for extending LBM to solving more complex fluid, for instance in high speed flows, large temperature/density variation flow, and high density ratio multiphase flows, it is not well established.

CHAPTER 3. A VOLUMETRIC LBM BOUNDARY SCHEME

The LBM boundary scheme proposed by Chen et al. (1998) realizes the bounce-back condition and specular-reflection condition in a “volumetric” form and preserves the local mass exactly, this avoids the problems introduced by the point-wise interpolation based schemes. In this chapter, the basic elements to implement this volumetric scheme on arbitrary geometries are described, numerical algorithms for realizing bounce-back reflection and specular reflection conditions are presented, this is followed by discussion on major shortcoming associated with the original volumetric scheme. Numerical examples are presented to illustrate these problems.

3.1 Volumetric LBM boundary scheme

In this section, we’ll follow Chen et al. (1998) to discuss the detailed volumetric algorithms for implementing two basic LBM boundary conditions: the bounce-back reflection boundary condition and the specular reflection boundary condition.

3.1.1 Cartesian Grid for LBM Simulation

In LBM, a Cartesian grid system with regular cubic lattice cells is generally used to discretize the spatial fluid domain, for solid boundary, it is represented by a surface grid and is directly overlaid on top of the underlying Cartesian mesh to generate exact fluid/solid interface. The surface grids used are triangulations of the original CAD surface, the resulting planar surface elements are written in common STL or NASTRAN format for direct user import during simulation setup. Figure 3.1 shows an example of such surface grid and the underlying Cartesian fluid cells.

With specified user input, generation of volumetric grid for the fluid domain and creation of fluid/solid database for boundary conditions is completely automated during the discretization process. Such a

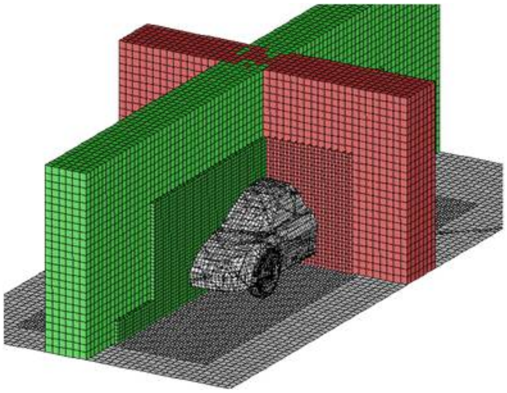


Figure 3.1 Surface grid used for a automobile model and the applied underlying Cartesian grid (from PowerFLOW Training Notes (2008))

Cartesian grid approach minimizes the needs for the mesh translation in the engineering CAD/CFD analysis, and is very effective in reducing the overall cost related to generation of body-conforming grids with very complex geometries (Aftosmis (1997)).

3.1.2 Surfel and pgram

The basic elements to enforce the volumetric LB boundary condition are purely particle dynamics based. For a solid surface that intersects the underlying fluid lattice cells, its surface elements (from now on the name of "surfel" is used for simplicity) interacts with its neighboring fluid cells to generate the particle-surface dynamics. Along given particle velocity, the particles that can advect to the surfel location within a unit timestep can only come from a bounded spatial domain that uniquely defined by its velocity and the surfel geometry. In numerical simulations with a discrete particle velocity model, such a domain can be mathematically defined as a set of extruded parallelogram (parallelepiped in three dimensions, named as "Pgram" in the following contents) that originated from the related surfel. The Pgram extends along the reverse direction of incoming particle velocity, and forms a closed volume containing all the incoming particles with specified advection velocity. Figure 3.2 gives such a geometry definition on a general curved surface boundary: for surfel α with unit normal $\hat{\boldsymbol{n}}$ and surface area of A^{α}, the Pgram for particle velocity $\boldsymbol{c}_i$ is defined by a volume $V^{\alpha}_{i,pgram} = |\boldsymbol{c}_i \cdot \hat{\boldsymbol{n}}| A^{\alpha} \Delta t$. Since the surfel orientation and location can be arbitrary with respect to the discrete particle velocity, only a subset of the discrete velocities can advect particles to the surfel, with corresponding velocity $\boldsymbol{c}_i$ satisfies condition

$\boldsymbol{c}_i \cdot \hat{\boldsymbol{n}} < 0$. For particles with velocity such that $\boldsymbol{c}_i \cdot \hat{\boldsymbol{n}} \geq 0$, they advect away from the surfel.

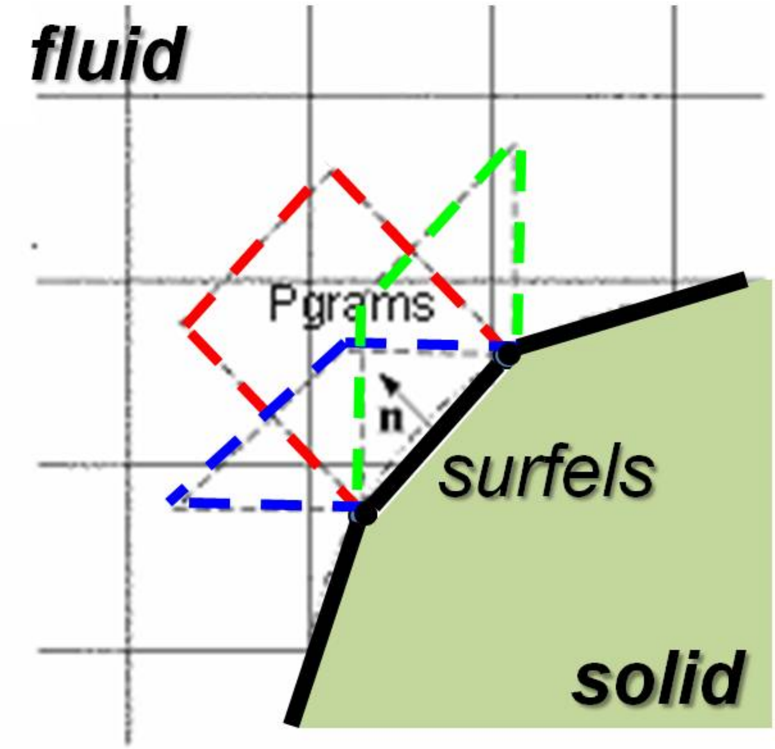


Figure 3.2   Definition of surfel and Pgram for a curved boundary

Let's illustrate this in a simple two-dimensional lattice configuration with the D2Q9 LBE model as shown in figure 3.3. Particle in the near wall lattice cell can advect along all its 9 velocity directions $\boldsymbol{c}_i (i = 0, 1, ..., 8)$ within a unit timestep. Among these discrete velocities, only the ones $\boldsymbol{c}_2, \boldsymbol{c}_3, \boldsymbol{c}_6$ that allow particle to directly impact surfel α, while for other velocities ($\boldsymbol{c}_1, \boldsymbol{c}_4, \boldsymbol{c}_5, \boldsymbol{c}_7$ and $\boldsymbol{c}_8$), particles directly advect to the neighboring fluid cells without interacting with surfel α.   Three Pgrams can be created for velocities $\boldsymbol{c}_2, \boldsymbol{c}_3, \boldsymbol{c}_6$, by simply gliding surfel α along their inverse directions with the same velocity magnitude (figure 3.4): Here the blue Pgram region represents the one formed for particle velocity $\boldsymbol{c}_3$, the red one represents Pgram for particle velocity $\boldsymbol{c}_6$, and the green one is for particle velocity $\boldsymbol{c}_2$.

With above volumetric boundary representation, surfels and Pgrams form the basic elements to realize particle/surface dynamics: Pgrams define confined spatial volumes that surfels can collect incoming particles with given velocity (the surfel gathering process), these surfel gathered particles can

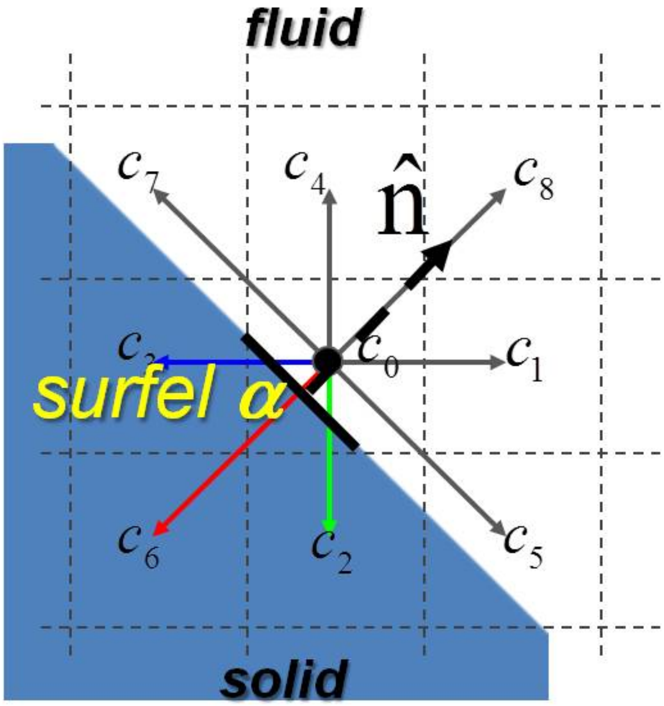


Figure 3.3 Particle advections along discrete velocities in a near-wall cell

redistributed back into the Pgram volumes according to the given boundary condition (the surfel scattering process). Both bounce-back reflection and specular reflection boundary conditions can be realized in such a surfel gathering/scattering process. Beside the processes of surfel gathering and scattering, intermediate surfel-particle interactions can also be constructed to enforce desired hydrodynamic boundary conditions (Chen et al. (1998)).

3.1.3 Pgram overlapped cells and near wall fluid dynamics

In the spatial domain, each Pgram related to surfel α and discrete particle velocity $\boldsymbol{c}_i$ overlaps m number of underlying fluid cells, these cells forms a region that they not only can receive particles advected from their neighborhood fluid cells, but also can receive surfel α scattered particles through this Pgram volume (figure 3.5). These cells are called Pgram-overlapped cells. Apparently, all the near wall fluid cells that are adjacent to surfels are Pgram-overlapped cells, the cells that are next to these near wall fluid cells (second layer near wall cells) may also be overlapped by a Pgram, depending on

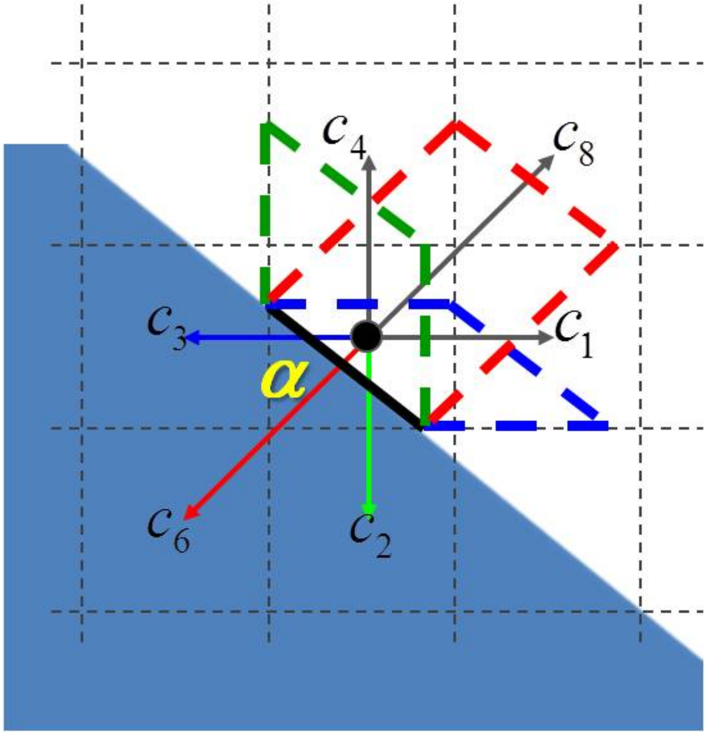


Figure 3.4   Three Pgrams are formed for surfel α with incoming particle velocities: $\boldsymbol{c}_2$, $\boldsymbol{c}_c$, $\boldsymbol{c}_6$

the actual surfel location and orientation.

The following geometry notations are used throughout the thesis content: For surfel α with incoming particle velocity $\boldsymbol{c}_i$, the Pgram has a geometrical volume $V^{\alpha}_{i,pgram}$, and this volume is splitted into m parts by the Pgram-overlapped cells $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ...$\boldsymbol{x}_m$ (figure 3.6), each part has a fraction volume of $V^{\alpha}_{i,pgram}(\boldsymbol{x}), \boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_m)$ such that:

$$V^{\alpha}_{i,pgram} = \sum_{\boldsymbol{x}=\boldsymbol{x}_1}^{\boldsymbol{x}_m} V^{\alpha}_{i,pgram}(\boldsymbol{x}) \tag{3.1}$$

The cell volume for a Pgram-overlapped cell is denoted by $V_{cell}(\boldsymbol{x})$, the cell volume is less than unity if a partial cell is formed by solid geometry intersection. For Pgram-overlapped cells, part of the total number of post-collide particles $N_i^{'}(\boldsymbol{x}, t)$ is directly advected to its neighborhood fluid cells, the remaining part is being collected by the related surfels through their Pgrams. The portion of these surfel collected particles is defined by

$$\begin{aligned} P^s_i(\boldsymbol{x}) &= \sum_{\alpha} \frac{V^{\alpha}_{i,pgram}(\boldsymbol{x})}{V_{cell}(\boldsymbol{x})} \\ &= \sum_{\alpha} P^{\alpha}_{i,pgram}(\boldsymbol{x}) \leq 1 \end{aligned} \tag{3.2}$$

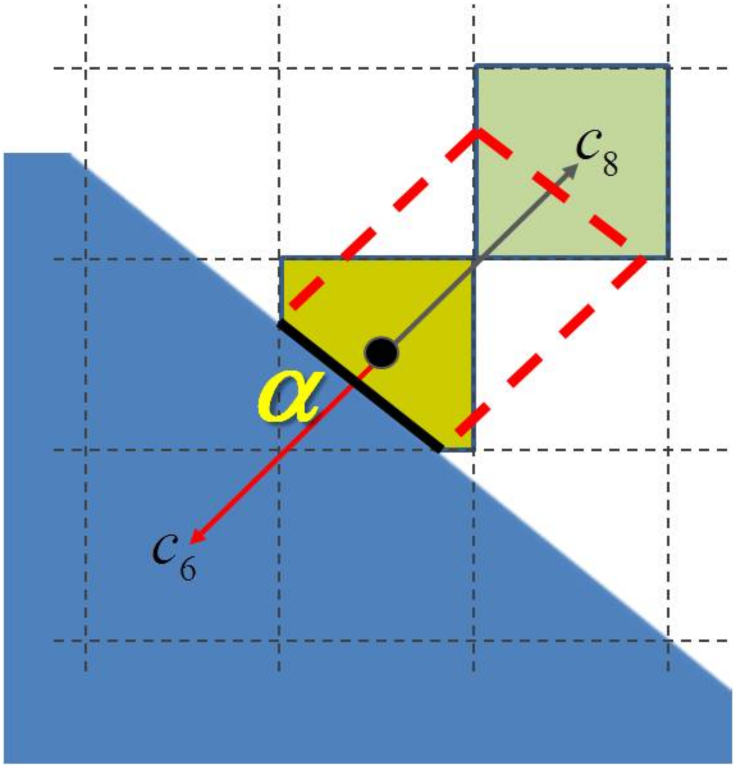


Figure 3.5 Illustration of near wall dynamics that involves Pgram (defined by surfel α and particle velocity $\boldsymbol{c}_6$ in the above example) overlapped cells: the fluid cells in green color receive particles along $-\boldsymbol{c}_6$ direction from both the yellow fluid cell as well as surfel α scattered particles within this Pgram region

With the above definitions, particle advections in these Pgram-overlapped cell can be expressed as:

$$N_i(\boldsymbol{x}+\boldsymbol{c}_i, t+1) = [1 - P_i^s(\boldsymbol{x})]N_i^{'}(\boldsymbol{x}, t) + Q_i(\boldsymbol{x}+\boldsymbol{c}_i, t) \tag{3.3}$$

In equation 3.3, the first term on the RHS represents the advected post-collide particles from its neighboring cell, and second term $Q_i(\boldsymbol{x}+\boldsymbol{c}_i, t)$ represents surfels scatted particles via all the overlapped Pgrams.

The calculation of surfels scattered particle numbers in equation 3.3 is straightforward: for fluid cell $\boldsymbol{x}$, there are N number of Pgrams overlapping it along the particle velocity $\boldsymbol{c}_i$ direction, which corresponds to N number of associated surfels. In each Pgram, among all the scattered particles $\Gamma_i^{out,\alpha}(t), (\alpha = \alpha_1, \alpha_2, \cdots, \alpha_N)$, the fraction (denoted by $\Gamma_i^{out,\alpha}(\boldsymbol{x}, t), (\alpha = \alpha_1, \alpha_2, \cdots, \alpha_N)$) that is redistributed into the corresponding overlapped Pgram volume $V_{i,pgram}^{\alpha}(\boldsymbol{x})$ can be expressed as a

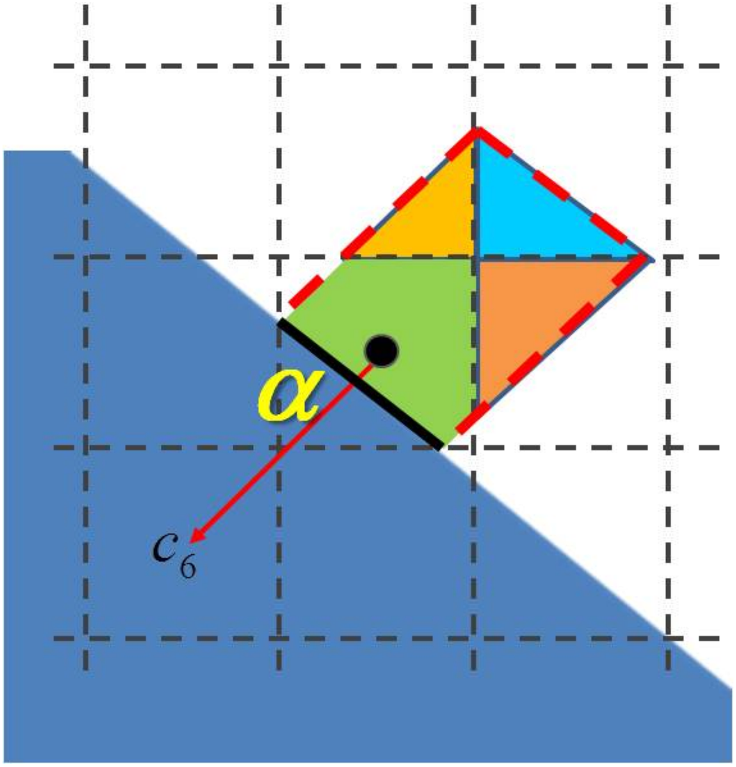


Figure 3.6 A Pgram defined by surfel α and particle velocity $\boldsymbol{c}_6$, it's splitted by the underlying fluid cells into four parts (each part is represented by different color)

function proportional to it's geometry weight

$$\Gamma_i^{out,\alpha}(\boldsymbol{x},t) = \frac{V_{i,pgram}^{\alpha}(\boldsymbol{x})}{V_{i,pgram}^{\alpha}} \Gamma_i^{out,\alpha}(t) \tag{3.4}$$

The overall surfel-scattered particles that being advected to this cell volume along $\boldsymbol{c}_i$ direction is simply a summation of all the contributions from related Pgrams (surfels):

$$\begin{aligned} Q_i(\boldsymbol{x},t) &= \sum_{\alpha} \Gamma_i^{out,\alpha}(\boldsymbol{x},t) \\ &= \sum_{\alpha} \frac{V_{i,pgram}^{\alpha}(\boldsymbol{x})}{V_{i,pgram}^{\alpha}} \Gamma_i^{out,\alpha}(t) \end{aligned} \tag{3.5}$$

The calculation of total outgoing particles $\Gamma_i^{out,\alpha}(t)$ for a particular surfel α and particle velocity $\boldsymbol{c}_i$ is directly related to the detailed surface boundary condition, this will be addressed in the next section.

In the above(equation 3.1 → 3.5), the extrusion of Pgram, the formation of partial cells, and the calculation of related Pgram volumes can be handled by standard geometry intersecting/clipping algorithms in computational geometry, In this study, these information is directly provided by the software

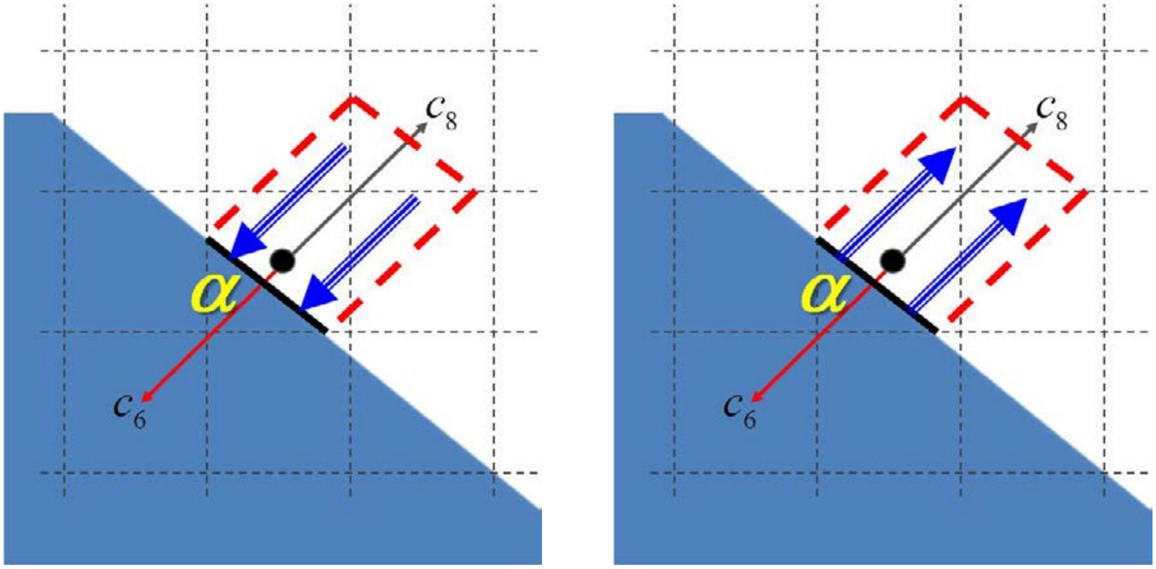


Figure 3.7 Surfel gathers incoming particles in the $\boldsymbol{c}_6$ direction (left) and scatters back outgoing particles in the $-\boldsymbol{c}_6$ direction through its Pgram

package of PowerFLOW, discussion of detailed implementation of these algorithms is beyond the scope of this thesis.

3.1.4 Volumetric bounce-back reflection algorithm

To realize volumetric bounce-back reflection boundary condition on surfel α, a two-step surfel gathering-scattering procedure is applied (figure 3.7):

1. Surfel gathering: Surfel α collects all the incoming particles $\Gamma_i^{in,\alpha}(t)$ along its $\boldsymbol{c}_i$ direction from the corresponding Pgram overlapped cells:

$$\Gamma_i^{in,\alpha}(t) = \sum_{\boldsymbol{x}} P_i^{\alpha}(\boldsymbol{x}) N_i(\boldsymbol{x}, t) \tag{3.6}$$

   Here $P_i^{\alpha}(\boldsymbol{x}) N_i(\boldsymbol{x}, t)$ represents the number of particles that comes from cell $\boldsymbol{x}$, since only $P_i^{\alpha}(\boldsymbol{x})$ fraction of its cell volume is overlapped with this Pgram and advect particles to the surfel α.

2. Surfel scattering: Surfel α scatters back all the incoming particles (with velocity $\boldsymbol{c}_i$) into the same Pgram volume, the outgoing particles revert their velocity direction to $-\boldsymbol{c}_i$

$$\Gamma_{i^*}^{out,\alpha}(t) = \Gamma_i^{in,\alpha}(t), \qquad \text{with} \qquad \boldsymbol{c}_{i^*} = -\boldsymbol{c}_i \tag{3.7}$$

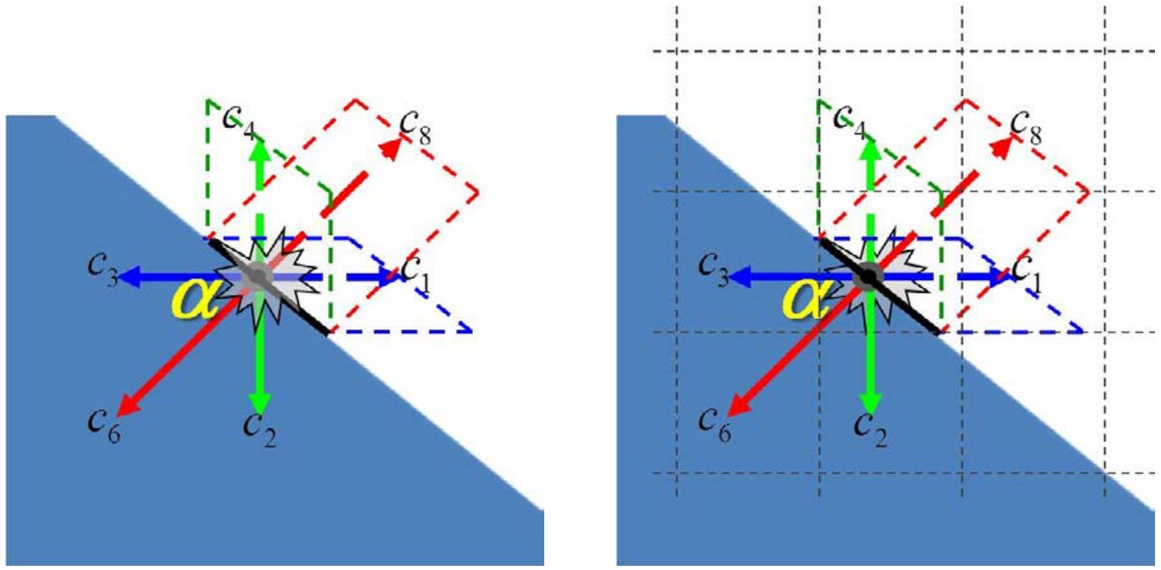


Figure 3.8 Surfel collision process and the constructed incoming/outgoing states

The scattered particles can then be redistributed within the Pgram overlapped cells according to equation 3.4.

3.1.5 Volumetric specular reflection algorithm

Due to the arbitrary location and orientation of actual surface, specular reflected particles from general surface could have velocity in any direction, it is thus very challenging to implement the specular reflection algorithm on an arbitrary curved surface with only a finite set of discrete particle velocities, since directly projecting these scattered particles onto the directions of discrete particle velocities is a non-trivial process.

Chen et al. (1998) introduced a novel idea to solve this problem: After surfel gathering process, an intermediate particle-surfel collision procedure was established, this was then followed by a surfel scatering process to scatter back the post-collided particles along the reversed velocity direction. To ensure zero change on the tangential momentum, the actual viscosity effect is set to 0 during the particle-surfel collision process, and surfel equilibrium distributions are constructed with local averaged density and tangential velocity. This allows to realize a volumetric specular-reflection condition with only a finite set of discrete particle velocities.

The procedure is detailed in the following steps:

1. Surfel gathering: surfel gathers the total incoming particles $\Gamma_i^{in,\alpha}(t)$, this step is the same as the one used in bounce-back reflection condition (equation 3.6).

2. Constructing surfel incoming states $f_{i^*}^{\alpha}$ and the corresponding equilibriums $f_{i^*}^{eq,\alpha}$. The surfel incoming states are constructed based on all the possible incoming particle velocities. For the simple case considered in figure 3.8, velocity $\boldsymbol{c}_2$, $\boldsymbol{c}_3$ and $\boldsymbol{c}_6$ form the incoming states for surfel α. The surfel incoming state distributions are calculated based on Pgram averaged distributions from its overlapped cells as:
$$f_{i^*}^{\alpha}(t) = \bar{f}_{i^*}^{\alpha}(t) \equiv \frac{\Gamma_i^{in,\alpha}(t)}{V_{i^*,pgram}^{\alpha}} \tag{3.8}$$
For these surfel states, the corresponding surfel equilibrium can be calcuated from the Pgram-averaged surfel density ρ^{α} and tangential velocity $\boldsymbol{u}_t^{\alpha}$ as:
$$\begin{aligned} f_{i^*}^{eq,\alpha}(t) \quad = \quad & \rho^{\alpha} w_i[1 + \frac{\boldsymbol{c}_{i^*} \cdot \boldsymbol{u}_t^{\alpha}}{T} + \frac{(\boldsymbol{c}_{i^*} \cdot \boldsymbol{u}_t^{\alpha})^2}{2T^2} \\ & - \frac{(\boldsymbol{u}_t^{\alpha})^2}{2T} + \frac{(\boldsymbol{c}_{i^*} \cdot \boldsymbol{u}_t^{\alpha})^3}{6T^3} - \frac{(\boldsymbol{c}_{i^*} \cdot \boldsymbol{u}_t^{\alpha})}{2T^2}(\boldsymbol{u}_t^{\alpha})^2] \end{aligned} \tag{3.9}$$

3. Surfel collision process. The in-coming particles collide with surfel, the post-collide particles revert their velocity direction and become out-going particles, their corresponding post-collide outgoing particle distribution obeys the BGK collision rule:
$$f_i^{\alpha'}(t) = f_i^{eq,\alpha}(t) + (1 - \frac{1}{\tau})[f_{i^*}^{\alpha}(t) - f_{i^*}^{eq,\alpha}(t)], \qquad \text{with} \qquad \boldsymbol{c}_{i^*} = -\boldsymbol{c}_i \tag{3.10}$$
In equation 3.10, the relaxation time is set to $\tau = 0.5$, which corresponds to a viscosity of $\nu = 0$. From the definition of $f_{i^*}^{eq,\alpha}(t)$ specified in equations 3.9, this zero viscosity value equivalently cancels the near wall shearing effect on the tangential momentum flux (Chen et al. (1998)), a frictionless boundary condition can thus be realized. The equilibrium distribution $f_i^{eq,\alpha}(t)$ can be obtained in the same manner as the one defined in equation 3.9.

4. Surfel scattering: Surfel α scatters the outgoing particles back into the related Pgram volume. Since the outgoing particle velocity is just a simple revert of the incoming particle velocity, same Pgram can be used for both surfel gathering and scattering.

The total amount of outgoing particles can be obtained via:

$$\Gamma_i^{out,\alpha}(t) = \bar{\Gamma}_i^{out,\alpha}(t) + \Delta_1 \Gamma_i^{out,\alpha}(t) \tag{3.11}$$

Here the first term on RHS represents the averaged number of out-going particles within the Pgram, with its value defined from the post-collide distributions:

$$\begin{aligned} \bar{\Gamma}_i^{out,\alpha}(t) &= V_{i,pgram}^{\alpha} f_i^{\alpha'}(t) \\ &= V_{i,pgram}^{\alpha}[f_i^{eq,\alpha}(t) + f_{i^*}^{eq,\alpha}(t)] - \Gamma_i^{in,\alpha}(t) \end{aligned} \tag{3.12}$$

The second term on the RHS of equation 3.11 corresponds to a mass correction procedure, since the process in equation 3.11 introduces a non-vanishing mass flux. This correction is defined as:

$$\begin{aligned} \Delta_1 \Gamma_i^{out,\alpha}(t) = & \frac{V_{i,pgram}^{\alpha}}{\sum_{i,\boldsymbol{c}_i\cdot\hat{\boldsymbol{n}}\leq 0,|\boldsymbol{c}_i|=\boldsymbol{c}_j} V_i^{\alpha}} \\ & [\sum_{i,\boldsymbol{c}_i\cdot\hat{\boldsymbol{n}}\leq 0,|\boldsymbol{c}_i|=\boldsymbol{c}_j} \Gamma_i^{in,\alpha}(t) - \sum_{i,\boldsymbol{c}_i\cdot\hat{\boldsymbol{n}}\geq 0,|\boldsymbol{c}_i|=\boldsymbol{c}_j} \bar{\Gamma}_i^{out,\alpha}(t)] \end{aligned} \tag{3.13}$$

The above algorithm can also be modified to include a finite friction force on the surfel with desired value. This was done by adding an extra term $\Delta_2 \Gamma_i^{out,\alpha}(t)$ into the RHS of equation 3.11:

$$\Delta_2 \Gamma_i^{out,\alpha}(t) = -K_f V_i^{\alpha} \boldsymbol{c}_i \cdot \hat{\boldsymbol{n}} \left[f_i^{eq,\alpha}(t) - f_{i^*}^{eq,\alpha}(t) \right] \tag{3.14}$$

K_f is the so-called skin friction factor and is connected to the tangential force $\boldsymbol{F}_f^{\alpha}$ via:

$$\boldsymbol{F}_f^{\alpha} = -K_f p^{\alpha} \boldsymbol{u}^{\alpha} \tag{3.15}$$

The generalized slip algorithm is the basis for incorporating turbulence wall shear stress model into LBM, particularly when simulating high Reynolds number turbulent flows with under-resolved near wall grid.

3.2 Connection to macroscopic boundary conditions

As already seen in section 2.4, volumetric bounce-back reflection can realize the desired wall friction in the hydrodynamic limit of the flow, here we only present the prove on connecting the above volumetric specular reflection algorithm to free-slip (zero tangential force) wall boundary condition.

Combining equation 2.43 and equation 3.11 together, we have:

$$\boldsymbol{F}^{\alpha}(t) = \frac{1}{A\Delta t}[\sum_{i^*} \boldsymbol{c}_{i^*}(\bar{\Gamma}_i^{out,\alpha}(t) + \Delta_1 \Gamma_i^{out,\alpha}(t)) - \sum_i \boldsymbol{c}_i \Gamma_i^{in,\alpha}(t)], \qquad \text{with} \qquad \boldsymbol{c}_{i^*} = -\boldsymbol{c}_i \quad (3.16)$$

Since the correction term $\Delta_1 \Gamma_i^{out,\alpha}(t)$ is a isotropic process, we can directly verify:

$$\sum_{i^*} \boldsymbol{c}_i \Delta_1 \Gamma_i^{out,\alpha}(t) = 0 \quad (3.17)$$

Substituting equation 3.12 into the remaining term of equation 3.16, and canceling out the in-coming particle flux, we have:

$$\boldsymbol{F}^{\alpha}(t) = \frac{1}{A\Delta t} = \frac{1}{A\Delta t} \sum_{i, \boldsymbol{c}_i \cdot \hat{\boldsymbol{n}} \geq 0} \boldsymbol{c}_i V_{i,pgram}^{\alpha} [f_i^{eq,\alpha}(t) + f_{i^*}^{eq,\alpha}(t)] \quad (3.18)$$

With the Pgram volume defined as $V_{i,pgram}^{\alpha} = |\boldsymbol{c}_i \cdot \hat{\boldsymbol{n}}^{\alpha}| A^{\alpha} \Delta t$, and applying the symmetry properties, we arrive at:

$$\begin{aligned} \boldsymbol{F}^{\alpha}(t) &= \hat{\boldsymbol{n}}^{\alpha} \cdot \sum_i \boldsymbol{c}_i \boldsymbol{c}_i f_i^{eq,\alpha}(t) \\ &= p^{\alpha} \hat{\boldsymbol{n}}^{\alpha} \quad (3.19) \end{aligned}$$

Since the resulting force only has component normal to the surface, this indicates that the tangential force is effectively zero and a free-slip boundary condition is thus achieved.

The local conservation of mass is also satisfied in the above defined specular reflection process since the mass correction in equation 3.13 exactly cancels the difference between incoming particles and out-going particles.

3.3 Drawback with the original volumetric scheme

The original volumetric boundary scheme can realize both the bounce-back reflection and specular reflection conditions on arbitrary geometry, it guarantees exact conservation of local mass for both boundary conditions, and can also conserve the local tangential momentum for specular reflection boundary condition. With a generalized slip algorithm, it can precisely control various hydrodynamic fluxes across the surel. The detailed balance condition is satisfied, and correspondingly the numerical surface noise associated with the boundary discretization is greatly reduced. The algorithm has been applied to many benchmark studies and the results are promising (Chen et al. (2003)).

However, this scheme has noticeable numerical diffusion near the solid surface, particularly when the solid boundary is arbitrarily oriented or offsets with respect to the underlying lattice cells (Li et al. (2004); Rohde (2004)). Rohde (2004) also reported that this scheme can achieve only first order of accuracy for flow problem with curved boundary.

The problem is directly linked to the surfel scattering process defined with equation 3.4 and equation 3.5. With such a definition, the scattered particles are evenly redistributed within its Pgram volume, this is equivalent to assume a constant particle distribution $f^{out,\alpha}_{i,pgram}(\boldsymbol{x},t)$ in the whole Pgram volume:

$$f^{out,\alpha}_{i,pgram}(\boldsymbol{x},t) = \bar{f}^{out,\alpha}_{i,pgram}(t) \equiv \frac{\Gamma^{out,\alpha}_{i}(t)}{V^{\alpha}_{i,pgram}} \tag{3.20}$$

However, the distribution function $f_i(\boldsymbol{x},t)$ can be decomposed into its equilibrium part $f^{eq}_i(\boldsymbol{x},t)$ and non-equilibrium part $f^{neq}_i(\boldsymbol{x},t)$ such that

$$f_i(\boldsymbol{x},t) = f^{eq}_i(\boldsymbol{x},t) + f^{neq}_i(\boldsymbol{x},t) \tag{3.21}$$

Considering only the equilibrium part, it can be further expanded up to second order as:

$$f^{eq}_i(\boldsymbol{x},t) = \rho(\boldsymbol{x},t)w_i[1 + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}(\boldsymbol{x},t)}{T} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}(\boldsymbol{x},t))^2}{2T^2} - \frac{\boldsymbol{u}(\boldsymbol{x},t)^2}{2T}] \tag{3.22}$$

Apparently the piece-wise constant distribution assumption is only valid when the formed Pgram is within one fluid cell (where $\rho(\boldsymbol{x},t)$ and $\boldsymbol{u}(\boldsymbol{x},t)$ just take the local cell averaged value), or particles distribute homogeneously within the Pgram volume. For problems with arbitrary located/oriented surface, the formed Pgram may overlaps several fluid cells, the second condition is only valid when the flow does not move, since the near wall flow region is highly viscous dominated, the wall shearing effect related to fluid motion results in a strong spatial variation of fluid quantities, the particle distribution thus differs cell by cell. Under such condition, enforcing an evenly redistribution of the out-going particles effectively mixes the solutions in the Pgram overlapped cells, this introduces high level of numerical diffusions into the system, and could seriously compromises the solution accuracy when wall shearing effect is not negligible. In the following numerical test, the effect of using equation 3.20 on solution accurary with general lattice-wall boundary configuration is illustated.

3.4 Convergence study with arbitrary oriented/offseted boundary

To study the accuracy of the original volumetric boundary scheme , the following numerical tests are constructed:

Considering a simple two dimensional channel flow situation where a constant gravity field g is applied to drive the flow. The channel height is H. When gravity is small the fully developed flow field is laminar and a well established parabolic profile of the stream wise velocity $U(y)$ can be obtained:

$$U(y) = -\frac{g}{2\nu}y^2 + \frac{g}{2\nu}Hy \tag{3.23}$$

Here y is the distance measured from the wall boundary and along the direction normal to the channel wall.

This problem can be easily solved on a grid system that has the near wall lattice cell boundaries exactly aligned with wall boundary. To make it more general, the grid system can be shifted in the wall normal direction so that the near wall cell has only a fraction of its volume being occupied by the fluid (lattice-offset configurations, see figure 3.9), or the grid system can rotate arbitrarily so that the wall boundaries have a inclined angle with respect to the fluid cell boundaries (the lattice-inclined configurations, see figure 3.10). Lattice offset and inclined arrangements are commonly encountered when an arbitrary geometry intersect with a Cartesian grid system.

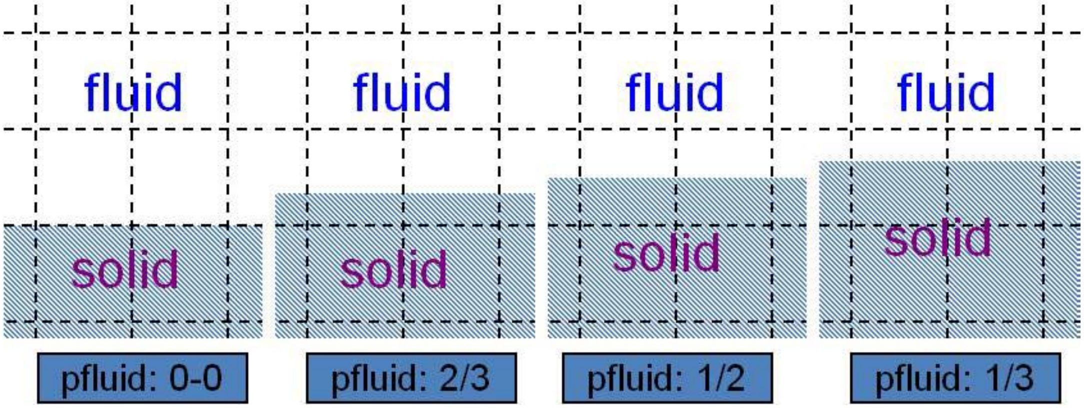


Figure 3.9 Lattice offset configurations. From left to right, configurations of: lattice aligned, lattice offset with pfluid $2/3$, lattice offset with pfluid $1/2$, lattice offset with pfluid $1/3$. Here pfluid measures the fraction of fluid in a boundary cell

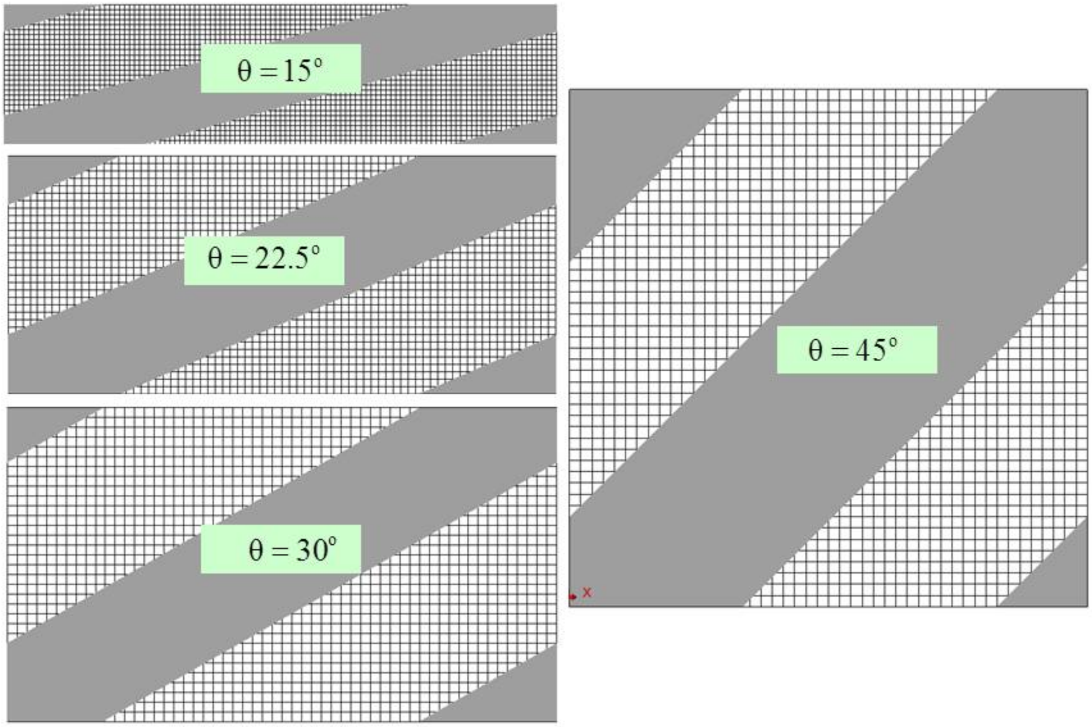


Figure 3.10 Lattice inclined configurations. Walls are inclined at an angle of θ with respect to the lattice cell boundary

A set of lattice offset configuration and lattice inclination configuration is studied, and the Chen et al. (1998) scheme is applied to identify the solution dependence on the lattice configurations. For the baseline study, a reference resolution between 16 and 17 cells is used for the length scale of channel height, the slightly variation of resolution is a results of geometry matching to create the desired lattice configuration.

The results of fluid streamwise velocity are plotted in figure 3.11 and figure 3.12. In all the simulations, a parabolic fluid velocity profile can be obtained, however, a strong lattice configuration dependence was observed: for the lattice-offset configuration with exact lattice-aligned wall boundary, simulation result agrees very well with theoretical value. When the solid boundary is slightly offsetted, numerical solutions start to deviate from the analytical ones. For all the the lattice-offset configurations, the solutions are under-predicted, and the solution is less accurate when fraction of fluid in the near wall cell (pfluid value) is close to $1/2$. For all the lattice-inclined configurations, very similar solutions are obtained and they are under-predicted by about 8% when compared with the theoretical value. It

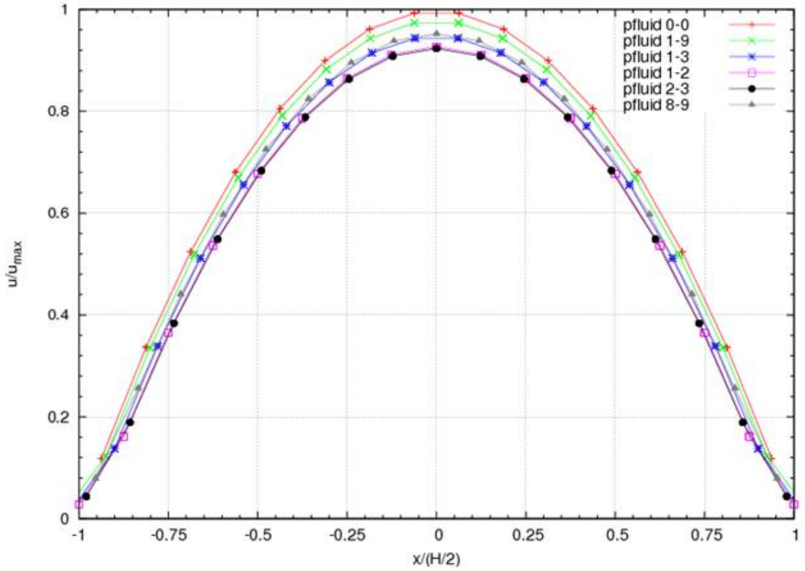


Figure 3.11 Normalized streamwise velocity for typical lattice-offset configurations

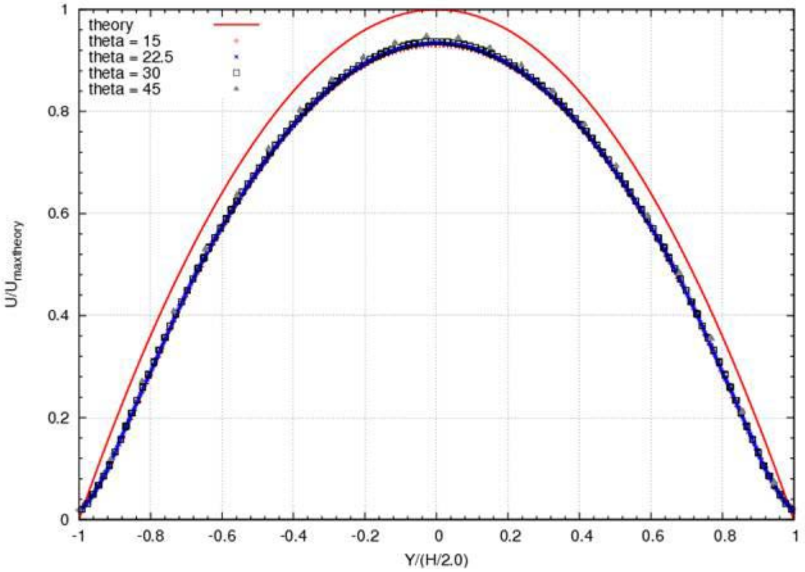


Figure 3.12 Normalized streamwise velocity for typical lattice-inclined configurations

should be noted that in all the non-lattice-aligned configurations, velocity in the first near wall point are significantly under-predicted, which directly shifts the predicted overall velocity profile.

Grid convergence study was also performed, and the convergence rate was measured as standard L_1 norm:

$$L_1 = \frac{\sum_n |Uy_n^{\text{lbm}} - Uy_n^{\text{theory}}|}{N} \tag{3.24}$$

The results are shown in figure 3.13 and figure 3.14 for lattice-offset configurations. Clearly, for exact lattice aligned configuration, a second order accuracy of the solution is obtained, while there is an lattice offset of the wall boundary, the solution reduced to first order accuracy. The solution shows strong pfluid dependence, errors for lattice-offseted cases are at least a factor of 3 larger than that of lattice-aligned case. When the fluid portion of the near wall cell is in the range between 0.5 and 0.9, the error is relative large compared to other lattice offset (pfluid) value.

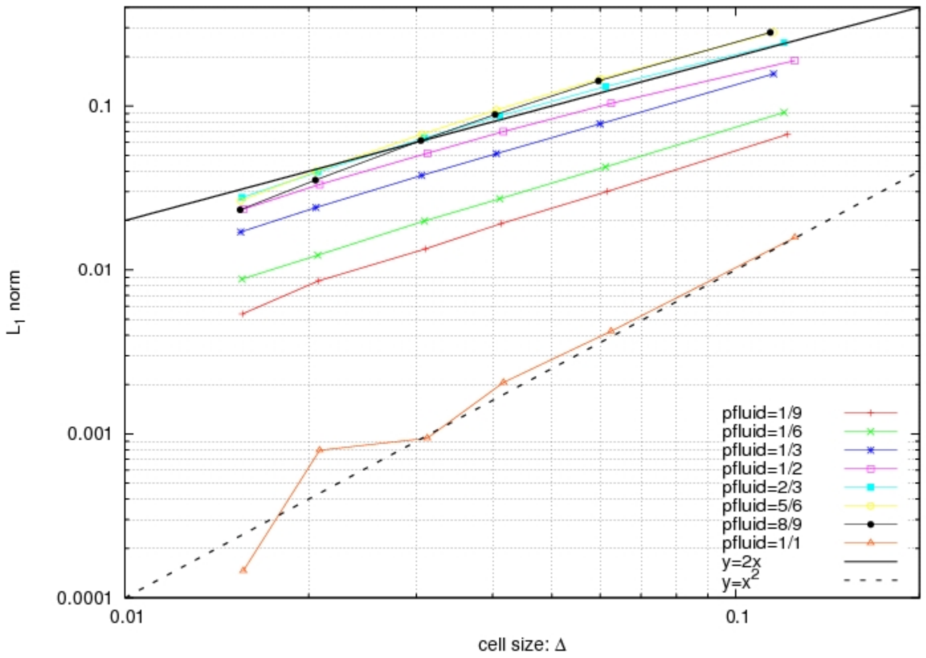


Figure 3.13 Grid convergence study for lattice-offset channel configurations: L_1 vs. resolution

For lattice-inclined configurations, the error are plotted in figure 3.15 and figure 3.16. The achieved order of accuracy is close to 1 for all the studied angles. The error shows small dependence on the inclined angles.

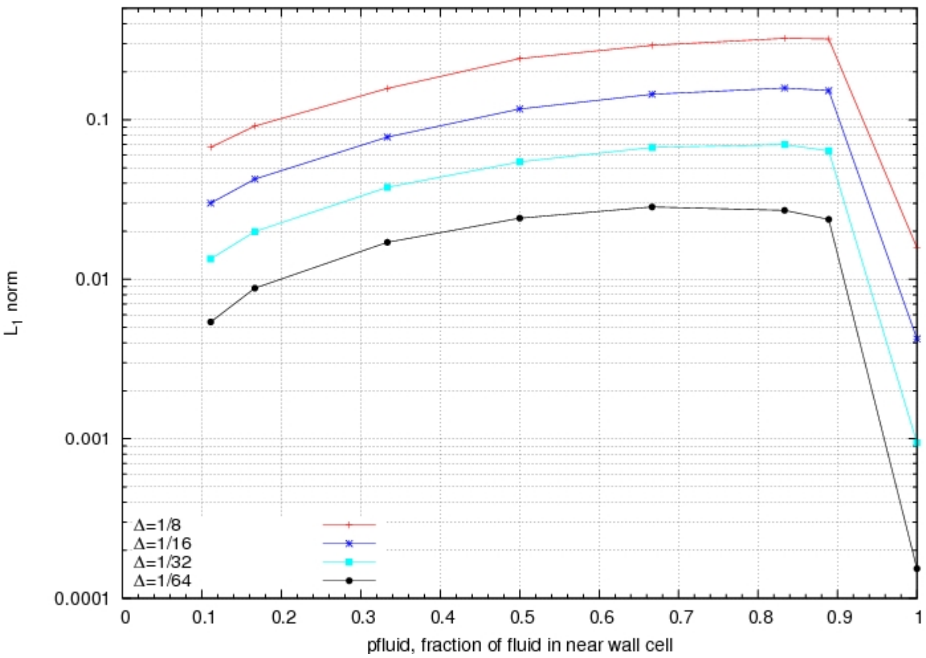


Figure 3.14 Grid convergence study for lattice-offset channel configurations: L_1 vs. offset (pfluid)

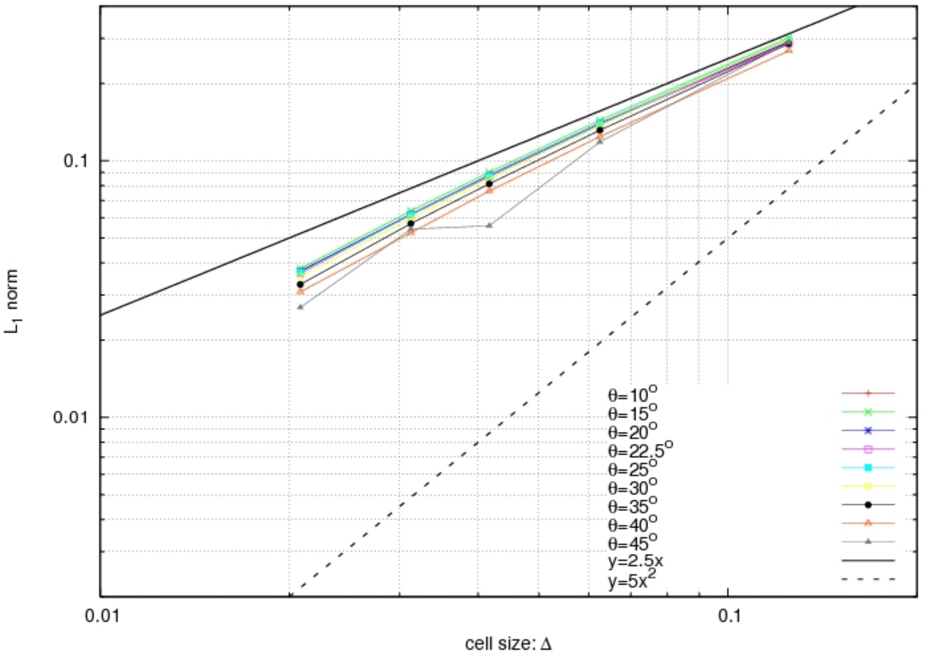


Figure 3.15 Grid convergence study for lattice-inclined channel configurations: L_1 vs. resolution

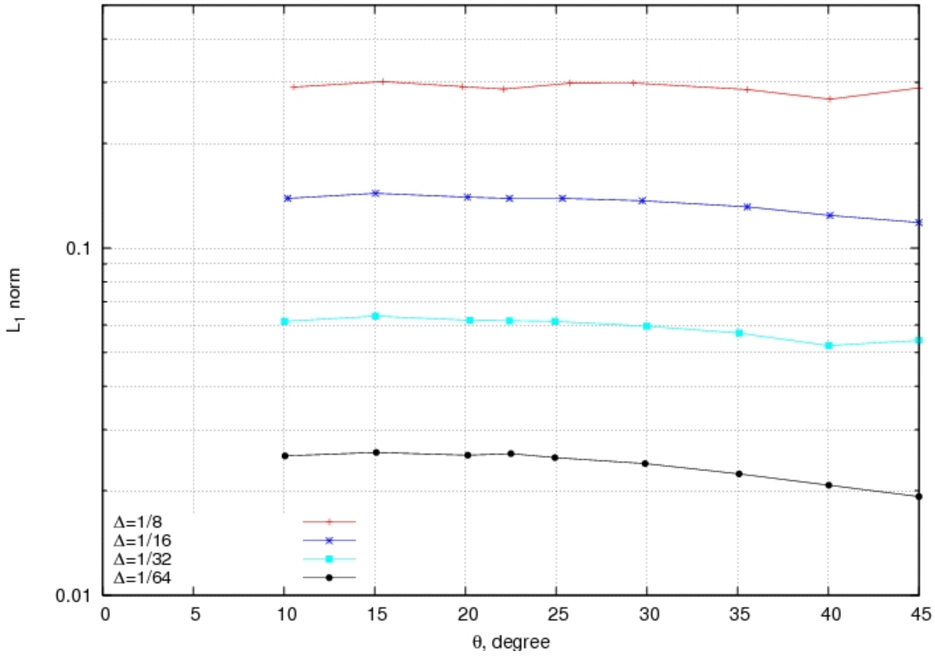


Figure 3.16 Grid convergence study for lattice-inclined channel configurations: L_1 vs. inclined angle

CHAPTER 4. AN IMPROVED VOLUMETRIC BOUNDARY SCHEME

In this chapter, an improved volumetric boundary scheme is proposed, it applies a surfel scattering correction process to improve solution accuracy on non-lattice aligned boundary configurations. The new scheme takes into account of the local flow velocity variations for particle redistributions in its scattered volume. Convergence study is performed to demonstrate the algorithm's improvement on solution accuracy. A number of validation cases are presented to prove its accuracy for flow predictions with curved boundaries.

4.1 An improved LBM volumetric boundary scheme

As discussed in the last chapter, the original volumetric boundary scheme is only first order accurate for non-lattice-aligned boundaries, the solution is also very sensitive to the wall boundary orientation and location, this is mainly due to the piece-wise constant particle distribution assumption used during surreal scattering process. To overcome this deficiency, the particle redistribution during surfel scattering can be modified by introducing a scatter correction term $\Delta\Gamma_i^{out,\alpha}(\boldsymbol{x},t)$ into equation 3.5:

$$Q(\boldsymbol{x},t) = \sum_{\alpha} \frac{V_{i,pgram}^{\alpha}(\boldsymbol{x})}{V_{i,pgram}^{\alpha}} [\Gamma_i^{out,\alpha}(\boldsymbol{x},t) + \Delta\Gamma_i^{out,\alpha}(\boldsymbol{x},t)] \tag{4.1}$$

By properly including the local flow variation into this correction term, it re-balances the scattered particle distributions in the corresponding Pgram overlapped cells and reflects the non-homogeneity of particle distributions in the Pgram volume. On the other hand, introducing such correction should not compromise the exact local conservations in the original form.

Clearly, if we assume a local variation of particle distribution $\Delta f_i^{\alpha}(\boldsymbol{x},t)$ inside the Pgram overlapped volume, the correction term can be directly obtained for each Pgram overlapped cells:

$$\Delta\Gamma_i^{out,\alpha}(\boldsymbol{x},t) = V_{i,pgram}^{\alpha}(\boldsymbol{x})\Delta f_i^{\alpha}(\boldsymbol{x},t) \tag{4.2}$$

To approximate $\Delta f_i^\alpha(\boldsymbol{x},t)$, one can rely on the Pgram equilibrium distribution in equation 3.22 and decompose the velocity field $\boldsymbol{u}_i^\alpha(\boldsymbol{x},t)$ inside Pgram volume into a mean value part $\bar{\boldsymbol{u}}_i^\alpha(t)$ and a spatial variation part $\Delta\boldsymbol{u}_i(\boldsymbol{x},t)$:

$$\boldsymbol{u}_i^\alpha(\boldsymbol{x},t) = \bar{\boldsymbol{u}}_i^\alpha(t) + \Delta\boldsymbol{u}_i(\boldsymbol{x},t) \tag{4.3}$$

By substitution and retaining only the leading order terms, one arrives:

$$\Delta f_i^\alpha(\boldsymbol{x},t) = \rho w_i \frac{\boldsymbol{c}_i \cdot \Delta\boldsymbol{u}_i(\boldsymbol{x},t)}{T} \tag{4.4}$$

For laminar flows the near wall velocity is linearly proportional to its distance from solid wall, $\Delta\boldsymbol{u}_i(\boldsymbol{x},t)$ can be calculated directly from the local velocity gradient $\frac{\partial \boldsymbol{u}}{\partial y}$ and its location as:

$$\Delta\boldsymbol{u}_i(\boldsymbol{x},t) = \frac{\partial \boldsymbol{u}}{\partial y}(y^\alpha(\mathbf{x}) - y_i^\alpha) \tag{4.5}$$

where $y^\alpha(\boldsymbol{x})$ is the distance from Pgram overlapped cell $\boldsymbol{x}$ to facet α and y_i^α is the volume weighted distance:

$$y_i^\alpha = \frac{\sum_x y^\alpha(\boldsymbol{x}) V_{i,pgram}^\alpha(\boldsymbol{x})}{\sum_x V_{i,pgram}^\alpha(\boldsymbol{x})} \tag{4.6}$$

This is the formed adopted by Li et al. (2004), the first-order scattering correction depends explicitly on local velocity gradient and gives accurate prediction of flow past an impulsively started cylinder. However, the near wall velocity gradient term $\frac{\partial \boldsymbol{u}}{\partial y}$ is difficult to evaluate for under resolved situations, especially for turbulent dominated near wall flow field. The issue is subsequently resolved via a convenient approximation of $\Delta f_i^\alpha(\boldsymbol{x},t)$ in the current study:

$$\Delta f_i^\alpha(\boldsymbol{x},t) \approx \rho^\alpha w_i (1 + \frac{\boldsymbol{c}_i \cdot \bar{\boldsymbol{u}}_i^\alpha(t)}{T}) \frac{\boldsymbol{c}_i \cdot (\boldsymbol{u}^\alpha(\boldsymbol{x},t) - \bar{\boldsymbol{u}}_i^\alpha(t))}{T} \tag{4.7}$$

where $\bar{\boldsymbol{u}}_i^\alpha(t)$ is an averaged velocity over Pgram-overlapped cells for surfel α with particle velocity $\boldsymbol{c}_i$:

$$\bar{\boldsymbol{u}}_i^\alpha = \frac{\sum_x \boldsymbol{u}^\alpha(\boldsymbol{x},t) V_{i,pgram}^\alpha(\boldsymbol{x})}{\sum_x V_{i,pgram}^\alpha(\boldsymbol{x})} \tag{4.8}$$

and $\boldsymbol{u}^\alpha(\boldsymbol{x},t)$ is the velocity in cell $\boldsymbol{x}$.

Equations 4.1, 4.2, 4.7 and 4.2 form a complete scattering correction procedure. With such a scattering correction, the outgoing particles are re-distributed based on local shear stress in overlapping cells within each Pgram. And it's straightforward to verify

$$\sum_i \Delta\Gamma_i^{out,\alpha}(t) = 0 \tag{4.9}$$

So adding such term does not alter the local conservation of mass.

4.2 Convergence study with the modified scheme

The proposed volumetric scheme was applied to the lattice gravity channel study with various lattice-wall configurations. With baseline resolution, figure 4.1 and figure 4.2 shows the predicted streamline velocity distribution for both the lattice-offset configurations and lattice-inclined configurations, current solution shows less dependence on the lattice-offset compared with solution from original volumetric scheme. Both set of configurations give consistent results, particularly, the results for lattice-offset configurations are almost overlapped with analytical solution, while the results for lattice-inclined studies shows improvement over previous predictions, but there are still a slight deviation from the theoretical one.

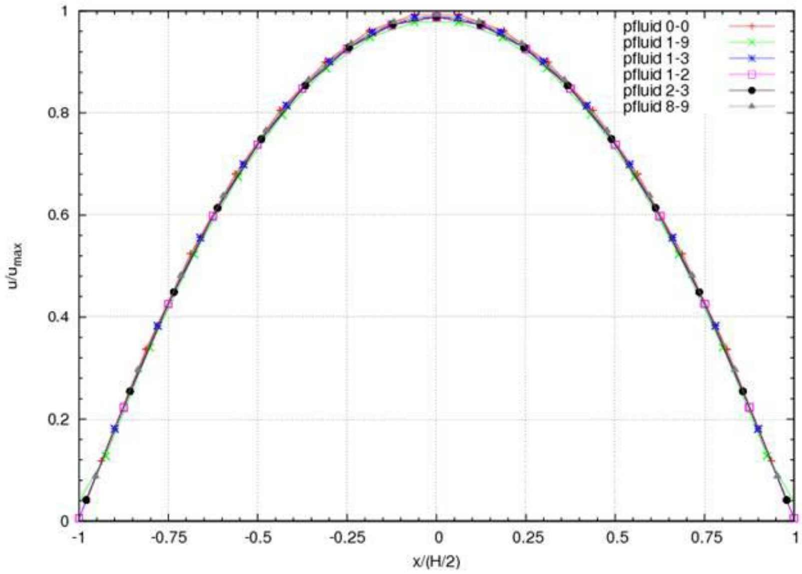


Figure 4.1 Normalized streamwise velocity for typical lattice-offset configurations with current scheme

Grid convergence study was performed for both the lattice-offset configurations (figure 4.3) and lattice-inclined configurations (figure 4.5), compared to the previous result in figure 3.13 and figure 3.15, the convergence rate is close to 2 for most of the lattice configurations, note for lattice offset configuration with very small pfluid value, the solution convergence rate is still close to 1, but the errors have been substantially reduced when compared with previous results.

Results are also directly plotted against the ones obtained from the original volumetric boundary scheme (figure 4.4 and figure 4.6). With the applied modification, the overall error level was reduced

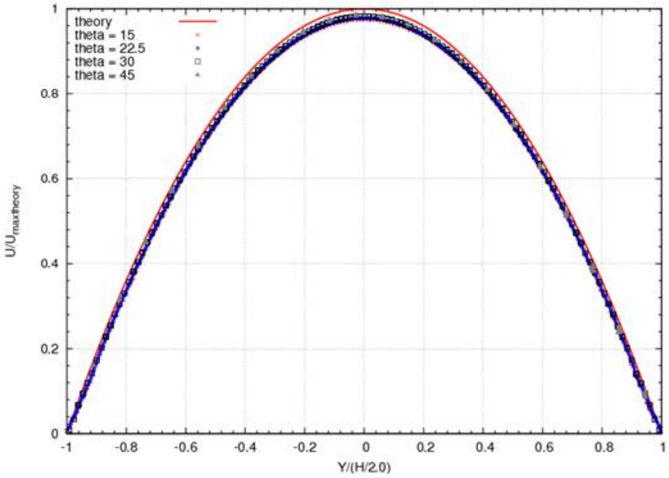


Figure 4.2 Normalized streamwise velocity for typical lattice-inclined configurations with current scheme

by a factor at least 2 for all the configurations.

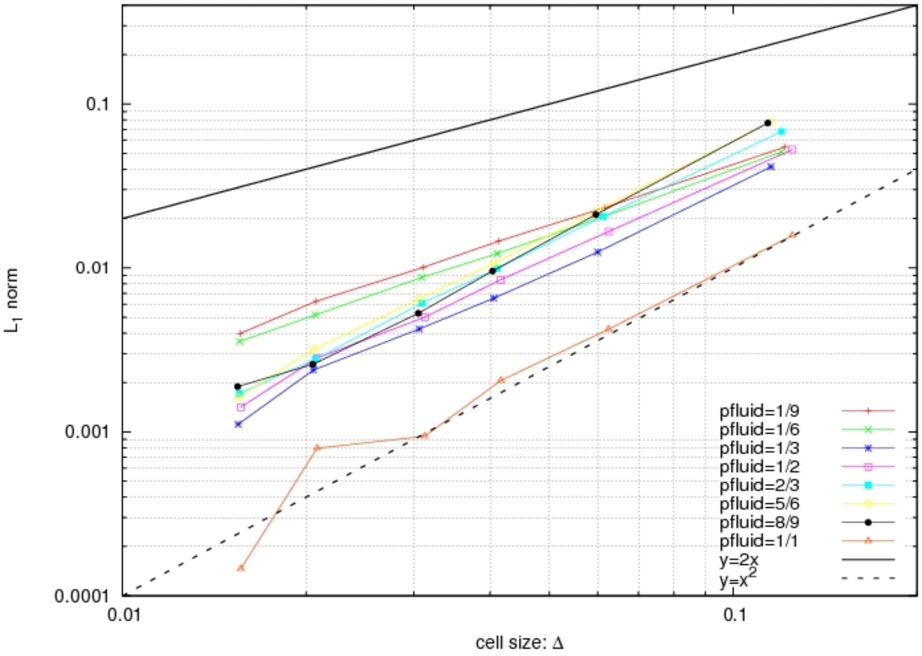


Figure 4.3 Grid convergence study for lattice-offset channel configurations: L_1 vs. resolution

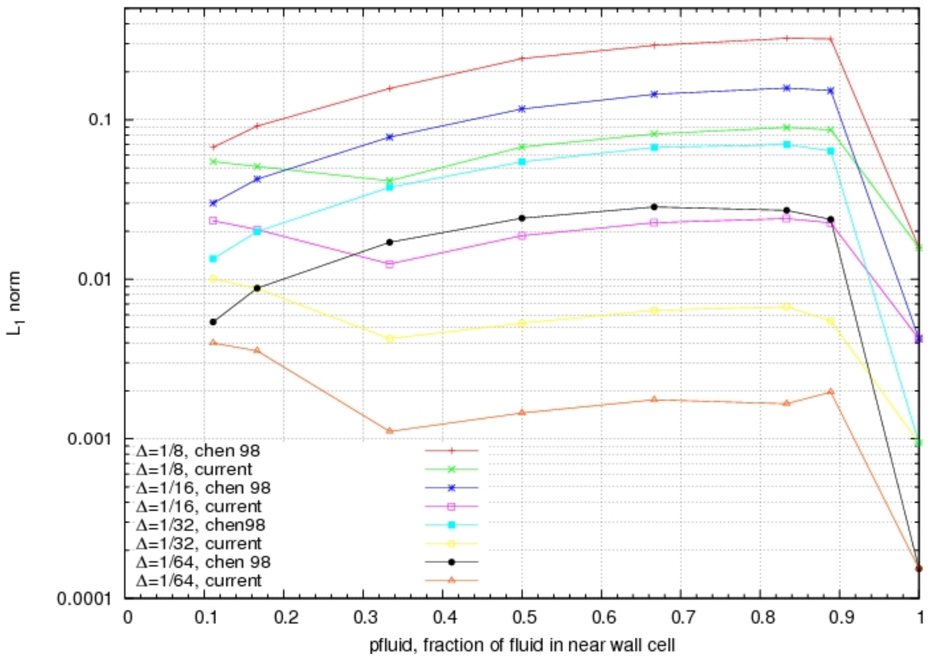


Figure 4.4 Grid convergence study for lattice-offset channel configurations: L_1 vs. offset

4.3 Benchmarked cases with the improved volumetric LBM boundary scheme

In the following, we present the studies on a number of benchmarked cases to demonstrate the accuracy of current scheme, particularly for flows involving curved boundaries.

4.3.1 Laminar flow past a circular cylinder with impulsive start

The first test case considered here is a time-dependent flow around an impulsively started circular cylinder. This is regarded as one of the classical prototypes of unsteady hydrodynamics, since the early stage of the flow development reveals a range of rather complex and subtle phenomena, such as boundary layer development, separation, vortex formation and secondary flows. In numerical simulations, solving such a flow is rather challenging, difficulties come from the prediction of separations on smooth curved geometry and steep changes of flow properties (such as the velocity and vorticity) on the surface that associated with the impulsive start. The nonlinear nature of this problem also poses many difficulties in terms of stability and accuracy for numerical methods.

The current study focuses on direct numerical simulations (DNS) at a moderate Reynolds number $Re = 550$, with $Re = U_o D/\nu$, where U_o is the free stream velocity, D is the cylinder diameter and

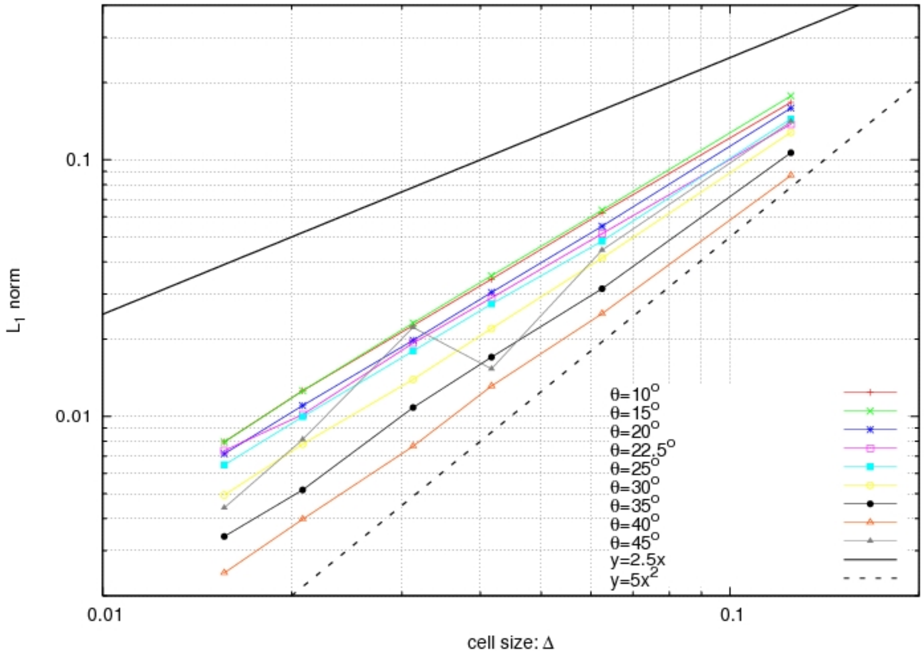


Figure 4.5 Grid convergence study for lattice-inclined channel configurations: L_1 vs. resolution

ν the kinetic viscosity. In numerical setup, the upstream boundary is 20 diameters away, and the down stream boundary is 40 diameters away from the center of the cylinder. The two vertical(side) boundaries are 17 diameters away in each direction (figure 4.7). The resolution across the cylinder is 160 finest cells, the total number of simulation cells is close to one million. simulation is run for a relatively short amount of time, $T \leq 7$, where T is the time non-dimensionlized by D/U_o, when the flow symmetry does not broken.

A uniform velocity profile is imposed on the inlet boundary and a constant static pressure boundary condition is imposed on the far field outlet, in the perpendicular y direction, a periodic boundary condition is used for simplicity. In the specification of initial condition, an irrotational potential flow solution is imposed at $T = 0^+$.

Simulations predicted a vorticity field and streamline field that are in good agreement with other numerical results and experiment observations (figure 4.9 and figure 4.8). It can be clearly observed that the secondary vortices appear within $T = 1$, it becomes bigger and stronger at $T = 3$ and $T = 5$. Small scale flow structures also continue to grow in time, but remain confined by the primary vortices up to $T = 5$.

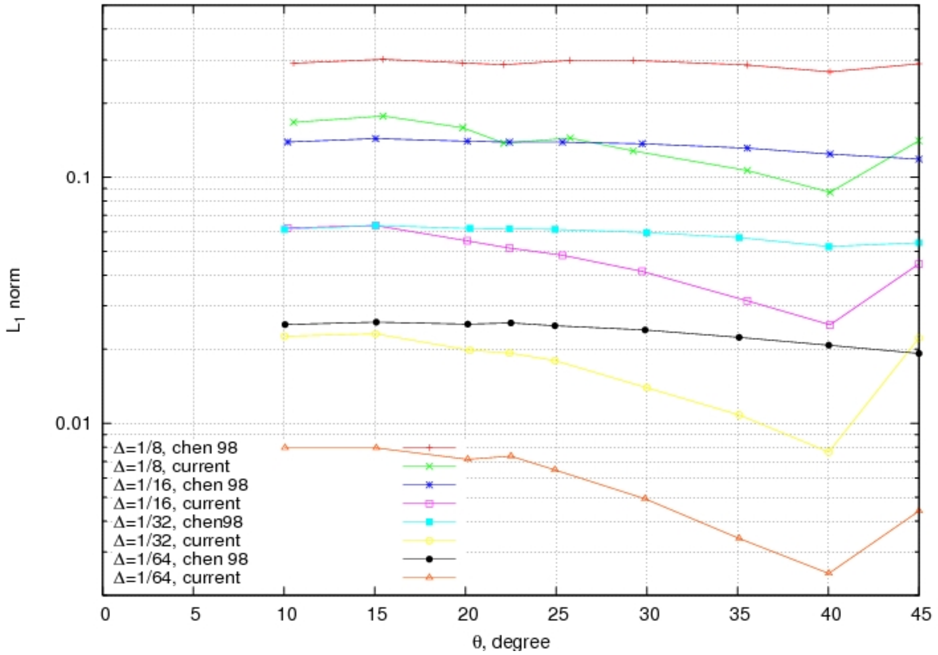


Figure 4.6 Grid convergence study for lattice-inclined channel configurations: L_1 vs. inclined angle

Figures 4.10 and 4.11 show the velocity evolution along the symmetry axes behind the cylinder. It is seen that the results match the experimental measurement well, especially in the enclosed wake region where streamwise velocity is in the negative x-direction. Results from Loc (1980) under-estimated the flow velocity in the wake region.

Figure 4.12 shows the time evolution of the separation/reattchement points on the cylinder surface. Flow separation/reattchement on the cylinder surface indicates a zero wall shear stress, which are tightly related to the generation and development of the primary and secondary vortices due to the impulsively start of the cylinder. The result compares reasonable well with the one obtained by Koumoutsakos and Leonard (1995), with a slightly under-estimation of the predicted separation/reattchement angles. For the onset of the primary vortex, current study predicts a rapid development of the first separation point (which occurs at $T = 0.44$), while vortex method (Koumoutsakos and Leonard (1995)) gives a rather slow change of the first separation point (occurs at $T = 0.20$).

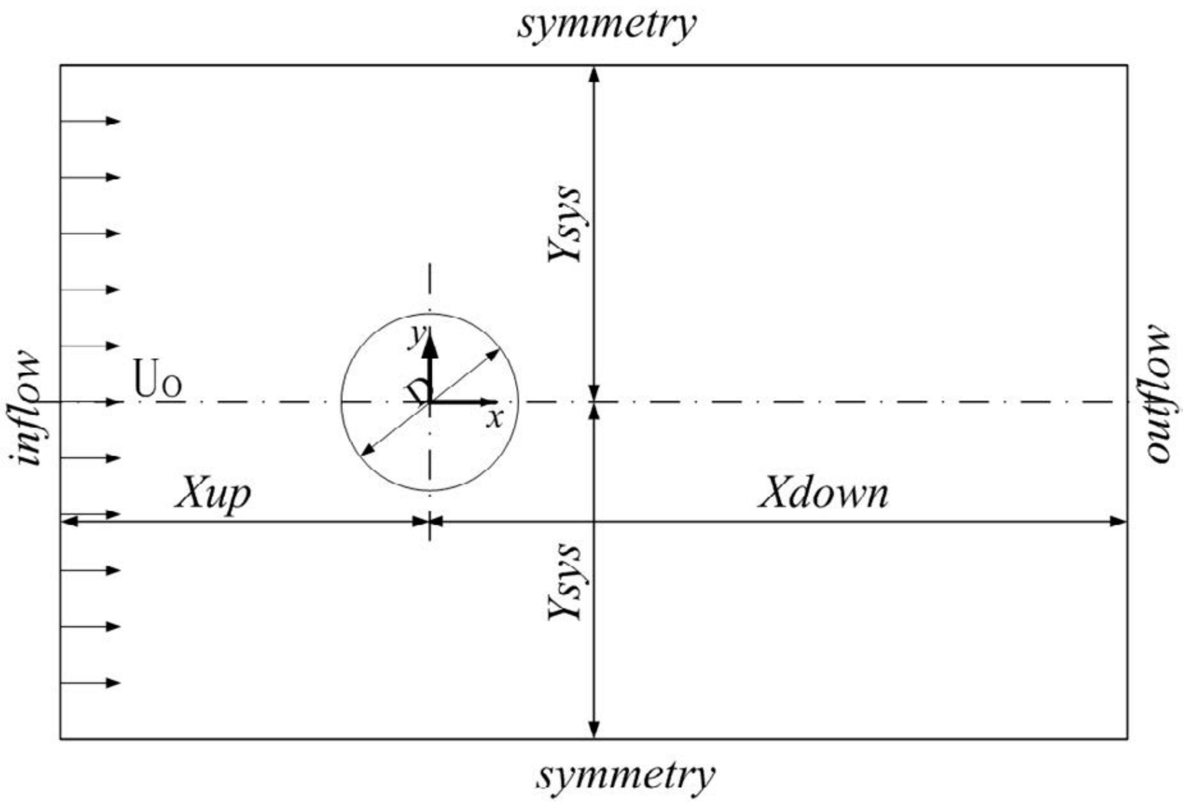


Figure 4.7 Sketch of the simulation setup for an impulsively started cylinder with Reynolds number $Re = 550$

4.3.2 Laminar flow past a circular cylinder with vortex shedding

The second problem studied here is a viscous flow past a circular cylinder at low Reynolds numbers ($50 \leq Re \leq 300$, with $Re = U_o D/\nu$, where U_o is the free stream velocity, D is the cylinder diameter and ν the fluid kinetic viscosity). In this Reynolds number range, the flow is unable to maintain steady state and vortex shedding happens when the flow loses symmetry. The vortex shedding phenomenon is regarded as one of the most challenging flow problems for unsteady flow prediction with numerical methods. A number of experimental investigations (Williamson (1989); Williamson and Roshko (1990)) as well as numerical studies have been conducted (He and Doolen (1997); Henderson (1995); Liu et al. (1998); Surmas et al. (2004)) for this type flows. The ability to capture major flow characteristics such as vortex shedding frequency, drag components, and pressure distributions, indicates the level of accuracy for the applied numerical methods.

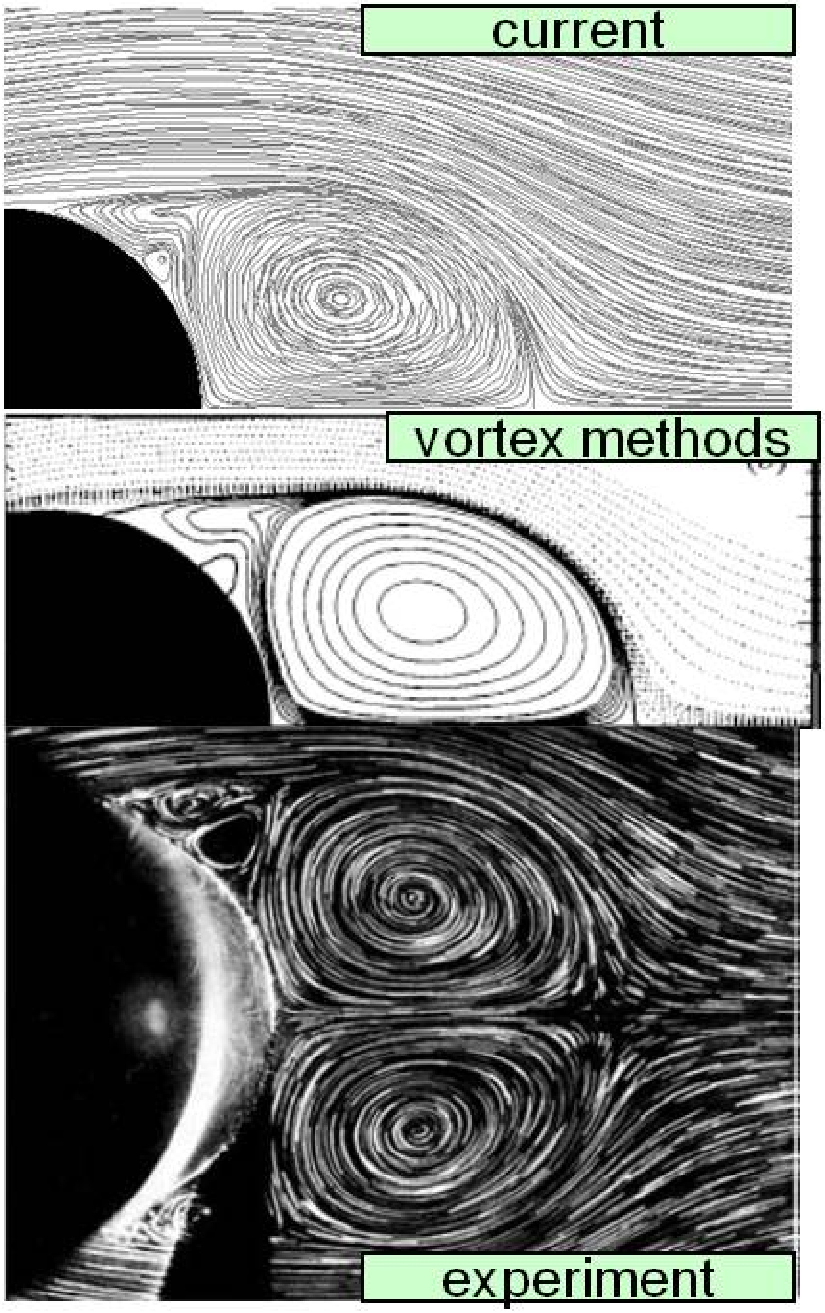


Figure 4.8 Streamline contours at $Re = 550$ and $T = 5$ for flow around an impulsively started cylinder. Top: current study; middle: vortex methods (Koumoutsakos & Leonard 1995); bottom: experiment (Bouard & Coutanceau 1980)

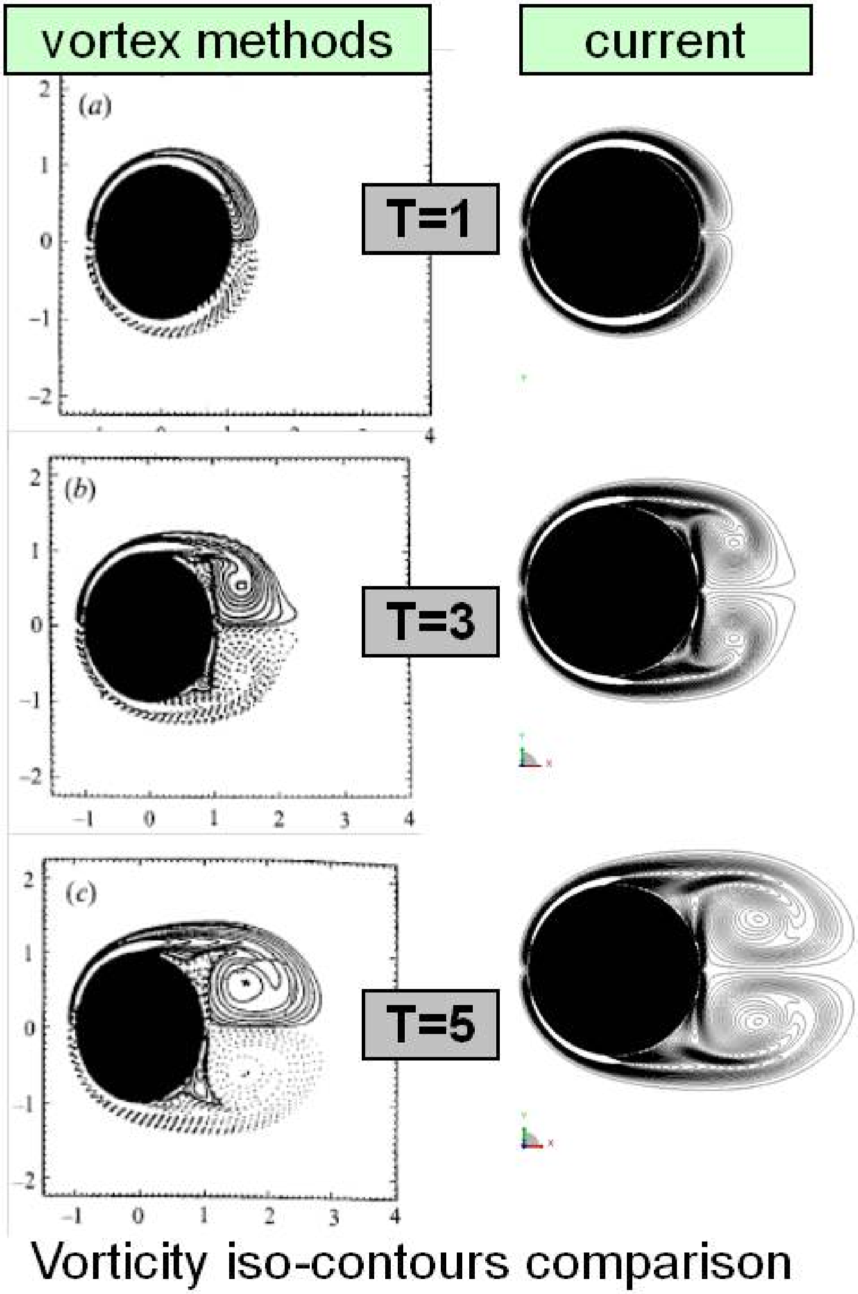


Figure 4.9 Vorticity iso-contours at $Re = 550$ for different time intervals T=$1, 3, 5$. Left: vortex methods (Koumoutsakos and Leonard (1995)); right: current study

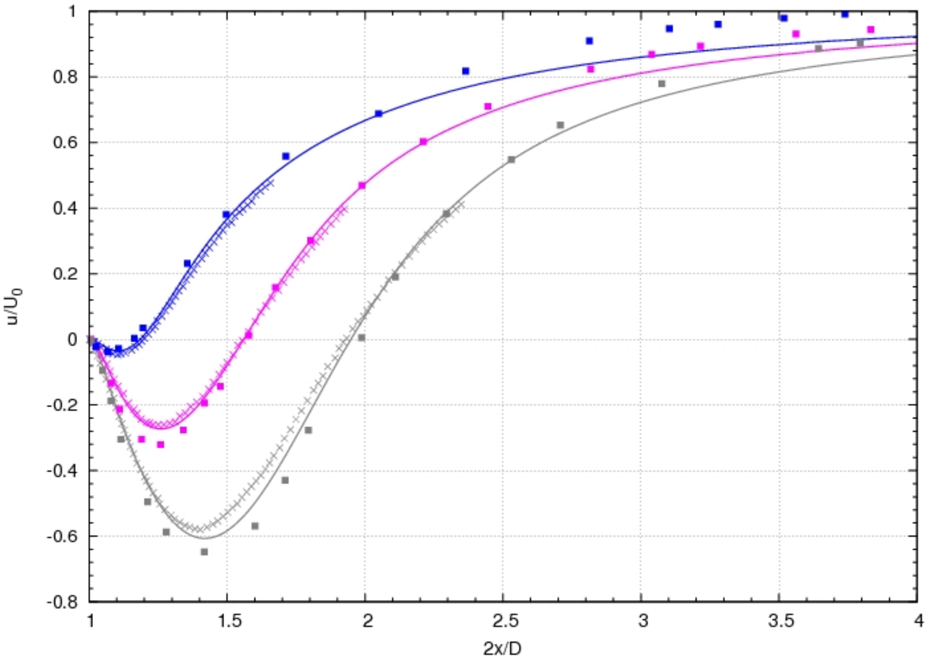


Figure 4.10 Comparison of streamvise velocity distributions on the flow axis behind cylinder at time intervals $T = 1$(red colors),$T = 2$ (pink colors) and $T = 3$ (grey colors). Lines: current study; filled squares: experiment (Bouard and Coutanceau (1980)); cross symbols: numerical method (Loc (1980))

In simulation, a large fluid domain was applied, with both the inlet and outlet boundary are placed $50D$ away from the cylinder (figure 4.13). On the inlet boundary, a uniform free stream velocity is imposed. On the far-field outlet, a static pressure boundary condition is specified. The lateral boundaries are also placed at $50D$ away from the cylinder center, where flow periodicity is assumed.

The initial flow field is specified by an irrotational flow field with small perturbations to quickly trigger the flow unsteadiness. To calculate the flow shedding frequency, flow probes are placed inside flow wake region to record the time histories of unsteady velocity/pressure variations. The simulations are run for a sufficiently long time to ensure that the flow field is fully developed, a sample frequency of $1/3$ Hz is used for the unsteady signals. The unsteady flow results are long time averaged to calculate the drag coefficient.

Convergence study has first been performed for the $Re = 100$ case. From the grid convergence study (table 4.1) which is based on a fixed simulation Mach number of 0.1, it can be seen that with increased resolution, results for major flow characteristics, such as Strouhal number ($St = fD/U_o$,

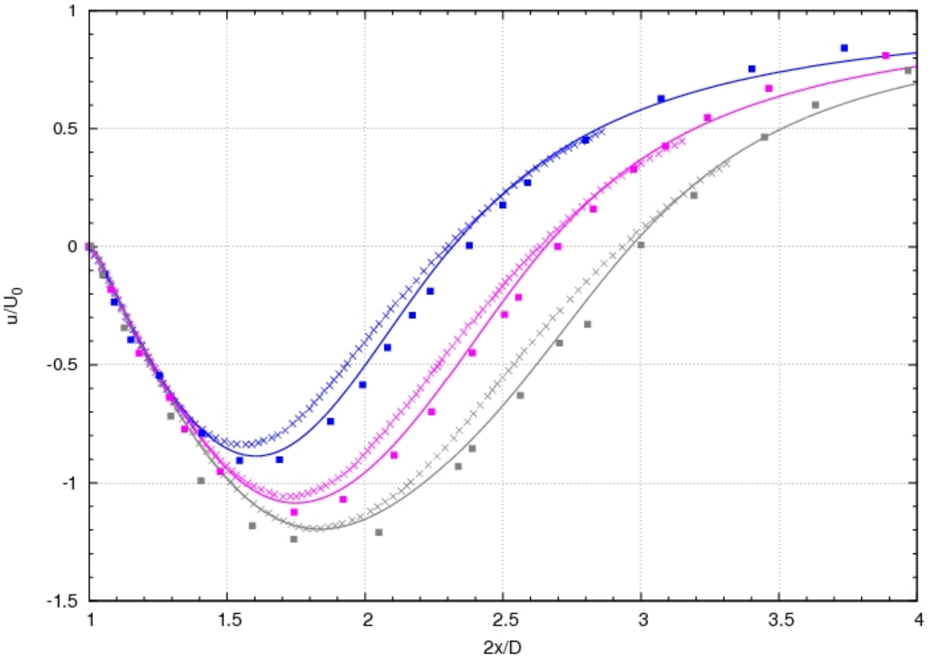


Figure 4.11 Comparison of streamvise velocity distributions on the flow axis behind cylinder at time intervals $T = 4$(red colors),$T = 5$ (pink colors) and $T = 6$ (grey colors). Lines: current study; filled squares: experiment (Bouard and Coutanceau (1980)); cross symbols: numberical method (Loc (1980))

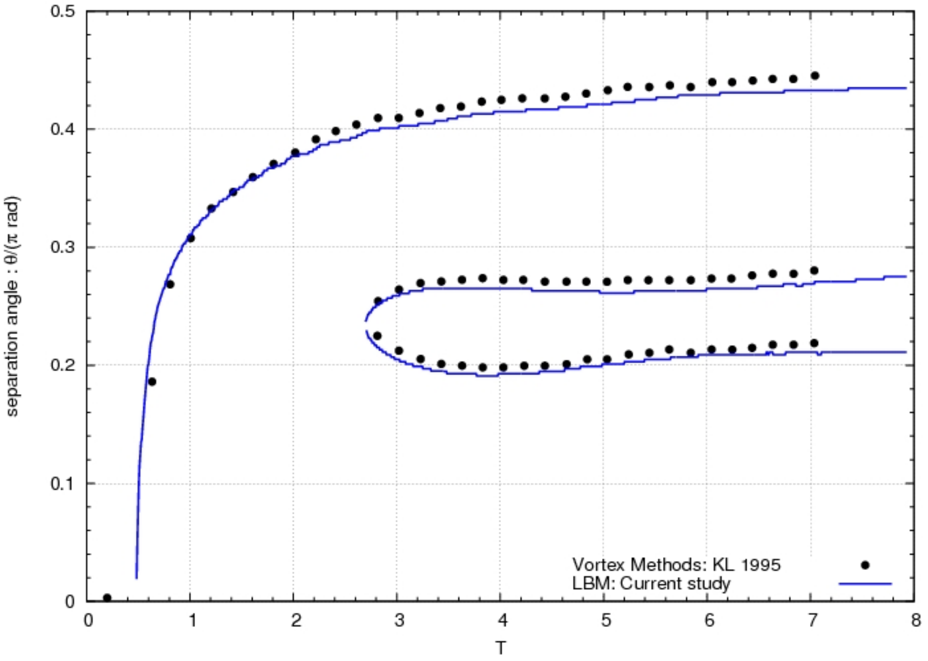


Figure 4.12 Time evolution of the separation/reattachment point on cylinder surface at $Re = 550$. Black dots: vortex method (Koumoutsakos & Leonard 1995), line: current study.

$\delta h/D$	C_D	C_{Dp}	C_{Df}	C_{pb}	St
1/90	1.345	0.999	0.346	-0.702	0.164
1/120	1.339	0.997	0.342	-0.701	0.164
1/160	1.336	0.995	0.341	-0.701	0.164

Table 4.1 Grid convergence study for laminar flow past circular cylinder at $Re = 100$

Mach	C_D	C_{Dp}	C_{Df}	C_{pb}	St
0.05	1.336	0.995	0.342	-0.696	0.164
0.10	1.336	0.995	0.341	-0.701	0.164
0.20	1.351	1.012	0.339	-0.717	0.164

Table 4.2 Mach convergence study for laminar flow past circular cylinder at $Re = 100$

where f is the vortex shedding frequency measured inside the wake region), drag coefficients (total drag-C_D, pressure drag-C_{Dp} and friction drag-C_{Df}), as well as base pressure coefficient (C_{pb}, the cylinder surface pressure coefficient measured 180^o degree from the front stagnation point), only show slight variations. This indicates that a grid independent solution can be obtained with a resolution of $\delta h/D = 1/160$ or higher. From the Mach convergence study (table 4.2), which is based on a fixed grid resolution of $\delta h/D = 1/160$, it can be seen that when the simulation Mach number is small (≤ 0.1), flow is nearly incompressible and the simulation results are almost unchanged. In the current study, a simulation Mach number of 0.1 and a grid resolution of $\delta h/D = 1/160$ are applied for simulation of all the studied Reynolds numbers.

The Strouhal number St and base pressure coefficient C_{pb} are the most import flow parameters for vortex shedding studies since they are very sensitive to errors induced in the numerical methods (He and Doolen (1997); Henderson (1995); Liu et al. (1998); Surmas et al. (2004); Braza et al. (1986); Calhoun (2002)). The computed results are shown in Fig.4.14 and Fig.4.15 and compared with experimental measurements (Williamson (1989); Williamson and Roshko (1990)) and other numerical predictions (He and Doolen (1997); Henderson (1995); Liu et al. (1998); Surmas et al. (2004); Braza et al. (1986); Calhoun (2002)). It can be clearly seen that the current approach gives accurate prediction when compared with experimental results. This indicates that both the flow separation and the recirculation region are well predicted. The LBM scheme with point-wise implementation of particle bounce back on the solid surface (Surmas et al. (2004)) over-predicts the flow shedding frequency, while the curvilinear LBM scheme (He and Doolen (1997)) gives too much dissipation and tends to under-predict the shedding frequency. The spectral element method (Henderson (1995)) slightly over-predicted the Strouhal number and base pressure coefficient, indicating the predicted flow separation is slightly delayed and the resulting wake size is relatively small when compared with experimental results. For $Re \geq 180$, all the 2D CFD methods can not predict the abrupt change of base pressure coefficient observed in experiment, which is mainly due to the fact that in reality three dimensional flow transitions happens in this Reynolds number range (Williamson (1989); Williamson and Roshko (1990)). Such phenomenon can only be captured through 3D simulations.

The predicted drag coefficients are plotted in Fig. 4.16 and Fig. 4.17. It can be seen that the current prediction compares well with results from spectral element method (Henderson (1995)), especially for the prediction of friction drag coefficient. The pressure drag coefficient (as well as the total drag coefficient) predicted by Henderson (1995) is slightly larger than the current predictions. This is because the spectral element method gives a slightly later prediction of separation location and a smaller wake size, both lead to a higher pressure drag since the pressure on the wake side will be reduced. The other two LBM methods either over-predicted the total drag coefficient (Surmas et al. (2004)) or give under-predicted values (He and Doolen (1997)), indicating they are less accurate than the current LBM algorithm.

On table 4.3, the predicted drag coefficient and Strouhal number at $Re = 100$ and $Re = 200$ is also compared with results from other numerical methods which solve the 2D incompressible Navier-Stokes equations with various algorithms (Henderson (1995); Liu et al. (1998); Braza et al. (1986); Calhoun (2002)). The predictions given by the current LBM approach can achieve the same accuracy level as other methods or even slightly better (e.g., for Strouhal number prediction at $Re = 100$).

Finally, the instantaneous vorticity field for two Reynolds numbers ($Re = 100$ and $Re = 200$) are presented in Fig. 4.18 and Fig. 4.19 as an illustration of predicted flow unsteadiness: Flow separation on the cylinder surface and the evolution of alternating vortices behind cylinder can be clearly observed for both cases. With a higher Reynolds number ($Re = 200$), flow tends to separate relatively late on the cylinder surface and forms relatively stronger shear layer vortices with higher evolution frequency. More complex secondary flow structures can be also observed in the near wake region. This is consistent

Table 4.3 Comparison of Drag Coefficient and Strouhal number for laminar flow past circular cylinder at low Reynolds number

Reference	$Re = 100$		$Re = 200$	
	Cd	St	Cd	St
Williamson (1989)[a]	–	0.164	–	–
Braza et al. (1986)[b]	1.364	0.160	1.400	0.200
Liu et al. (1998)[c]	1.350	0.164	1.310	0.192
Calhoun (2002)[d]	1.330	0.175	1.172	0.202
Henderson (1995)[e]	1.350	0.167	1.341	0.197
Current Method	1.336	0.164	1.342	0.196

[a] Experiment
[b] Finite volume method, 2nd order accurate
[c] Finite volume method with multigrid, 2nd order accurate
[d] Cartesian grid method
[e] Spectral element method

with other numerical predictions (He and Doolen (1997); Henderson (1995); Liu et al. (1998); Surmas et al. (2004); Calhoun (2002)).

4.3.3 Turbulent flow past a two dimensional airfoil geometry with leading edge ice shape

In the third test, the usefulness of the current approach is further examined by applying it to to simulate flows that involves relative complex geometries - a two dimensional turbulent flow over a business-jet airfoil (GLC 305) with rime ice (212 ice shape) accrued on its leading edge (Addy et al. (2003)). The iced airfoil profile and computational grid used in simulation is shown in figure 4.20. Since the shape of rime ice is quite complex, it is difficult to generate high quality mesh with body-fitted grid (Chi et al. (2005)).

On the flow condition, the freestream Mach number and static pressure are 0.12 and 20.5 psi, respectively, and the Reynolds number based on the freestream condition and the airfoil chord length is $Re = 3.5 \times 10^6$. In the numerical setup, the inflow, outflow and two side boundaries are located 15 chord lengths away from the airfoil. At the inflow boundary, the freestream velocity is specified. At the outflow boundary, a constant static pressure is imposed. At the other two boundaries above and below the airfoil, periodic condition is assumed. The computation grid used a resolution of 1024 cells per

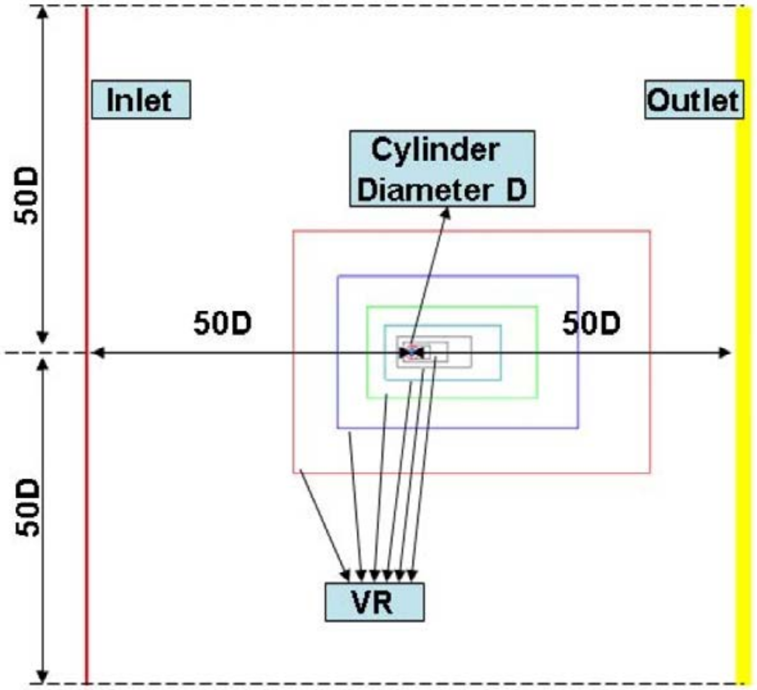


Figure 4.13 Simulation setup for flow past a two dimensional circular cylinder

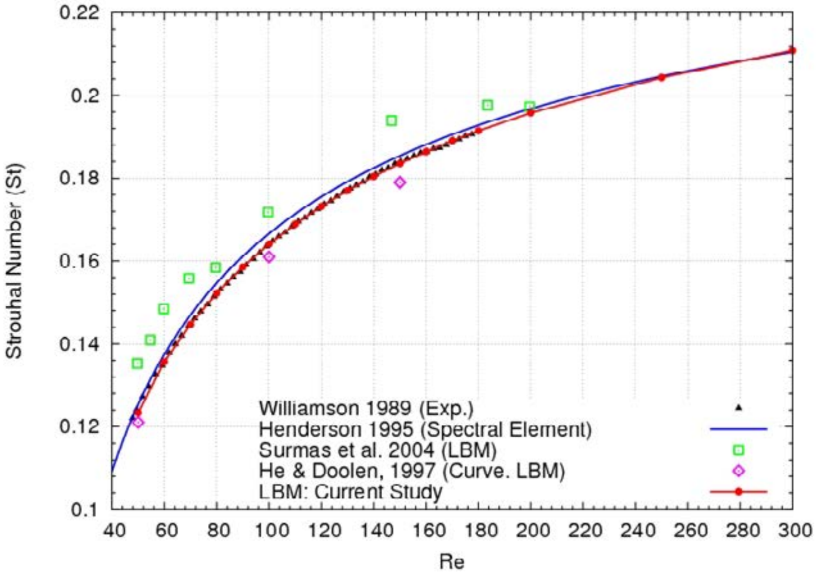


Figure 4.14 Strouhal number verus Reynolds number for laminar flow past circular cylinder at low Reynolds number

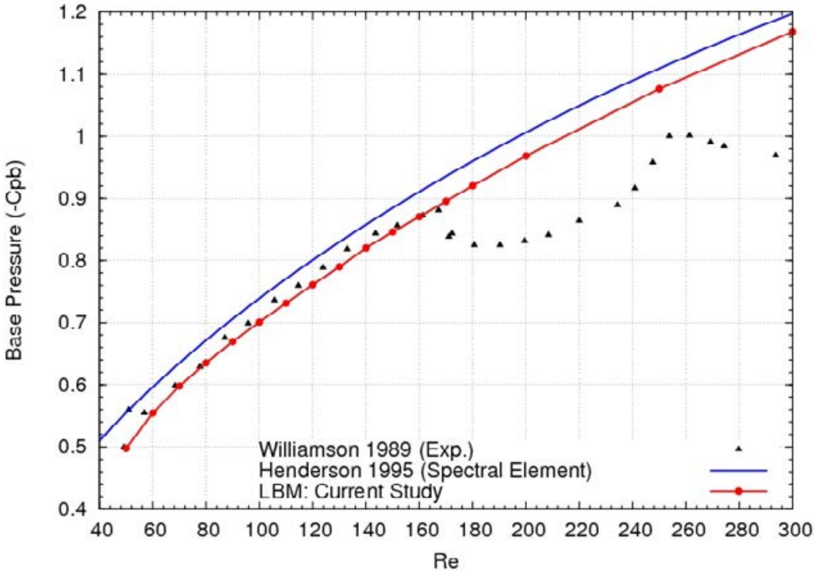


Figure 4.15 Base pressure coefficient verus Reynolds number for laminar flow past circular cylinder at low Reynolds number

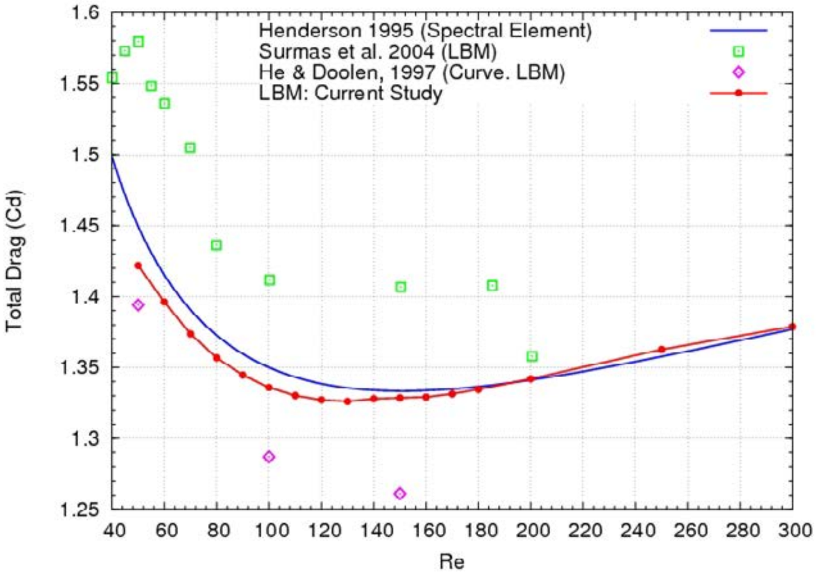


Figure 4.16 Total drag coefficient verus Reynolds number for laminar flow past circular cylinder at low Reynolds number

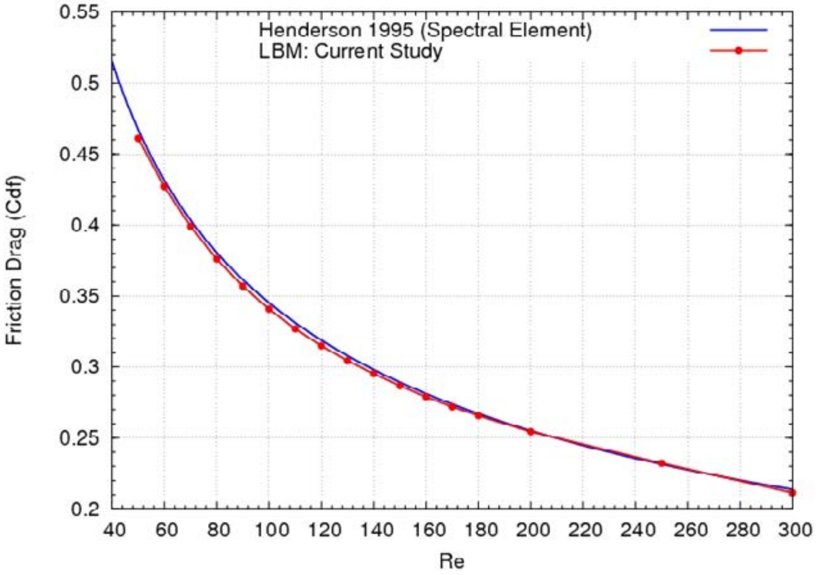


Figure 4.17 Friction drag coefficient and pressure drag coefficient verus Reynolds number for laminar flow past circular cylinder at low Reynolds number

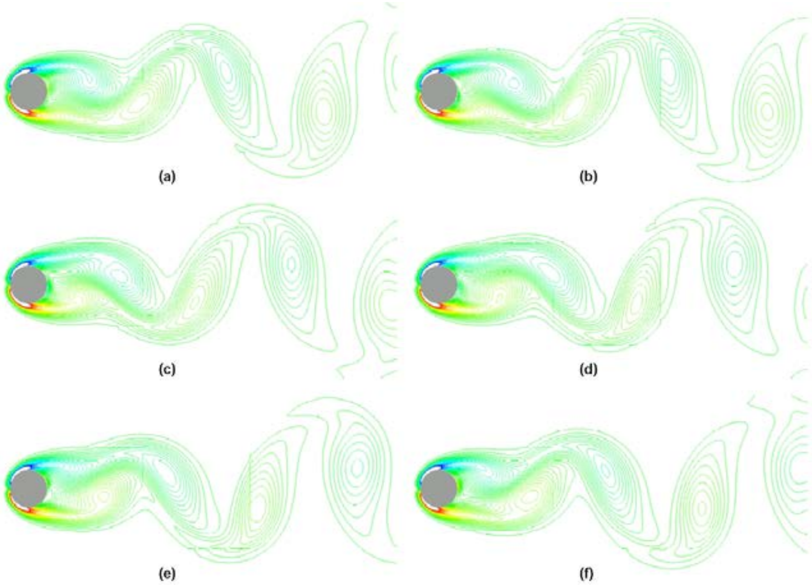


Figure 4.18 Instataneous vorticity fields for flow past a circular cylinder at $Re = 100$

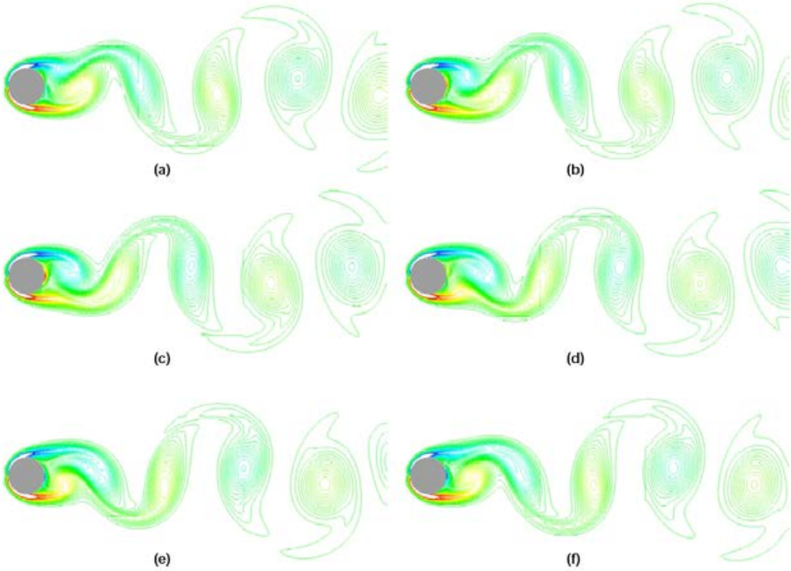


Figure 4.19   Instataneous vorticity fields for flow past a circular cylinder at $Re = 200$

chord length for the near wall flow region, this produced a y^+ value for the first cell between 30 and 150 for the attached flows. The total number of cells is about 130,000 for the entire flow domain.

Figure 4.21 shows that the predicted airfoil lift/drag coefficients compare well with experiment measurements by Addy et al. (2003). For the near stall situations, the airfoil lift coefficient is better predicted by the current approach than by the method presented by Chi et al. (2005), which used a one equation Spalart-Allmaras (S-A) (Spalart and Allmaras (1994)) turbulence model on a body fitted grid with much higher near wall resolution. However, the airfoil drag coefficient is slightly over-predicted at lower angles of attack and under-predicted at higher angles of attack.

Figures 4.22 and 4.23 show that the surface pressure coefficient distributions at several angle of attacks (AOAs), it can be seen that when angle of attack is low and the flow is fully attached to the airfoil surface, simulation can predict the surface pressure distributions very accurately. At higher angle of attack where flow is separated from the lead edge near suction surface, the current simulation over predicts the pressure coefficient (Cp) distributions. This could be due to the fact that in reality the separated flows are fully three dimensional, while two dimensional simulations tends to over-predict the size of the re-circulation zone since the physics of vortex stretching can not be captured. Nevertheless, the overall predictions of pressure distribution are quite encouraging thereby validating current method.

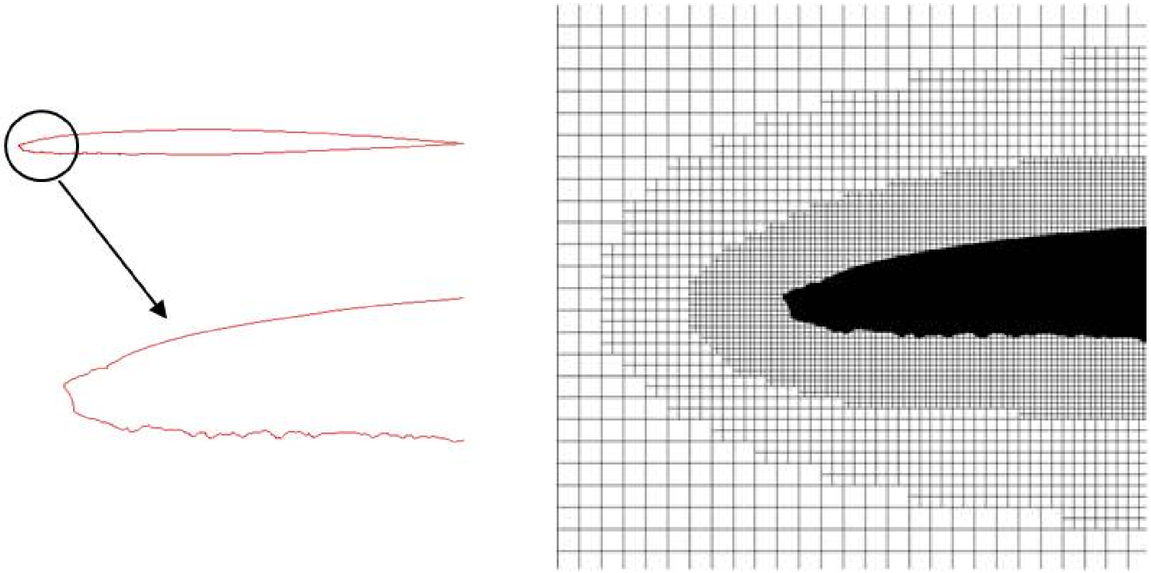


Figure 4.20 The studied airfoil profile (left) and computational grid used (right)

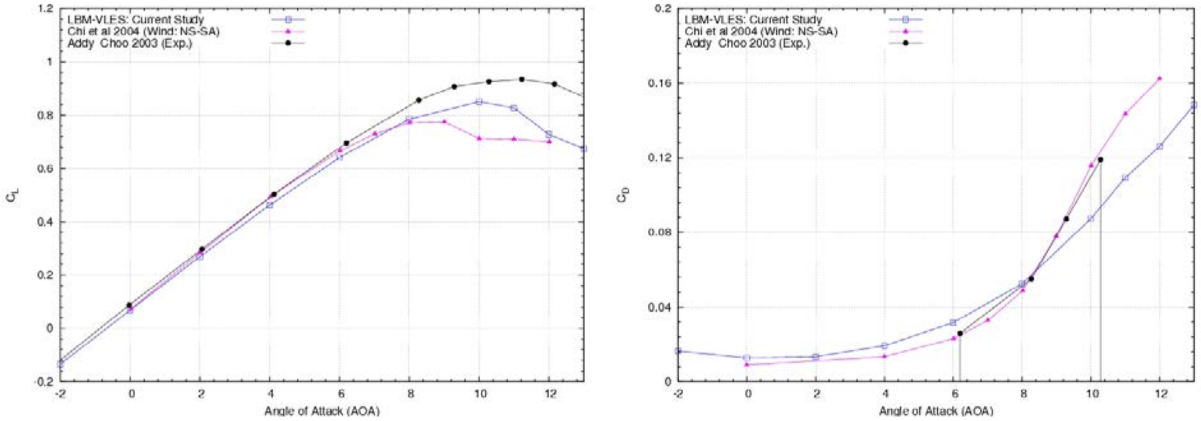


Figure 4.21 Lift (left) and drag (right) coefficient verus angle of attack (AOA) for the studied iced-air-foil

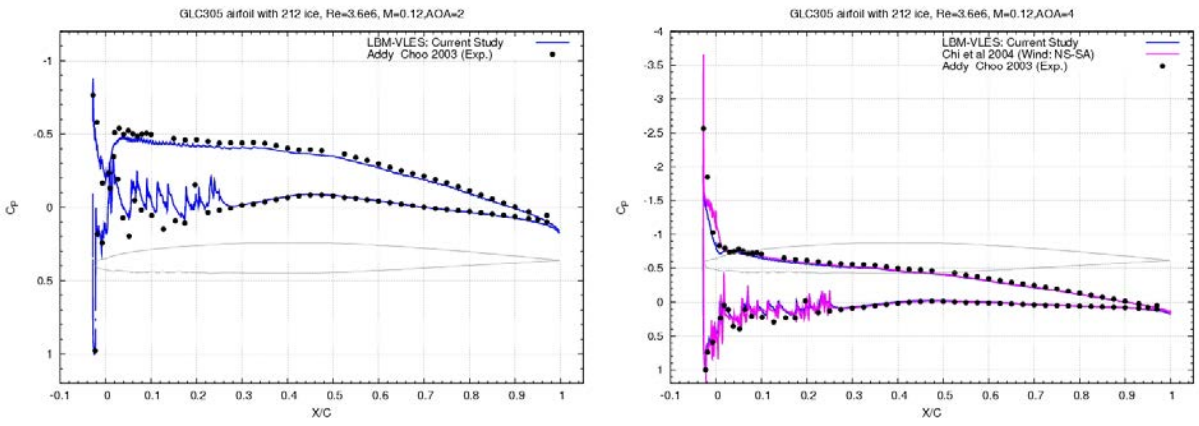


Figure 4.22 Surface pressure coefficient (Cp) distributions at AOA = 2 (left) and AOA = 4 (right)

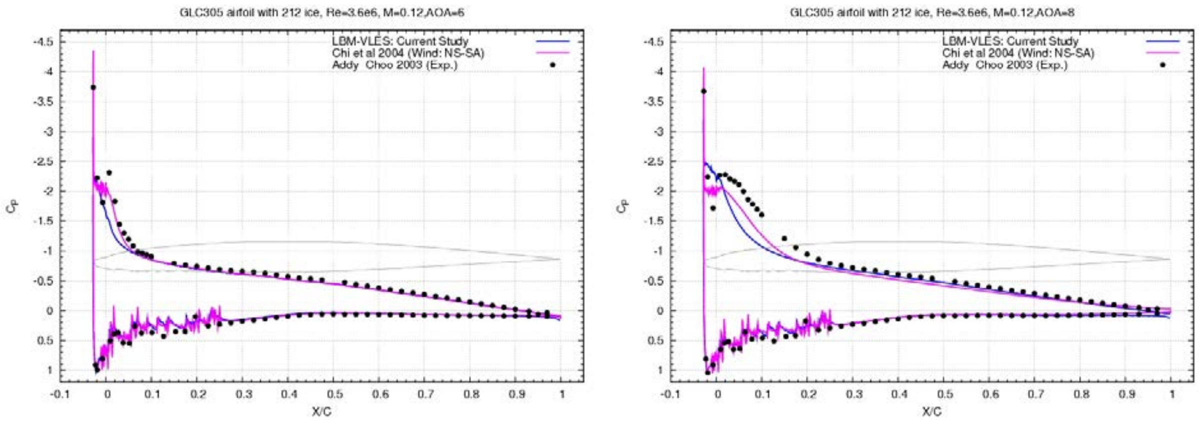


Figure 4.23 Surface pressure coefficient (Cp) distributions at AOA = 6 (left) and AOA = 8 (right)

4.3.4 Turbulent flow past a three dimensional trapezoidal wing configuration

The next problem studied here involves three dimensional complex geometry configurations: the trapezoidal wing (trapwing) with full span flap (FSF) configurations (figure 4.24), this generic high lift configuration has been widely used in aerospace industry to improve the aerodynamic performance of aircraft. It consists of four components: slat, main, flap and base pod elements (Johnson et al. (2000); McGinley et al. (2005)). Both the main element and flap elements are swept and tapered. The simulated model has a mean aerodynamic chord of 1.006 m and a span of 2.162 m. More detailed model dimensions are provided in figure 4.25. As seen from the table, the flap deflection was 25 degrees and

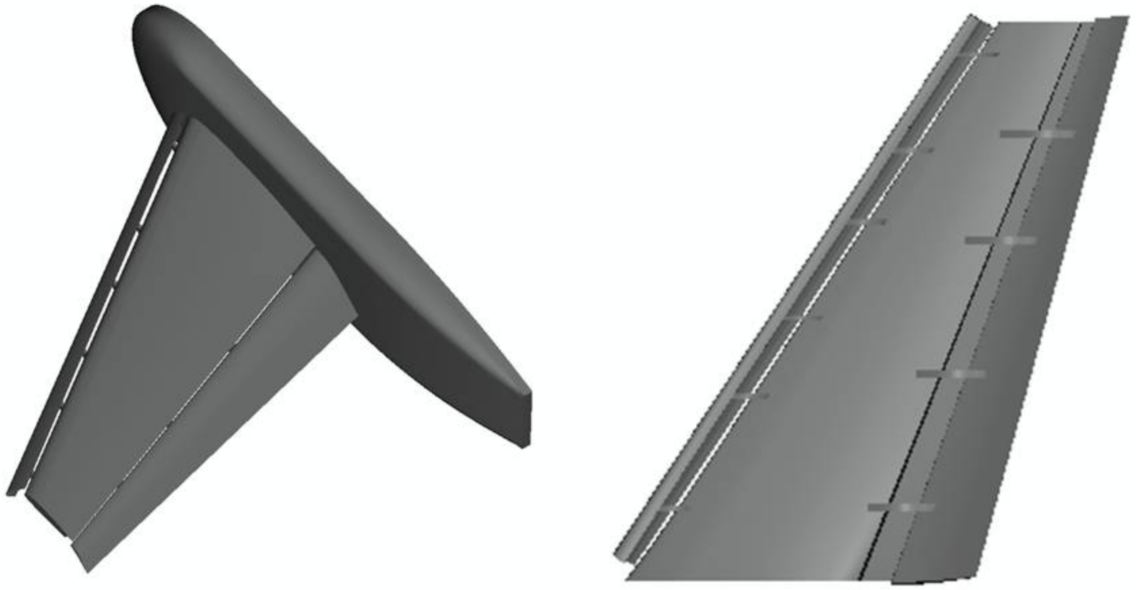
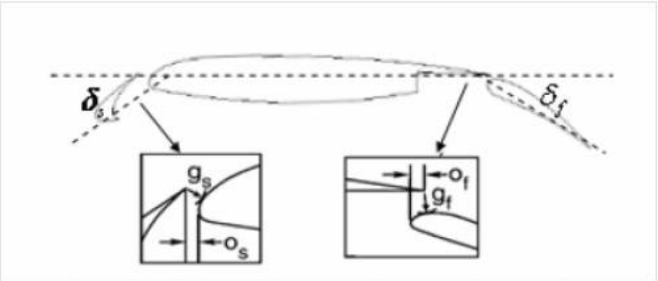


Figure 4.24 Schematic of the studied trapwing FSF model



Case	Δ_s	Δ_f	g_s	o_s	g_f	o_f
FSF	30°	25°	0.015	0.015	0.015	0.005

Figure 4.25 Geometric settings of the studied trapwing FSF model (from Khorrami et al. (2002))

slat deflection was 30 degrees for the model studied here. In current configuration, wing brackets that are used for structure support are included in the simulation, where available CFD studies tend to ignore this geometry to simplifying the meshing process (Khorrami et al. (2002)).

Computations were conducted for a wide range of angles of attack based on the available experimental database. All simulations were conducted under free-air conditions and results were compared to the free air corrected experimental data. The freestream velocity was 60 m/s with an Mach number of 0.2, the Reynolds number is 4.3 million based on the mean aerodynamic chord and freestream

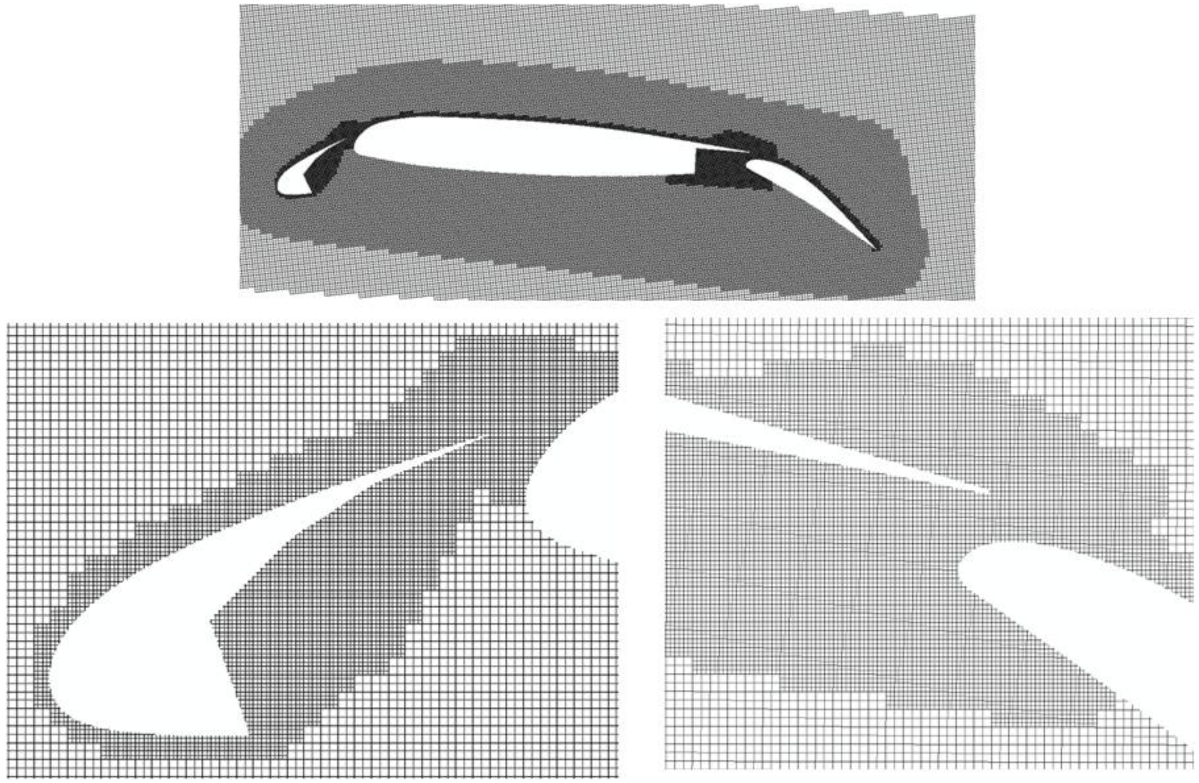


Figure 4.26 Computational grid used for the studied trapwing FSF model (cross-sectional view at 50% span location)

velocity, this exactly matches the experimental conditions conducted in the NASA Langley's 14 by 22 ft wind tunnel (Johnson et al. (2000)). The LBM-VLES approach described in Chapter 2 is used for simulating such an high Reynolds number flow. Figure 4.26 shows the computational grid used in the current study, a resolution of 1.25 mm per cell-size is used for the flow region around the wing elements surfaces, away for the wing elements, grid resolution was subsequently coarsed to reduced overall size of the computation. There are $140M$ fluid cells and $8.7M$ surface elements used in one simulation. All the simulations were run for around $100,000$ timesteps (corresponding to a physical time of 0.25 second) to achieve convergence of integrated forces. It took a wall clock time of 49.2 hours to finish one simulation on a 128 processor AMD Dual-Core Opteron280 cluster.

Figure 4.27 and 4.28 show the variation of the simulated lift coefficient versus the angle of attack (AOA) and the drag polar curve as compared with experimental datas from Johnson et al. (2000). It can be seen that drag and lift are in general well predicted, though the under-predictions increase at higher

AOA. Particularly, Maximum lift (CLmax) is under-predicted by about 8% compared to experiments. Also, note that CLmax occurs at $AOA = 30$ in simulations compared to the experimental occurrence at $AOA = 33$. To further assess the capability of LBM-VLES simulations, detailed surface pressure predictions are also presented in figure 4.29 and 4.30, results are plotted against experimental data for two different angle of attack: AOA $=$ 6 and AOA $=$ 20, at different spanwise locations for each element. Results for the lower angle of attack (figure 4.29) indicate excellent agreement of computed and experimental data, particularly in the wing tip region. At higher angle of attack (figure 4.30), results again show good comparisons with experimental data, though the wing tip region is under-predicted. Such under-predictions at higher AOA is directly related to the under-prediction of CLmax in figure 4.27 and may also be due to under-resolved wing tip vortex. The effects of flow transitioning from laminar to turbulence that happens in reality could also impact the prediction of wing tip vortex, since in simulation flow is assumed to be fully turbulent. Nevertheless, the overall predictions of pressure distribution are quite encouraging thereby validating our simulation and build confidence in the application of current approach for flow predictions that involves complex geometries.

Figure 4.31 shows instantaneous snapshots of predicted three dimensional flow vortical structures (isosurfaces of λ_2 colored by vorticity magnitude, where λ_2 is a vortex identication criteria proposed by Jeong and Hussain (1995)). Several flow features can be highlighted: a large wing tip vortex is formed from the trailing edge of slat element outboard region, it is lifted off from the main element top surface and extends downstream of the flow field. There are also substantial small structures of flow unsteadiness in the slat cove region as well as in the gaps of slat-main element and main-flap element. The effect of bracket geometries is also visible with intensified local flow unsteadiness nearby. The ability to capture these unsteady flow phenomena for such complex geometry configuration could provide insights into the underlying flow physics that related to the airframe noise generation.

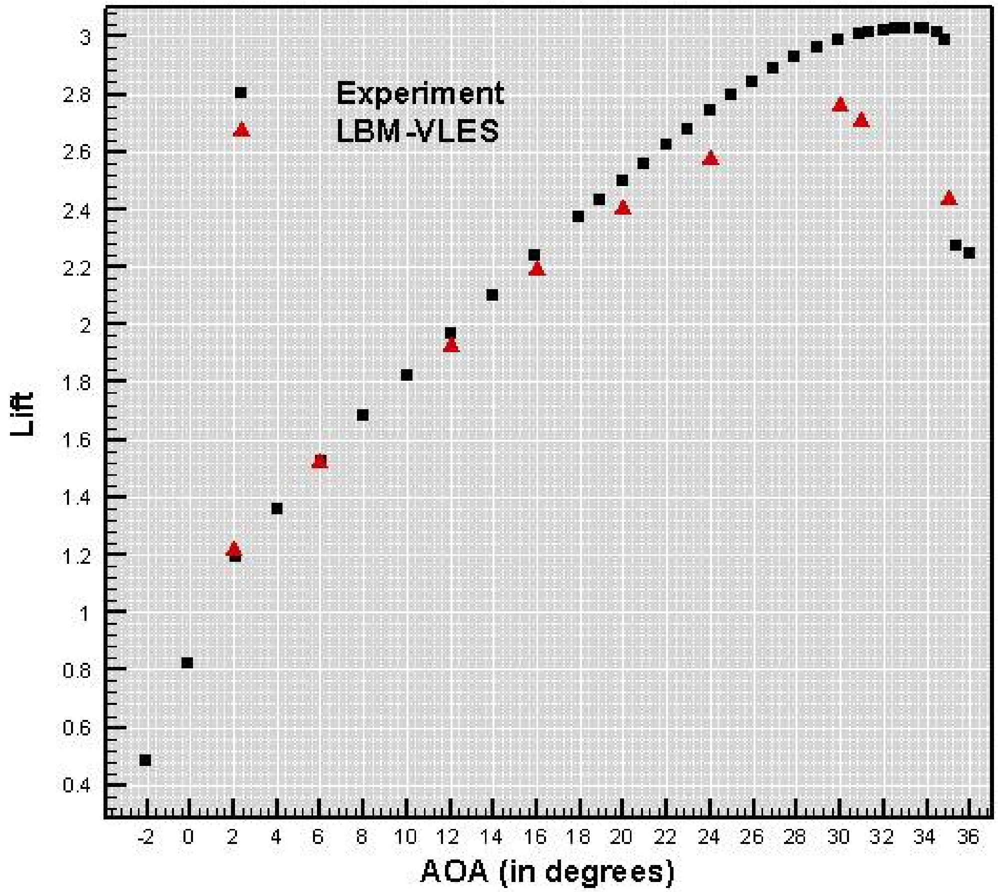


Figure 4.27 Lift versus angle of attack (AOA) for the studied trapwing FSF model

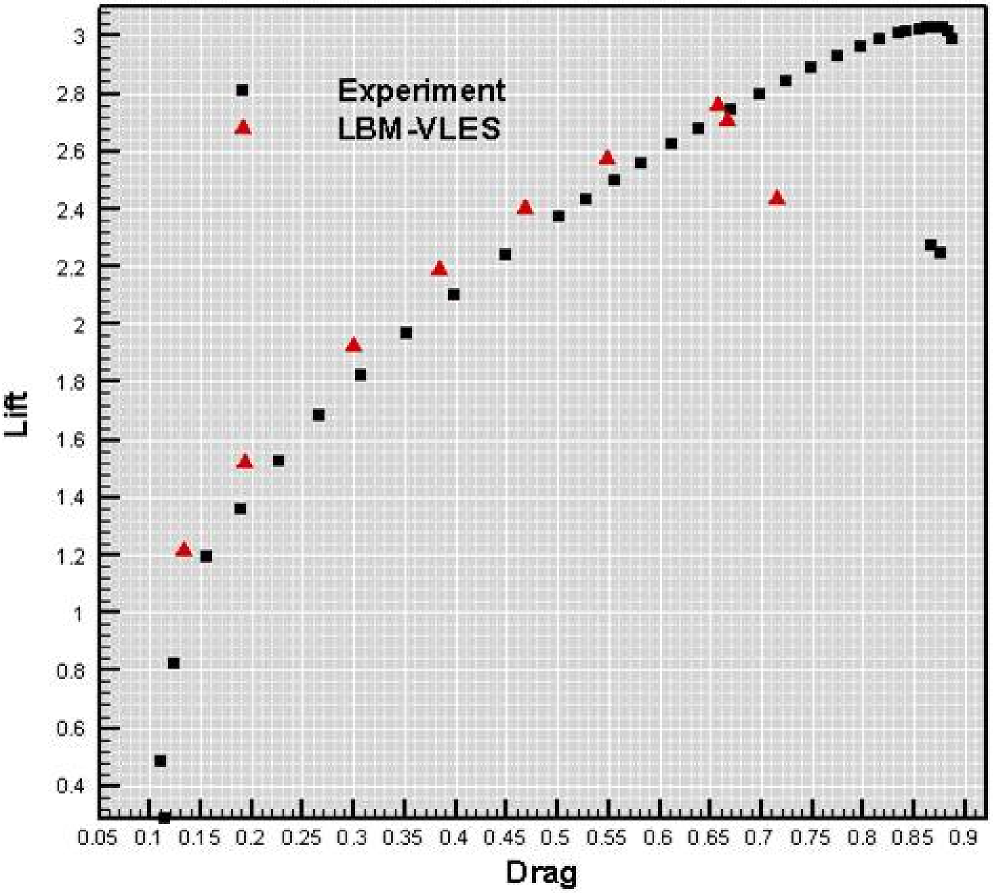


Figure 4.28 Drag versus angle of attack (AOA) for the studied trapwing FSF model

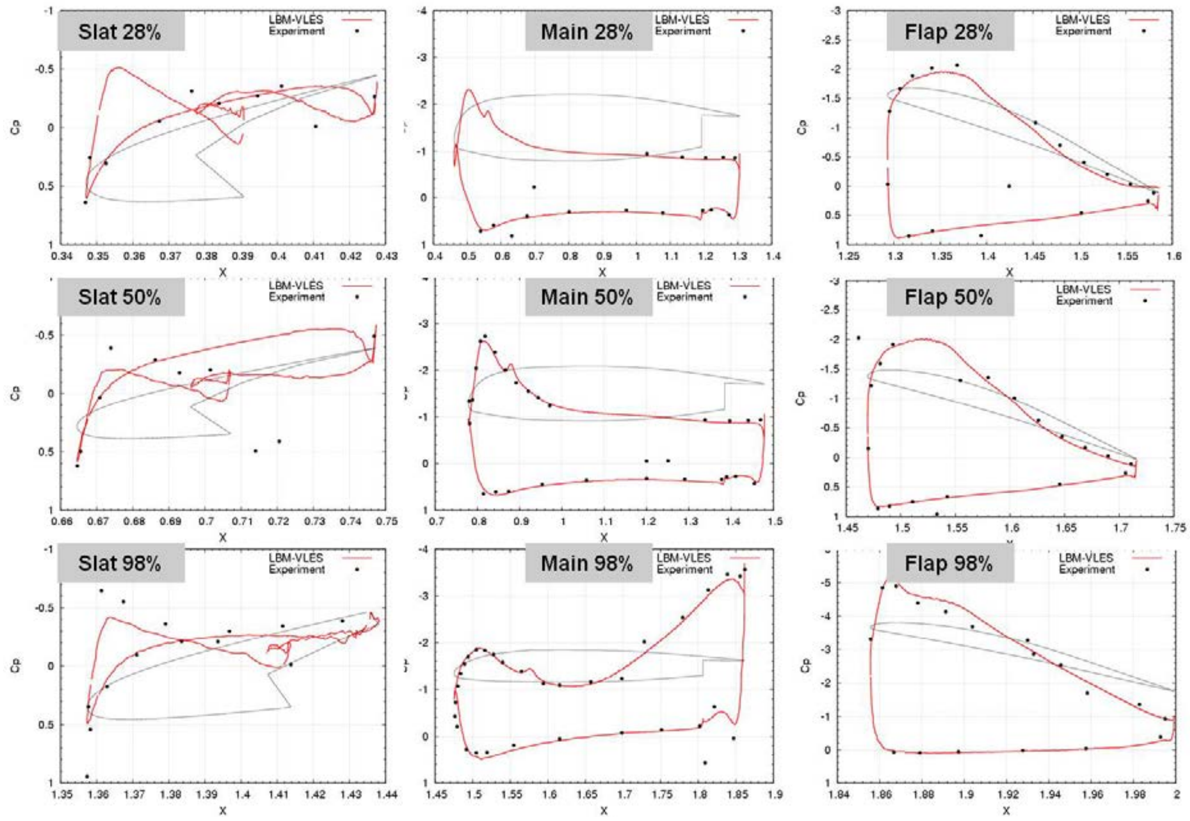


Figure 4.29 Surface pressure coefficient (Cp) distributions at AOA= 6 for the studied trapwing FSF model. Columnwise, left: slat element; middle: main element; right: flap element. Rowwise, top: 28% span location; middle: 50% span location; bottom: 98% span location

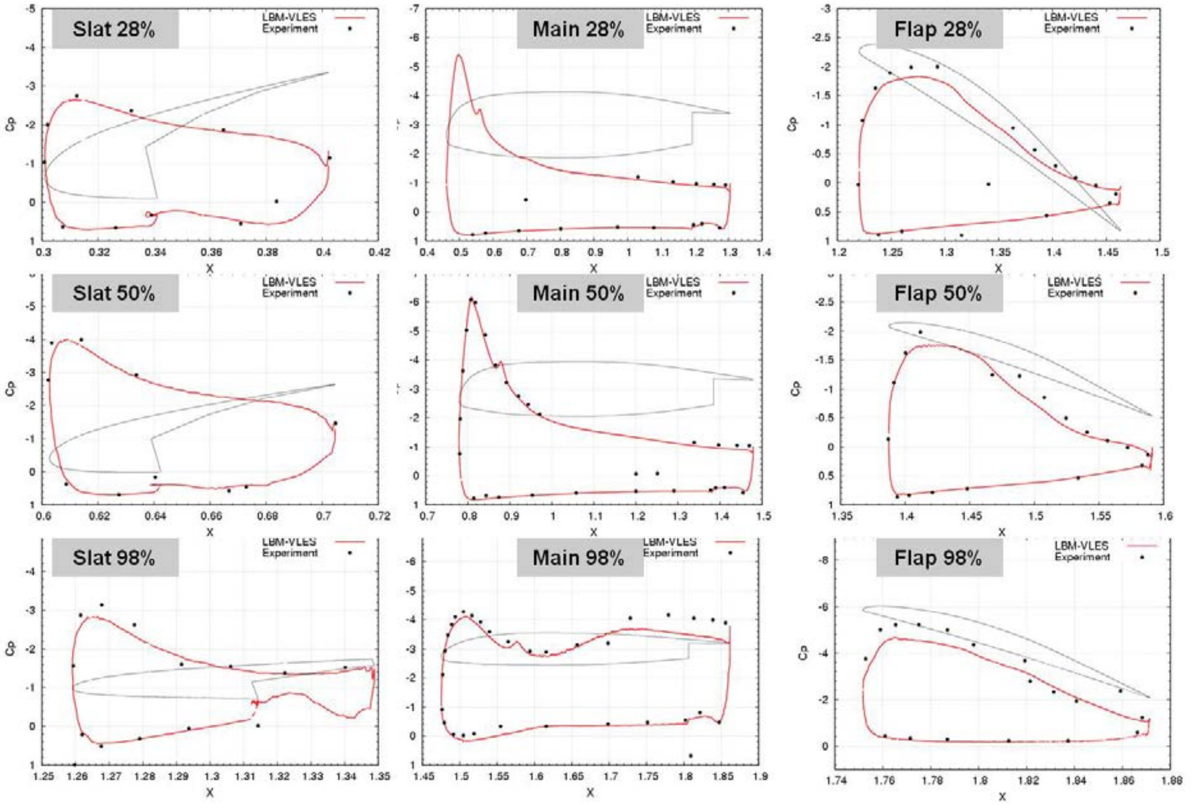


Figure 4.30 Surface pressure coefficient (Cp) distributions at $AOA = 20$ for the studied trapwing FSF model. Columnwise, left: slat element; middle: main element; right: flap element. Rowwise, top: 28% span location; middle: 50% span location; bottom: 98% span location

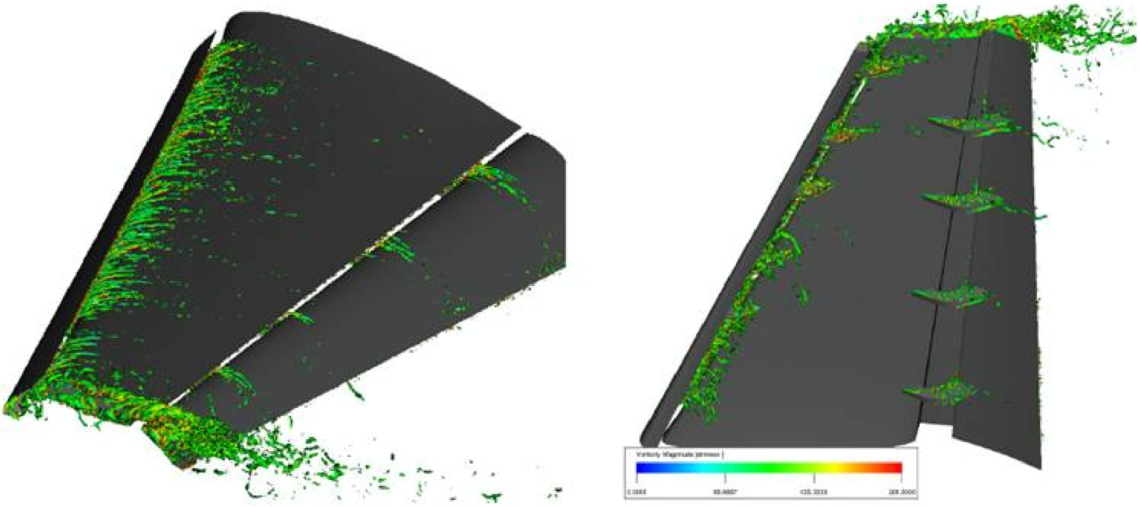


Figure 4.31 Instantaneous vortical flow structures (iso-surface of λ_2 with a value of -300) colored by vorticity magnitude at AOA$= 10$ for the studied trapwing FSF model

CHAPTER 5. An EXTENSION OF VOLUMETRIC LBM BOUNDARY SCHEME FOR SLIDING MESH SIMULATION

In this chapter, we discuss the motivations for developing an volumetric boundary scheme for enforcing sliding mesh interface condition in LBM sliding mesh simulation. The general concept of sliding mesh approach is discussed first, followed by the details of LBM sliding mesh algorithms. Volumetric boundary schemes are developed to enforce interface condition on the slidng-interface, its implementations on a Cartesian grid is illustrated. Benchmark validations are provided to verify the algorithm and demonstrate the accuracy and robustness of the applied scheme.

5.1 Motivation

The fluid flows associated with rotating systems are quite complex and characterized by a variety of unsteady flow phenomena such as laminar-turbulent transitions, boundary layer separation and reattachment, formation and evolution of vortices and above all, mixing and entrainment processes. Such flows are found in almost every industrial process involving fans, propellers, blowers, pumps, stirred tanks, turbo machinery components etc. In these situations, since the involved flow geometry configurations are very complex and include both rotating and stationary parts (where the shape of related geometry can not be generally described by a surface of revolution), it is extremely difficult to solve the problem on a fixed grid system within a single reference frame.

However, for such problems, because the regularity of a rotational motion, the spatial domain may be divided into two sub-domains, and each is represented by a separate grid system, e.g. an inner grid that rotates with the rotating component and an outer grid that stays fixed in laboratory reference frame. The rotation the inner grid produces a relative slide motion on the interface of the two grids. The problem can then be solved on each grid system in a time-dependent manner, with an interface algorithm to couple the sub-domain flow solutions together (Rai (1986); Luo et al. (1993); Murthy et al. (1994)). This is the so-called sliding mesh approach, it has been widely used in CFD for numerical simulation of flows involving rotating geometry (Rai (1989a,b); Daskopoulos and Harris (1996); Hartmann et al. (2004)).

Existing sliding mesh approaches solve the Navier-Stokes on body-fitted grids, with sliding-interface well defined on the grid boundary. For LBM with non-body-fitted Cartesian grid system, there is a lack of study on applying the sliding mesh approach, since it's very challenging to define the sliding mesh interface, and applying the corresponding interface algorithm to couple the flow solutions across it is even hard. Fortunately, with the volumetric boundary scheme that being discussed in Chapter 3 & 4, boundary conditions on curved surfaces can be realized through straightforward particle bounce-back reflection or specular reflection, local conservations and detailed balancing on arbitrary boundaries can be ensured by the simple particle surfel gathering/scattering process. The ability to exact conserve of the hydrodynamic fluxes across across curved boundary allows to further extend the volumetric LB boundary scheme for solving the the sliding-interface problem in LBM. Developing such a LBM sliding mesh approach is beneficial: it not only enables accurate LBM predictions for this type of flow problem, but also allows the use of very convenient Cartesian grid for the solution.

5.2 General concept of sliding mesh

The essential idea behind sliding mesh is straightforward, this can be illustrated with a simple two-dimensional case where an eight-blade mixer is rotating inside a stationary tank, which has six stationary baffles mounted on the wall (Figure 5.1).

5.2.1 Domain sub-division

The flow domain can be divided into two non-overlapping sub-domains, each can be treated as a separate solution regime: the outer one that includes both the stationary tank walls and baffles is fixed in the laboratory reference frame (inertial reference frame), and the the inner one that contains the rotating mixer assumes a cylindrical shape and rotates with the moving geometry. The two sub-domains abut with each other along the interface defined by the cylindrical shape, and the axis of this cylindrical

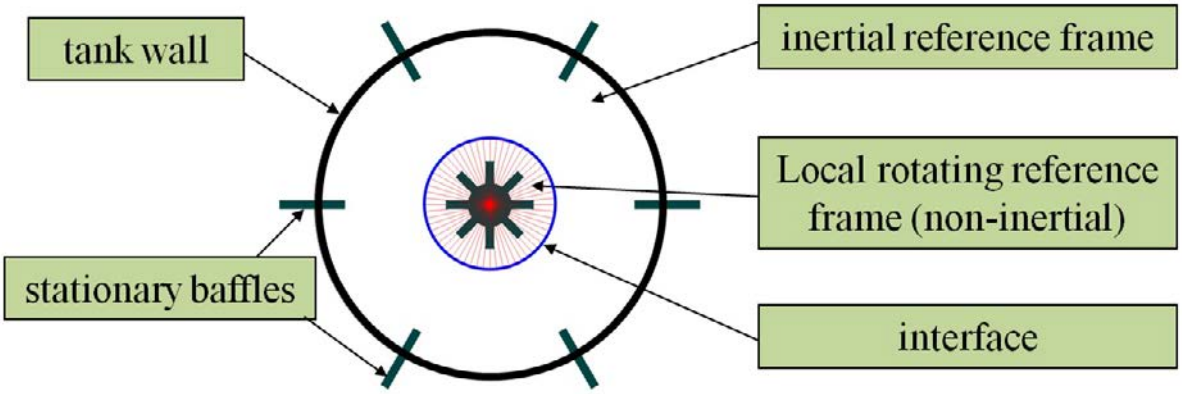


Figure 5.1 Two dimensional example of sliding mesh simulation of flow around a rotating mixer inside a baffled tank

shape coincide with the rotational axis of the mixer. By solving the problem in these two sub-domains and exchange the flow information across the abutting interface (the so called “sliding mesh interface”), the whole flow field solution can be obtained.

5.2.2 Local rotating reference frame and grids for the sub-domains

The solution in the inner domain can be directly transformed into a local rotating reference frame (non-inertial reference frame) that being fixed with the rotating mixers, and the corresponding non-inertial forces is applied to the flow governing equations in this domain. In this way the rotating geometries stay fixed when viewed from the non-inertial reference frame, so that they can be treated as standard walls inside. It should be noted here that any type of solid walls can be included into this sub domain, as long as they share the same rotation motion as the non-inertial reference frame that has. For wall that are not stationary when observed in the non-inertial reference frame, its shape has to be a surface of revolution about the rotation axis, so that a rotating wall boundary condition can be directly applied to it in the simulation.

Through such a reference frame transformation, a fixed grid system can then be applied to discretize the inner domain, so that there is no relative motion between geometry and underlying grid. In other words, when viewed from the laboratory reference frame (inertial reference frame), the grid system for inner domain rotates with the mixer. The discretization of outer-domain is straightforward since no moving parts involved, and a separate grid system can be used.

5.2.3 Interface coupling

By this domain sub-division and applying the frame of reference transformation for the inner domain, both sub-domains can be represented by a separate fixed grid system in its own reference frame, thus avoids the problem of handling relative motions between the rotating components and the underlying grid. Time-dependent flow solutions can be obtained for each individual domain, coupled by proper sliding-interface algorithm that account for the sliding motion of the two grids. In this way, the sliding mesh interface can be viewed as a transparent fictitious boundary separating the two domains, if the neighboring computational cell boundaries are exactly matched (paired) on both sides, flow information can be directly exchanged from the two sides (with proper velocity transformations). However, due to the rotation of the inner grid, the resulting sliding motion along the cylindrical interface between the two grids could produce a mismatch of corresponding computation cell boundary on each side, this requires sliding-interface algorithm to be able to exchange the flow information along the non-matching cell boundaries as well.

In the solution process, the sliding-interface algorithm exchanges flow information between the two domains at each time step, and can effectively couple the solutions on both sides of the interface. Since the sliding mesh interface serves only for computational purpose, it should produce minimum disturbance on the flow, this requires the sliding-interface algorithm to be able to maintain a smooth solution across the interface, as well can ensure the corresponding local conservations of flow field variables such as density and momentum. These are the key factors for accurate and robust prediction with sliding mesh approach.

5.3 LBM sliding mesh algorithm

5.3.1 Solving LBE in a non-inertial reference frame

For flow field inside the local rotating reference frame, the solution is obtained by introducing a force population $\boldsymbol{F}_i(\boldsymbol{x}, t)$ into the collision term of the D3Q19 LBGK model:

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i, t + 1) = f_i(\boldsymbol{x}, t) - \frac{1}{\tau}(f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)) + \boldsymbol{F}_i(\boldsymbol{x}, t) \tag{5.1}$$

Where the force population $\boldsymbol{F}_i(\boldsymbol{x},t)$ can be directly connected to a external body force field $\boldsymbol{A}(\boldsymbol{x},t)$ via (Guo et al. (2002a)):

$$\begin{aligned} \boldsymbol{F}_i &= (1-\frac{1}{2\tau})\rho w_i(\frac{(\boldsymbol{c}_i-\boldsymbol{u}^{'}(\boldsymbol{x},t))\cdot\boldsymbol{A}(\boldsymbol{x},t)}{T} \\ &+\frac{(\boldsymbol{c}_i-\boldsymbol{u}^{'})\cdot(\boldsymbol{c}_i-\boldsymbol{A}(\boldsymbol{x},t)}{T^2}) \end{aligned} \tag{5.2}$$

it is worth pointing out that such a body force treatment has a second order accuracy in velocity space and is essential to the accuracy of numerical simulation with non-inertial forces.

The equilibrium distribution f_i^{eq} take the same form as in equation 2.4, except that the velocity $\boldsymbol{u}(\boldsymbol{x},t)$ is replaced by $\boldsymbol{u}^{'}(\boldsymbol{x},t)$ which is a simple shift of pre-collide velocity by $\boldsymbol{A}(\boldsymbol{x},t)/2$:

$$\boldsymbol{u}^{'}(\boldsymbol{x},t)=\boldsymbol{u}(\boldsymbol{x},t)+\frac{\boldsymbol{A}(\boldsymbol{x},t)}{2} \tag{5.3}$$

For a non-inertial reference frame that rotate with angular velocity of $\Omega(\boldsymbol{x},t)$ about its axis, the corresponding inertial force is defined as:

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{x},t) &= -\underbrace{\boldsymbol{\Omega}(\mathbf{x},\mathbf{t})\times(\boldsymbol{\Omega}(\mathbf{x},\mathbf{t})\times\mathbf{r}(\mathbf{x},\mathbf{t}))}_{\text{centrifugal force}} \\ &-\underbrace{2\boldsymbol{\Omega}(\boldsymbol{x},t)\times\boldsymbol{u}^{'}(\boldsymbol{x},t)}_{\text{Coriolis force}}-\underbrace{\frac{d\boldsymbol{\Omega}(\boldsymbol{x},t)}{dt}\times\boldsymbol{r}(\boldsymbol{x},t))}_{\text{Euler force}} \end{aligned} \tag{5.4}$$

where $\boldsymbol{r}(\boldsymbol{x},t)$ is the position vector measured from the origin of the non-inertial reference frame. and velocity vector are the ones observed in the non-inertial reference frame.

Here the first term on RHS of equation 5.4 represents the centrifugal force Coriolis force, and the second term is the Coriolis force, the third term corresponds to the Euler force. For cases with constant rotation, the Euler force goes to zero.

The velocity transformation between inertial reference frame and the non-inertial reference frame takes the following form:

$$\boldsymbol{u}_{\mathbb{B}}(\boldsymbol{x},t)=\boldsymbol{u}_{\mathbb{A}}(\boldsymbol{x},t)-\boldsymbol{\Omega}(\boldsymbol{x},t)\times\boldsymbol{r}(\boldsymbol{x},t) \tag{5.5}$$

where $\boldsymbol{u}_{\mathbb{B}}(\boldsymbol{x},t)$ is the relative velocity observed in the non-inertial reference frame $\mathbb{B}$ and $\boldsymbol{u}_{\mathbb{A}}(\boldsymbol{x},t)$ is the absolute velocity observed in the inertial reference frame $\mathbb{A}$.

The following notations are also adopted to avoid further confusion: the computational domain for outer part of the fluid region is represented by A and is simulated in the inertial reference frame $\mathbb{A}$, while the computational domain for inner part is represented by B and is simulated in the non-inertial reference frame $\mathbb{B}$. For flow variable with the subscript of $_{\mathbb{A}}$ or $_{\mathbb{B}}$, it means that the variable (vectors) is observed in the inertial reference frame $\mathbb{A}$ or non-inertial reference frame $\mathbb{B}$, while the use of subscript of A or B means that flow variable is defined based on information from the outer computational domain A or inner computational domain B.

5.3.2 Discretization of sliding mesh interface

Handling flow transitions across the sliding mesh interface is the most challenging part of an sliding mesh simulation. A good representation of the sliding mesh interface is very critical for both accuracy and efficiency of the applied algorithms. In this study, a cylindrical double layered transparent surface is applied to represent the sliding mesh interface, with each layer stay fixed to its neighboring computational domain, e.g. the inner layer of the interface is fixed with the inner domain B and the outer one is fixed with domain A. Flow transition can be dynamically realized between these two layers when the two grids slide.

The discretization of both layers results in an identical set of surface mesh: each of the cylindrical layer is divided into m sets of rings orthogonal to the rotation axis, and each set of ring layer is faceted into a set of surface elements of the same size within the ring. Since the ring surface is axisymmetric, the surface elements are faceted with identical shape and size with each ring. At time $T = 0$, there is no gap between the inner and outer layer surface elements, the initial flow field can then be covered by one set of the Cartesian grid, with the two layers of surface mesh separating the two computational domains. The two layers of surface mesh serve as fictitious boundaries for the associated computational domain, and the volumetric LBM boundary algorithm discuss in Chapter 3 & 4 can be applied to construct the "interface" dynamics for flow transition across the sliding mesh interface.

Figure 5.2 gives an example of the discretized double layer interface in two dimension (only part of the interface is shown here). The red solid line represents the discretized outer layer of interface, while the blue dashed line is the inner layer interface mesh. Please note surfel α in domain A forms a matching pair with surfel β in domain B. A Pgram set along a diagonal direction is also outlined to illustrate possible interface dynamics, e.g. surfel α can gather/scatter particles from its (red) Pgram in domain A, while surfel β can collect/redistribute particles via the blue Pgram in domain B.

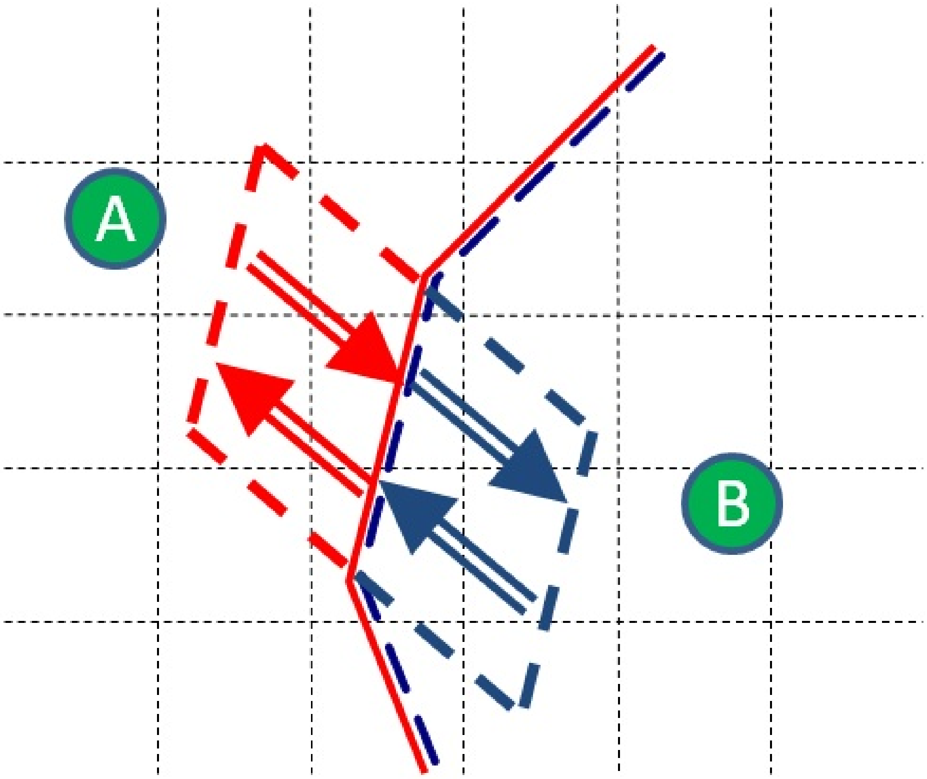


Figure 5.2   Sliding mesh interface is represented by a set of double-layer surfels

5.3.3 A volumetric boundary algorithm for sliding mesh interface condition

The two-layer sliding mesh interface surfels can interact with their neighboring particles via surfel gathering/scattering process to generate the desired flow dynamics. In other word, the sliding mesh interface algorithm can be viewed as a special case for realizing surface dynamics in LBM. There are two fundamental requirement for such a sliding mesh interface algorithm: first, it need to communicate accurately of the flow information on both sides of the interface so that a closed form solution can be obtained for the whole flow domain; second, it should ensure the local conservations of mass and momentum when exchanging information through the interface.

To construct such a surface dynamics, we can directly utilize the specular reflection algorithm discussed in Chapter 3 & 4, since it have nice features to conserve both the local mass and tangential momentum. It can be shown that by simply replacing the tangential velocity with the full velocity vector in the constructed surfer equilibrium, the resulting algorithm is able to conserve both the mass and total momentum flux across the sliding mesh interface. This modified specular reflection algorithm forms the baseline for the current sliding mesh interface algorithm.

In the following, since most of the process details regarding the specular reflection boundary condition, such as surface gather and scattering procedure, surface collision and mass correction for local mass conservation, have already been addressed in Chapter 3 & 4, the details will be skipped and only the most relevant ones of the sliding-interface algorithm will be discussed.

5.3.3.1 Construction equilibrium state for interface surfels

For surfel α in domain A on figure 5.2, its surfel equilibrium state along ith directly can be constructed as:

$$\begin{aligned} f_{i,\mathbb{A}}^{\alpha,eq}(t) &= \bar{\rho}^{\alpha} w_i[1 + \frac{\boldsymbol{c}_i \cdot \bar{\boldsymbol{u}}_{\mathbb{A}}^{\alpha}}{T} - \frac{(\boldsymbol{c}_i \cdot \bar{\boldsymbol{u}}_{\mathbb{A}}^{\alpha})}{2T} \\ &\quad + \frac{(\boldsymbol{c}_i \cdot \bar{\boldsymbol{u}}_{\mathbb{A}}^{\alpha})^2}{2T^2} + \frac{(\boldsymbol{c}_i \cdot \bar{\boldsymbol{u}}_{\mathbb{A}}^{\alpha})^3}{6T^3} - \frac{(\boldsymbol{c}_i \cdot \bar{\boldsymbol{u}}_{\mathbb{A}}^{\alpha})}{2T^2}(\bar{\boldsymbol{u}}_{\mathbb{A}}^{\alpha})^2] \end{aligned} \tag{5.6}$$

As noted already, the use of full velocity vector here is essential for the conservation of mass and total momentum flux across the sliding mesh interface.

Here $\bar{\rho}^{\alpha}$ is the averaged density based on sampled values from both sides:

$$\bar{\rho}^{\alpha} = \frac{1}{2}(\rho^{\alpha,A} + \rho^{\alpha,B}) \tag{5.7}$$

and $\bar{\boldsymbol{u}}_{\mathbb{A}}^{\alpha}$ are mass averaged velocity based on sampled values from both sides:

$$\bar{\boldsymbol{u}}_{\mathbb{A}}^{\alpha} = \frac{1}{2\bar{\rho}^{\alpha}}(\rho^{\alpha,A}\boldsymbol{u}_{\mathbb{A}}^{\alpha,A} + \rho^{\alpha,B}\boldsymbol{u}_{\mathbb{A}}^{\alpha,B}) \tag{5.8}$$

In the above, $\rho^{\alpha,A}$ represents sampled density value for surfel α from the domain A, which can be calculated directly based on its Pgram. $\rho^{\alpha,B}$ represents sampled density value for surfel α from the domain B and its calculation is done through its surfel match pair β. $\boldsymbol{u}_{\mathbb{A}}^{\alpha,A}$ represents the sampled velocity value for surfel α from the domain A, and $\boldsymbol{u}_{\mathbb{A}}^{\alpha,B}$ is the sampled one for surfel α from domain B. Both velocities are observed in reference frame $\mathbb{A}$.

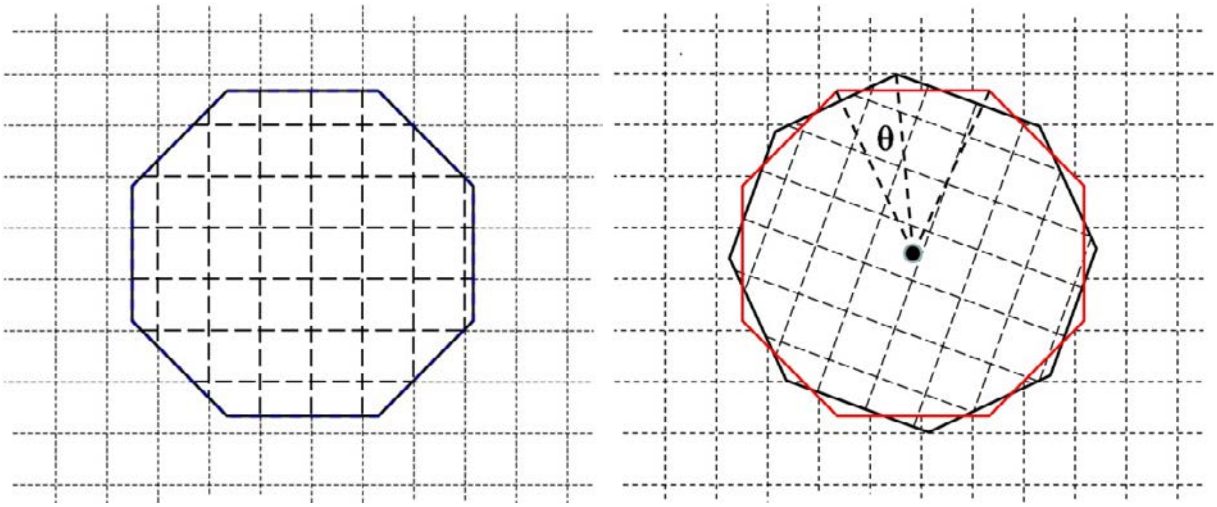


Figure 5.3 Illustration of mis-matching of surface elements on sliding mesh interface: Grid system at T=0 (Left) and T > 0 (Right)

For surfel β in domain B, the procedure to construct surfel equlibrium states is the same, except the velocities need to be observed in reference frame $\mathbb{B}$.

It should be pointed out that by using the two-sides averaged values in the equilibrium distribution, the solutions in both domain A and domain B are closely coupled together.

5.3.3.2 Surfel matching algorithm

To calculate the averaged density and velocity value in equation 5.7 and equation 5.8, samplings of density and velocities in both domains on each side of surfel are needed . If the sampled region is in the same domain as surfel is, the process is straightforward and can be directly calculated via surfel's Pgram (as described in chapter 3). However, if the the sampled region is not in the same domain, the sampling process can only be done through surfel's matching pair in the neighboring domain. The process is non-trival, since the underling surface mesh for both layers may not necessary coincide with each other during the simulation.

An example is used here to illustrate this surfel mismatch situation (Figure 5.3): At $T = 0$, both the inner and outer layer of surface mesh coincide with each other, and when $T > 0$, the inner grid rotates about an angle of θ, this results in a mis-match of inner and outer layer of surface mesh.

For mis-matched situation, due to the identical size and shape for surface elements inside a given ring, each surface element in the outer domain A can only intercept at most two surface elements in the inner domain B any given time (Figure 5.4). In this example, surfel α in domain A establishes a matching pair set of surfel $\beta 1$ and $\beta 2$ in domain B.

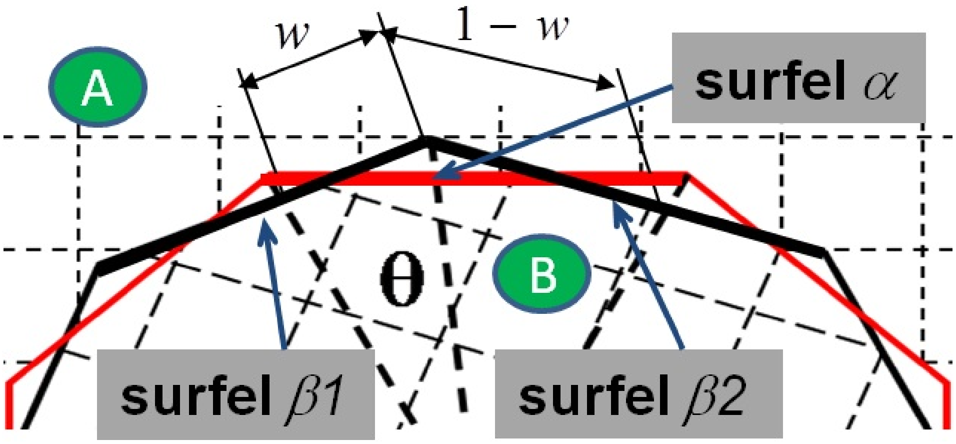


Figure 5.4 Matching surfel α in domain A with surfel $\beta 1$ and $\beta 2$ in domain B

A matching ratio $\omega(t)$ can be defined based on the geometrical information of the matching pair set as:

$$\omega(t) = \frac{A_{\beta 1^B \to \alpha^A}}{A_\alpha} \tag{5.9}$$

where $A_{\beta 1^B \to \alpha^A}$ is the projected area of surfel $\beta 1$ to surfel α.

Clearly,

$$A_\alpha = A_{\beta 1^B \to \alpha^A} + A_{\beta 2^B \to \alpha^A} \tag{5.10}$$

or

$$1 - \omega(t) = \frac{A_{\beta 2^B \to \alpha^A}}{A_\alpha} \tag{5.11}$$

with $A_{\beta 2^B \to \alpha^A}$ being the projected area of surfel $\beta 2$ to surfel α.

With this definition, the sampling of density and velocity from domain B side for surfel α (located in domain A) can be directly linked to the sampling of density and velocity for its matched pair surfel $\beta 1$ and $\beta 2$:

$$\begin{aligned} \rho^{\alpha,B} &= \frac{A_{\beta 1^B \to \alpha^A} \rho^{\beta 1,B} + A_{\beta 2^B \to \alpha^A} \rho^{\beta 2,B}}{A_\alpha} \\ &= \omega(t)\rho^{\beta 1,B} + (1-\omega(t))\rho^{\beta 2,B} \end{aligned} \tag{5.12}$$

$$
\begin{aligned}
\boldsymbol{u}_{\mathbb{A}}^{\alpha,B} &= \frac{A_{\beta 1^B \rightarrow \alpha^A} \rho^{\beta 1,B} \boldsymbol{u}_{\mathbb{A}}^{\beta 1,B} + A_{\beta 2^B \rightarrow \alpha^A} \rho^{\beta 2,B} \boldsymbol{u}_{\mathbb{A}}^{\beta 2,B}}{A_{\alpha} \rho^{\alpha,B}} \\
&= \frac{\omega(t) \rho^{\beta 1,B} \boldsymbol{u}_{\mathbb{A}}^{\beta 1,B} + (1-\omega(t)) \rho^{\beta 2,B} \boldsymbol{u}_{\mathbb{A}}^{\beta 2,B}}{\rho^{\alpha,B}} \qquad (5.13)
\end{aligned}
$$

In this way, sampling across different domain can be realized through its match surfel pair in that corresponding domain.

The matching ratio $\omega(t)$ also defines a physical process on the amount of flow field information that the surfel can receive from its matching pair: in this example, surfel α receives $\omega(t)$ part of flow field information from surfel $\beta 1$ and $1 - \omega(t)$ part of flow field information from surfel $\beta 2$. Its value is uniquely defined for each interface surfel element at every timestep with given mesh size and rotating speed, and it is bounded by $0 \leq \omega(t) \leq 1$. It should be noted here that due to the equal-size of all the surface elements in each ring, their corresponding matching ratio are the same at every time sequence, and these values can be pre-computed before the simulation so that the overall computation cost can be minimized.

5.3.3.3 Solution procedure

The standard surfel gathering, collision and scattering process can be directly applied to both sides of the interface surfel, and the overall solution procedure is outlined in the following steps:

1. Gather incoming states and sample mass and momentum from neighbor cells on both sides of sliding mesh interface.

2. Match sliding mesh interface elements between outer and inner surfaces. As mentioned earlier, each surface element on one side communicates with two surface elements on the other side at any given time. The matching ratio of two surface elements depends on the sliding mesh rotational position.

3. Transform the surface velocity from local rotating reference to stationary laboratory reference frame inside sliding mesh.

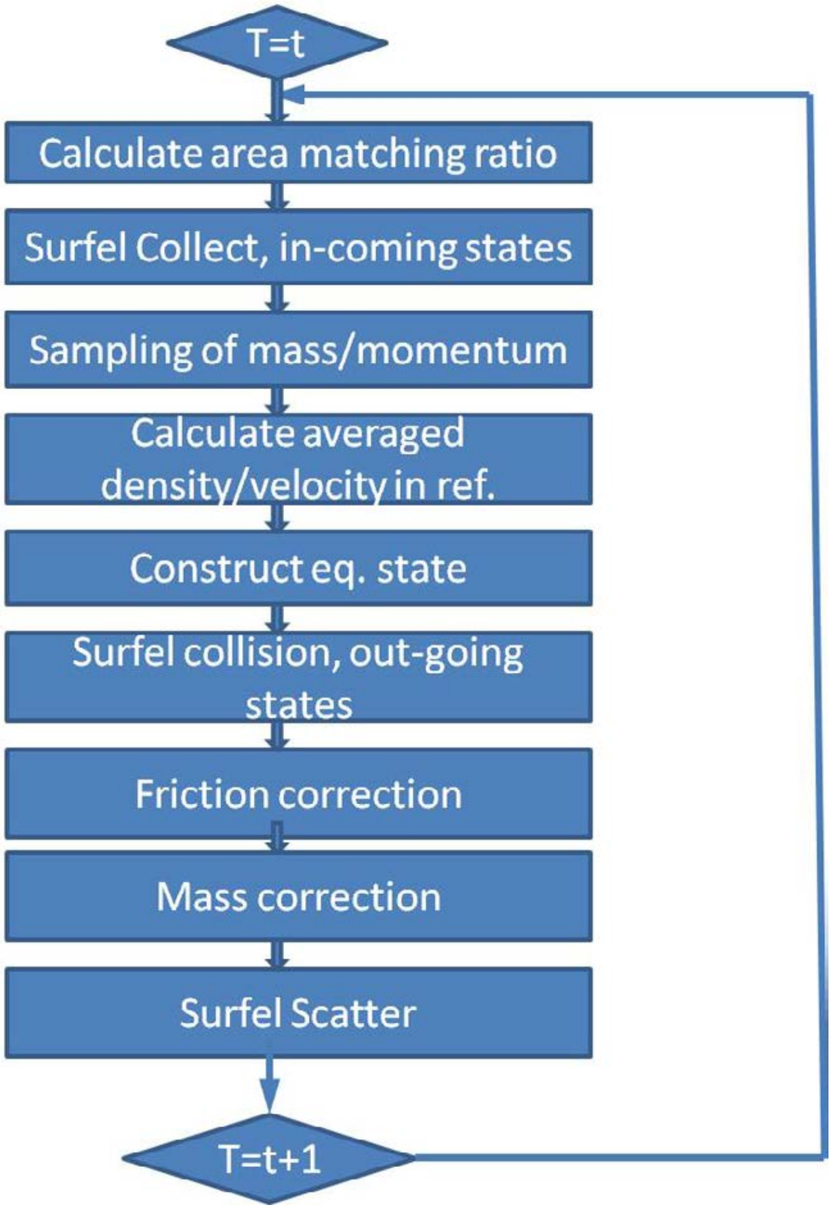

T=t
Calculate area matching ratio
Surfel Collect, in-coming states
Sampling of mass/momentum
Calculate averaged density/velocity in ref.
Construct eq. state
Surfel collision, out-going states
Friction correction
Mass correction
Surfel Scatter
T=t+1

Figure 5.5 Solution procedure for the proposed LBM sliding mesh approach

4. Calculate averaged values of mass ρ and momentum $\rho\boldsymbol{u}$for each matched sliding interface element pairs in ground fixed reference frame. Then the momentum values inside sliding mesh are also calculated

5. Construct equilibrium states on interface surfel elements in both domains

6. Apply the surfel collision in all surfels.

7. Ensure local mass conservation.

8. Scatter outgoing particles back to its neighboring cells.

The process is sketched in figure 5.5:

5.4 Benchmarked cases with the LBM sliding mesh approach

Based on the sliding mesh algorithm presented in the previous, time accurate numerical validations were conducted for three different cases: 2D direct numerical simulation of co-rotating cylinder flow, 2D direct numerical simulation of cross flow past a single rotating blade and a 3D turbulent flow past a benchmark D4119 Propeller.

5.4.1 Two dimensional laminar flow driven by a rotating cylinder

As a first test, a simple test case of a two dimensional couette flow being driven by two concentric rotating cylinders (figure 5.6) is considered to examine the accuracy of the scheme. The configuration includes two coaxial cylinders with radius of r and $2r$ each, the inner cylinder is rotating at a constant angular velocity ω_o and the outer cylinder is fixed. The simulation Reynolds number based on cylinder radius r and a characteristic velocity magnitude of $u_{\text{char}} = r\omega_o$ is set to 147, For such a low Reynolds number flow, the resulting flow field is axi-symmetric with a well-known couette flow profile. In simulation, the characteristic velocity is mapped to a fixed lattice velocity with corresponding mach number of $M = 0.15$. This results in a fluid viscosity of $\nu = 0.01$ in lattice units when a reference resolution of 16 cells is applied to across the length of cylinder radius. The corresponding relaxation time of $\tau = 0.53$.

Grid convergence studies were conducted to verify the accuracy of the scheme. For this flow configuration, since it's fully axisymmetric, the location of sliding interface can be placed at any arbitrary distance way for the solid wall, and the resulting sliding mesh prediction should be less dependent on the sliding mesh interface location. This was also being verified in the current study. Figure 5.7 and figure 5.9 show comparison of normalized tangential velocity and pressure distribution along the flow axis at three grid resolutions ($r/\delta = 16, 32, 64$) for a sliding mesh interface location at $d/r = 1/32$ for the inner cylinder. Figure 5.8 and figure 5.10 show comparison of normalized tangential velocity and

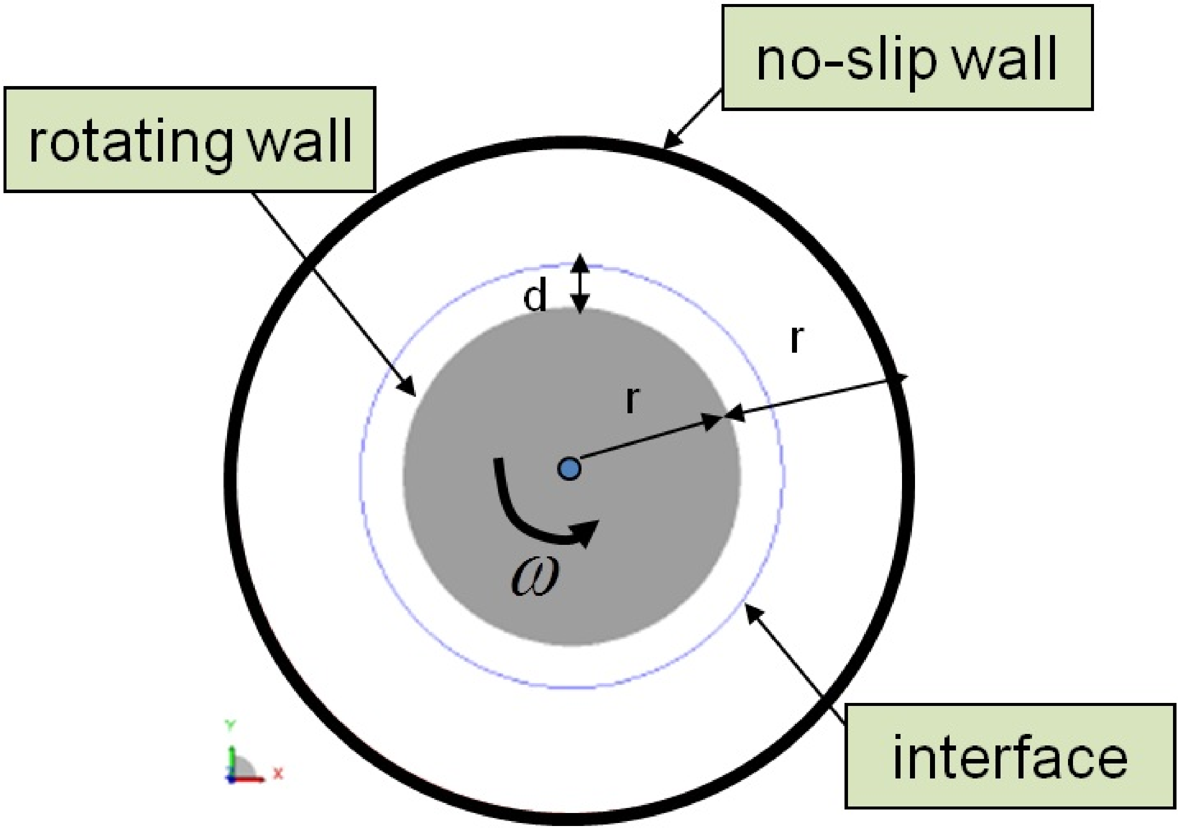


Figure 5.6 Flow configuration for LBM sliding mesh simulation of two-dimensional rotating flow. The sliding mesh interface is represented by a blue line and is set at a distance of d from the inner cylinder surface

pressure distribution along the flow axis at three grid resolutions for a sliding mesh interface location at $d/r = 1/2$ for the inner cylinder. The results are plotted against the predictions without sliding mesh as well (so that a rotating wall boundary condition is directly applied to inner cylinder wall). It can be observed that all the sliding mesh results agree well with the one that without sliding mesh applied, and a good grid convergence can be obtained for both sliding mesh interface locations. The solutions near sliding mesh interface does not show apparent discontinuity, which indicates that a smooth transition of flow field across sliding mesh interface can be obtained for such a stressful situation with very low lattice viscosity. Figure 5.11 measures the error of velocity field as a function of grid resolution, and the error is calculated by comparing the sliding mesh solution (denoted by $\boldsymbol{u}_n^{\text{sldm}}$), with the one that no-sliding mesh approach was applied (denoted by $\boldsymbol{u}_n^{\text{no-sldm}}$):

$$L_2 = \sqrt{\frac{\sum_n (\boldsymbol{u}_n^{\text{sldm}} - \boldsymbol{u}_n^{\text{no-sldm}})^2}{N}} \tag{5.14}$$

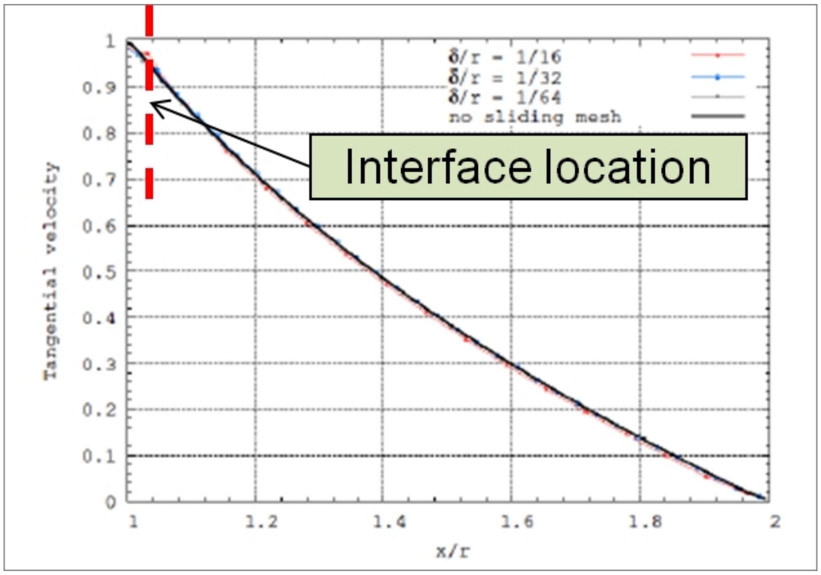


Figure 5.7 Tangential velocity distribution along flow axis for sliding mesh interface located at a distance of $r/32$ away from the inner cylinder

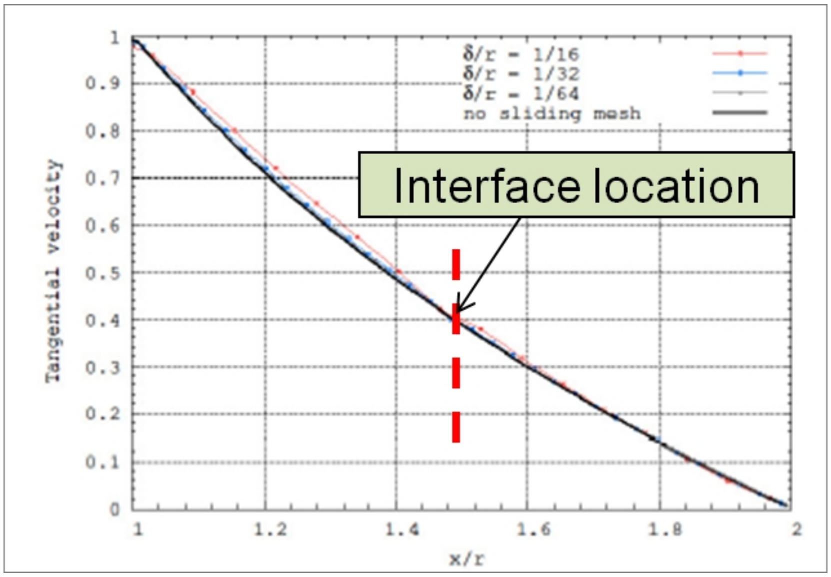


Figure 5.8 Tangential velocity distribution along flow axis for sliding mesh interface located at a distance of $r/2$ away from the inner cylinder

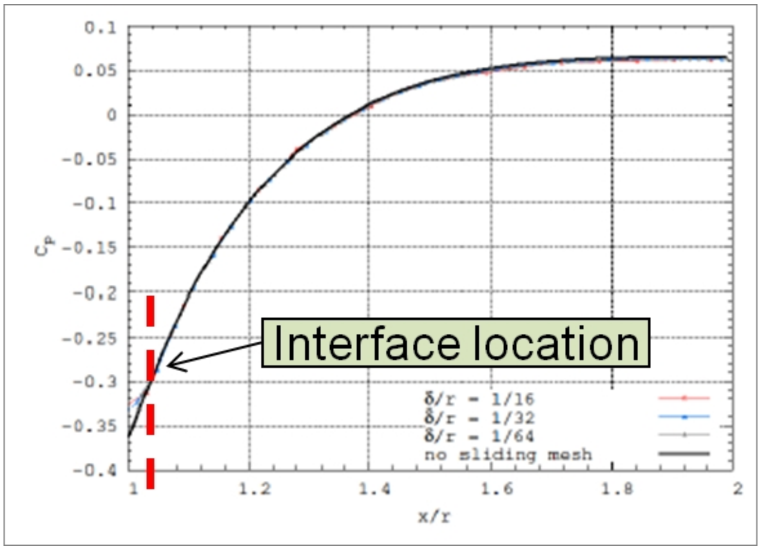


Figure 5.9 Non-dimensional pressure distribution along flow axis for sliding mesh interface located at a distance of $r/32$ away from the inner cylinder

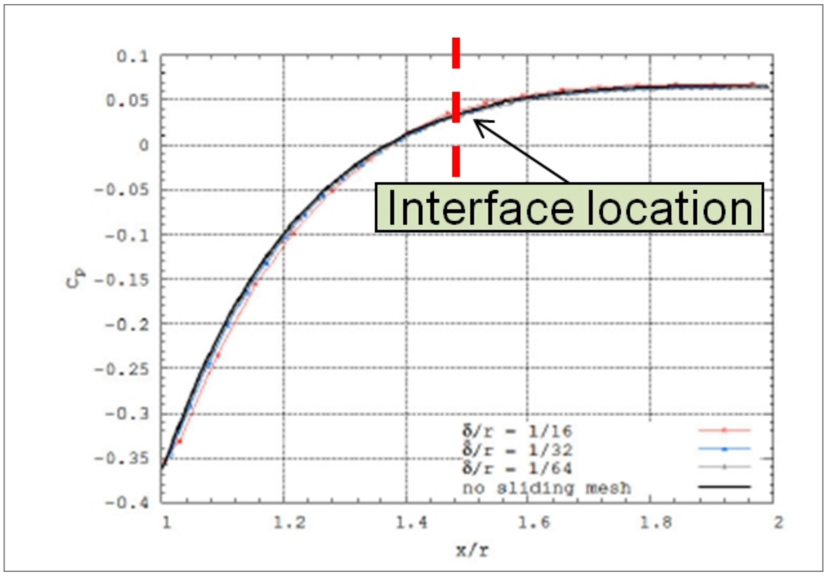


Figure 5.10 Non-dimensional pressure distribution along flow axis for sliding mesh interface located at a distance of $r/2$ away from the inner cylinder

For all the three tested sliding mesh interface location, the errors show nearly exponential decay with increased grid resolution, the slope of the decay rates indicate that current scheme is second order accurate in space. With same resolution, the error does not show strong dependence on the sliding mesh interface location, this also proves that current scheme is capable of achieving good results consistently

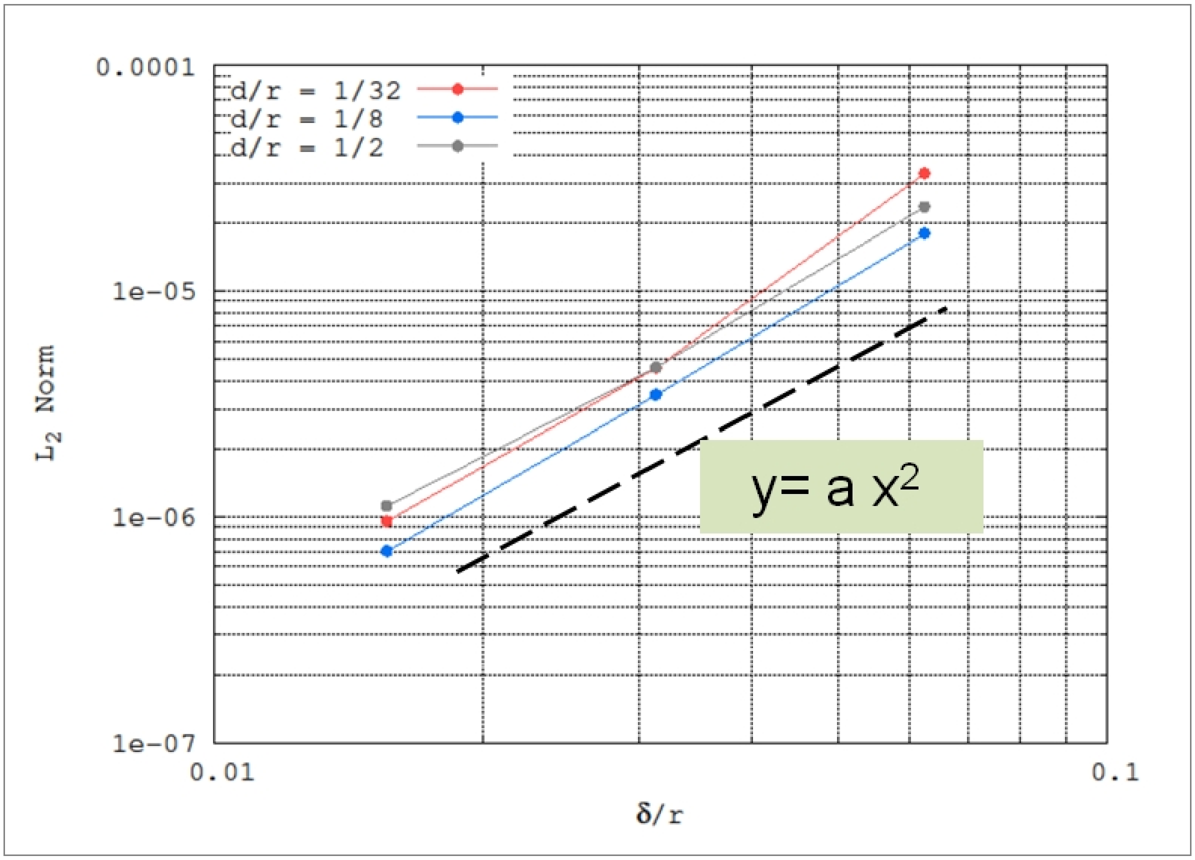


Figure 5.11   Grid convergence study for three typical sliding mesh interface locations

even the distance of sliding mesh interface is very close to the rotating surface.

5.4.2 Two dimensional laminar flow past a rotating single-blade baffle

In the second test, a two dimensional crossing flow past a single rotating blade configuration is considered to verify the implementation of current sliding mesh approach. The blade is rotating at clockwise direction and the flow is also confined by stationary walls on both the top and bottom side (figure 5.12). The flow domain is covered by uniform size grid with 32 cells being used for the full length of the blade, and a rotating angular velocity of 0.0327 radius per lattice time step is chosen, the flow Reynolds number based on blade radius and blade tip velocity is 115. Contours of instantaneous flow field velocity magnitude is presented in figure 5.13. It can be seen that a reasonable flow field is obtained for this configuration, and local velocity magnitude is high when the the blade tip passes by the region close to top wall, this corresponds to the time that blade tip velocity are in-line with the incoming flow velocity, for the flow region between bottom wall and rotating blade, the flow is de-accelerated due to the counter directions of blade tip velocity and incoming free stream velocities. As seen from the snapshots, the smooth distribution of velocity field also indicate the treatment of sliding mesh interface in current approach is accurate.

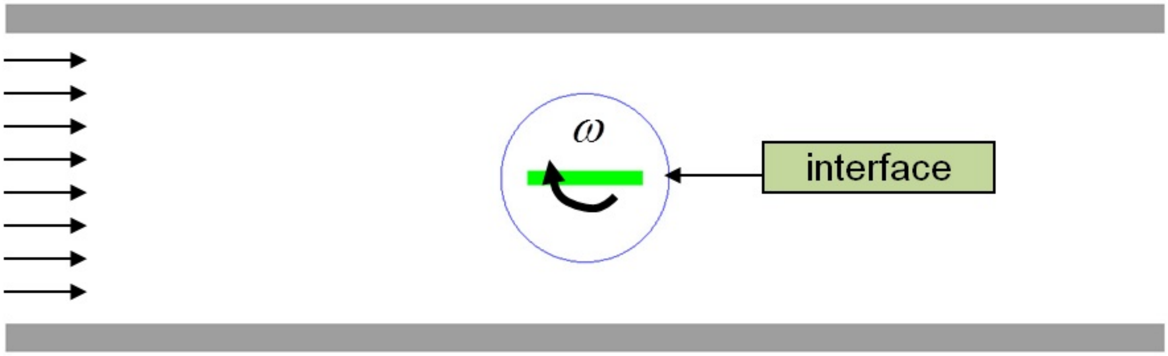


Figure 5.12 Flow configuration for LBM sliding mesh simulation of a two-dimensional cross flow past single rotating blade. The sliding mesh interface is indicated by a red color

5.4.3 Three dimensional turbulent flow past an DTMB 4119 Propeller

In the third test, a complex benchmark case of a propeller rotating in a flow field with uniform incoming velocity was selected to validate the accuracy of current sliding mesh approach. The propeller involved in this study is a symmetric three bladed propeller designed in the 1960s, known as DTMB 4119 propeller, it has zero skewed section profiles and operates at high speeds of rotation. This propeller has been widely used as standard benchmark geometry to validate lifting surface design methods (Jessup (1998); Gindroz et al. (1998)).

Figure 5.14 shows a schematic of the DTMB 4119 propeller geometry, the propeller has a radius of $R = 152.4$ mm and rotates at a speed of $n = 10$ rps (revolutions per seconds). In the open water test, the propeller has a advance velocity magnitude V, which corresponds to a propeller advance coefficient J that generally being used to describe the its operation:

$$\text{Advance coefficient:} \qquad J = \frac{V}{nD} \tag{5.15}$$

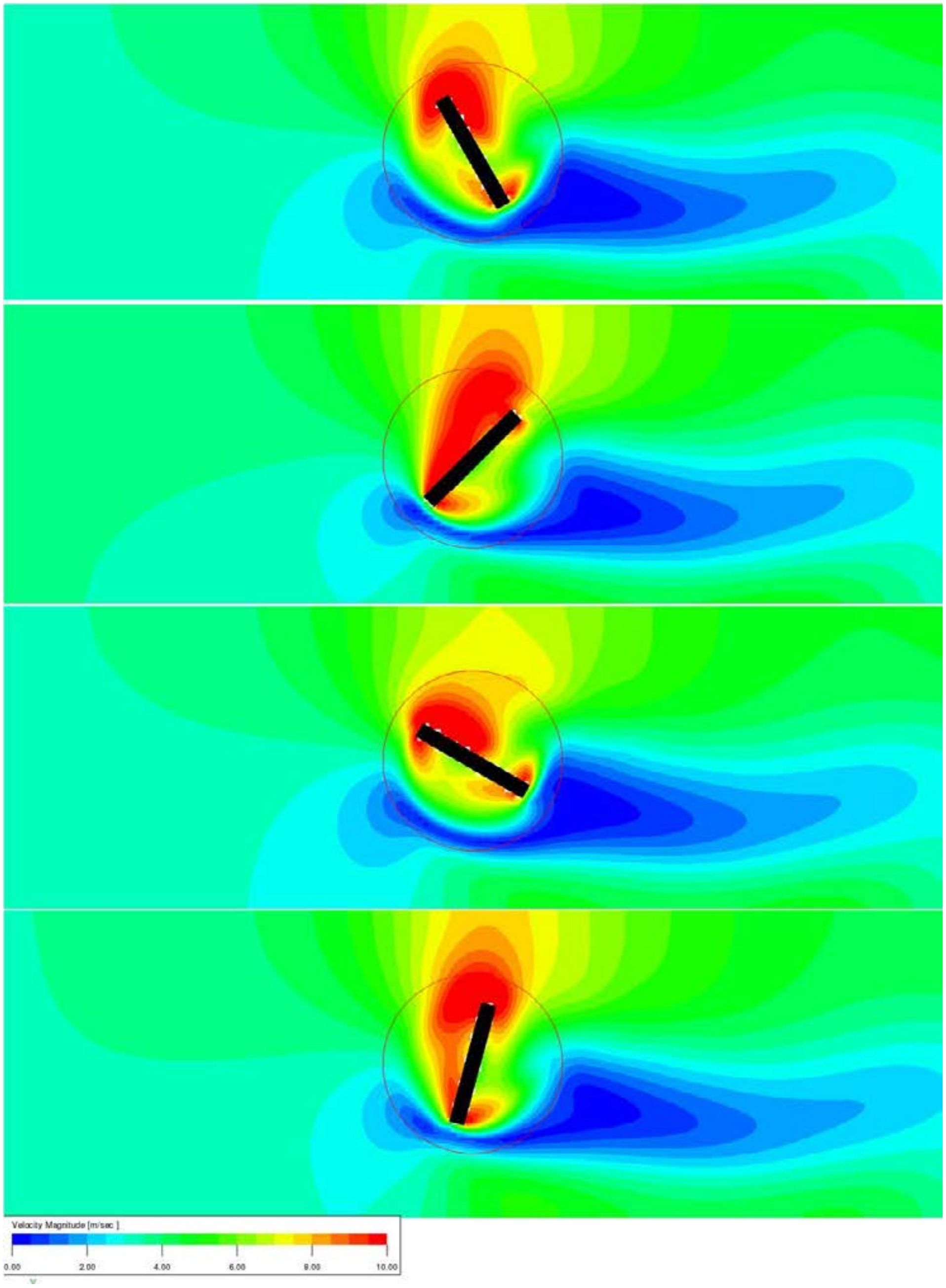


Figure 5.13 Instantaneous vorticity field for 2D cross flow past single rotating blade

In simulation, the advancing motion of propeller in open water test is translated into a flow problem with uniform incoming velocity so that propeller only has a relative rotating motion with respect to the simulation domain. A cylindrical sliding mesh region that tightly enclose the propeller was used in simulation (Figure 5.15), a fine grid resolution with 200 cells per blade radius length was applied

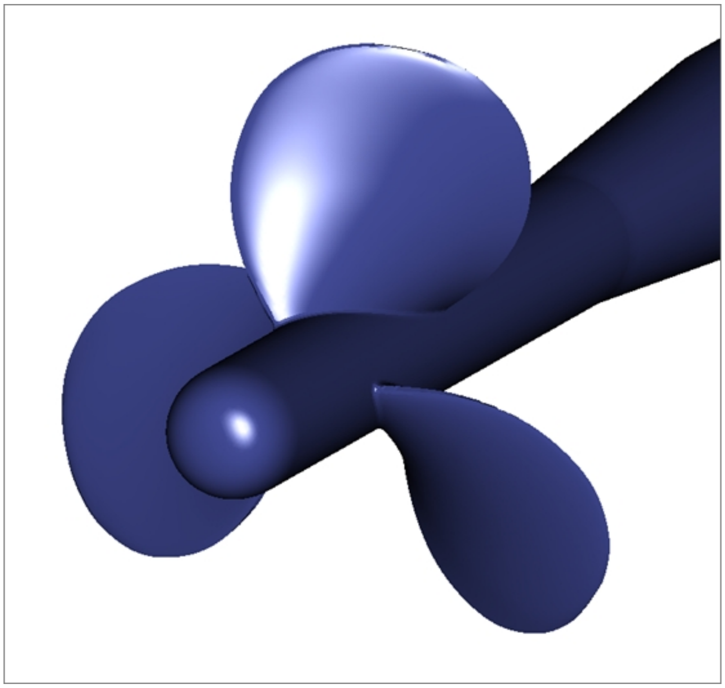


Figure 5.14 Schematic of the DTMB 4119 propeller geometry

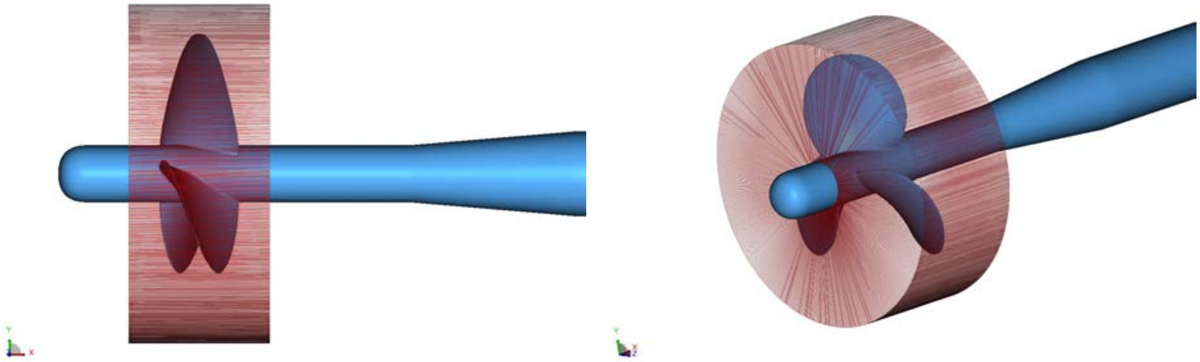


Figure 5.15 Sliding mesh configuration for the DTMB 4119 propeller. The red cylindrical region is defined as sliding mesh domain

for this region. Variable resolutions are applied and the grid system used is shown in figure 5.16. The overall fluid cells are $52.4M$ and surfels are $1.3M$, on a Linux Xeon 5160 (dual-core) cluster with 108 cores, it takes 19.4 hours to complete 12 revolution (corresponding to a physical time of 1.2 second, or $13.4k$ timesteps).

The propeller is simulated in a wide range of advance coefficient around the design operational

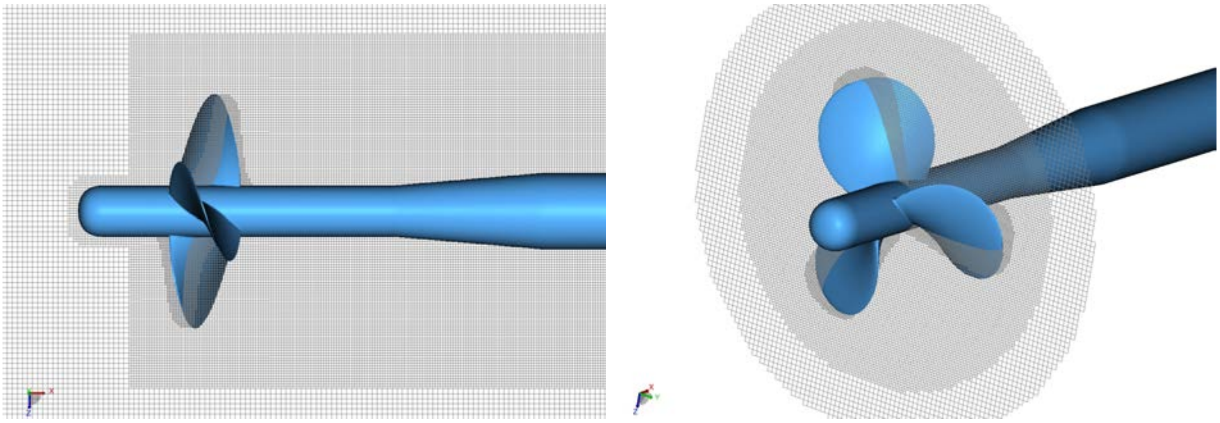


Figure 5.16 Grid system used for current DTMB 4119 sliding mesh simulation

point $J = 0.833$, a typical Reynolds number based on the blade tip velocity $2rn\pi$ and blade radius is 1.46×10^{-6}. For such a high Reynolds number turbulent flow, the sliding mesh approach here is coupled through the LBM-VLES turbulence modeling approach for the solution. The related PDE sliding mesh algorithm can be referenced at Sun et al. (2009).

Our simulation results are compared directly with the open water test data of Jessup (1989, 1998).

At given operating condition, the performance of the propeller is characterized by a set of non-dimensional coefficients based on the generated propeller thrust T and torque Q:

$$\text{Thrust coefficient:} \quad K_T = \frac{T}{\rho n^2 D^4} \tag{5.16}$$

$$\text{Torque coefficient:} \quad K_Q = \frac{Q}{\rho n^2 D^5} \tag{5.17}$$

$$\text{Propeller efficiency:} \quad \eta_o = \frac{K_T J}{K_Q 2\pi} \tag{5.18}$$

The predicted propeller performance result is directly compared with available open water data and it's shown in figure 5.17, 5.18 and 5.19.

It can be seen that current LBM sliding mesh approach gives reasonable good predictions on propeller thrust coefficient, torque coefficient and efficiency, especially when the propeller advance coefficient is low. When $J = 1.0$, large discrepancy between the prediction and experiment can be observed. It could be due to the fact that cavitation effects that occurs in experiment can not be predicted by current single phase simulation.

Figure 5.20 shows snapshots of the three dimensional vortical flow structures around the propeller,

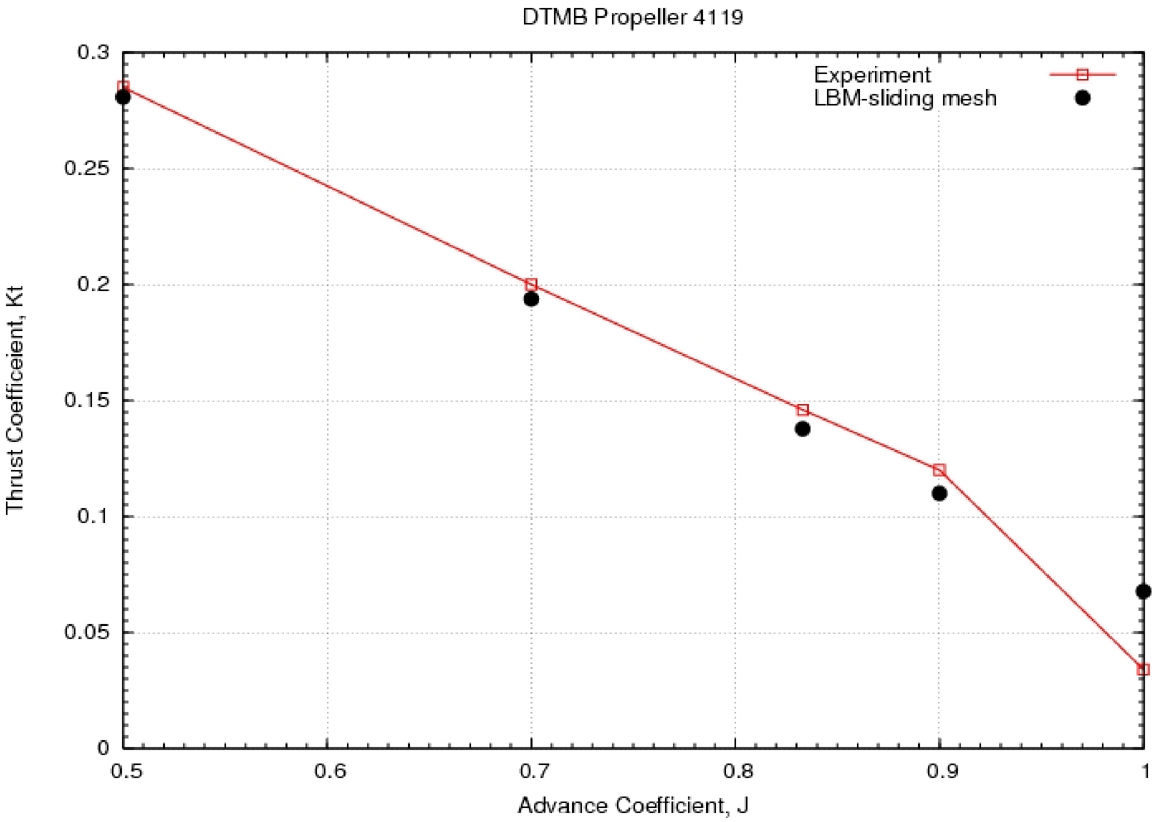


Figure 5.17 Thrust coefficient comparisons: LBM sliding mesh results vs. experimental data

the vortical flow structure is represented by iso-surfaces of λ_2 and is colored by the local pressure. These images clearly demonstrate the complexity of the flow.

The instantaneous vorticity distributions across propeller center plane is shown in figure 5.21, the resulting vorticity field is very smooth and shows no discontinuity across the sliding mesh boundary.

Figure 5.22 shows comparisons of circumferential averaged velocity components between the experiments and the LBM based sliding mesh simulations for the operating condition corresponding to J=0.833. The velocities are compared downstream of the propeller at X/R=0.3281, where X is positive downstream, with its origin at the propeller reference line. Note that all velocities are normalized by the advance speed, V. The data in figure 5.22 is presented as line plots of velocity versus radius of the propeller. It shows good agreements with experimental data at all locations, although the axial velocity is slightly over predicted away from the center.

Figure 5.23 shows similar analysis of the phase-averaged velocity at the same downstream location.

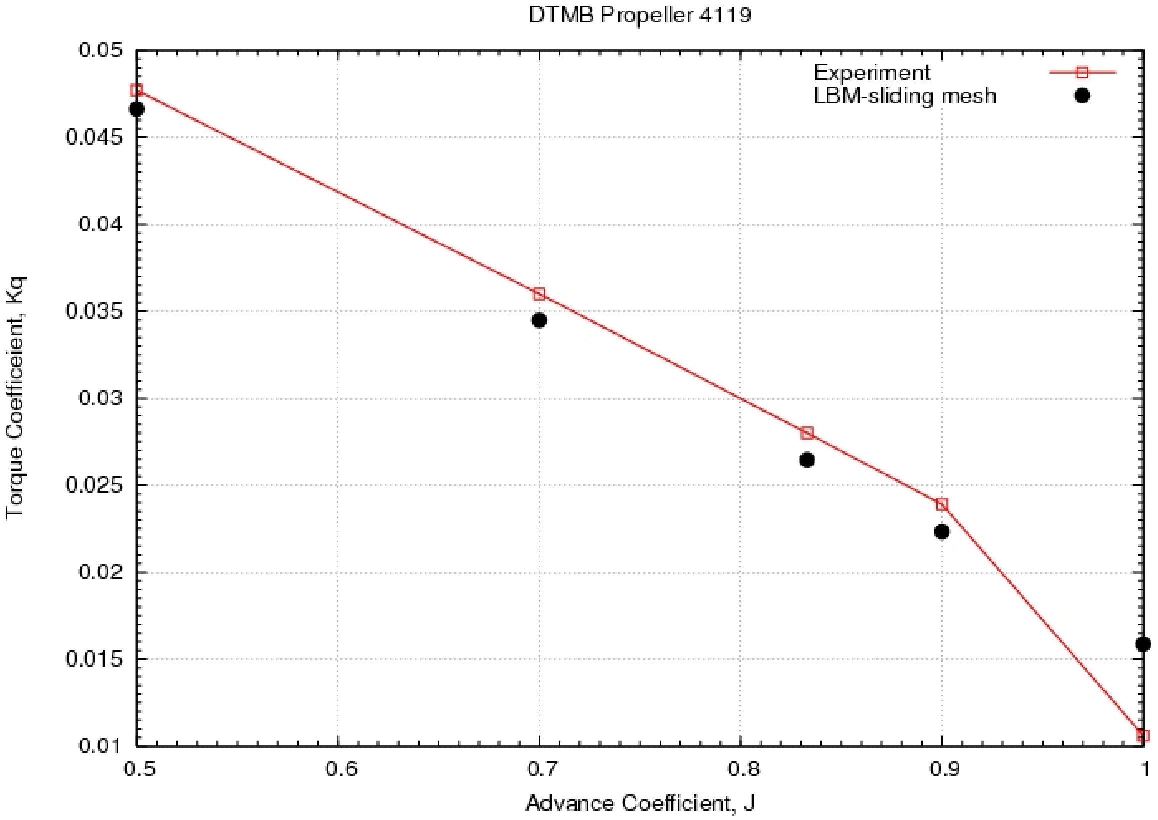


Figure 5.18 Torque coefficient comparisons: LBM sliding mesh results vs. experimental data

All the data is presented as line plots of velocity versus blade angle. Representative measurements are compared at r/R=0.3, 0.7, 0.9 and 0.924. Again the computed data compares reasonably well to the experimental data.

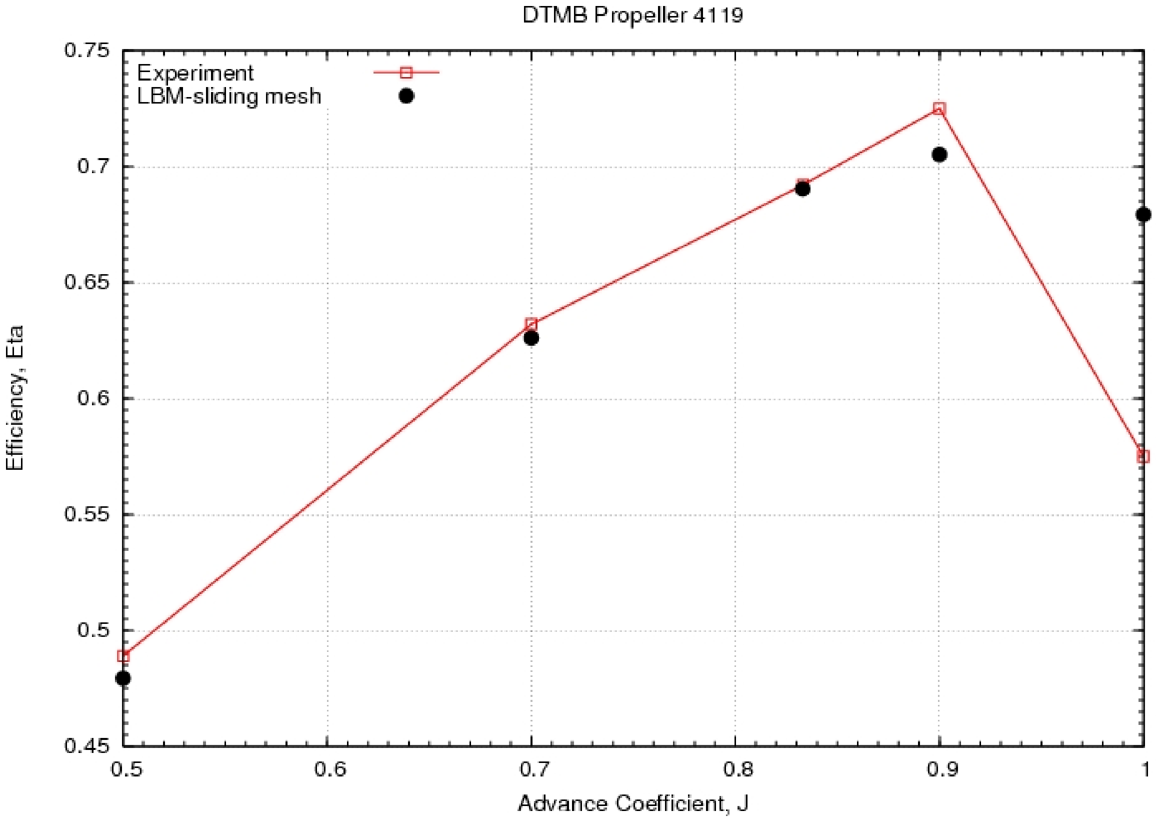
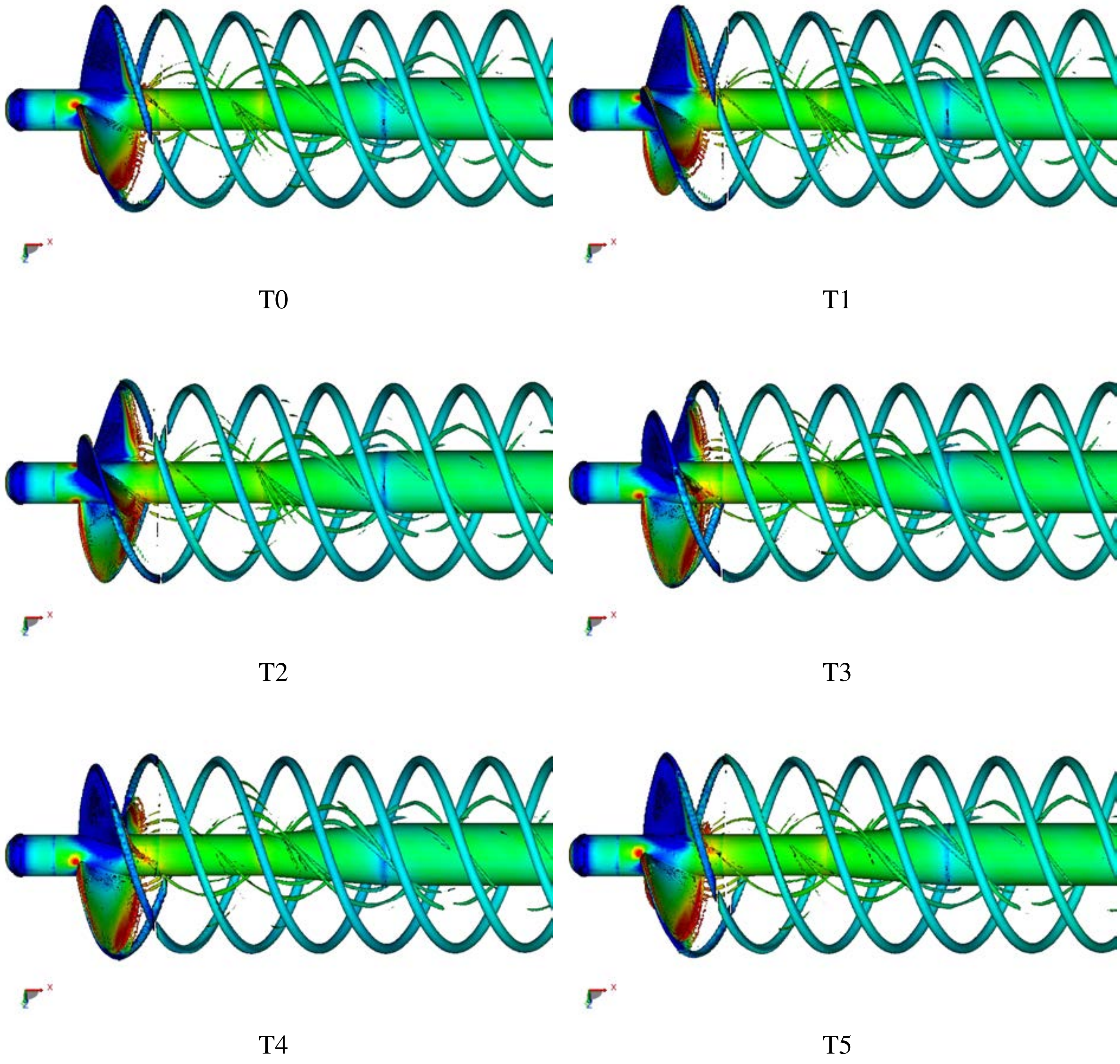


Figure 5.19 Propeller efficiency comparisons: LBM sliding mesh results vs. experimental data


T0

T1

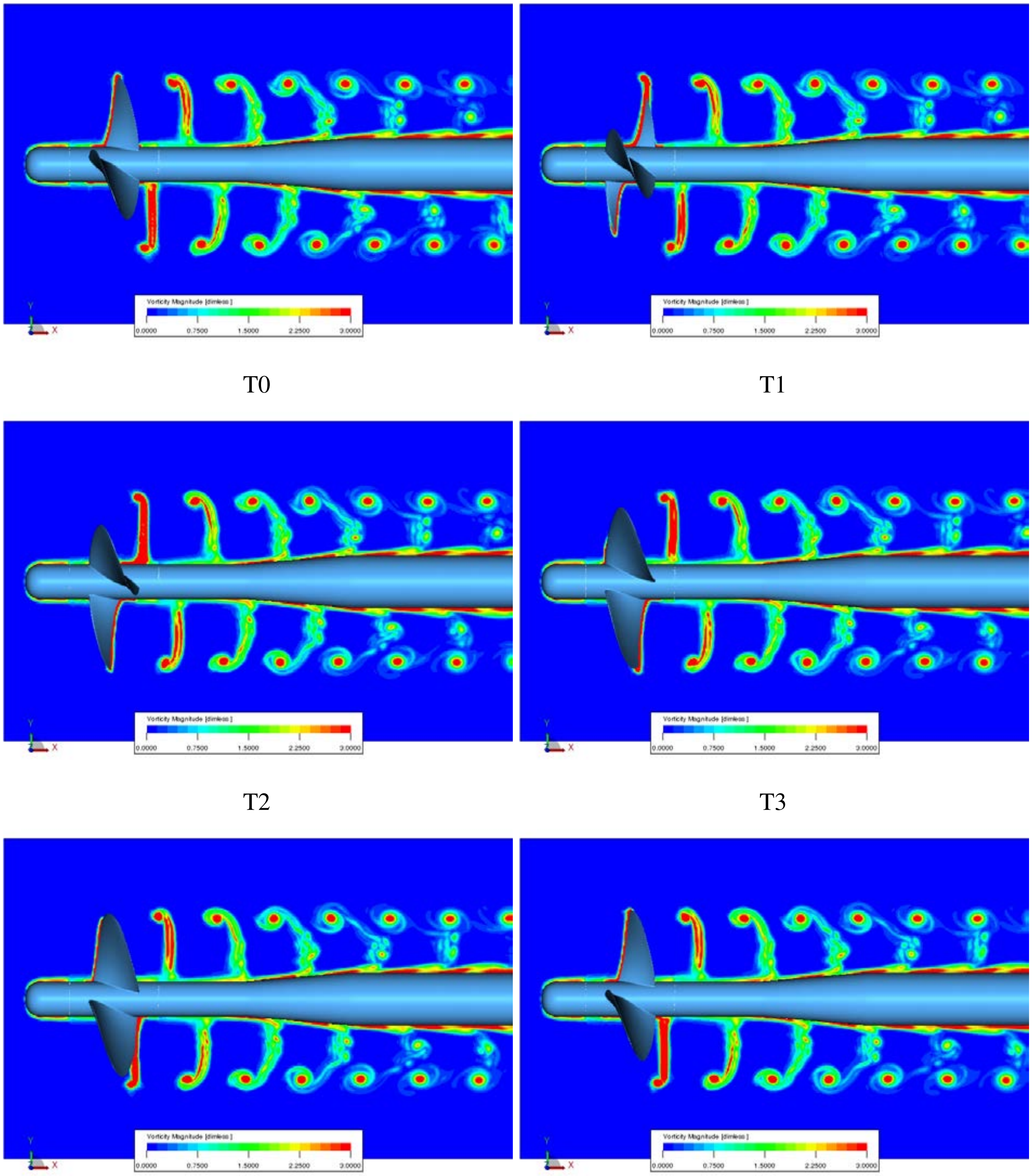
T2

T3

T4

T5

Figure 5.20 Iso-surface of instantaneous vortical flow structures (λ_2 magnitude of -10 colored by pressure)



T0

T1

T2

T3

T4

T5

Figure 5.21 Instantaneous vorticity magnitude distribution across propeller center plane

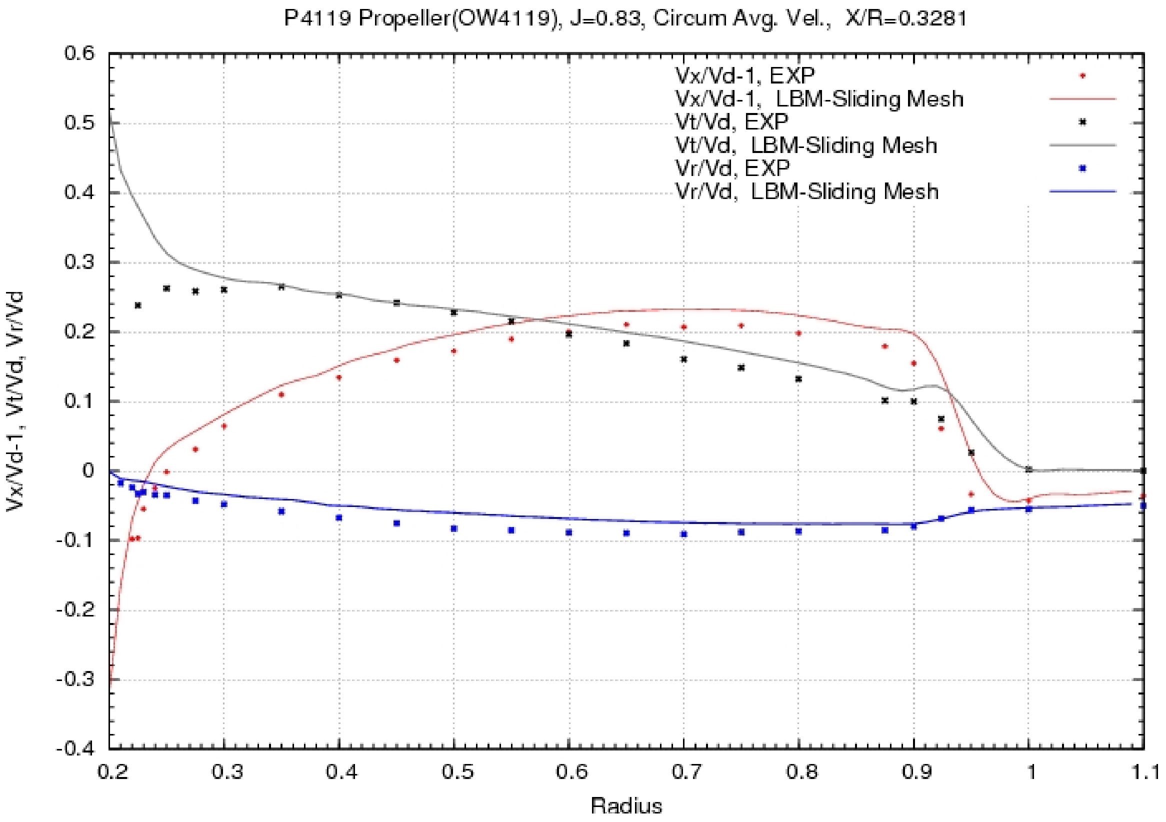
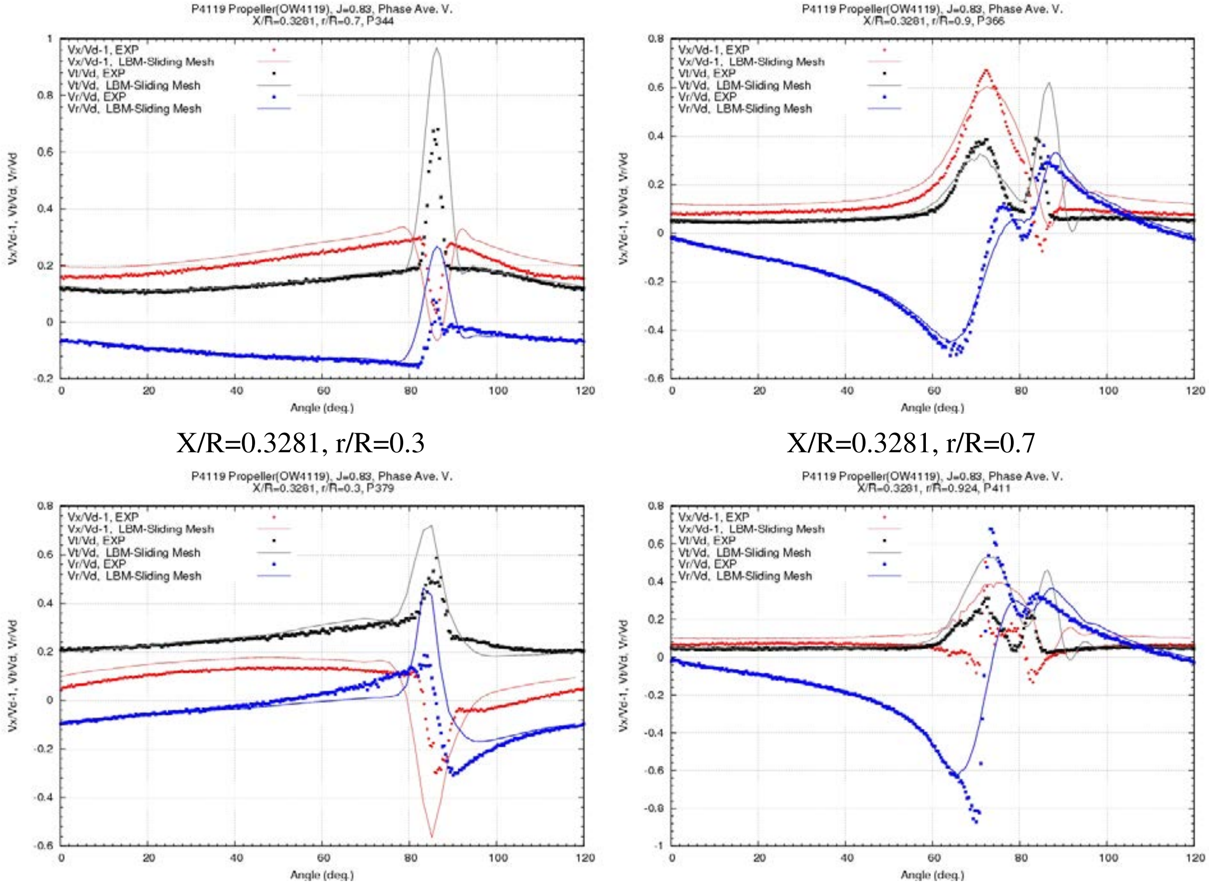


Figure 5.22 Comparison of circumferential averaged velocities at location of X/R=0.3281



X/R=0.3281, r/R=0.3

X/R=0.3281, r/R=0.7

X/R=0.3281, r/R=0.9

X/R=0.3281, r/R=0.924

Figure 5.23 Comparison of phase averaged velocities at location of X/R=0.3281

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

6.1 Major accomplishment

In this dissertation, several contributions have been made, including the improvement of a volumetric LBM boundary conditions and the formulation of a volumetric LBM sliding-mesh interface approach for performing sliding mesh simulations. The following is a summary of these accomplishments.

6.1.1 Development of an improved volumetric LBM boundary scheme

An improved volumetric LBM boundary scheme has been successfully developed for improving the solution accuracy of flow with an arbitrarily located solid wall. The proposed scheme abandons the assumption of homogeneous particles distributions for scattered backed particles in a boundary condition process, and introduces a scattering correction procedure to redistribute particles scattered back from the surface, the correction procedure incorporates local flow velocity variations to account for the inhomogeneity of the particle distribution within the affected volume, and does not compromise the local conservation of mass. It demonstrates close to second order accuracy for flows with arbitrarily located surface, and shows less solution sensitivity to actual lattice/wall configuration. A number of typical flow benchmark problems have been validated, the studies show that the current boundary approach can obtain accurate flow solutions on problems with curved boundary.

6.1.2 Development of a volumetric LBM sliding mesh interface approach for sliding mesh simulation

The proposed volumetric LBM boundary condition has been extended to enforce sliding-mesh interface condition with Cartesian grids, where a modified volumetric specular reflection boundary scheme is applied to realize the sliding mesh interface condition. The scheme applies two-sides averaged surfel velocity and density values to construct surfel equilibrium distribution and couples the flow solutions from both sides of the sliding mesh interface, the conservation of local mass is ensured by applying an appropriate mass correction in the scheme, and conservation of local momentum across the sliding-mesh interface is achieved by imposing zero viscosity for the modified specular reflection boundary scheme. An overall LBM sliding mesh procedure is presented to enable simulation of flow involving rotating geometries. A flow past a rotating propeller case has been simulated to demonstrate the current approach's capability and accuracy for performing sliding mesh simulation with true rotating geometry.

6.2 Recommendations for future work

The following topics are suggested for future work:

- Further algorithm refinement to improve solution dependence on lattice-wall configuration.

  It should be noted that with current improvement, there is still a visible solution dependence on lattice-wall configurations, particularly when comparing solutions from lattice-aligned configuration and non-lattice aligned configuration. To improve this, one of the possible ways is to include the flow non-equilibrium information into the scattering correction procedure, since current improvement is purely based on an equilibrium approximation of particle distribution within the scattered volume, including the flow non-equilibrium information is expected to give more accurate presentation of local particle distributions, thus may provide improved solution.

- Extension of current volumetric scattering correction idea to LBM multiphase wetting wall model.

  We have conducted some preliminary research in this multiphase area and found that the Shan-Chen type wall potential wetting model is also very sensitive to actual lattice-wall configuration. How to incorporate the ideas from current study to accurately model the wall wetting phenomena is needed.

- Extension of current sliding mesh interface algorithm to handle more general interface motion.

  The current LBM sliding mesh interface algorithms is presented for solving problems involving geometry with rigid body rotation, however, the fundamental ideas behind this numerical scheme is not limited to this constraint and can be applied to any interface problem that involves relative motions between flow solution domains. Developing suitable interface algorithm to handle more general interface motion is a nature extension of current work.

BIBLIOGRAPHY

Addy, H. E. J., Broeren, A. P., Zoeckler, J. G., and Lee, S. (2003). A wind tunnel study of icing effects on a business jet airfoil. Technical Report NASA/TM-2003-212124, National Aeronautics and Space Administration, Glenn Research Center, Cleveland, OH USA.

Aftosmis, M. J. (1997). Solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries. In *von Karman Institute for Fluid Dynamics*, Providence, Rhode Island. Lecture Series 1997-02.

Agarwal, R. (1999). Computational fluid dynamics of whole-body aircraft. *Annual Review of Fluid Mechanics*, 31:125–169.

Aidun, C. K. and Clausen, J. R. (2010). Lattice-Boltzmann method for complex flows. *Annual Review of Fluid Mechanics*, 42:439–472.

Anderson, A., Zheng, X., and Cristini, V. (2005). Adaptive unstructured volume remeshing-I: The method. *Journal of Computational Physics*, 208(2):616 – 625.

Bakker, A., Laroche, R. D., Wang, M. H., and Calabrese, R. V. (1997). Sliding mesh simulation of laminar flow in stirred reactors. *Chemical Engineering Research and Design*, 75(1):42–44.

Bhatnagar, P., Gross, E., and Krook, M. (1954). A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component system. *Physical Review*, 94:511–525.

Bird, G. A. (1994). *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford, Claredon.

Blades, E. L. and Marcum, D. L. (2007). A sliding interface method for unsteady unstructured flow simulations. *International Journal for Numerical Methods in Fluids*, 53:507–529.

Bode, J. (1994). Computational fluid dynamics applications in the chemical industry. *Computers & Chemical Engineering*, 18, Supplement 1(0):S247 – S251.

Bouard, R. and Coutanceau, M. (1980). The early stage of development of the wake behind an impulsively started cylinder for $40 < Re < 10^4$. *Journal of Fluid Mechanics*, 101:583–607.

Bouzidi, M., Firdaouss, M., and Lallemand, P. (2001). Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13(11):3452–23459.

Braza, M., Chassaing, P., and Minh, H. H. (1986). Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics*, 165:79–130.

Brucato, A., Ciofalo, M., Grisafi, F., and Rizzuti, L. (1990). Computer simulation of turbulent fluid flow in baffled and unbaffled tanks stirred by radial impellers. In *Proceeding of International Conference on Computer Applications to Batch Processes*, pages 69–86, Cengio, Italy. ed. V. G. Dovi.

Calhoun, D. (2002). A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. *Journal of Computational Physics*, 176(2):231 – 275.

Cercignani, C. (1989). Kinetic theory with “bounce-back” boundary conditions. *Transport Theory and Statistical Physics*, 18:125–131.

Chan, W. M. (2009). Overset grid technology development at NASA Ames Research Center. *Computers & Fluids*, 38(3):496 – 503.

Chan, W. M. and Buning, P. G. (1995). Surface grid generation methods for overset grids. *Computers & Fluids*, 24(5):509 – 522.

Chapman, S. and Cowling, T. G. (1970). *The Mathematical Theory of Non-Uniform Gases, 3rd edn.* Cambridge University Press., London.

Chen, H. (1998). Volumetric formulation of the lattice Boltzmann method for fluid dynamics: Basic concept. *Physical Review E*, 58:3955–3963.

Chen, H., Chen, S., and Matthaeus, W. H. (1992). Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. *Physical Review A*, 45:R5339–R5342.

Chen, H., Kandasamy, S., Orszag, S., Shock, R., Succi, S., and Yakhot, V. (2003). Extended Boltzmann kinetic equation for turbulent flows. *Science*, 301(5633):633–636.

Chen, H., Orszag, S. A., Staroselsky, I., and Succi, S. (2004). Expanded analogy between Boltzmann kinetic theory of fluids and turbulence. *Journal of Fluid Mechanics*, 519:301–314.

Chen, H., Teixeira, C., and Molvig, K. (1998). Realization of fluid boundary conditions via discrete Boltzmann dynamics. *International Journal of Modern Physics C*, 9(8):1281–1292.

Chen, S., Bao, S., Liu, Z., Li, J., Yi, C., and Zheng, C. (2010). A heuristic curved-boundary treatment in lattice Boltzmann method. *EPL (Europhysics Letters)*, 92(5):54003.

Chen, S., Chen, H., Martnez, D., and Matthaeus, W. (1991). Lattice Boltzmann model for simulation of magnetohydrodynamics. *Physical Review Letters*, 67:3776–3779.

Chen, S. and Doolen, G. D. (1998). Lattice Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30:329–364.

Chen, S., Martinez, D., and Mei, R. (1996). A non-slip boundary condition for lattice Boltzmann simulations. *Physics of Fluids*, 8:2527–2536.

Chi, X., Li, Y., Chen, H., Addy, H. E., Choo, Y., and Shih, T. (2005). A comparative study using CFD to predict iced airfoil aerodynamics. In *43rd Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA. Paper No. AIAA-2005-1371.

Chun, B. and Ladd, A. J. C. (2007). Interpolated boundary condition for lattice Boltzmann simulations of flows in narrow gaps. *Physical Review E*, 75:066705.

Clausen, J. R., Jr., D. A. R., and Aidun, C. K. (2010). Parallel performance of a lattice-Boltzmann/finite element cellular blood flow solver on the IBM Blue Gene/P architecture. *Computer Physics Communications*, 181(6):1013 – 1020.

Cornubert, R., d'Humières, D., and Levermore, D. (1991). A Knudsen layer theory for lattice gases. *Physica D: Nonlinear Phenomena*, 47(1-2):241 – 259.

Daskopoulos, P. and Harris, C. K. (1996). Three dimensional CFD simulations of turbulent flow in baffled stirred tanks: an assessment of the current position. *Institution of Chemical Engineers Symposium Series*, 140:1–14.

Derksen, J. (2001). Assessment of large eddy simulations for agitated flows. *Transactions of the Institution of Chemical Engineers (IChemE)*, 79(A):824–830.

Derksen, J., Harry, E. A., and den Akker, V. (1999). Large eddy simulations on the flow driven by a rushton turbine. *The American Institute of Chemical Engineers (AIChE) Journal*, 45(2):209–221.

Eggels, J. G. (1996). Direct and large-eddy simulation of turbulent fluid flow using the lattice-Boltzmann scheme. *International Journal of Heat and Fluid Flow*, 17(3):307 – 323.

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

Filippova, O. and Hã́nel, D. (1997). Lattice-Boltzmann simulation of gas-particle flow in filters. *Computers & Fluids*, 26(7):697 – 712.

Filippova, O. and Hã́nel, D. (1998). Grid refinement for lattice-BGK models. *Journal of Computational Physics*, 147(1):219 – 228.

Fox, R. (2008). A quadrature-based third-order moment method for dilute gas-particle flows. *Journal of Computational Physics*, 227(12):6313 – 6350.

Frisch, U., d'Humières, D., Hasslacher, B., Lallemand, P., Pomeau, Y., and Rivet, J. P. (1987). Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, 1:649–707.

Frisch, U., Hasslacher, B., and Pomeau, Y. (1986). Lattice-gas automata for the Navier-Stokes equations. *Physical review letters*, 56:1505–1508.

Gad-el Hak, M. (1999). The fluid mechanics of microdevices - the Freeman Scholar Lecture. *Journal of Fluids Engineering*, 121(1):5–33.

Gad-el Hak, M. (2006). *The MEMS Handbook (second edition)*. CRC Press/Taylor & Francis,, Boca Raton, FL.

Gallivan, M. A., Noble, D. R., Georgiadis, J. G., and Buckius, R. O. (1997). An evaluation of the bounce-back boundary condition for lattice Boltzmann simulations. *International Journal for Numerical Methods in Fluids*, 25(3):249–264.

Gindroz, B., Hoshino, T., Holtrop, J., Jessup, S. D., Ligtelijn, J. T., Mewis, F., Poustoshniy, A., Pylkkanen, J. V., and Wang, G.-Q. (1998). The propulsion committee: Final report and recommendations to the 22nd ITTC. In *22nd ITTC Propulsion Committee Propeller RANS/Panel Method Workshop Proceedings*, Grenoble, France.

Ginzbourg, I. and Adler, P. M. (1994). Boundary flow condition analysis for the three-dimensional lattice Boltzmann model. *Journal De Physique II*, 75(2):191–214.

Ginzburg, I. and d'Humières, D. (2003). Multireflection boundary conditions for lattice Boltzmann models. *Physical Review E*, 68:066614.

Gosman, A. D., Lekakou, C., Politis, S., Issa, R. I., and Looney, M. K. (1992). Multidimensional modeling of turbulent two-phase flows in stirred vessels. *The American Institute of Chemical Engineers (AIChE) Journal*, 38:1946–1956.

Gundy-Burlet, K. (1992). Unsteady two- and three-dimensional Navier-Stokes simulations of multistage turbomachinery flows. *Computing Systems in Engineering*, 3(1-4):231 – 240.

Guo, Z., Zheng, C., and Shi, B. (2002a). Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E*, 65(4):046308.

Guo, Z., Zheng, C., and Shi, B. (2002b). An extrapolation method for boundary conditions in lattice Boltzmann method. *Physics of Fluids*, 14(6):2007–2010.

Guo, Z., Zheng, C., and Shi, B. (2002c). Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method. *Chinese Physics*, 11(4):366–374.

Hartmann, H., Derksen, J., Montavon, C., Pearson, J., Hamill, I., and van den Akker, H. (2004). Assessment of large eddy and RANS stirred tank simulations by means of LDA. *Chemical Engineering Science*, 59(12):2419 – 2432.

Harvey, P. S. and Greaves, M. (1982). Turbulent flow in an agitated vessel, Part I: A predictive model. *Chemical Engineering Research and Design*, 60(a):195–200.

He, X., Chen, S., and Doolen, G. D. (1998). A novel thermal model for the lattice Boltzmann method in incompressible limit. *Journal of Computational Physics*, 146(1):282 – 300.

He, X. and Doolen, G. D. (1997). Lattice Boltzmann method on a curvilinear coordinate system: Vortex shedding behind a circular cylinder. *Physical Review E*, 56:434–440.

Henderson, R. D. (1995). Details of the drag curve near the onset of vortex shedding. *Physics of Fluids*, 7(9):2102–2104.

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

Jameson, A. (1999). Re-engineering the design process through computation (SP). *Journal of Aircraft*, 36(1):36 –50.

Jeong, J. and Hussain, F. (1995). On the identication of a vortex. *Journal of Fluid Mechanics*, 285:69–94.

Jessup, S. (1989). *An Experimental Investigation of Viscous Effects of Propeller Blade Flow*. PhD thesis, The Catholic University of America.

Jessup, S. D. (1998). Experimental data for RANS calculations and comparisons (DTMB p4119). In *22nd ITTC Propulsion Committee Propeller RANS/Panel Method Workshop Proceedings*, Grenoble, France.

Johnson, F. T., Tinoco, E. N., and Yu, N. J. (2005). Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle. *Computers & Fluids*, 34(10):1115 – 1151.

Johnson, P. L., Jones, K. M., and Madson, M. D. (2000). Experimental investigation of a simplified 3D high lift configuration in support of CFD validation. In *18th AIAA Applied Aerodynamics Conference*, Denver, CO, USA. Paper No. AIAA-2000-4217.

Kandhai, D., Koponen, A., Hoekstra, A., Kataja, M., Timonen, J., and Sloot, P. (1999). Implementation aspects of 3D lattice-BGK: Boundaries, accuracy, and a new fast relaxation method. *Journal of Computational Physics*, 150(2):482 – 501.

Kao, P.-H. and Yang, R.-J. (2008). An investigation into curved and moving boundary treatments in the lattice Boltzmann method. *Journal of Computational Physics*, 227(11):5671 – 5690.

Katz, J. (2006). Aerodynamics of race cars. *Annual Review of Fluid Mechanics*, 38:27–63.

Khorrami, M. R., Berkman, M. E., Li, F., and Singer, B. A. (2002). Computational simulations of a three-dimensional high-lift wing. In *20th AIAA Applied Aerodynamics Conference*, St. Louis, MO, USA. Paper No. AIAA 2002-2804.

Koumoutsakos, P. and Leonard, A. (1995). High resolution simulations of the flow around and impulsively started cylinder using vortex methods. *Journal of Fluid Mechanics*, 296:1–38.

Ladd, A. J. C. (1994). Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. numerical results. *Journal of Fluid Mechanics*, 271:311–339.

Ladd, A. J. C. and Verberg, R. (2001). Lattice-Boltzmann simulations of particle-fluid suspensions. *Journal of Statistical Physics*, 104(5-6):1191–1251.

Lallemand, P. and Luo, L.-S. (2003). Lattice Boltzmann method for moving boundaries. *Journal of Computational Physics*, 184(2):406 – 421.

Lavallee, P., Boon, J. P., and Noullez, A. (1991). Boundaries in lattice gas flows. *Physica D: Nonlinear Phenomena*, 47(1-2):233 – 240.

Lee, J. and Lee, S. (2010). Boundary treatment for the lattice Boltzmann method using adaptive relaxation times. *Computers & Fluids*, 39(5):900 – 909.

Li, Y., Shock, R., Zhang, R., and Chen, H. (2004). Numerical study of flow past an impulsively started cylinder by the lattice-Boltzmann method. *Journal of Fluid Mechanics*, 519:273–300.

Liu, C., Zheng, X., and Sung, C. (1998). Preconditioned multigrid methods for unsteady incompressible flows. *Journal of Computational Physics*, 139(1):35 – 57.

Loc, T. (1980). Numerical analysis of unsteady secondary vortices generated by an impulsively started cylinder. *Journal of Fluid Mechanics*, 100:111–128.

Lu, Z., Liao, Y., Qian, D., McLaughlin, J., Derksen, J., and Kontomaris, K. (2002). Large eddy simulations of a stirred tank using the lattice Boltzmann method on a nonuniform grid. *Journal of Computational Physics*, 181(2):675 – 704.

Luo, J. Y., Gosman, A. D., Issa, R. I., C., M. J., and K., F. M. (1993). Full flow field computation of mixing in baffled stirred vessels. *Transactions of the Institution of Chemical Engineers (IChemE)*, 71(A):342–344.

Mavriplis, D. J. (1997). Unstructured grid techniques. *Annual Review of Fluid Mechanics*, 29(7):473 – 514.

Maxwell, C. J. (1879). On stresses in rarified gases arising from inequalities of temperature. *Philosophical Transactions of the Royal Society of London*, 170:231–256.

McCorkle, D., Bryden, K., and Carmichael, C. (2003). A new methodology for evolutionary optimization of energy systems. *Computer Methods in Applied Mechanics and Engineering*, 192(44-46):5021 – 5036.

McGinley, C. B., Jenkins, L. N., Watson, R. D., and Bertelrud, A. (2005). 3-D high-lift flow-physics experiment transition measurements. In *35th AIAA Fluid Dynamics Conference and Exhibit*, Toronto, Ontario, Canada. Paper No. AIAA 2005-5148.

Meakin, R. L. (2001). Object X-rays for cutting holes in composite overset structured grids. In *15th AIAA Computational Fluid Dynamics Conference*, pages 1–8, Anaheim, CA, USA. Paper AIAA 2001-2537.

Mei, R., Luo, L.-S., and Shyy, W. (1999). An accurate curved boundary treatment in the lattice Boltzmann method. *Journal of Computational Physics*, 155(2):307 – 330.

Murthy, J. Y., Mathur, S. R., and Choudhury, D. (1994). CFD simulation of flows in stirred tank reactors using a sliding mesh technique. *Institution of Chemical Engineers Symposium Series*, 136:341–348.

Noble, D. R., Chen, S., Georgiadis, J. G., and Buckius, R. O. (1995). A consistent hydrodynamic boundary condition for the lattice Boltzmann method. *Physics of Fluids*, 7:203–209.

Passalacqua, A. and Fox, R. (2011). Advanced continuum modelling of gas-particle flows beyond the hydrodynamic limit. *Applied Mathematical Modelling*, 35(4):1616 – 1627.

Peng, Y. (2005). *Thermal Lattice Boltzmann Two-Phase Flow Model for Fluid Dynamics*. PhD thesis, University of Pittsburgh, Pittsburgh, PA, USA.

Pericleous, K. A. and Patel, M. K. (1986). The modeling of tangential and axial agitators in chemical reactors. *PhysicoChemical Hydrodynamics*, 8:105–123.

Perng, C. Y. and Murthy, J. Y. (1993). A moving-deforming-mesh technique for simulation of flow in mixing tanks. *The American Institute of Chemical Engineers Symposium Series*, 293:37–41.

Pervaiz, M. M. and Teixeira, C. M. (1999). Two equation turbulence modeling with the lattice Boltzmann method. In *2nd International Symposium on Computer Technology For Fluid/Thermal/Chemical Systems with Industrial Applications. ASME PVP Division Conference*, Boston, MA, USA.

Peskin, C. S. (2002). The immersed boundary method. *Acta Numerica*, 11:479 – 517.

Placek, J. and Tavlarides, L. L. (1985). Turbulent flow in stirred tanks. Part I: Turbulent flow in the turbine impeller region. *The American Institute of Chemical Engineers (AIChE) Journal*, 31:1113–1120.

Pohl, T. K., Wilke, M., Iglberger, J., and K. Rude, U. (2003). Optimization and profiling of the cache performance of parallel lattice Boltzmann codes. *Parallel Processing Letters*, 13(4):549–560.

PowerFLOW Training Notes (2008). Powerflow training notes.

Qian, Y., d'Humières, D., and Lallemand, P. (1992). Lattice BGK models for Navier-Stokes equation. *Europhysics Letters*, 17(6):479 – 484.

Qu, K., Shu, C., and Chew, Y. (2010). Lattice Boltzmann and finite volume simulation of inviscid compressible flows with curved boundary. *Advances in Applied Mathematics and Mechanics*, 2(5):573–586.

Rai, M. M. (1984). A conservative treatment of zonal boundaries for Euler equation calculations. In *AIAA 22nd Aerospace Sciences Meeting*, Reno, NV, USA. AIAA Paper 1984-0164.

Rai, M. M. (1985). A conservative treatment of zonal boundaries for Euler equation calculations. In *AIAA 23rd Aerospace Sciences Meeting*, Reno, NV, USA. AIAA Paper 1985-0488.

Rai, M. M. (1986). Patched-grid calculations with the Euler and Navier-Stokes equations: Theory and applications. Technical Report NASA/TM-88228, National Aeronautics and Space Administration, Ames Research Center, Moffett Field, CA, USA.

Rai, M. M. (1989a). Three-dimensional Navier-Stokes simulations of turbine rotor-stator interaction; Part I - methodology. *Journal of Propulsion and Power*, 5(3):305–311.

Rai, M. M. (1989b). Three-dimensional Navier-Stokes simulations of turbine rotor-stator interaction; Part II - results. *Journal of Propulsion and Power*, 5(3):312–319.

Rai, M. M. and Madavan, N. K. (1990). Multi-airfoil Navier-Stokes simulations of turbine rotor-stator interaction. *Journal of Turbomachinery*, 112(3):377–384.

Revstedt, J., Fuchs, L., Kovcs, T., and Trgrdh, C. (2000). Influence of impeller type on the flow structure in a stirred reactor. *The American Institute of Chemical Engineers (AIChE) Journal*, 46(2):2373–2382.

Rohde, M. (2004). *Extending the lattice-Boltzmann method: novel techniques for local grid refinement and boundary conditions*. PhD thesis, Delft University of Technology.

Rohde, M., Derksen, J. J., and Van den Akker, H. E. A. (2002). Volumetric method for calculating the flow around moving objects in lattice-Boltzmann schemes. *Physical Review E*, 65:056701.

Schetz, J. A. (2001). Aerodynamics of high-speed trains. *Annual Review of Fluid Mechanics*, 33:371–414.

Schnute, J. and Shinbrot, M. (1973). Kinetic theory and boundary conditions for fluids. *Canadian Journal of Mathematics*, 25(6):1183–1215.

Shan, X. and Chen, H. (1993). Lattice Boltzmann model for simulating flows with multiple phases and components. *Physical Review E*, 47:1815–1819.

Shan, X. and Chen, H. (1994). Simulation of nonideal gases and liquid-gas phase transitions by the lattice Boltzmann equation. *Physical Review E*, 49:2941–2948.

Shan, X., Yuan, X., and Chen, H. (2006). Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation. *Journal of Fluid Mechanics*, 550:413–441.

Shih, T. I.-P. (2008). CFD analysis of turbine cooling (invited keynote). In *19th National & 8th ISHMT-ASME Heat and Mass Transfer Conference*, JNTU Hyderabad, India. Paper No. US-22.

Skordos, P. A. (1993). Initial and boundary conditions for the lattice Boltzmann method. *Physical Review E*, 48:4823–4842.

Spalart, P. R. and Allmaras, S. R. (1994). A one-equation turbulence model for aerodynamic flows. *La Recherche Aerospatiale*, 1(1):5–21.

Steijl, R. and Barakos, G. (2008). Sliding mesh algorithm for CFD analysis of helicopter rotor-fuselage aerodynamics. *International Journal for Numerical Methods in Fluids*, 58:527–549.

Succi, S. (2001). *The lattice Boltzmann equation for fluids dynamics and beyond*. Oxford, Clarendon.

Sun, C., Zhang, R., and Chen, H. (2009). A PDE sliding mesh algorithm for turbulent flow simulations with PowerFLOW. Technical report, EXA Corporation.

Surmas, R., dos Santos, L. O., and Philippi, P. C. (2004). Lattice Boltzmann simulation of the flow interference in bluff body wakes. *Future Generation Computer Systems*, 20(6):951 – 958.

Tabor, G., Gosman, A. D., and Issa, R. I. (1996). Numerical simulation of the flow in a mixing vessel stirred by a rushton turbine. *Institution of Chemical Engineers Symposium Series*, 140:25–34.

Teixeira, C. M. (1998). Incorporating turbulence models into the lattice-Boltzmann method. *International journal of modern physics C*, 9(8):1159–1176.

Velivelli, A. C. and Bryden, K. M. (2004). A cache-efficient implementation of the lattice Boltzmann method for the two-dimensional diffusion equation. *Concurrency and Computation: Practice and Experience*, 16(14):1415–1432.

Verberg, R. and Ladd, A. J. C. (2000). Lattice-Boltzmann model with sub-grid-scale boundary conditions. *Physical Review Letters*, 84:2148–2151.

Wang, Z. (2007). High-order methods for the Euler and Navier-Stokes equations on unstructured grids. *Progress in Aerospace Sciences*, 43(1-3):1 – 41.

Wang, Z. and Parthasarathy, V. (2000). A fully automated chimera methodology for multiple moving body problems. *International Journal for Numerical Methods in Fluids*, 33(7):919 – 938.

Wang, Z. J. and Srinivasan, K. (2002). An adaptive Cartesian grid generation method for Dirty geometry. *International Journal for Numerical Methods in Fluids*, 39(8):703 – 717.

Whitfield, D. L., Swafford, T. W., Mulac, R. A., Belk, D. M., and Janus, J. M. (1987). Three-dimensional unsteady euler solutions for propfans and counter-rotating propfans in transonic flow. In *AIAA 19th Fluid Dynamics, Plasma Dynamics, and Lasers Conference*, Honolulu, HI, USA.

Williamson, C. H. K. (1989). Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, 206:579–627.

Williamson, C. H. K. and Roshko, A. (1990). Measurements of base pressure in the wake of a cylinder at low Reynolds numbers. *Zeitschrift fuer Flugwissenschaften und Weltraumforschung*, 14(1-2):38–46.

Wolfram, S. (1986). Cellular automaton fluids 1: Basic theory. *Journal of Statistical Physics*, 45(3-4):471–526.

Yakhot, V. and Orszag, S. A. (1986). Renormalization group analysis of turbulence. I. Basic theory. *Journal of Scientific Computing*, 1(1):3–51.

Yakhot, V. and Smith, L. M. (1992). The renormalization group, the ϵ-expansion and derivation of turbulence models. *Journal of Scientific Computing*, 7(1):35–61.

Yu, D., Mei, R., Luo, L.-S., and Shyy, W. (2003). Viscous flow computations with the method of lattice Boltzmann equation. *Progress in Aerospace Sciences*, 39(5):329 – 367.

Ziegler, D. P. (1993). Boundary conditions for lattice Boltzmann simulations. *Journal of Statistical Physics*, 71:1171–1177. 10.1007/BF01049965.

Zitney, S. E. and Syamlal, M. (2002). Integrated process simulation and CFD for improved process engineering. In Grievink, J. and van Schijndel, J., editors, *European Symposium on Computer Aided Process Engineering-12 35th European Symposium of the Working Party on Computer Aided Process Engineering*, volume 10 of *Computer Aided Chemical Engineering*, pages 397 – 402. Elsevier.