

IMPLICIT LATTICE BOLTZMANN METHOD FOR LAMINAR/TURBULENT
FLOWS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FATİH ÇEVİK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

MAY 2016

Approval of the thesis:

IMPLICIT LATTICE BOLTZMANN METHOD FOR LAMINAR/TURBULENT FLOWS

submitted by **FATİH ÇEVİK** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Tuna Balkan
Head of Department, **Mechanical Engineering**

Prof. Dr. Kahraman Albayrak
Supervisor, **Mechanical Engineering Department, METU**

Examining Committee Members:

Prof. Dr. M. Haluk Aksel
Mechanical Engineering Department, METU

Prof. Dr. Kahraman Albayrak
Mechanical Engineering Department, METU

Prof. Dr. Nafiz Alemdaroğlu
Airframe and Powerplant Maintenance Department, AU

Assoc. Prof. Dr. Selin Aradağ Çelebioğlu
Mechanical Engineering Department, TOBB ETU

Assist. Prof. Dr. Nilay Sezer Uzol
Aerospace Engineering Department, METU

Date:

30.05.2016

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: FATİH ÇEVİK

Signature :

ABSTRACT

IMPLICIT LATTICE BOLTZMANN METHOD FOR LAMINAR/TURBULENT FLOWS

Çevik, Fatih

Ph.D., Department of Mechanical Engineering

Supervisor : Prof. Dr. Kahraman Albayrak

May 2016, 182 pages

Lattice Boltzmann Method is an alternative computational method for fluid physics problems. The development of the method started in the late 1980s and early 1990s. Various numerical schemes like stream and collide, finite difference, finite element and finite volume schemes are used to solve the discrete Lattice Boltzmann Equation. Almost all of the numerical schemes in the literature are explicit schemes to exploit the natural features of the discrete Lattice Boltzmann Equation like parallelism and easy coding.

In this thesis, an Implicit Finite Volume Lattice Boltzmann Method (*IFVLBM*) is developed. The method is limited for the incompressible fluid simulation, however loosely coupled Spalart-Allmaras turbulence model is incorporated for the simulations for high Reynolds numbers. Moreover, local time stepping techniques and dual time stepping techniques are also implemented for convergence acceleration to use in steady state and unsteady problems respectively. The IFVLBM demonstrates improvements in stability characteristics and convergence is accelerated as the limitation of CFL number is eased compared to the classical Lattice Boltzmann Methods.

The test case results for laminar, turbulent, steady and unsteady flows are compared with either experimental or numerical data in the literature. Also, numerical data available in the literature from the CFL3D software, which is a Reynolds averaged

Navier Stokes solver developed by NASA, is used for flow field comparisons. The results of the developed method are in good agreement with the data given in the literature.

Keywords: Finite Volume Method, Lattice Boltzmann Equation, Implicit Methods, Turbulent Flows

ÖZ

LAMİNER/TÜRBÜLANSLI AKIŞLAR İÇİN ÖRTÜK KAFES BOLTZMANN YÖNTEMİ

Çevik, Fatih

Doktora, Makina Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. Kahraman Albayrak

Mayıs 2016 , 182 sayfa

Kafes Boltzmann yöntemi, akışkan fiziği problemler için bir alternatif bir sayısal yöntemdir. 1980'lerin sonu ve 1990'ların başında yöntemin geliştirilmesine başlanmıştır. Akma ve çarpışma, sonlu farklar, sonlu elemanlar ve sonlu hacimler yöntemi gibi değişik nümerik yöntemler, ayrık kafes Boltzmann denkleminin çözümü için kullanılmıştır. Literatürdeki nümerik yöntemlerin neredeyse tamamı, ayrık kafes Boltzmann denkleminin paralelleştirilebilme ve kolay kodlama gibi doğal özelliklerinden yararlanmak için açık yöntemdir.

Bu tezde, bir Örtük Sonlu Hacimler Kafes Boltzmann Yöntemi (*ÖSHKBY*) geliştirilmiştir. Yöntem, sıkıştırılamaz akışkanlar ile sınırlıdır, ancak yüksek Reynolds sayısı simülasyonları için Spalart Allmaras türbülans modeli yönteme dahil edilmiştir. Dahası bölgesel zaman adımı tekniği ve ikili zaman adımı tekniği, yakınsamanın hızlandırılması için sırası ile durağan ve kararsız problemlerde kullanılmak üzere uygulanmıştır. *ÖSHLBY*, kararlılık karakteristiğinde gelişim göstermiş ve klasik kafes Boltzmann yönteminde kullanılan CFL sayısı limitlerinin rahatlatılmasından dolayı yakınsama hızı artmıştır.

Laminer, türbülanslı, düzenli ve düzensiz akışlar için yapılan çalışmalar literatürdeki nümerik ve deney verileri ile karşılaştırılmıştır. Ayrıca, NASA tarafından geliştirilen Reynolds ortalamalı Navier Stokes çözücü olan CFL3D yazılımının, literatürde

açık bulunan test çözümleri akış alanı karşılaştırmaları için kullanılmıştır. Geliştirilen yöntemin sonuçları literatürdeki veriler ile uyum içindedir.

Anahtar Kelimeler: Sonlu Hacimler Yöntemi, Kafes Boltzmann Denklemi, Örtük Yöntemler, Türbülanslı Akışlar

*To the ladies of my life, my wife Işın
my daughters İlke and İlgi*

ACKNOWLEDGMENTS

I would like to thank my supervisor Professor Kahraman Albayrak for accepting me as a Ph.D. student and giving me the chance to work with him. His positive attitude during these years of study was always the encouraging and motivating.

I also want to thank my thesis review committee members Professor M. Haluk Aksel and Professor Nafiz Alemdarođlu. Their support and suggestion during the thesis reviews has guided me during the process not only in scientific context but also in my working environment.

I want to thank to the instructors of the courses that I attended, Professor Hafit Yüncü, Professor Zafer Dursunkaya, Professor Faruk Arınç, Associate Professor Tuba Okutucu Özyurt, Associate Professor Dilek Funda Kurtuluş, Professor İ. Sinan Akmandor and finally my supervisor Professor Kahraman Albayrak. During the doctoral process, the the courses helped me refresh my past background and build a solid fundamental that allowed me to overcome the problems that I have faced.

I will like to thank my colleagues Dr. Bedri Yađız and Mr. Sinan Pakkan for their friendship and suggestions about my thesis subject and typesetting tools.

I have a special thanks to my colleague in ASELSAN, Mr. Göktan Güzel, a brilliant engineer who has a great knowledge in Computational Fluid Dynamics subject and coding in FORTRAN. His suggestions on my thesis and help in finding the test cases are appreciated. Our discussions on LBM was the most fruitful for me. It was a privilege to know him.

I also want to thank to ASELSAN A.Ş. the company I've been proud to be an employee since 2010. The support that ASELSAN A.Ş. gave to the employees is encouraging and an excellent opportunity for continuing education.

I thank to my brother, sister, mom and dad for their prayers. I want to thank to Dr. Hüdai Özdamar, Mrs. Berna Özdamar, Dr. Belgin Ünal whom I consider as family for their support and being always reachable for me and my family.

I want to thank to my loving wife Işın Ünal Çevik MD,Ph.D., a Neurology Professor in Hacettepe University, for her patience and encouragement during these years and believing in me.

Finally I thank to my daughters İlke Çevik, İlgi Gülten Çevik and nephew Kuzey Özdamar, making the life beautiful and worth living for.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xvi
LIST OF FIGURES	xviii
LIST OF ABBREVIATIONS & ACRONYMS	xxiii
LIST OF SYMBOLS	xxv
CHAPTERS	
1 INTRODUCTION	1
2 LATTICE BOLTZMANN METHOD	5
2.1 Introduction	5
2.2 Kinetic Theory	7
2.2.1 Particle Dynamics	7
2.2.2 Maxwell - Boltzmann Distribution Function	8
2.3 Boltzmann Transport Equation	10

2.3.1	Bhatnagar, Gross, Krook, BGK Approximation . . .	13
2.3.2	Equilibrium Distribution Function	14
2.4	Solution Methods	14
2.4.1	Stream and Collide Method	18
2.5	Boundary Conditions	21
2.6	Turbulence Models	22
2.6.1	Classes of Turbulence Models	25
2.6.2	Turbulence Equations	25
2.6.3	Eddy Viscosity Hypothesis	28
3	DEVELOPMENT OF IFVLBM	29
3.1	Complete Set of Equations	30
3.2	Non Dimensionalization	31
3.3	Finite Volume Formulation	34
3.3.1	MUSCLE Scheme for 2D and 3D faces	36
3.4	Boundary Conditions	41
3.4.1	Velocity Inlet Boundary Condition	42
3.4.2	Pressure Outlet Boundary Condition	44
3.4.3	Wall / Moving Wall Boundary Condition	45
3.5	Implicit Formulation for Steady State	46
3.6	Local Time Stepping for Acceleration	51

3.7	Dual Time Stepping for Transient/Unsteady Problems	51
3.8	Turbulence Modeling	53
3.9	Approximate Factorization	55
4	2D MODEL VALIDATION	57
4.1	Laminar Steady Flow	59
4.1.1	Lid Driven Cavity Flow	59
4.1.2	Steady Laminar Flow Over a Flat Plate	66
4.1.3	Steady Laminar Flow Over a Cylinder	73
4.2	Turbulent Flow	77
4.2.1	Turbulent Flow Over Flat Plate	77
4.2.2	Flow Over NACA0015 Airfoil	81
4.3	Unsteady Flow	86
4.3.1	Flow Past a Circular Cylinder	86
4.4	Grid Convergence	91
4.4.1	Cylinder Grid	92
4.4.2	NACA0015 Grid	93
5	3D MODEL VALIDATION & MODEL ASSESSMENT	95
5.1	Flows in 3D	97
5.1.1	Flow over a Infinite AR NACA0015 3D Wing Section	97
5.1.2	Laminar Flow over a Infinite AR Confined Cylinder	103
5.1.3	Uniform Flow Past a Sphere	109

5.1.4	Flow over Delta Wing	114
5.2	Model Assessment	123
5.2.1	Spatial Accuracy	123
5.2.2	Temporal Accuracy	125
5.2.3	Convergence Acceleration	126
5.2.4	Computational Performance	127
6	RESULTS & DISCUSSION	131
6.1	Summary	131
6.2	Discussion of the Results	132
6.3	Future Work	137
6.4	Conclusion	138
	REFERENCES	139
	APPENDICES	
A	INCOMPRESSIBLE NS AND LBE	147
A.1	From LBE to Incompressible NSE	147
A.1.1	Chapman Enskog Expansion	147
A.1.2	Derivation of NS from LBE	151
B	LATTICE BOLTZMANN STENCILS	155
B.1	D1Q3 Model	156
B.2	D1Q5 Model	156
B.3	D2Q5 Model	157

B.4	D2Q9 Model	158
B.5	D3Q15 Model	159
B.6	D3Q19 Model	160
B.7	D3Q27 Model	161
C	THOMAS ALGORITHM	163
C.1	Tridiagonal System Solution	163
C.2	Block Tridiagonal System Solution	165
D	APPROXIMATE FACTORIZATION METHOD	167
D.1	General Convection Equation	167
D.2	Alternating Direction Implicit Method	168
D.3	Diagonally Dominant Alternating Direction Implicit Method	170
D.4	Lower Upper Symmetric Gauss-Seidel Method	171
E	CONTROL VOLUME AND GEOMETRIC QUANTITIES	173
E.1	2D Control Volume	173
E.2	3D Control Volume	175
	CURRICULUM VITAE	181

LIST OF TABLES

TABLES

Table 3.1	Dimensional and non-dimensional forms	32
Table 3.2	Spalarat Allmaras one equation model constants	54
Table 4.1	Comparison of center coordinates of eddies	65
Table 4.2	Boundary layer parameters for $Re=10000, 100000$	72
Table 4.3	Characteristic parameters for flow over a cylinder	76
Table 4.4	Unsteady results comparison table for $Re = 100$	89
Table 4.5	Grid dependency for cylinder	92
Table 4.6	Grid dependency for NACA0015	93
Table 5.1	Coefficient comparison for 2D and 3D solution	98
Table 5.2	Confined Cylinder Eddies	105
Table 5.3	Grid dependency for 3D sphere	109
Table 5.4	Comparison of computed data with literature data for $Re = 100$. . .	113
Table 5.5	Static aerodynamic coefficients for CFL3D and IFVLBM	116
Table 5.6	Pressure coefficient center coordinates	120
Table 5.7	Temporal accuracy simulation parameters	125
Table 5.8	Temporal accuracy simulation results	125
Table 5.9	Solution domains for the validation cases	128
Table 5.10	Computational performance data for validation cases	129
Table B.1	Parameters of D1Q3 model	156

Table B.2	Parameters of D1Q5 model	156
Table B.3	Parameters of D2Q5 model	157
Table B.4	Parameters of some D2Q9 model	158
Table B.5	Parameters of some D3Q15 model	159
Table B.6	Parameters of some D3Q19 model	160
Table B.7	Parameters of some D3Q27 model	161

LIST OF FIGURES

FIGURES

Figure 1.1	Fluid model, geometric scales and numerical approaches	3
Figure 2.1	Knudsen number limits on some mathematical models	5
Figure 2.2	Probability density distributions of some noble gases	10
Figure 2.3	Position and velocity change of a particle	11
Figure 2.4	Stream and collide process	19
Figure 2.5	Turbulence model hierarchy according to complexity level (Level 0 - most complex)	26
Figure 3.1	2D Quadrilateral control volume elements	35
Figure 3.2	3D hexagonal control volume	38
Figure 3.3	Ghost cell implementation	41
Figure 3.4	Velocity inlet boundary condition	43
Figure 3.5	Pressure outlet boundary condition	44
Figure 3.6	Wall / moving wall boundary condition	45
Figure 3.7	Implicit operator matrix	50
Figure 4.1	Simple algorithm for the codes	58
Figure 4.2	Case setup for cavity flow	59
Figure 4.3	Grids for 129 x 129 and 257 x 257 (Every 10 th gridline is shown)	60
Figure 4.4	Streamlines inside the cavity by IFVLBM $Re = 400$	61
Figure 4.5	Streamlines IFVLBM and Ghia et.al $Re = 1000$	61

Figure 4.6 Streamlines IFVLBM and Ghia et.al $Re = 3200$	62
Figure 4.7 Streamlines IFVLBM and Ghia et.al $Re = 5000$	62
Figure 4.8 Horizontal and vertical velocity distribution at vertical and horizontal geometric center-line	63
Figure 4.9 Horizontal and vertical velocity distribution at vertical and horizontal geometric center-line	63
Figure 4.10 Horizontal and vertical velocity distribution at vertical and horizontal geometric center-line	64
Figure 4.11 Horizontal and vertical velocity distribution at vertical and horizontal geometric center-line	64
Figure 4.12 Flat plate problem setup	66
Figure 4.13 Flat plate 273×193 grid	67
Figure 4.14 Local friction coefficients C_f along the flat plate	68
Figure 4.15 Velocity distributions in the boundary layer for $Re = 10000$	69
Figure 4.16 Velocity distributions in the boundary layer for $Re = 100000$	70
Figure 4.17 Velocity distributions in the boundary layer for $Re = 10000$ and 100000	71
Figure 4.18 Cylinder problem setup	73
Figure 4.19 Solution domain for steady cylinder problems.	74
Figure 4.20 Eddy formations and separation streamline for the flow over cylinder for Reynolds numbers $Re = 10$	75
Figure 4.21 Eddy formations and separation streamline for the flow over cylinder for Reynolds numbers $Re = 20$	75
Figure 4.22 Eddy formations and separation streamline for the flow over cylinder for Reynolds numbers $Re = 40$	76
Figure 4.23 Flat plate problem setup	77
Figure 4.24 Velocity distributions in the boundary layer at various locations	78
Figure 4.25 Contours of the ratio of turbulent viscosity to dynamic viscosity ratio	79
Figure 4.26 Ratio of turbulent viscosity to laminar viscosity at $x = 0.97008$	80

Figure 4.27 C_f values along the flat plate	80
Figure 4.28 NACA0015 problem setup	81
Figure 4.29 NACA0015 Solution domain	82
Figure 4.30 CM vs α of NACA0015	82
Figure 4.31 CL vs α of NACA0015	83
Figure 4.32 CD vs α of NACA0015	83
Figure 4.33 TE separation of NACA0015 airfoil $AoA = 10.0$ degrees	84
Figure 4.34 TE separation of NACA0015 airfoil $AoA = 12.5$ degrees	84
Figure 4.35 TE separation of NACA0015 airfoil $AoA = 15.0$ degrees	85
Figure 4.36 TE separation of NACA0015 airfoil $AoA = 18.0$ degrees	85
Figure 4.37 Velocity distribution on the surface and vicinity of airfoil at $AoA =$ 18	86
Figure 4.38 Cylinder problem setup	87
Figure 4.39 Drag coefficient change wrt iteration	87
Figure 4.40 Lift coefficient change wrt iteration	88
Figure 4.41 Vorticity contours for $Re = 100$	90
Figure 4.42 Asymptotic behavior of C_d for cylinder grids	92
Figure 4.43 Asymptotic behavior of C_l for NACA0015 grids	93
Figure 5.1 Problem setup for NACA0015	97
Figure 5.2 3D NACA0015 airfoil sections	98
Figure 5.3 Residuals for NACA0015 solution	98
Figure 5.4 Pressure contours for NACA0015 solutions	99
Figure 5.5 Velocity contours for NACA0015 solutions	100
Figure 5.6 Streamlines for NACA0015 solutions	101
Figure 5.7 Pressure coefficient over 2D and 3D airfoil sections	102
Figure 5.8 Problem setup for confined cylinder	103

Figure 5.9 3D problem setup for confined cylinder	104
Figure 5.10 Residuals for confined cylinder solution	104
Figure 5.11 Confined cylinder pressure coefficients	106
Figure 5.12 Confined cylinder velocity contours	107
Figure 5.13 Confined cylinder Streamlines	108
Figure 5.14 3D problem setup for flow past a sphere	109
Figure 5.15 Sphere solution domain and surface meshes	110
Figure 5.16 Drag coefficients of IFVLBM and literature data	111
Figure 5.17 Pressure coefficient around the sphere at $z=0$ plane	111
Figure 5.18 360° toroidal vortex $Re = 100$	112
Figure 5.19 A section of toroidal vortex $Re = 100$	112
Figure 5.20 Velocity contours on some sections $Re = 100$	113
Figure 5.21 3D delta wing problem setup	114
Figure 5.22 Delta wing solution domain and surface meshes	115
Figure 5.23 Pressure coefficient at upper surface $x = 0.20$	116
Figure 5.24 Pressure coefficient at upper surface $x = 0.40$	116
Figure 5.25 Pressure coefficient at upper surface $x = 0.60$	117
Figure 5.26 Pressure coefficient at upper surface $x = 0.80$	117
Figure 5.27 Pressure coefficient at upper surface for various chord locations	118
Figure 5.28 Pressure coefficient contours on upper surface	118
Figure 5.29 Pressure coefficient contours at various x locations	119
Figure 5.30 Stream ribbons on delta wing	120
Figure 5.31 x component of Vorticity contours on delta wing	121
Figure 5.32 Stream ribbons on delta wing	122
Figure 5.33 Numerical spatial accuracy of IFVLBM	124
Figure 5.34 Density residuals for $Re = 400$ (cavity flow)	124

Figure 5.35 Temporal error of IFVLBM	126
Figure 5.36 Convergence Acceleration of IFVLBM	127
Figure B.1 1 dimensional D1Q3 stencil	156
Figure B.2 1 dimensional D1Q5 stencil	156
Figure B.3 1 dimensional D2Q5 stencil	157
Figure B.4 2 dimensional D2Q9 stencil	158
Figure B.5 3 dimensional D3Q15 stencil	159
Figure B.6 3 dimensional D3Q19 stencil	160
Figure B.7 3 dimensional D3Q27 stencil	161
Figure E.1 2D control volume element	174
Figure E.2 3D Control Volume Element	176

LIST OF ABBREVIATIONS & ACRONYMS

ADI	Alternating Direction Implicit
AR	Aspect Ratio
ARS	Algebraic Reynolds Stress
BC	Boundary Condition
BGK	Bhatnager Gross Krook
BR	Blockage Ratio
BTE	Boltzmann Transport Equation
CFD	Computational Fluid Dynamics
CFL	Courant Friedrichs Lewy number
CPU	Central Processing Unit
DDADI	Diagonally Dominant Alternating Direction Implicit
DF	Distribution Function
DNS	Direct Numerical Simulation
DRT	Dual Relaxation Time
EDF	Equilibrium Distribution Function
FVLBM	Finite Volume Lattice Boltzmann Method
GHz	Giga Hertz
GPU	Graphics Processing Unit
IFVLBM	Inverse Finite Volume Lattice Boltzmann Method
LBE	Lattice Boltzmann Equation
LBM	Lattice Boltzmann Method
LES	Large Eddy Simulation
LGA	Lattice Gas Automata
LUSGS	Lower Upper Symmetric Gauss-Seidel
MDS	Molecular Dynamics Simulation
MRT	Multi Relaxation Time
MUSCL	Monotonic Upstream-Centered Scheme for Conservation Laws
NEDF	Non-equilibrium Distribution Function

NS	Navier Stokes
NSE	Navier Stokes Equations
RAM	Random Access Memory
RANS	Reynolds Averaged Navier Stokes
RST	Reynolds Stress Transport
S&C	Stream and Collide
SRT	Single Relaxation Time

LIST OF SYMBOLS

Latin

A	Area
c	lattice velocity constant
c_s	speed of sound
e	intenal energy
f	density distribution function
f	frequency
F	numerical flux
f^{eq}	equilibrium distribution function
f^{neq}	non-equilibrium distribution function
H	Shape factor
I, J, K	cell center indices
i, j, k	cell vertex indices, lattice indices
k	turbulent kinetic energy
k	heat transfer coefficient
K	Kelvin
KE	kinetic energy
Kn	Knudsen Number
L_{ref}	representative physical length scale, reference length
M_{ref}	lattice velocity, speed of sound relation
Ma	Mach number
N	number of molecules
n	number of moles
p	pressure
Re	Reynolds number
St	Strouhal number
t	time
T	temperature

U_{ref}	reference speed
V	volume
\vec{c}	lattice velocity
\vec{e}	lattice velocity
\vec{F}	external force
$\vec{i}, \vec{j}, \vec{k},$	unit vectors
<i>Greek</i>	
α	lattice direction
β	sideslip angle
δ^*	displacement thickness
κ	Boltzmann constant
λ	mean free path
μ	dynamic viscosity
ν	kinematic viscosity
Ω	collision operator
Ω	control volume
ρ	density
σ	CFL number
τ	non-dimensional relaxation
θ	momentum thickness
<i>Operators</i>	
$\vec{\nabla}$	$\frac{\partial}{\partial x}\vec{i} + \frac{\partial}{\partial y}\vec{j} + \frac{\partial}{\partial z}\vec{k}$
<i>Special</i>	
$\hat{}$	non-dimensional quantity
$\vec{}$	vector quantity

CHAPTER 1

INTRODUCTION

Fluids are the integral part of our daily life. Air and water, the most known fluids to mankind has been in a continuous contact by humans or devices has been built. Moreover, we have been surrounded by different types of fluids in different areas of life and encounter with them frequently like the gasoline used by our cars, the blood in our circulatory system or the lubricants used for operating most of the mechanical devices.

Fluids have weaker inter-molecular forces compared to the solids. As a matter of fact, the scientific definition of the fluid is given as follows; *"fluid is a substance that deforms continuously under the application of a (tangential) shear stress, no matter how small the shear stress be"* [1]. Fluid that has been studied for a long period of time is a general classification of liquids and gases (or vapors), and it has to be studied continuously and thoroughly to use them efficiently and properly in our daily life.

Generally, the motion dynamics of fluids can be modeled and simulated mathematically depending on time and length scales. The length scale is classified by Knudsen number Kn named after the Danish physicist "*Martin Knudsen*". Knudsen number ($Kn = \lambda/L$) is the ratio of the mean free path λ to the representative characteristic length L used to describe the flow problem. The motion of the fluid can be divided into three subgroups with respect to length scale. These subgroups are named as:

- macroscopic scale
- mesoscopic scale
- microscopic scale

The major divisions due to length scales for fluid modeling given in reference [2] is shown in Figure 1.1. The microscopic motion of the fluid is very complicated and non-homogeneous. In the microscopic level, the equation of motion is the fundamental mathematical model and solved for each molecule in the system. Molecular Dynamics Simulation methods are used for finite number of molecules in the order of thousands to a few millions. The Molecular Dynamics Simulation for a mole of matter is almost impossible to handle with the calculation and memory capacity of modern computing tools, since each mole of any pure substance consists of 6.02×10^{23} (Avogadro's number) atoms/molecules.

On the other hand the macroscopic motion of the fluid can be considered as the average result of the molecules and it is homogeneous and continuous. In the macroscopic level, conservation laws for mass, momentum and energy are applied to a control volume. The obtained partial differential equations are the governing equations that describes the fluid flow. The conventional Computational Fluid Dynamics (CFD) methods are used to solve these governing equations (Navier Stokes, Euler equations) by using various numerical methods. For some simple flows, it is also possible to have analytical solutions.

Although there are some other statistical methods for mesoscopic motion of fluids, scientist and researchers had started to develop the The Lattice Boltzmann Method (LBM) approximately three decades ago to simulate flows at mesoscopic scale. LBM is an approach at mesoscopic scale which studies the micro-dynamics of imaginary particles by simplified kinetic models. It is an alternative way to simulate fluid flows. LBM has many natural features that can be accepted as advantage. The basic streaming and collision process of simulated fluid particles gives a clear view, the implementations of the algorithm and boundary conditions are easy and it can be coded in parallel with a very little effort. These natural features make LBM an alternative and tempting numerical tool for simulating fluid systems.

As a CFD method, the classical solution approach to Lattice Boltzmann Equation (LBE) is the Stream and Collide (S&C) method, which is easy to apply and can be parallelized almost linearly with the calculation core capacity of the target machine. For the Stream and Collide method, distribution functions in each lattice node for

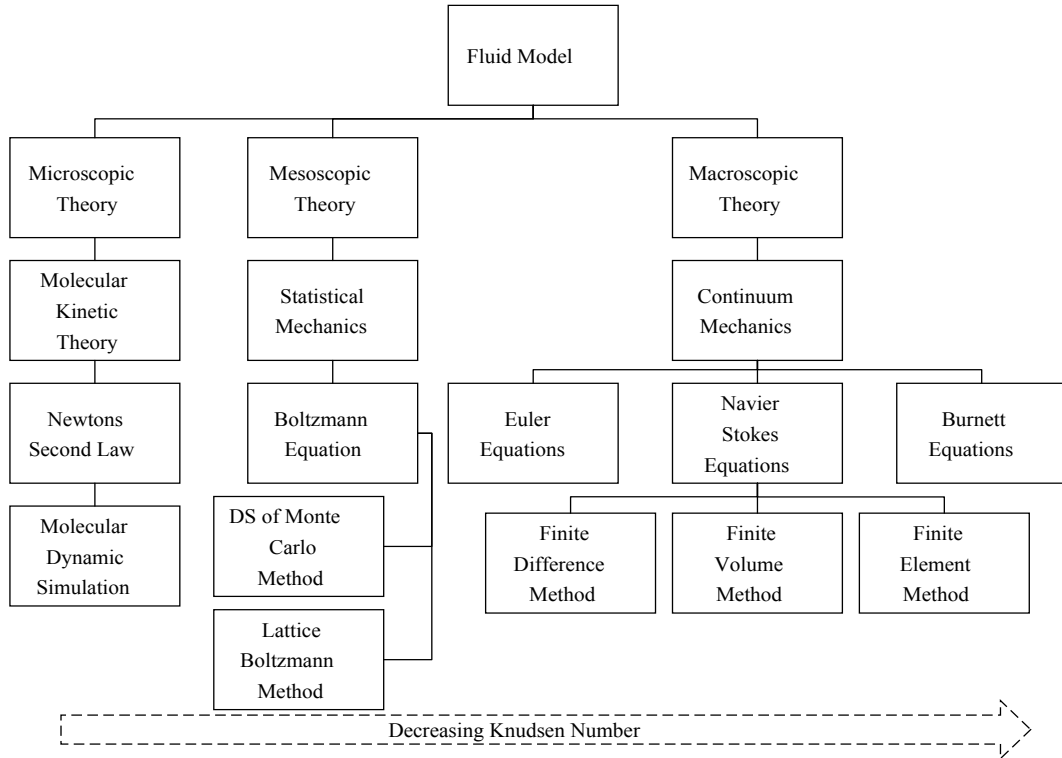


Figure 1.1: Fluid model, geometric scales and numerical approaches

all lattice directions can be solved independently, that is the reason why many researchers are studying on Graphical Processing Unit (GPU) implementation of LBE. However, there are two major drawbacks for the S&C method. The first one is the the Courant–Friedrichs–Lewy (*CFL*) condition; the solution algorithm dictates the condition that the *CFL* number should be less or equal to 1. The second drawback is that, uniform grid must be used throughout the solution domain. The fine grid requirement where the gradients are large will be a burden for the solution of the whole domain.

To shorten the solution time, different techniques based on S&C method are investigated by the researchers. These techniques will be discussed briefly in Chapter 2. Also researchers realize that the LBE is a partial differential equation, which can be solved by application of well known methods like Finite Difference, Finite Element, Finite Volume, Meshless and Spectral methods. It is possible to find information for all different methods for the solution of LBE in the literature. However, there is a commonality for almost all researches, they are **explicit**. To use the advantage of the parallelization and ease of implementation, explicit solution methods are preferred by

the researchers studying on LBMs. While acceleration of convergence is the aim, the stability condition for the solution methods are the obstacle to be avoided.

In this thesis, a cell centered Finite Volume approach applied on structured grids is used to solve the LBE for isothermal incompressible flows. An **implicit** solution technique is implemented to solve the derived Finite Volume formulation. The independence of the discrete LBE in all velocity directions in each grid cell is sacrificed to exploit the stability properties of the implicit method. Also Finite Volume formulation will let body fitted grids and fine mesh where the gradients are higher. A backward (Backward Euler Method) first order time integration is used for steady state solutions where temporal accuracy is not needed. Dual time stepping method is used for the problems where the transient solution is important. The subiterations are performed by using a second order accurate time integration method. The literature about the LBM is given and the details of the method will be described thoroughly in the forthcoming chapters.

The proceeding chapters of this manuscript is organized as follows:

- Chapter 2 of this thesis focuses on the literature about the kinetic theory, LBE, different solution methods, boundary conditions for LBEs and turbulence models.
- Chapter 3 focuses on the derivation of the Implicit Finite Volume Lattice Boltzmann Method for structured grids.
- Chapter 4 presents the validation cases including 2 dimensional steady state laminar, steady state turbulent, unsteady laminar cases with grid dependency for some validation cases.
- Chapter 5 presents the results on selected validation cases for 3D implementation. Furthermore, some numerical studies on stability, accuracy and convergence acceleration are also presented in Chapter 5.
- Chapter 6 summarizes the results and presents some suggestions for the future work which can be an extension of this study. The appendices are organized to present some additional auxiliary techniques, methods and materials used in the thesis.

CHAPTER 2

LATTICE BOLTZMANN METHOD

2.1 Introduction

One way of classifying the fluid motion is the use of the non-dimensional Knudsen number (Kn) as described in chapter 1 which is the ratio of the molecular mean free path and the characteristic length:

$$Kn = \frac{\lambda}{L} \tag{2.1}$$

The characteristic length L can be any physical dimension that describes the flow like the diameter of a cylinder, the height of a channel or the chord length of a wing section. The molecular mean free path λ is defined as the distance traveled by an atom or a molecule between successive collisions. The Knudsen number gives information about the the mathematical models that can be used to solve the fluid problem. Figure 2.1 adopted from reference [3], gives a comparison of mathematical models that can be applied to different scales with respect to Knudsen number.

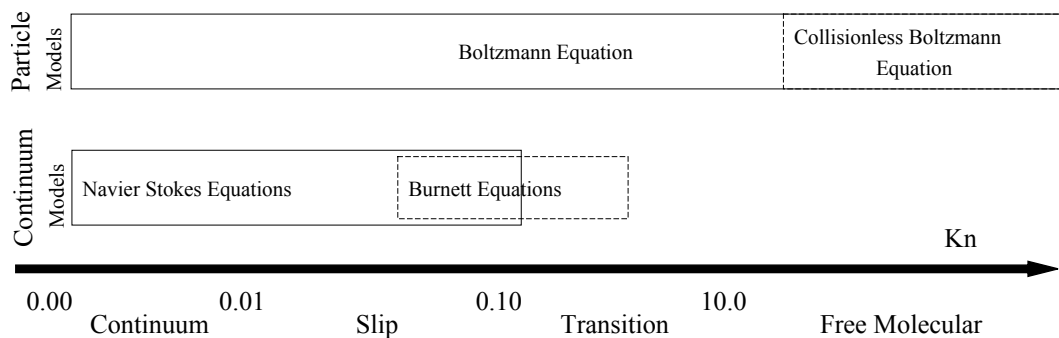


Figure 2.1: Knudsen number limits on some mathematical models

As seen from Figure 2.1, Boltzmann transport equation can be used to govern all fluid problems for Kn ranging from zero to infinity, where the NS equations can only be used up to the point where the Kn number is close to 0.1 covering continuum and slip flow region. More and more scientists and researchers are working on Lattice Boltzmann Methods for that reason.

In most of the articles related with LBM, there are some standard introductory phrases. One example to that is, “The Lattice Boltzmann Methods are originated from the Lattice Gas Automata methods” where Lattice Gas Automata methods are subclasses of cellular automata to simulate the fluid motion by using the discrete microscopic models [4, 5]. The first Lattice Gas Automata, the HPP model (named after Hardy, Pomeau and de Pazzis), for fluids was proposed by Hardy, Pomeau and de Pazzis [6]. The HPP model satisfies the conservation of mass and momentum, but it can not reproduce the Navier-Stokes equation because the square lattice lacks sufficient rotational invariance. But the FHP model (named after Frisch, Hasslacher and Pomeau) proposed by Frisch, Hasslacher and Pomeau [7] by using a triangular lattice with hexagonal symmetry has solved the issue of reproducing the NS equations. The Navier-Stokes equation is fully recovered by the FHP model, however it is only applicable to 2D simulations. The application of LGA to simulate 3 dimensional fluid flow was not easy but solution came by using a four dimensional lattice arrangement [8]. The face-centered hypercubic (FCHC) has sufficient symmetry; it can be projected in three dimensional space and can recover incompressible Navier Stokes equations [9].

The LBM derived from the LGA is a powerful technique for the simulation of single phase and multi-phase flows in complex geometries. Unlike Navier–Stokes solvers, LBMs consider flows to be composed of a collection of pseudo-particles that are represented by a velocity distribution function. These fluid particles reside and interact on the nodes of a grid. System dynamics and complexity emerge by the repeated application of local rules for the motion, collision and redistribution of these particles. The Lattice Boltzmann method, therefore, is an ideal approach for mesoscale and scale-bridging simulations. The scale-bridging consists of a two-scale decomposition in which the coarse scale is simulated using continuum methods, while the fine scale is simulated using Lattice Boltzmann methods[10]. In order to understand the Lattice Boltzmann method, it is better to visit the basic statistical mechanics and Boltzmann

equation.

2.2 Kinetic Theory

Kinetic theory is the underlying logic behind the LBM. The atoms and molecules are the fundamentals of substances and assumed to be perfect spheres moving and vibrating randomly in free space. The moving particles (atoms & molecules) are colliding with each other and the boundaries constantly. Kinetic theory explains the macroscopic properties of gases, such as pressure, temperature, viscosity, thermal conductivity, and volume by taking molecular composition and motion into account. The conservation laws are applicable, i.e. the conservation of mass, the conservation of momentum and the conservation of energy are also satisfied by the random motions of atoms & molecules at molecular level.

2.2.1 Particle Dynamics

The Newton's second law states that the net force applied to a body will change its linear momentum which is also applicable at the molecular level.

$$\vec{F} = \frac{d(m\vec{c})}{dt} \quad (2.2)$$

where \vec{F} is the net force which is the sum of external forces and intermolecular forces, m is the mass and \vec{c} is the velocity of the particle. If the mass is constant the equation will turn into:

$$\vec{F} = m \frac{d\vec{c}}{t} = m\vec{a} \quad (2.3)$$

where \vec{a} is the acceleration. The velocity vector is obtained by differentiating the position vector of the particle.

$$\vec{c} = \frac{d\vec{r}}{dt} \quad (2.4)$$

The equations, 2.2, 2.3, 2.4 are solved for MDS (molecular dynamics simulation) such that \vec{F} is a known function¹. An applied external force changes the velocity of the particle from \vec{c} to $\vec{c} + \frac{\vec{F}}{m}dt$ and position of the particle from \vec{r} to $\vec{r} + \vec{c}dt$. In the absence of an external force, the particle will continue to move without changing its direction and speed until the next collision.

If the internal energy of the system is increased (such as adding heat), the magnitude of particle velocity increases, and as a result the interaction between the particles also increases. Temperature, a macroscopic quantity, is the indication of increase in the kinetic energy of the system. The fluid particles also interacts with the boundaries of the system. As a result of the interaction between the fluid particles and the boundaries of the system, pressure (another macroscopic quantity) is observed.

The following equations are the relation of the pressure and temperature with the kinetic energy, where n is the number of moles, KE is the kinetic energy, κ is the Boltzmann constant[12, p. 6].

$$p = \frac{2}{3}nKE \quad (2.5)$$

$$T = \frac{2}{3}\frac{1}{\kappa}KE \quad (2.6)$$

2.2.2 Maxwell - Boltzmann Distribution Function

The Maxwell–Boltzmann distribution is a probability distribution named after James Clerk Maxwell and Ludwig Boltzmann. The distribution is developed by Maxwell in 1859 [12, p. 7]. When Maxwell realized that, dealing huge number of molecules are difficult even everything is known, he simply put forward the idea that the information of velocity and position of each particle at every instant of time is not important. Instead, a distribution function can be used to define the percentage of particles in a container at a specific location, having velocities within a specified range at some specific time. Upon after that, Ludwig Boltzmann continued to investigate into the physical attributes of the distribution function.

¹ As an example, Leonard Jones et al. [11] define a potential energy function that is used to calculate the force acting on each molecule for MDS

Statistically speaking, a particle speed probability distribution indicates which speeds are more likely. A particle will have a speed selected randomly from the distribution, and is more likely to be within one range of speeds than another. The distribution depends on the mass of the particle and the temperature of the system. The Maxwell–Boltzmann distribution applies to the classical ideal gas, which is an idealization of real gases. In real gases, there are various factors (e.g., van der Waals interactions, vortical flow, relativistic speed limits, and quantum exchange interactions) that can make their speed distribution different from the Maxwell–Boltzmann form. However, rarefied gases at ordinary temperatures behave very close to an ideal gas and the Maxwell speed distribution is an excellent approximation for such gases. Thus, it forms the basis of the Kinetic theory of gases, which provides a simplified explanation of many fundamental gaseous properties, including pressure and diffusion. The Maxwell- Boltzmann velocity distribution function is given as:

$$f(c) = \sqrt{\left(\frac{m}{2\pi\kappa T}\right)^3} 4\pi c^2 e^{-\frac{mc^2}{2\kappa T}} \quad (2.7)$$

where, m is the mass of the particle, κ is the Boltzmann constant ($\kappa = 1.38 \times 10^{-23} J/K$), T is the temperature and c is the velocity. The distribution function defines the probability to find a particle near speed c .

The equation is plotted for some noble gases for $T = 298K$ and the plot is given in Figure 2.2. The noble gases used as example are Helium, Neon, Argon and Xenon with atomic weight of 4, 20, 40, 132 *gram/mole* respectively. As seen from the figure if the molecular weight is smaller, the particle will have higher average speed. When the particle becomes heavier, the average speed is getting smaller. It must be also noted that the area under each curve is equal to 1.

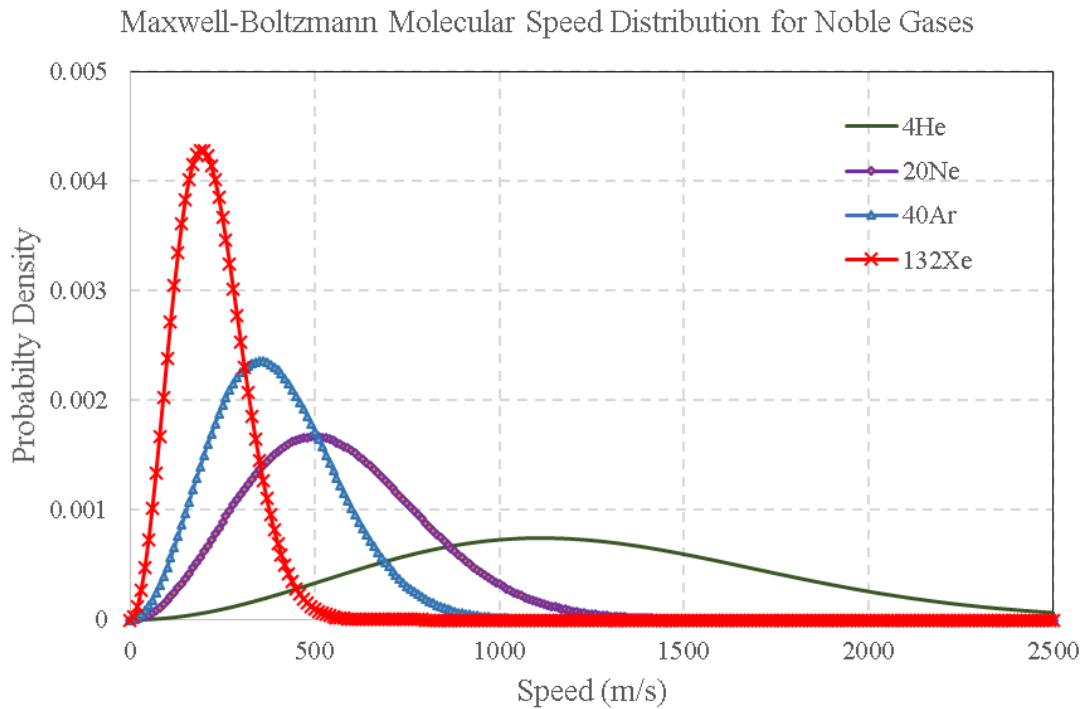


Figure 2.2: Probability density distributions of some noble gases

2.3 Boltzmann Transport Equation

Before deriving the Boltzmann Transport Equation, the following assumptions are made:

- The total volume of the particles are small compared to the system in consideration.
- All particles have the same mass.
- The number of particles is large enough to treat statistically.
- These particles are in constant, random, and rapid motion.
- The rapidly moving particles constantly collide among themselves and with the boundaries.
- Collisions are perfectly elastic.

- The interactions between particles are negligible except collision.
- The average kinetic energy of the gas particles depends only on the absolute temperature of the system.
- The collisions are assumed to be instantaneous.
- The gas particles are affected by gravity.

A system can be explained statistically, such that $f(\vec{r}, \vec{c}, t)$ is a distribution function of any quantity. It is indeed the number of molecules at time t , positioned between the distance \vec{r} and $\vec{r} + \vec{d}\vec{r}$ and having speed in the interval c and $\vec{c} + \vec{d}\vec{c}$. A force \vec{F} acting on a particle of mass m will change its position from \vec{r} to $\vec{r} + \vec{c}dt$ and change its speed from \vec{c} to $\vec{c} + (\vec{F}/m)dt$. Figure 2.3 summarizes how an external force affects the particle speed and position and direction.

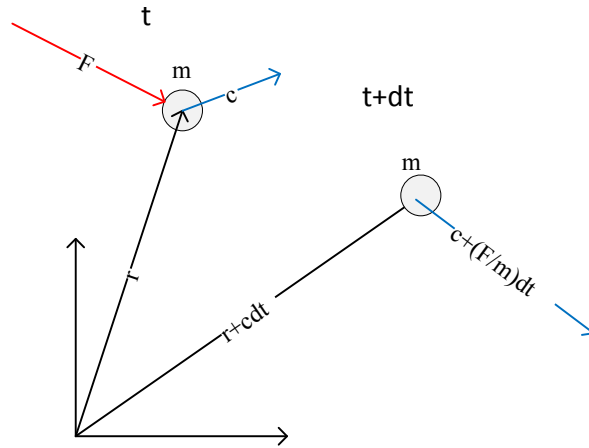


Figure 2.3: Position and velocity change of a particle

For a system we can say that the number of molecules “ N ” is not changing with time. In the absence of external force, if there is no collision between the molecules it can be simply stated that:

$$f(\vec{r} + \vec{c}dt, \vec{c} + \vec{F}/m dt, t + dt) - f(\vec{r}, \vec{c}, t) = 0 \quad (2.8)$$

However, if there are molecular collisions, the equation 2.8 is needed to be modified since there will be a difference before and after the collision for the interval $drdc$.

The rate of change of the distribution function between post and pre collision state is called the collision operator. Equation 2.8 is modified as below:

$$f(\vec{r} + \vec{c}dt, \vec{c} + \vec{F}/mdt, t + dt) - f(\vec{r}, \vec{c}, t) = \Omega(f)dt \quad (2.9)$$

Equation 2.9 can further be modified by using some algebra such that as $dt \rightarrow 0$

$$\frac{f(\vec{r} + \vec{c}dt, \vec{c} + \vec{F}/mdt, t + dt) - f(\vec{r}, \vec{c}, t)}{dt} = \frac{df(\vec{r}, \vec{c}, t)}{dt} = \Omega(f) \quad (2.10)$$

Equation 2.10 states that the rate of change of distribution function is equal to the rate of collision. The distribution function has the arguments, \vec{r} , \vec{c} and t , the total change of distribution function can be rewritten as follows:

$$df = \frac{\partial f}{\partial \vec{r}} \cdot \vec{dr} + \frac{\partial f}{\partial \vec{c}} \cdot \vec{dc} + \frac{\partial f}{\partial t} dt \quad (2.11)$$

Substituting equation 2.11 into the equation 2.10, the following equation is obtained:

$$\frac{\partial f}{\partial \vec{r}} \cdot \frac{\vec{dr}}{dt} + \frac{\partial f}{\partial \vec{c}} \cdot \frac{\vec{dc}}{dt} + \frac{\partial f}{\partial t} \frac{dt}{dt} = \Omega(f) \quad (2.12)$$

Recall that, the derivative of position with respect to time is velocity and derivative of velocity is with respect to time is acceleration which nothing but the force divided to mass ($\frac{d\vec{r}}{dt} = \vec{c}$, $\frac{d\vec{c}}{dt} = \vec{a} = \frac{\vec{F}}{m}$).

Equation 2.12 can further be modified as follows:

$$\frac{\partial f}{\partial t} + \vec{c} \cdot \frac{\partial f}{\partial \vec{r}} + \frac{\vec{F}}{m} \frac{\partial f}{\partial \vec{c}} = \Omega(f) \quad (2.13)$$

In the absence of external force \vec{F} the above equation can be rewritten as:

$$\frac{\partial f}{\partial t} + \vec{c} \cdot \vec{\nabla} f = \Omega(f) \quad (2.14)$$

where $\vec{\nabla} = \frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k}$ for Cartesian coordinates.

Looking at the equation 2.14, we can classify it as an advection equation with a source term. The equation can be solved if the collision operator is known, however, the collision operator itself is function of the distribution function, which makes the solution procedure much more complicated.

2.3.1 Bhatnagar, Gross, Krook, BGK Approximation

It is very difficult to solve the Boltzmann Equation since the collision term is very complicated. However, if the collision operator can be replaced by a more simple operator, than it can be possible to cope with Boltzmann equation. Bhatnagar Gross and Krook [13] came up with a simple approximation for the collision operator in 1954. In many research studies on Lattice Boltzmann Methods, their approximation for the collision operator is used.

The collision operator is defined as:

$$\Omega(f) = -\omega(f - f^{eq}) = -\frac{1}{\tau}(f - f^{eq}) \quad (2.15)$$

The coefficient ω is called the collision frequency where its reciprocal τ is the relaxation coefficient. The equilibrium distribution function (EDF) f^{eq} is the Maxwell-Boltzmann distribution function.

Since new collision operator is introduced, equation 2.14, can be rewritten as follows:

$$\frac{\partial f}{\partial t} + \vec{c} \cdot \vec{\nabla} f = -\frac{1}{\tau}(f - f^{eq}) \quad (2.16)$$

The above equation is the continuous Boltzmann Equation. In Lattice Boltzmann Methods, equation 2.16 is assumed to be valid along lattice directions. Thus the discrete Boltzmann Equation is given as follows:

$$\frac{\partial f_\alpha}{\partial t} + \vec{c}_\alpha \cdot \vec{\nabla} f_\alpha = -\frac{1}{\tau}(f_\alpha - f_\alpha^{eq}) \quad (2.17)$$

where α denotes the lattice direction.

The discrete Lattice Boltzmann Equation 2.17 is the heart of the Lattice Boltzmann Methods. This equation is solved for every linkage stated in the selected solution stencil (In Appendix B, some stencils are given as reference). Then, the macroscopic quantities are calculated by the following relations:

$$\rho = \sum_{\alpha} f_{\alpha} \quad (2.18)$$

$$\rho \vec{u} = \sum_{\alpha} \vec{c}_{\alpha} f_{\alpha} \quad (2.19)$$

2.3.2 Equilibrium Distribution Function

The equilibrium distribution function must be defined to complement the discrete LBE. For BGK lattice Boltzmann Methods, equilibrium distribution function is given as follows [14, p. 35]:

$$f_{\alpha}^{eq} = \rho w_{\alpha} \left[1 + \frac{\vec{c}_{\alpha} \cdot \vec{u}}{c_s^2} + \frac{1}{2} \frac{(\vec{c}_{\alpha} \cdot \vec{u})^2}{c_s^4} - \frac{1}{2} \frac{(\vec{u})^2}{c_s^2} \right] \quad (2.20)$$

where w_{α} is the weighting factor, ρ is the density, \vec{c}_{α} is the velocity along the lattice direction, \vec{u} is the macroscopic velocity at the lattice point and c_s is the speed of sound.

2.4 Solution Methods

Standard Lattice Boltzmann Method has the idea of dealing with the particle distributions over a discrete lattice mesh. The LBM can be considered as a simple molecular dynamics model and fills the gap between the microscopic fluid simulations and macroscopic fluid simulations [15]. By simple stream and collide algorithm, particle distributions are calculated for each time advancement. The macroscopic flow quantities are then calculated by using the particle distributions. Since the equations are solved locally, the LBM has a high potential for parallel implementations and has an advantage over other CFD methods. However, there is a strict requirement for the

particles to move from one lattice point to the next in one time step. This situation dictates the CFL condition $CFL = 1$. Then the cell size limitations, which is coming from the need of setting Courant Fredrich Lewy number to 1, exposes an adverse influence on solution time and computational memory requirements. Its simplicity allows a small amount of coding effort while the computational and memory cost for large scale problems are significant. Moreover, the pressure and velocity solutions are not coupled in Lattice Boltzmann Method, which saves computational time compared to NS methods that requires the solution of Poisson's equation for pressure and velocity coupling for incompressible flows.

The standard LBM is limited to regular meshes. When bodies with complex boundaries are in focus, the resolution of the mesh is increased to have accurate solutions and that yields increase in computational time and memory requirements. Some alternative techniques are developed to overcome the significant disadvantage of the LBM. The techniques [16, p. 78] can be listed as follows:

- Grid refinement and multi-block methods
- Interpolation methods
- Finite Difference Lattice Boltzmann Method (FD-LBM)
- Finite Element Lattice Boltzmann Method (FE-LBM)
- Taylor Series Expansion Least Square based Lattice Boltzmann method (TS-LBM)
- Finite Volume Lattice Boltzmann method (FV-LBM)

The grid refinement technique is to increase the grid density where large gradients exist near the boundaries. The information exchange between different levels of grids takes place according to the set of rules defined by the different algorithms. The boundary information can be spread to the flow domain rapidly by coarser grids where the grid refinement is not needed so, the computational cost is reduced significantly by comparison to the standard LBM [16]. There are different studies and different algorithms about the application of grid refinement and multi-block techniques. Filippova and Hänel [17] are the first to study about the subject. The approach is called

FH method and relies on information exchange after the post collision step. The FH method has known to have instabilities when relaxation parameters of fine or coarse grids are close to unity ($\tau_c \approx 1$ or $\tau_f \approx 1$). Based on the FH method, Dupuis and Chopard [18] develop another method called DC method. Both fine and coarse grid exist at the same time in the solution domain where the refinements are required. The solution information of coarse grid is passed into the fine grid by common grid points. This process has increased the computational efficiency significantly. However, the information exchange between coarse and fine grids requires special treatment to overcome the undesired results.

Yu et al. [19] introduced an alternative method that uses grid blocks with different resolutions. The method is called the multi-block method. The solution domain is decomposed into blocks with different grid resolution. Each block is solved by standard LBM and the information exchange is performed on block interfaces. The information exchange method is similar to the one used in FH method. Guo et al. [20] has proposed another domain decomposition method. In this more flexible approach, sub-domains may overlap or not. Also the lattice stencil in each domain is not required to be the same stencil. The nonequilibrium extrapolation method is used to treat the ghost boundaries of the sub-domains. The information exchange is performed by using the equilibrium distribution functions. The macroscopic quantities (density and velocity) are interpolated from the sub-domains.

The use of adaptive grids are described by Crouse et al. [21]. Their method depends on a sensor variable. The sensor variable depends on derived quantities of the flow or on heuristic approach to detect the region where a refinement is required. The quantities in the refined cells are calculated by interpolation methods using the parent cells. Crouse's method is generalized by Tölke et al. [22] for multiple relaxation time LBMs. More studies on the subject can be found on references [23, 24, 25].

The second technique, which incorporates with some interpolation methods first proposed by He et al. [26], is known as Interpolation Supplemented LBM (ISLBM). The idea is the computational mesh and the lattice points (where the discrete velocities are set) are separated. Then a calculation process similar to the standard LBM is performed with an additional interpolation step. The ISLBM is still bounded by the

CFL=1 condition. Moreover, the interpolation schemes needs to be at least second order accurate to minimize the dissipation errors. The second order accurate interpolation schemes do not influence the viscosity of ISLBM. Another advantage of the ISLBM is that the simulated Reynolds number can be increased. That is, a coarser computational mesh can be used for the same level of accuracy with respect to the standard LBM for same Reynolds number [27].

The third technique is the Finite Difference Lattice Boltzmann Method. The standard LBM is the upwind finite difference scheme of the Lattice Boltzmann Equation with BGK approximation [13] to the collision operator. Cao et al. [28] first use the finite difference method to discretize the LBE. They used central difference scheme for the spatial derivatives in discrete velocity direction. Moreover a second order explicit Runge Kutta integration scheme is used to advance in time. Mei and Shyy [29] proposed a more general semi implicit scheme applicable to curvilinear coordinate systems. Although the implicit treatment of the collision term has some improvement on stability , the extrapolation used in the equilibrium distribution function has negative impact on the stability. Moreover, the three time step algorithm requires more memory. An alternative semi implicit approach is proposed by Guo and Zhao [20] which does not have the mentioned disadvantages on stability.

The fourth technique is the Finite Element Lattice Boltzmann Method. Lee and Lin [30] proposed the method characteristic Galerkin FE-LBM. Their approach is to integrate the discrete velocity LBE equation along the characteristic line. They used a second order prediction correction method to advance in time. The flow domain is decomposed into a set of non-overlapping elements [20]. Then, the distribution function is projected onto a set of localized piece-wise polynomials associated with the elements. There are different approaches for the FE-LBM, like least squares finite element method [31], and spectral element discontinuous Galerkin method [32].

The fifth technique is called the Taylor Series Expansion and the Least Squares based Lattice Boltzmann Method (TS-LBM). This method is developed by a group of scientists [33, 34, 35]. In this method the standard LBE goes under Taylor expansion in spatial direction. The expansion is truncated to the second order derivatives. An algebraic set of equations, in which, the coefficients depend on the coordinates of the

grids and lattice velocity, is obtained. The method is applicable to different lattice models and reduces to the standard LBM for the uniform grids. The unknowns are distribution function f and its spatial derivatives to the second order on grid points. The least squares method takes place when there is a need to solve an overdetermined system [36]. That is there are more grid points used than the number of unknowns. The application of least squares optimizes the error in calculating the unknowns for each grid point.

The final technique is the Finite Volume Lattice Boltzmann Method. The model is developed by Nannelli and Succi. They define a coarse grained distribution function \bar{f} where the standard distribution function f is averaged over a control volume [37]. Simple Euler rule is used in time advancement and the flux of distribution function in standard LBM is evaluated from the coarse grained distribution functions by using some interpolation schemes. More advanced alternative finite volume techniques are used later by different researchers [38, 39, 40]. These methods are first order accurate in time. Higher order time advancement scheme as fourth order Runge Kutta method is proposed by Ubertini and Succi [41]. In flux calculations, some different methods are used such as least squares linear reconstruction method by Stiebler et al. [42] and TVD approach by Patil and Lakshmisha [43]. Guzel and Koc [44] applied Implicit Explicit Runge Kutta (IMEX) [45] method for the FV-LBM and showed that the stability is increased with respect to fully explicit schemes. In IMEX scheme for LBM, the characteristic of the collision invariant [46] is used to eliminate the implicit terms, where the time advancement reduces an explicit scheme.

2.4.1 Stream and Collide Method

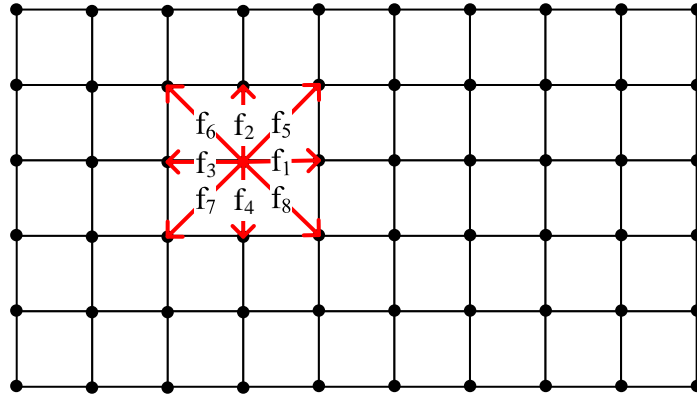
As mentioned above there are many solution methods for the LBE with BGK collision operator. But the original method developed from the Lattice Gas Automata is the Stream and Collide Method. The method is valid for incompressible flows $Ma_\infty < 0.3$ and for small Reynolds numbers. The method will be discussed here with some of its drawbacks. The solution procedure starts with the discrete Lattice Boltzmann Equation.

$$f_\alpha(\vec{r} + \vec{c}dt, t + dt) - f_\alpha(\vec{r}, t) = -\frac{1}{\tau} (f_\alpha(\vec{r}, t) - f_\alpha^{eq}(\vec{r}, t)) \quad (2.21)$$

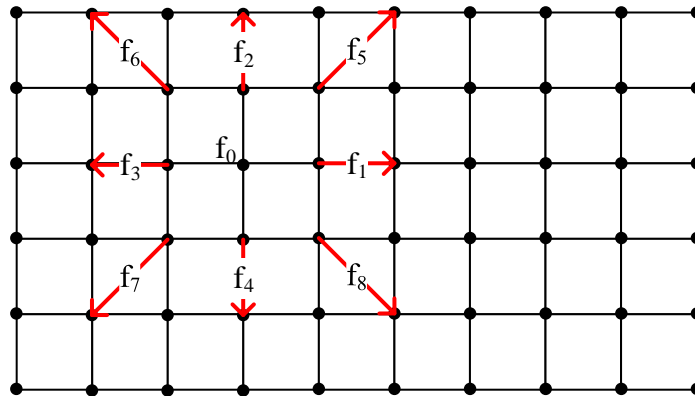
where $\tau = \frac{\tau_c}{dt}$ is the non dimensional relaxation time. The time advancement step is divided into 2 levels. The first level is the collision step where $\tilde{f}_\alpha(\vec{r}, t + dt)$ is calculated. Then the second step is the streaming step, where particle density probability function is moved from the lattice in focus to the neighboring lattice. Figure 2.4 is showing the streaming process.

$$\tilde{f}_\alpha(\vec{r}, t + dt) = f_\alpha(\vec{r}, t) - \frac{1}{\tau} (f_\alpha(\vec{r}, t) - f_\alpha^{eq}(\vec{r}, t)) \quad (2.22)$$

$$f_\alpha(\vec{r} + \vec{c}dt, t + dt) = \tilde{f}_\alpha(\vec{r}, t + dt) \quad (2.23)$$



(a) Collision process



(b) Streaming process

Figure 2.4: Stream and collide process

After streaming local macroscopic quantities are calculated using the following relations:

$$\rho = \sum_{\alpha} f_{\alpha} \quad (2.24)$$

$$\rho \vec{u} = \sum_{\alpha} \vec{c}_{\alpha} f_{\alpha} \quad (2.25)$$

Then the new equilibrium function is calculated for each lattice point.

$$f_{\alpha}^{eq} = \rho w_{\alpha} \left[1 + \frac{\vec{c}_{\alpha} \cdot \vec{u}}{c_s^2} + \frac{1}{2} \frac{(\vec{c}_{\alpha} \cdot \vec{u})^2}{c_s^4} - \frac{1}{2} \frac{(\vec{u})^2}{c_s^2} \right] \quad (2.26)$$

Also the kinematic viscosity is related to the relaxation parameter τ with the following relation for Stream and Collide method:

$$\nu = \left(\frac{\tau}{3} - \frac{1}{6} \right) \frac{(dx)^2}{dt} \quad (2.27)$$

The equation 2.27 is used to relate the relaxation parameter with the Reynolds number through the kinematic viscosity.

Looking at the equations 2.22 and 2.23 , it can be seen that the collision step and the stream step for all lattice points can be solved independently. Hence, it can be concluded that, the solution can be performed in parallel as far as there is a suitable computational hardware. Also, it is simple to code the solution algorithm. However, despite its advantages, there are some disadvantages of Stream and Collide method given in the literature:

- The particles must travel from one lattice point to the neighboring lattices in one time step ($CFL = 1$ condition). Therefore, all the solution domain will have a uniform lattice resolution. This brings the fact that, a huge number of lattice point will be required to solve the whole domain.
- As τ closes to 0.5 numerical instabilities occur in the solution process.

- Boundary nodes require special treatment since the boundary conditions are defined by macroscopic quantities (either Dirichlet or Neumann). Also, the distribution functions at the boundaries should be modified to reflect correct physical boundary conditions.

2.5 Boundary Conditions

Boundary conditions are very important to solve the differential equations numerically. The accuracy of the solution strongly relies on the boundary conditions. It is also the same for the LBMs. It is essential to impose the proper boundary conditions for the distribution functions.

In general, there exists three types of boundary conditions for the differential equations. The first one is the “Dirichlet” type boundary condition where the function itself is defined at the boundary. The second one is the “Neumann” type boundary condition where the normal derivative of the function is defined at the boundary. The third type is the mixed boundary condition where it is the linear combination of “Dirichlet” and “Neumann” boundary condition.

The situation is a little bit different for the lattice Boltzmann Method, since the unknown is the distribution function in the Lattice Boltzmann Equation and most of the times the boundary conditions are defined by use of macroscopic quantities such as density, pressure, velocity, etc.

There are different approaches that define the link between the macroscopic quantities and the distribution functions. In this section, some of the approaches will be described briefly. First, “Distribution Modification” and then “Distribution Reconstruction” will be discussed.

Distribution Modification

The unknown distribution functions are obtained by applying physical rules. These rules are defined as the bounce back, mass and momentum conservation or any combination. The most basic rule for the solid boundaries are fullway bounce back

and halfway bounce back rule defined in references [47, 48]. Inamuro boundary conditions [49], Zou/He boundary conditions [50] can be given as other examples.

Distribution Reconstruction

As the name implies in this approach, all the distribution functions on the boundary is reconstructed for each iteration during the solution procedure from the macroscopic flow quantities and their derivatives. Regularized boundary condition [51], finite difference boundary conditions [52] and Guo boundary conditions [53] are the examples to this type.

Curved Boundaries

All the above mentioned boundary conditions are derived for straight boundaries. However, in real life, curved boundaries are more likely to be encountered. Thus some interpolation or extrapolation schemes have to be incorporated with the curved boundary conditions. Remembering that for the Stream and Collide method, the lattice is uniform for the whole domain which will bring the situation that some lattice nodes will not be on the boundary.

For the Distribution Modifications approach, some interpolation methods were proposed by Fillipova et.al [17] and a different method proposed by Mei et. al [54]. Also, simple fitting methods were proposed by Bouzidi et.al [55] and Yu et. al [56].

For the Distribution Reconstruction approach, regularized curved boundary conditions were proposed by Verschaeve et.al [57].

2.6 Turbulence Models

The definition of turbulence made by von Karman [58] in 1937 is:

“Turbulence is an irregular motion which in general makes its appearance in fluids, gaseous or liquid, when they flow past solid surfaces or even when neighboring

streams of the same fluid flow past or over one another.”

Almost all flows, that we encounter in our daily life are turbulent. Turbulence phenomena can be described as random and chaotic. The velocity fluctuates in all directions and has infinite number of scales. Solving the NS equations for a turbulent flow is impossible because the equations are elliptic, non-linear, velocity-pressure and velocity-temperature is coupled. Moreover, the flow is three dimensional, chaotic, diffusive, dissipative, and intermittent. Turbulent flows typically develop as an instability of laminar flows. For a viscous fluid, these instabilities result from the interactions of the non-linear inertial terms and the viscous terms contained in the Navier-Stokes equations. These interactions are very complex because the turbulence is rotational, three-dimensional, and time-dependent.

Turbulence is a continuous phenomenon that occurs on a large range of length and time scales. However, the length scales are bigger compared to the molecular scales. In order to visualize turbulent flows, turbulent eddies must be observed. Turbulent eddies can be thought of as a local swirling motions with a characteristic dimension on the order of the local turbulence length scale. Turbulent eddies can overlap in space where larger eddies carry smaller ones. It can be taught that the larger eddies are decomposed to smaller eddies until the smallest scale eddies, where the energy is dissipated into heat by molecular viscosity. That brings the conclusion that turbulent flows are always dissipative.

Turbulent flows also exhibit a largely enhanced diffusivity where the turbulent diffusion greatly enhances the transfer of mass, momentum, and energy. Therefore, the apparent stresses may be of several orders of magnitude greater compared to the corresponding laminar case.

As stated above, the rotational and three-dimensional natures of turbulence are closely linked, as vortex stretching is required to maintain the constantly fluctuating vorticity. As vortex stretching is absent in two-dimensional flows, turbulence must be three-dimensional. This implies that there are no two-dimensional approximations, thus making the problem of resolving turbulent flows, a difficult problem.

The time-dependent nature of turbulence means that statistical averaging techniques

are required to approximate random fluctuations in velocity. However, time averaging leads to unknown and non-analytical correlations in the equations of motion. This explains the classic closure problem of turbulence, which requires modeled expressions to account for the additional unknowns. And developing models for the additional unknowns is the primary focus of turbulence modeling.

As a summary, Tennekes [59, p. 1] stated that the definition of turbulence is easy yet better to define some characteristics that every turbulent flow employs. Below is the list of the nature of turbulence:

- Irregularity or randomness which makes it impossible to have a deterministic approach to the turbulent flows. This nature forces to have statistical methods.
- Diffusivity causes rapid mixing in turbulent flows. Rapid mixing increases rates of momentum, heat and mass transfer in turbulent flows with respect to laminar flows.
- High Reynolds numbers is another nature of turbulent flows. Turbulence often occurs at high Reynolds numbers. The turbulence is often originated from the instabilities of laminar flows when the Reynolds number becomes too large. The instabilities are associated with the interaction of nonlinear inertia terms in the equations of motion and viscous terms.
- Three dimensional vorticity fluctuations plays an important role on defining turbulent flows. The flow must be three dimensional and rotational since the important mechanism called as vortex stretching is absent in two dimensional flow.
- Dissipation is another important nature of turbulent flows. The kinetic energy is transformed to internal energy while viscous shear stresses perform deformation work on the fluid.
- Continuum is also an important nature of the turbulent flows. The motion is governed by equations of fluid mechanics and does not depend on the molecular scales, which means that the smallest scale of turbulence is much more larger than the molecular scales.

2.6.1 Classes of Turbulence Models

It is very difficult to solve the turbulent flows. Even with the most powerful computers, a direct simulation of turbulence is very limited to the order of $Re\ 10^4 - 10^5$. In reference [60, p. 214], the grid resolution and CPU time is related to the Re like $Re^{9/4}$ and Re^3 respectively for DNS. The impracticality of the situation leads engineers and researchers to seek approximate models to handle turbulence in flow problems. The major classification of turbulent models can be given as:

- Algebraic models
- One-equation models
- Multiple equation models
- Second order closures (Reynolds Stress Models)
- Large Eddy Simulation

The algebraic, one equation and multiple equation models are called first order closures. They are based on Boussinesq eddy viscosity hypothesis and non-linear eddy viscosity formulations. In Figure 2.5, the turbulence models are presented according to the complexity level by Blazek [60, p. 215].

2.6.2 Turbulence Equations

For incompressible fluids, by using the Einstein summation the governing equations can be written as follows:

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (2.28)$$

$$\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\mu}{\rho} \nabla^2 v_i \quad (2.29)$$

$$\frac{\partial T}{\partial t} + v_j \frac{\partial T}{\partial x_j} = k \nabla^2 T \quad (2.30)$$

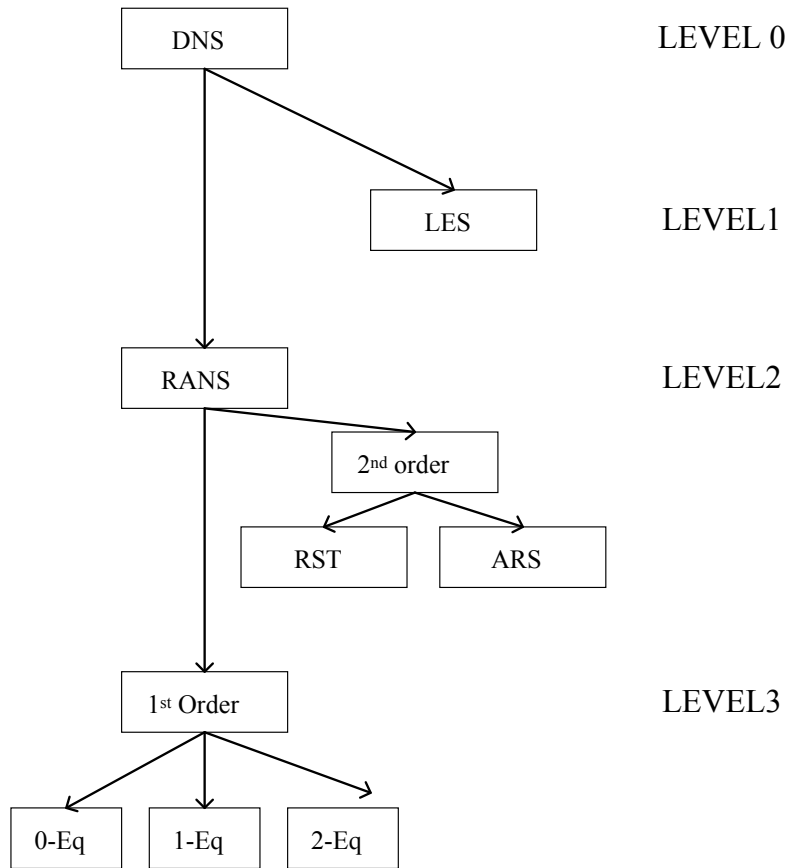


Figure 2.5: Turbulence model hierarchy according to complexity level (Level 0 - most complex)

where ∇^2 is the Laplace operator, ρ is the density, μ is the dynamic viscosity, k is the heat transfer coefficient, v is the velocity, p is the pressure and T is the temperature.

Reynolds Averaging

The origin of the time-averaged Navier-Stokes equations dates back to the late nineteenth century when Reynolds (1895) published results from his research on turbulence. The flow quantities can be written as the mean value of the quantity plus a fluctuation at any time for the turbulent flows.

$$v_i = \bar{v}_i + v'_i, \quad p = \bar{p} + p' \tag{2.31}$$

The mean values of any quantity can be averaged by different techniques.

- Time averaging is used for statistically steady turbulence.

$$\text{they } \bar{v}_i = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} v_i dt \quad (2.32)$$

The flow quantity is no longer function of time but only space. The time T must be taken large enough when compared with the time scale of the turbulent fluctuations.

- Spatial averaging is used for homogeneous turbulence.

$$\bar{v}_i = \lim_{\Omega \rightarrow \infty} \frac{1}{\Omega} \int_{\Omega} v_i d\Omega \quad (2.33)$$

The average flow quantity is uniform inside the control volume but still changing with time.

- Ensemble averaging is used for general turbulence.

$$\bar{v}_i = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{m=1}^N (v_i)_m \quad (2.34)$$

The flow quantity is still function of time and space.

For above averaging approaches, the average of the fluctuating part of any quantity is zero ($\bar{\phi}'_i = 0$). But the average of the any combined quantity can be different than zero for the correlated quantities. That is $v_i \bar{v}_i \neq 0$ and $v_i \bar{v}_j \neq 0$.

Favre Averaging

Favre averaging technique is another technique that is used for varying density flows [60, p. 218]. However it is just not right to use Favre averaging for all flow quantities because it will bring additional complexity. The most convenient application in the literature is to use Reynolds averaging for density and pressure, and Favre averaging for other flow quantities. For velocity the Favre averaging formula is given as follows:

$$\bar{v}_i = \frac{1}{\bar{\rho}} \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \rho v_i dt \quad (2.35)$$

where $\bar{\rho}$ is the Reynolds-averaged density.

The flow quantities in NS equations can be replaced by either Reynolds averaged quantities or Favre averaged quantities to obtain the governing equations for the turbulent flows. However, for the LBM the governing equation is not continuum equation, yet the averaged quantities are not taken in consideration. But as specified in the following section, eddy viscosity hypothesis that replaces the dynamic viscosity as a sum of laminar and turbulent parts will be used to incorporate turbulence effects.

2.6.3 Eddy Viscosity Hypothesis

Boussinesq presented one of the most significant contribution to turbulence modeling in 1877. Based on the observation, that the momentum transfer in a turbulent flow is dominated by the mixing caused by the large eddies, he hypothesized the turbulent shear stress depends on the mean rate of strain as in laminar flows.

Without going in to further details, the viscosity is replaced by the equation

$$\mu = \mu_L + \mu_T \quad (2.36)$$

where μ_L is the laminar part of the viscosity which is calculated by Sutherland formula and μ_T is the turbulent viscosity calculated by some model.

The eddy-viscosity concept of Boussinesq is very convenient since it requires only the determination of μ_T . Once turbulent eddy viscosity μ_T is known, μ can easily be calculated by equation 2.36, and using the relation $\tau_c = \mu / (\rho c_s^2)$ it can be incorporated to the Lattice Boltzmann Method.

Boussinesq's approach became the basis for a large variety of first-order turbulence closures. More information on the details about the turbulence modeling can be found in the references [59, 61, 60, 62, 63, 64, 65].

CHAPTER 3

DEVELOPMENT OF IFVLBM

As presented in Chapter 2, it is possible to solve the LBE by a bunch of different methods. Besides the classical Stream and Collide method and its derivatives which requires uniform lattice resolution throughout the domain, researchers and scientists realized that the LBE is a differential equation, so that other techniques could be employed to solve it. In this chapter, a finite volume formulation of LBE will be derived. Then for the time discretization, a backward Euler method will be used for steady state problems. And to increase accuracy a second order backward integration formula will be used for the transient solutions. All the solutions are performed on structured grids.

The advantage of the finite volume formulation over S&C method can be described as follows:

- The solution domain consists of body fitted meshes. Hence, there will not be any need of special treatment of the probability density function on the boundaries.
- The uniform mesh will not be necessary, hence the number of cells required to describe the solution domain can be reduced significantly by using much coarser grids where the derivatives are smaller.
- The CFL number can be eased, where the solution schemes for S&C methods require $CFL = 1$.
- It is possible to use techniques like local time stepping for faster convergence.

3.1 Complete Set of Equations

The continuous Boltzmann Equation with Bhatnagar-Gross-Krook [13] approximation for the collision term is the starting point of the method presented in this study like many other studies. The continuous Lattice Boltzmann Equation (LBE) is given as:

$$\frac{\partial f}{\partial t} + \vec{e} \cdot \vec{\nabla} f = \Omega(f) \quad (3.1)$$

where f is ($f = f(\vec{x}, \vec{e}, t)$) the particle density distribution function. The arguments of the distribution function are the position \vec{x} , particle velocity \vec{e} and time t .

The lattice velocity \vec{e} can be defined as follows:

$$\vec{e} = c \left(e_x \vec{i} + e_y \vec{j} + e_z \vec{k} \right) \quad (3.2)$$

where e_x, e_y and e_z are either 1 or 0 depending on the lattice direction. The collision operator $\Omega(f)$ given in equation 3.1 is defined as

$$\Omega(f) = -\frac{1}{\tau_c} (f - f^{eq}) \quad (3.3)$$

In equation 3.3, f^{eq} is the equilibrium distribution function and τ_c is the relaxation time. When the collision operator is substituted into the Boltzmann Equation, the following continuous LBE is obtained:

$$\frac{\partial f}{\partial t} + \vec{e} \cdot \vec{\nabla} f = -\frac{1}{\tau_c} (f - f^{eq}) \quad (3.4)$$

Equation 3.4 can be discretized [16, p. 18] in a set of direction in velocity space where f_α is the density distribution function, e_α is the discrete velocity and f_α^{eq} is the equilibrium distribution function in the α^{th} direction.

$$\frac{\partial f_\alpha}{\partial t} + e_\alpha \cdot \vec{\nabla} f_\alpha = -\frac{1}{\tau_c} (f_\alpha - f_\alpha^{eq}) \quad (3.5)$$

The equilibrium function for each α^{th} direction is given in equation. 3.6.

$$f_{\alpha}^{eq} = \rho w_{\alpha} \left[1 + \frac{\vec{e}_{\alpha} \cdot \vec{u}}{c_s^2} + \frac{1}{2} \frac{(\vec{e}_{\alpha} \cdot \vec{u})^2}{c_s^4} - \frac{1}{2} \frac{(\vec{u})^2}{c_s^2} \right] \quad (3.6)$$

where \vec{u} is the macroscopic velocity and the w is the weighting factor of the α^{th} direction .

The physical quantities called as macroscopic density and the momentum are calculated by the formulas [16, p. 18], respectively:

$$\rho = \sum_{\alpha} f_{\alpha} \quad \text{and} \quad \rho \vec{u} = \sum_{\alpha} f_{\alpha} \vec{e}_{\alpha} \quad (3.7)$$

where the local macroscopic velocity is defined as $\vec{u} = u_x \vec{i} + u_y \vec{j} + u_z \vec{k}$.

The lattice velocities are defined as a function of arbitrary constant c which is related to the speed of sound with the following relation for D2Q9 and D3Q19 stencils:

$$M_{ref} = \frac{c}{c_s} = \sqrt{3} \quad (3.8)$$

It should be noted that M_{ref} is the nondimensional lattice speed and can be thought as a pseudo Mach number and it is different from the free stream Mach number Ma_{∞} . It takes value depending on the selected solution stencil. The values are given in Appendix B for different types of stencils in 1D, 2D and 3D.

Pressure is related with density by the the isentropic relation, $p = c_s^2 \rho$. The relaxation parameter is a function of kinematic viscosity and the relation is given by

$$\tau_c = \nu / c_s^2 \quad (3.9)$$

Equation 3.9 relates the LBE with the physical world equations. The derivation of Navier Stokes Equations from LBE is given in Appendix A.

The equations in this section are the complete set of equations to apply the Lattice Boltzmann Method to any flow problem of single phase, single substance medium.

3.2 Non Dimensionalization

The Lattice Boltzmann Equation is the representation of the physics of a real system. However, during implementation, some aspects should be considered as in all CFD

problems. The units of the simulated quantities must be carefully chosen to reflect the real physical quantities. Also, the parameters like the grid resolution, time step etc. should be tuned to obtain a desired accuracy. It is a common practice to use nondimensional form of the governing equations in CFD to relate the problem to some nondimensional parameters like Reynolds number, Mach number etc..

The nondimensional form of the Discrete Velocity Boltzmann Equation is obtained by choosing the variables L_{ref} as the reference length, c as the reference velocity and free stream density ρ_∞ as the reference density. The time used for non-dimensionalization is the ratio of reference length to reference velocity. The dimensional and non-dimensional quantities are given in Table 3.1.

Table 3.1: Dimensional and non-dimensional forms

	dimensional	non-dimensional
length	x	$\hat{x} = \frac{x}{L_{ref}}$
time	t	$\hat{t} = \frac{tc}{L_{ref}}$
velocity	\vec{u}	$\hat{u} = \frac{\vec{u}}{c}$
discrete velocity	\vec{e}_α	$\hat{e}_\alpha = \frac{\vec{e}_\alpha}{c}$
distribution function	f_α	$\hat{f}_\alpha = \frac{f_\alpha}{\rho_\infty}$
macroscopic density	ρ	$\hat{\rho} = \frac{\rho}{\rho_\infty}$
pressure	p	$\hat{p} = \frac{P}{\rho c_s^2}$
kinematic viscosity	ν	$\hat{\nu} = \frac{\nu}{cL_{ref}}$
nondimensional relaxation	τ_c	$\hat{\tau} = \frac{\tau_c}{L_{ref}/c}$
free stream speed	U_∞	$\hat{U}_\infty = \frac{U_\infty}{c}$
pseudo Mach number		$M_{ref} = \frac{c}{c_s}$
free stream Mach number	M_∞	$\frac{U_\infty}{c_s}$

Starting with equation 3.4 combined with the continuous form of equation 3.6, the following equation is obtained:

$$\frac{\rho_\infty \partial \frac{f}{\rho_\infty}}{\frac{L_{ref}}{c} \partial \frac{t}{t^*}} + \frac{\vec{e}}{c} \cdot \rho_\infty \frac{L_{ref} \vec{\nabla} f}{L_{ref} \rho_\infty} = -\frac{1}{\tau_c} \left[f - w\rho \left[1 + \frac{\vec{e} \cdot \vec{u}}{c_s^2} + \frac{1}{2} \frac{(\vec{e} \cdot \vec{u})^2}{c_s^4} - \frac{1}{2} \frac{\vec{u}^2}{c_s^2} \right] \right] \quad (3.10)$$

where the reference time $t^* = \frac{L_{ref}}{c}$.

$$\frac{\rho_\infty \partial \hat{f}}{t^* \partial \hat{t}} + \frac{\rho_\infty}{t^*} \hat{e} \cdot \hat{\nabla} \hat{f} = -\frac{1}{\tau_c} \left[\hat{f} - w\rho \left[1 + \frac{\hat{e} \cdot \hat{u}}{c_s^2} + \frac{1}{2} \frac{(\hat{e} \cdot \hat{u})^2}{c_s^4} - \frac{1}{2} \frac{\hat{u}^2}{c_s^2} \right] \right] \quad (3.11)$$

After doing some algebra the continuous nondimensional Lattice Boltzmann Equation is obtained:

$$\frac{\partial \hat{f}}{\partial \hat{t}} + \hat{e} \cdot \hat{\nabla} \hat{f} = -\frac{1}{\hat{\tau}} \left[\hat{f} - w\hat{\rho} \left[1 + M_{ref}^2 (\hat{e} \cdot \hat{u}) + \frac{M_{ref}^4}{2} (\hat{e} \cdot \hat{u})^2 - \frac{M_{ref}^2}{2} \hat{u}^2 \right] \right] \quad (3.12)$$

Equation 3.12 can be written for each lattice direction depending on the stencil selected for the solution and the physical problem itself.

$$\frac{\partial \hat{f}_\alpha}{\partial \hat{t}} + \hat{e}_\alpha \cdot \hat{\nabla} \hat{f}_\alpha = -\frac{1}{\hat{\tau}} \left[\hat{f}_\alpha - \hat{f}_\alpha^{eq} \right] \quad (3.13)$$

The complementary equations for the system can be written as follows:

$$\hat{\rho} = \sum_{\alpha} \hat{f}_\alpha \quad \text{and} \quad \hat{\rho} \hat{u} = \sum_{\alpha} \hat{f}_\alpha \hat{e}_\alpha \quad (3.14)$$

$$\tau_c = \nu / c_s^2 \Rightarrow \hat{\tau} = \hat{\nu} M_{ref}^2 \quad (3.15)$$

The relation between the physical simulation and the nondimensional set of equations are obtained from the Reynolds number through the nondimensional kinematic viscosity $\hat{\nu}$.

$$\begin{aligned}
Re &= \frac{\rho_\infty U_\infty L_{ref}}{\mu} = \frac{U_\infty L_{ref}}{\nu} \\
Re &= \frac{U_\infty L_{ref}}{\nu} \frac{c_s}{c_s} \frac{c}{c} = \frac{\frac{U_\infty}{c_s}}{\frac{\nu}{c L_{ref}}} \frac{c_s}{c} = \frac{M_\infty}{\hat{\nu}} \frac{1}{M_{ref}} \\
\Rightarrow \hat{\nu} &= \frac{M_\infty}{Re} \frac{1}{M_{ref}}
\end{aligned} \tag{3.16}$$

M_∞ is the Mach number of the free stream and Re is the Reynolds number depending on the reference length L_{ref} and free stream velocity magnitude U_∞ .

The solutions to the incompressible Navier-Stokes equations depend only on the Reynolds number (Re). Hence, for the LBM M_∞ is chosen arbitrarily in the incompressible region ($M \leq 0.3$, usually taken $M = 0.2$) when the transient is not important.

For simplicity, in the rest of this manuscript, the hats are omitted and all parameters as well as the variables are assumed to be nondimensional unless stated otherwise explicitly.

3.3 Finite Volume Formulation

A cell centered finite volume formulation used by Peng et al. [38], Stiebler et al. [42], Patil et al. [43] and Zarghami et al. [66] is implemented in this manuscript. The standard Lattice Boltzmann Equation is integrated over the control volume with the following assumptions:

- In the cell centered finite volume method the cells serve directly as control volumes containing the unknown distribution functions.
- The equilibrium distribution function and the distribution function are assumed to be constant inside the control volume for each α direction.
- The fluxes are calculated by the distribution function values on the boundary of the control volumes. The value of the distribution function is calculated by some interpolation schemes using the neighboring cells.

The control volume with the quadrilateral elements are shown in Figure 3.1.

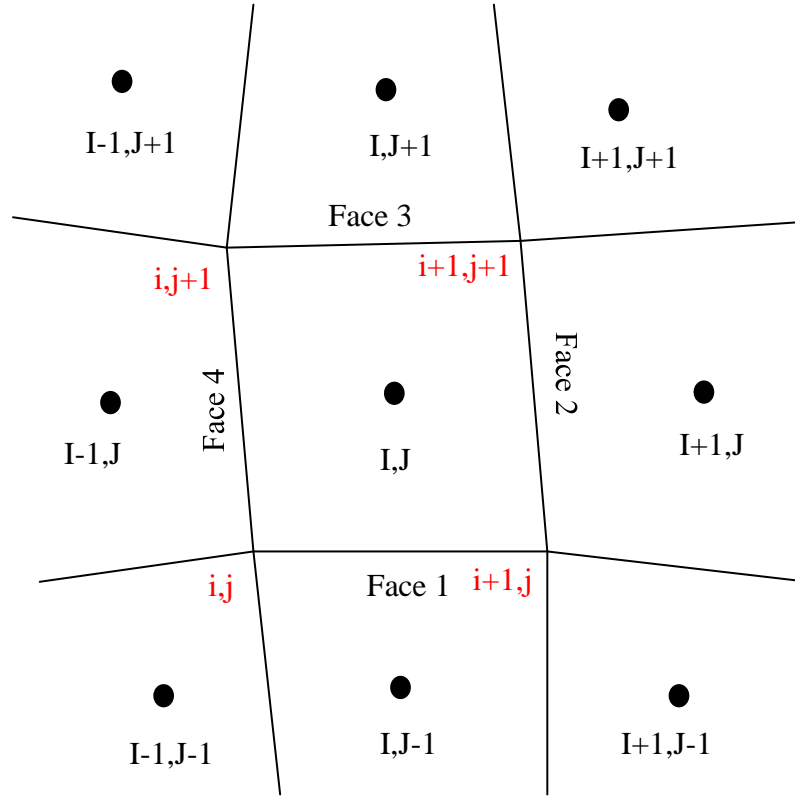


Figure 3.1: 2D Quadrilateral control volume elements

The standard discrete velocity Lattice Boltzmann Equation when integrated over a control volume turns into equation 3.17 . The derivation is assumed to be performed for a unit depth 3D element.

$$\int_V \left[\frac{\partial f_\alpha}{\partial t} \right] dV + \int_V \left[\vec{e}_\alpha \cdot \vec{\nabla} f_\alpha \right] dV = \int_V \left[-\frac{1}{\tau} (f_\alpha - f_\alpha^{eq}) \right] dV \quad (3.17)$$

the second term can be rewritten using the following relation:

$$\vec{\nabla} (f_\alpha \vec{e}_\alpha) = \vec{e}_\alpha \cdot \vec{\nabla} f_\alpha + f_\alpha \vec{\nabla} \cdot \vec{e}_\alpha \quad (3.18)$$

Since \vec{e}_α is constant vector for each alpha direction throughout the solution domain for all cells, equation 3.18 can be written as

$$\vec{\nabla} (f_\alpha \vec{e}_\alpha) = \vec{e}_\alpha \cdot \vec{\nabla} f_\alpha \quad (3.19)$$

Equation 3.19 is substituted in equation 3.17.

$$\int_V \left[\frac{\partial f_\alpha}{\partial t} \right] dV + \int_V \left[\vec{\nabla} \cdot (f_\alpha \vec{e}_\alpha) \right] dV = \int_V \left[-\frac{1}{\tau} (f_\alpha - f_\alpha^{eq}) \right] dV \quad (3.20)$$

Recall that the Gauss divergence theorem states that the outward flux of a vector field through a closed surface is equal to the volume integral of the divergence of the vector field over the volume enclosed by the surface:

$$\int_S (\vec{r} \cdot \vec{n}) dS = \int_\Omega (\vec{\nabla} \cdot \vec{r}) d\Omega \quad (3.21)$$

The second term, which is the advection of the distribution function, turns in to a flux term by applying the Gauss divergence theorem. After performing some algebra for the quadrilateral elements used in implementation, equation 3.17 can be rewritten as:

$$\int_V \left[\frac{\partial f_\alpha}{\partial t} \right] dV + \int_S [(f_{\alpha,i} \vec{e}_\alpha \cdot \vec{n}_i)] dS = \int_V \left[-\frac{1}{\tau} (f_\alpha - f_\alpha^{eq}) \right] dV \quad (3.22)$$

where for the quadrilateral or hexagonal control volume the integral of the second term can be rewritten by using a finite summation rather than using the integral:

$$\frac{\partial f_\alpha}{\partial t} + \frac{1}{V} \sum_{i=1}^{\#faces} f_{\alpha,i} (\vec{e}_\alpha \cdot \vec{n}_i) A_i = -\frac{1}{\tau} (f_\alpha - f_\alpha^{eq}) \quad (3.23)$$

where V is the volume of the cell, A is the face area and \vec{n} is the unit vector pointing outward of the cell on the related face. The summation is performed on all faces of the control volume, in which $\#faces = 4$ for the quadrilateral element for the 2D implementation and $\#faces = 6$ for the hexagonal element in 3D implementation.

3.3.1 MUSCLE Scheme for 2D and 3D faces

In equation 3.23, the distribution functions on the faces are needed to evaluate the flux term. It is assumed that the distribution functions f_α are constant on the face. In order to determine the distribution function values on the faces, 2^{nd} order accurate MUSCL (Monotone Upstream centered Scheme for Conservation Laws) scheme is used. In Figure 3.1, quadrilateral cell and face numbering are shown. The cells to be

used in flux calculations are defined by the sign of the product of lattice velocity and the face normal. The cell in focus has the indexes I, J . For each face the f_α values are calculated as follows:

$$\text{face 1 } f_{\alpha[I,J]} = \begin{cases} f_{\alpha[I,J]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J]} - f_{\alpha[I,J-1]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J+1]} - f_{\alpha[I,J]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face1} \geq 0 \\ f_{\alpha[I,J-1]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J]} - f_{\alpha[I,J-1]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J-1]} - f_{\alpha[I,J-2]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face1} < 0 \end{cases} \quad (3.24)$$

$$\text{face 2 } f_{\alpha[I,J]} = \begin{cases} f_{\alpha[I,J]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I+1,J]} - f_{\alpha[I,J]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J]} - f_{\alpha[I-1,J]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face2} \geq 0 \\ f_{\alpha[I+1,J]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I+1,J]} - f_{\alpha[I,J]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I+2,J]} - f_{\alpha[I+1,J]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face2} < 0 \end{cases} \quad (3.25)$$

$$\text{face 3 } f_{\alpha[I,J]} = \begin{cases} f_{\alpha[I,J]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J+1]} - f_{\alpha[I,J]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J]} - f_{\alpha[I,J-1]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face2} \geq 0 \\ f_{\alpha[I,J+1]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J+1]} - f_{\alpha[I,J]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J+2]} - f_{\alpha[I,J+1]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face2} < 0 \end{cases} \quad (3.26)$$

$$\text{face 4 } f_{\alpha[I,J]} = \begin{cases} f_{\alpha[I,J]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J]} - f_{\alpha[I-1,J]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I+1,J]} - f_{\alpha[I,J]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face4} \geq 0 \\ f_{\alpha[I-1,J]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J]} - f_{\alpha[I-1,J]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I-1,J]} - f_{\alpha[I-2,J]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face4} < 0 \end{cases} \quad (3.27)$$

The term ε can be set to 0 for a first order accurate interpolation. For $\varepsilon = 1$ the term κ determines the spatial accuracy. A second order spatial accuracy is obtained by setting κ term to $1/3$ [60, p. 96]. Using equations 3.24, 3.25, 3.26 and 3.27, the second term of equation 3.23 for cell $[I, J]$ of a quadrilateral element can be rewritten as follows:

$$F_{\alpha[I,J]} = \begin{aligned} & \left[(f_{\alpha})_{@face1} (\vec{e}_{\alpha} \cdot \vec{n}_{face1}) A_{face1} \right]_{[I,J]} + \dots \\ & \left[(f_{\alpha})_{@face2} (\vec{e}_{\alpha} \cdot \vec{n}_{face2}) A_{face2} \right]_{[I,J]} + \dots \\ & \left[(f_{\alpha})_{@face3} (\vec{e}_{\alpha} \cdot \vec{n}_{face3}) A_{face3} \right]_{[I,J]} + \dots \\ & \left[(f_{\alpha})_{@face4} (\vec{e}_{\alpha} \cdot \vec{n}_{face4}) A_{face4} \right]_{[I,J]} \end{aligned} \quad (3.28)$$

For a 3D hexagonal element, the flux calculation can be formulated in a similar manner. In Figure 3.2, a control volume (cell) having I, J, K indexes is shown.

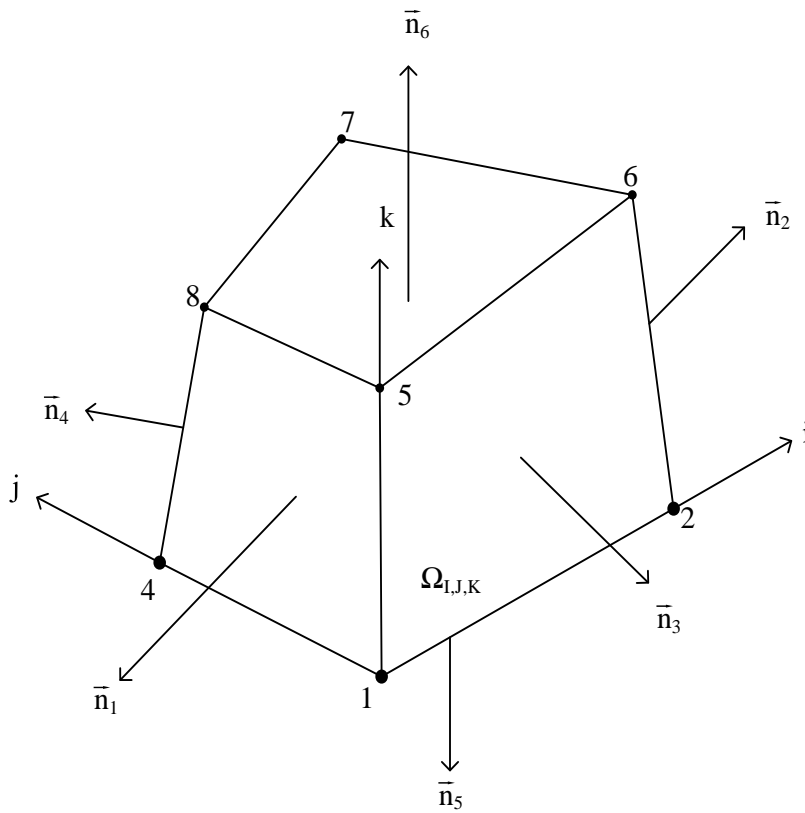


Figure 3.2: 3D hexagonal control volume

$$\text{face 1 } f_{\alpha[I,J,K]} = \begin{cases} f_{\alpha[I,J,K]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J,K]} - f_{\alpha[I-1,J,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I+1,J,K]} - f_{\alpha[I,J,K]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face1} \geq 0 \\ f_{\alpha[I-1,J,K]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J,K]} - f_{\alpha[I-1,J,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I-1,J,K]} - f_{\alpha[I-2,J,K]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face1} < 0 \end{cases} \quad (3.29)$$

$$\text{face 2 } f_{\alpha[I,J,K]} = \begin{cases} f_{\alpha[I,J,K]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I+1,J,K]} - f_{\alpha[I,J,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J,K]} - f_{\alpha[I-1,J,K]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face2} \geq 0 \\ f_{\alpha[I-1,J,K]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I+1,J,K]} - f_{\alpha[I,J,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I+2,J,K]} - f_{\alpha[I+1,J,K]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face2} < 0 \end{cases} \quad (3.30)$$

$$\text{face 3 } f_{\alpha[I,J,K]} = \begin{cases} f_{\alpha[I,J,K]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J,K]} - f_{\alpha[I,J-1,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J+1,K]} - f_{\alpha[I,J,K]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face3} \geq 0 \\ f_{\alpha[I,J-1,K]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J,K]} - f_{\alpha[I,J-1,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J-1,K]} - f_{\alpha[I,J-2,K]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face3} < 0 \end{cases} \quad (3.31)$$

$$\text{face 4 } f_{\alpha[I,J,K]} = \begin{cases} f_{\alpha[I,J,K]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J+1,K]} - f_{\alpha[I,J,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J,K]} - f_{\alpha[I,J-1,K]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face4} \geq 0 \\ f_{\alpha[I,J+1,K]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J+1,K]} - f_{\alpha[I,J,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J+2,K]} - f_{\alpha[I,J+1,K]}) \right) & \text{for } \vec{e}_\alpha \cdot \vec{n}_{face4} < 0 \end{cases} \quad (3.32)$$

$$\text{face 5 } f_{\alpha[I,J,K]} = \begin{cases} f_{\alpha[I,J,K]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J,K]} - f_{\alpha[I,J,K-1]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J,K+1]} - f_{\alpha[I,J,K]}) \right) & \text{for } \vec{e}_{\alpha} \cdot \vec{n}_{face5} \geq 0 \\ f_{\alpha[I,J,K-1]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J,K]} - f_{\alpha[I,J,K-1]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J,K-1]} - f_{\alpha[I,J,K-2]}) \right) & \text{for } \vec{e}_{\alpha} \cdot \vec{n}_{face5} < 0 \end{cases} \quad (3.33)$$

$$\text{face 6 } f_{\alpha[I,J,K]} = \begin{cases} f_{\alpha[I,J,K]} + \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J,K+1]} - f_{\alpha[I,J,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J,K]} - f_{\alpha[I,J,K-1]}) \right) & \text{for } \vec{e}_{\alpha} \cdot \vec{n}_{face6} \geq 0 \\ f_{\alpha[I,J,K+1]} - \frac{\varepsilon}{4} \left((1 + \kappa) \cdot (f_{\alpha[I,J,K+1]} - f_{\alpha[I,J,K]}) + \dots \right. \\ \quad \left. (1 - \kappa) \cdot (f_{\alpha[I,J,K+2]} - f_{\alpha[I,J,K+1]}) \right) & \text{for } \vec{e}_{\alpha} \cdot \vec{n}_{face6} < 0 \end{cases} \quad (3.34)$$

Using equations 3.29, 3.30, 3.31, 3.32, 3.33 and 3.34 the second term of equation 3.23 for cell $[I, J, K]$ of a hexagonal element can be rewritten as follows:

$$F_{\alpha[I,J,K]} = \begin{aligned} & \left[(f_{\alpha})_{@face1} (\vec{e}_{\alpha} \cdot \vec{n}_{face1}) A_{face1} \right]_{[I,J,K]} + \dots \\ & \left[(f_{\alpha})_{@face2} (\vec{e}_{\alpha} \cdot \vec{n}_{face2}) A_{face2} \right]_{[I,J,K]} + \dots \\ & \left[(f_{\alpha})_{@face3} (\vec{e}_{\alpha} \cdot \vec{n}_{face3}) A_{face3} \right]_{[I,J,K]} + \dots \\ & \left[(f_{\alpha})_{@face4} (\vec{e}_{\alpha} \cdot \vec{n}_{face4}) A_{face4} \right]_{[I,J,K]} + \dots \\ & \left[(f_{\alpha})_{@face5} (\vec{e}_{\alpha} \cdot \vec{n}_{face5}) A_{face5} \right]_{[I,J,K]} + \dots \\ & \left[(f_{\alpha})_{@face6} (\vec{e}_{\alpha} \cdot \vec{n}_{face6}) A_{face6} \right]_{[I,J,K]} \end{aligned} \quad (3.35)$$

Equation 3.28 for 2D and equation 3.35 for 3D collect all the flux from the surfaces under a numerical flux integral term which is the net transport of the distribution function through the boundaries. Then the equation 3.17 is reconstructed by substituting equation 3.35.¹

$$\frac{\partial f_{\alpha[I,J,K]}}{\partial t} + \frac{1}{V_{[I,J]}} F_{\alpha[I,J,K]} = -\frac{1}{\tau} (f_{\alpha[I,J,K]} - f_{\alpha[I,J,K]}^{eq}) \quad (3.36)$$

¹ K index is dropped or set to 1 for 2D usage

3.4 Boundary Conditions

Guo and Zheng [67] proposed an extrapolation method, to implement the boundary conditions. In the proposed method the boundary conditions are defined by the known macroscopic quantities instead of the distribution functions. Furthermore, Guo and Zheng has also shown that the extrapolation method is second order accurate and has better numerical stability compared to the alternative methods presented in their paper.

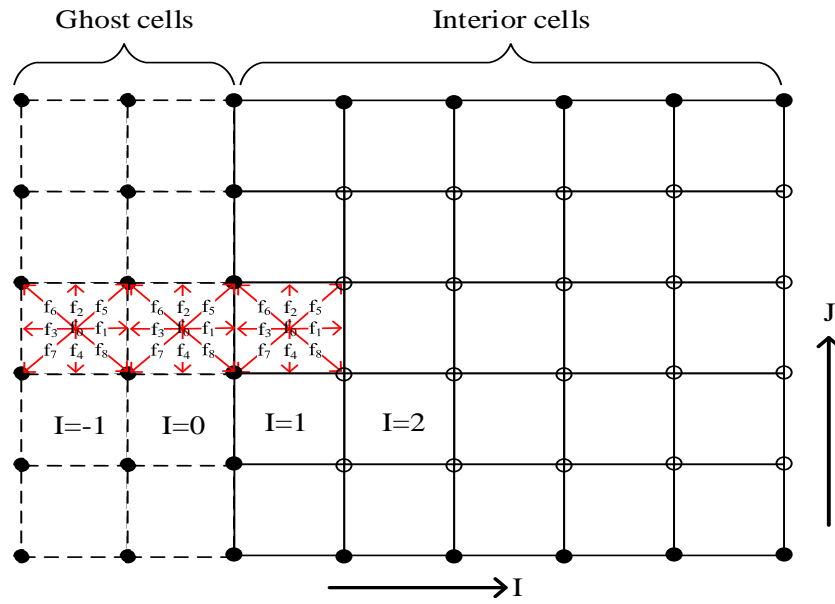


Figure 3.3: Ghost cell implementation

Two levels of ghost cells are used for for the boundary conditions of the solution domain and appropriate macroscopic variables are set depending on the boundary condition given. A representative solution domain for a 2D boundary is shown in Figure 3.3. The dots are representing the vertices of the cells. The empty dots are the fluid inside the solution domain, and the filled ones are either the boundary vertices or ghost cell vertices. The distribution function, equilibrium distribution function and the macroscopic quantities are calculated at the center of the control volume. The equilibrium distribution functions of the ghost cells are calculated using the macroscopic quantities and the distribution function is calculated by adding the 0^{th} order extrapolation (ghost cells shown by dashed lines in Figure 3.3 use the same nonequi-

librium distribution functions as the neighboring interior cells) of the non-equilibrium distribution function.

$$f_{\alpha[ghost]} = f_{\alpha[ghost]}^{eq} + f_{\alpha[ghost]}^{neq} \quad (3.37)$$

where $f_{\alpha[ghost]}^{neq} = f_{\alpha[interior]} - f_{\alpha[interior]}^{eq}$.

The term $f_{\alpha[ghost]}^{neq}$ in equation 3.37 is the nonequilibrium part of the distribution function of the cells next to the boundary (interior) inside the flow domain. In the proceeding subsections, how the boundary conditions are treated by using a quadrilateral cell will be described. The examples are given only for faces in either “I” direction or “J” direction, however implementation for other directions are easy. It has to be noted that, the calculation stencil is fixed with respect to a inertial frame, however cells might be oriented differently. That is all coordinates of the vertices can be changed when moving from vertex $P_{i,j,k}$ to $P_{i+1,j,k}$. The examples are for given for a 2D problem but it is straight forward to apply the boundary conditions to 3D.

3.4.1 Velocity Inlet Boundary Condition

In Figure 3.4, it is assumed that an inlet velocity U_{ref} is defined as a velocity inlet boundary condition. To calculate the numerical integral flux for the cell I, J , the distribution function value at the inlet face must be known. It is stated early in the manuscript that, MUSCL scheme will be used to obtain the value of the distribution function at the faces. Therefore, the values of distribution functions at the cell center of the ghost layers must be determined.

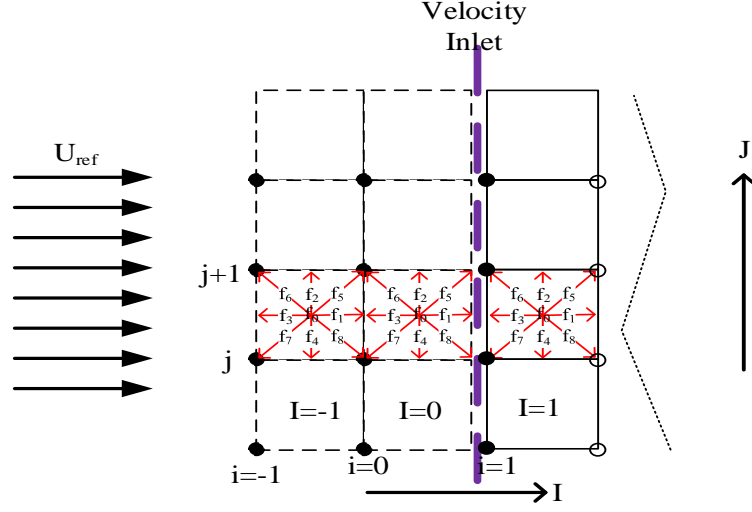


Figure 3.4: Velocity inlet boundary condition

$$\begin{aligned} f_{\alpha[I-1,J]} &= f_{\alpha[I-1,J]}^{eq} + f_{\alpha[I,J]}^{neq} \\ f_{\alpha[I-2,J]} &= f_{\alpha[I-2,J]}^{eq} + f_{\alpha[I,J]}^{neq} \end{aligned} \quad (3.38)$$

$f_{\alpha[I,J]}^{neq}$ is calculated as follows:

$$\begin{aligned} f_{\alpha[I,J]}^{neq} &= \left[f_{\alpha[I,J]} - f_{\alpha[I,J]}^{eq} \right] \\ f_{\alpha[I,J]}^{neq} &= \left[f_{\alpha[I,J]} - \rho_{[I,J]} w_{\alpha} \left[1 + 3 (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]} + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]}^2 - \frac{3}{2} \vec{u}_{[I,J]}^2 \right] \right] \end{aligned} \quad (3.39)$$

and the equilibrium distribution function for the ghost cells are calculated as follows:

$$\begin{aligned} f_{\alpha[I-1,J]}^{eq} &= \rho_{[I-1,J]} w_{\alpha} \left[1 + 3 (\vec{e}_{\alpha} \cdot \vec{U}_{ref})_{[I-1,J]} + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{U}_{ref})_{[I-1,J]}^2 - \frac{3}{2} \vec{U}_{ref}^2 \right] \\ f_{\alpha[I-2,J]}^{eq} &= \rho_{[I-2,J]} w_{\alpha} \left[1 + 3 (\vec{e}_{\alpha} \cdot \vec{U}_{ref})_{[I-2,J]} + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{U}_{ref})_{[I-2,J]}^2 - \frac{3}{2} \vec{U}_{ref}^2 \right] \end{aligned} \quad (3.40)$$

where \vec{U}_{ref} is the free stream (reference) velocity and must be defined as a vector with respect to the inertial reference frame. It has to be noted that the density $\rho_{[I-2,J]} = \rho_{[I-1,J]} = \rho_{[I,J]}$ is taken from the interior cell.

3.4.2 Pressure Outlet Boundary Condition

In Figure 3.4, it is assumed that an pressure outlet boundary condition is given. To calculate the numerical integral flux for the cell I, J , the distribution function value at the outlet face must be known. Again, MUSCL scheme will be used to obtain the value of the distribution function at the faces, that is why the values of distribution functions at the cell center of the ghost layers must be determined.

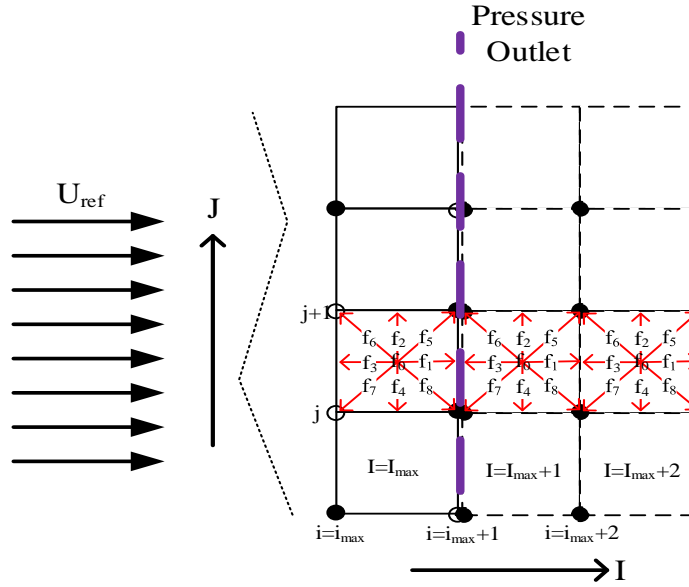


Figure 3.5: Pressure outlet boundary condition

$$\begin{aligned} f_{\alpha[I+1,J]} &= f_{\alpha[I+1,J]}^{eq} + f_{\alpha[I,J]}^{neq} \\ f_{\alpha[I+2,J]} &= f_{\alpha[I+2,J]}^{eq} + f_{\alpha[I,J]}^{neq} \end{aligned} \quad (3.41)$$

$f_{\alpha[I,J]}^{neq}$ is calculated as follows:

$$\begin{aligned} f_{\alpha[I,J]}^{neq} &= \left[f_{\alpha[I,J]} - f_{\alpha[I,J]}^{eq} \right] \\ f_{\alpha[I,J]}^{neq} &= \left[f_{\alpha[I,J]} - \rho_{[I,J]} w_{\alpha} \left[1 + 3 (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]} + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]}^2 - \frac{3}{2} \vec{u}_{[I,J]}^2 \right] \right] \end{aligned} \quad (3.42)$$

and the equilibrium distribution function for the ghost cells are calculated as follows:

$$\begin{aligned} f_{\alpha[I+1,J]}^{eq} &= \rho_{[I+1,J]} w_{\alpha} \left[1 + 3 (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]} + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]}^2 - \frac{3}{2} \vec{u}^2 \right] \\ f_{\alpha[I+2,J]}^{eq} &= \rho_{[I+2,J]} w_{\alpha} \left[1 + 3 (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]} + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]}^2 - \frac{3}{2} \vec{u}^2 \right] \end{aligned} \quad (3.43)$$

It has to be noted that the density $\rho_{[I+2,J]} = \rho_{[I+1,J]} = \rho_{\infty}$ is taken from the free stream while the macroscopic velocity for the ghost cells are taken from the the interior cell.

3.4.3 Wall / Moving Wall Boundary Condition

In Figure 3.6, it is assumed that a wall (either stationary or moving) boundary condition is given. To calculate the numerical integral flux for the cell I, J , the distribution function value at the wall face must be known. Using the same method, as in previous sections, the value of the distribution function at the faces are obtained by using the values of distribution functions at the cell center of the ghost layers. The distribution functions at the ghost cell centers are calculated by equation 3.44.

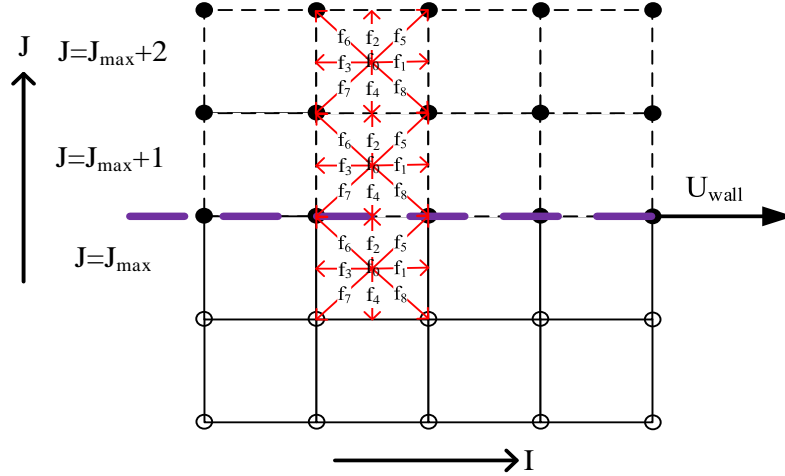


Figure 3.6: Wall / moving wall boundary condition

$$\begin{aligned} f_{\alpha[I,J+1]} &= f_{\alpha[I,J+1]}^{eq} + f_{\alpha[I,J]}^{neq} \\ f_{\alpha[I,J+2]} &= f_{\alpha[I,J+2]}^{eq} + f_{\alpha[I,J]}^{neq} \end{aligned} \quad (3.44)$$

$f_{\alpha[I,J]}^{neq}$ is calculated as follows:

$$f_{\alpha[I,J]}^{neq} = \left[f_{\alpha[I,J]} - f_{\alpha[I,J]}^{eq} \right] \quad (3.45)$$

$$f_{\alpha[I,J]}^{neq} = \left[f_{\alpha[I,J]} - \rho_{[I,J]} w_{\alpha} \left[1 + 3 (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]} + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{u})_{[I,J]}^2 - \frac{3}{2} \vec{u}_{[I,J]}^2 \right] \right]$$

and the equilibrium distribution function for the ghost cells are calculated as follows:

$$f_{\alpha[I,J+1]}^{eq} = \rho_{[I,J+1]} w_{\alpha} \left[1 + 3 (\vec{e}_{\alpha} \cdot \vec{U}_{wall}) + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{U}_{wall})^2 - \frac{3}{2} \vec{U}_{wall}^2 \right] \quad (3.46)$$

$$f_{\alpha[I+2,J]}^{eq} = \rho_{[I+2,J]} w_{\alpha} \left[1 + 3 (\vec{e}_{\alpha} \cdot \vec{U}_{wall}) + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{U}_{wall})^2 - \frac{3}{2} \vec{U}_{wall}^2 \right]$$

It has to be noted that the density $\rho_{[I,J+2]} = \rho_{[I,J+2]} = \rho_{[I,J]}$ is taken from interior cell while the wall velocity is used for the calculation of the equilibrium distribution function for the ghost cells. If the wall is stationary then the equilibrium distribution functions for the ghost cells became as follows:

$$f_{\alpha[I,J+1]}^{eq} = \rho_{[I,J]} w_{\alpha} \quad (3.47)$$

$$f_{\alpha[I+2,J]}^{eq} = \rho_{[I,J]} w_{\alpha}$$

3.5 Implicit Formulation for Steady State

Finite volume formulation for the discrete Lattice Boltzmann Equation for α^{th} direction is given as follows:

$$\frac{\partial f_{\alpha}}{\partial t} + \frac{1}{V} F_{\alpha} = -\frac{1}{\tau} (f_{\alpha} - f_{\alpha}^{eq}) \quad (3.48)$$

and it can be rearranged as follows:

$$\frac{\partial f_{\alpha}}{\partial t} = -\frac{1}{V} F_{\alpha} - \frac{1}{\tau} (f_{\alpha} - f_{\alpha}^{eq}) \quad (3.49)$$

The above equation needs to be solved for each lattice direction α and f^{eq} is calculated by using the known macroscopic quantities ρ , and \vec{u} for present time level. For the explicit time integration scheme distribution function for each direction can

be calculated independently for the next time step. The link between the distribution functions is equilibrium distribution function f_α^{eq} . However, for the implicit formulation, the right hand side of equation is calculated for the next time step. The density ρ and the velocity \vec{u} is unknown. It is possible to formulate each discrete velocity direction separately and linearize the right hand side of equation 3.49. Although, the solution seems to be straight forward, but it does not work. The correct method would be to solve the distribution functions for all directions simultaneously. The first step is to combine all the equations for each α^{th} direction. The distribution functions, equilibrium distribution functions and numerical flux integral term are written as vectors with n elements:

$$\vec{f} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f_{n-1} \end{bmatrix} \quad \vec{f}^{eq} = \begin{bmatrix} f_0^{eq} \\ f_1^{eq} \\ f_2^{eq} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f_{n-1}^{eq} \end{bmatrix} \quad \vec{F} = \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ F_{n-1} \end{bmatrix}$$

The length of the vectors depends on the chosen lattice stencil. In this manuscript, D2Q9 and D3Q19 stencils are utilized. Using the vectorial form, the discrete Lattice Boltzmann Equations for a cell can be written as follows:

$$\frac{\partial}{\partial t} \vec{f} + \frac{1}{V} \vec{F} = -\frac{1}{\tau} (\vec{f} - \vec{f}^{eq}) \quad (3.50)$$

The equation 3.50 can be re-organized by collecting all the terms to the right hand side.

$$\frac{\partial}{\partial t} \vec{f} = -\frac{1}{V} \vec{F} - \frac{1}{\tau} (\vec{f} - \vec{f}^{eq}) \quad (3.51)$$

A general discretization scheme for equation 3.36 is given in equation 3.52. By setting

the parameter $\beta = 1$, a fully implicit backward difference method is obtained.

$$\frac{\vec{f}^{(n+1)} - \vec{f}^{(n)}}{\Delta t} = \beta \cdot R\vec{H}S^{(n+1)} + (1 - \beta) \cdot R\vec{H}S^{(n)} \quad (3.52)$$

In equation 3.52, the $R\vec{H}S$ term is equal to $-\frac{1}{V}\vec{F} - \frac{1}{\tau}(\vec{f} - \vec{f}^{eq})$ and is needed to be calculated at the $(n + 1)$ time level where the macroscopic quantities and distribution functions are unknown. A linear approximation to the RHS of the equation can be written as follows:

$$R\vec{H}S^{n+1} \equiv R\vec{H}S^n + \left(\frac{\partial}{\partial \vec{f}} R\vec{H}S \right)^n (\Delta \vec{f})^n \quad (3.53)$$

where $\Delta \vec{f}^{(n)}$ is defined as $\vec{f}^{(n+1)} - \vec{f}^{(n)}$. Combining equations 3.52 and 3.53 and substituting the RHS into combined equation, equation 3.54 is obtained and it is written more explicitly in equation 3.55.

$$\frac{\Delta \vec{f}^{(n)}}{\Delta t} = \left[-\frac{1}{V}\vec{F} - \frac{1}{\tau}(\vec{f} - \vec{f}^{eq}) \right]^{(n)} + \left[\frac{\partial}{\partial \vec{f}} \left(-\frac{1}{V}\vec{F} - \frac{1}{\tau}(\vec{f} - \vec{f}^{eq}) \right) \right]^{(n)} \cdot \Delta \vec{f}^{(n)} \quad (3.54)$$

$$\Delta \vec{f} = -\frac{\Delta t}{V}\vec{F} - \frac{\Delta t}{\tau}\vec{f} + \frac{\Delta t}{\tau}\vec{f}^{eq} - \frac{\Delta t}{V}\frac{\partial}{\partial \vec{f}}\vec{F} \cdot \Delta \vec{f} - \frac{\Delta t}{\tau}\frac{\partial}{\partial \vec{f}}\vec{f} \cdot \Delta \vec{f} + \frac{\Delta t}{\tau}\frac{\partial}{\partial \vec{f}}\vec{f}^{eq} \cdot \Delta \vec{f} \quad (3.55)$$

Since all the terms of the equation 3.55 is at time level (n) , the superscript is dropped for simplicity. Collecting the $\Delta \vec{f}$ terms in the left side equation 3.56 is obtained:

$$\underbrace{\Delta \vec{f} + \frac{\Delta t}{\tau}\frac{\partial}{\partial \vec{f}}\vec{f} \cdot \Delta \vec{f}}_1 - \underbrace{\frac{\Delta t}{\tau}\frac{\partial}{\partial \vec{f}}\vec{f}^{eq} \cdot \Delta \vec{f}}_2 + \underbrace{\frac{\Delta t}{V}\frac{\partial}{\partial \vec{f}}\vec{F} \cdot \Delta \vec{f}}_3 = -\underbrace{\frac{\Delta t}{V}\vec{F}}_4 - \underbrace{\frac{\Delta t}{\tau}\vec{f}}_5 + \underbrace{\frac{\Delta t}{\tau}\vec{f}^e}_6 \quad (3.56)$$

The terms numbered as 4 to 6 on the right hand side can be considered as the Residual term \vec{R} and there is no challenge to calculate them.

The term numbered as 1 includes the Jacobian of the distribution function itself which is an identity matrix.

The 2nd term includes the Jacobian of the Equilibrium Distribution Function (JEDF) which is a $n \times n$ non-zero matrix. The JEDF is the link between the distribution functions of discrete directions through macroscopic quantities. The formula of each element of the JEDF is given in equation 3.57 where α and β are dummy indexes.

$$\text{JEDF}_{\alpha,\beta} = \frac{\partial f_{\alpha}^{(eq)}}{\partial f_{\beta}} = \omega_{\alpha} \begin{bmatrix} 1 + 3\vec{e}_{\alpha} \cdot \vec{u} + \frac{9}{2} (\vec{e}_{\alpha} \cdot \vec{u}) - \frac{3}{2} (\vec{u} \cdot \vec{u}) + \dots \\ (3\vec{e}_{\alpha} + 9 (\vec{e}_{\alpha} \cdot \vec{u}) \vec{e}_{\alpha} - 3\vec{u}) \cdot (\vec{e}_{\beta} - \vec{u}) \end{bmatrix} \quad (3.57)$$

However, the term numbered as 3 includes the Jacobian of the numerical flux integral term $\frac{\partial}{\partial \vec{f}} \vec{F}$. It is a square $n \times n$ matrix where all the non-diagonal elements are zero. The Jacobian of the numerical flux integral term must be treated as an operator for $\Delta \vec{f}$. The $\Delta \vec{f}$ is taken from the cell in focus or the neighboring (first order upwind) cells according to sign of the face $\vec{n} \cdot \vec{e}_{\alpha}$ property [60, p. 194]. For NS equations Jacobian of the fluxed are divided into 2 as convective flux and viscous flux. For viscous flux at any face, the average value of the neighboring cells is used. For convective flux, first order upwind scheme is used.

Equation 3.56 can be rearranged as:

$$\left[I + \frac{\Delta t}{\tau} I - \frac{\Delta t}{\tau} \text{JEDF} \right] \Delta \vec{f} + \frac{\Delta t}{V} \frac{\partial}{\partial \vec{f}} \vec{F} \cdot \Delta \vec{f} = - \frac{\Delta t}{V} \vec{F} - \frac{\Delta t}{\tau} \vec{f} + \frac{\Delta t}{\tau} \vec{f}^{eq} \quad (3.58)$$

The generic resulting system of equations for a m by n element 2D domain is given in equation 3.59.

$$\left[A \right]_{m \cdot n, m \cdot n} \left[X \right]_{m \cdot n} = \left[B \right]_{m \cdot n} \quad (3.59)$$

where A is the square coefficient matrix, also called implicit operator matrix [60, p. 195]. The A matrix is a sparse penta-diagonal matrix. X is the unknown vector

and B is the residual vector. All the elements of A are 9×9 submatrices. Only the main diagonal, the upper and lower neighbor of the main diagonal and some distinct elements due to spatial derivatives are non zero. The shapes of matrix A , vector X and vector B are shown in Figure 3.7. All elements of X and B (each round shape) are 9 element column vectors of $\Delta \vec{f}$ and \vec{R} , respectively.

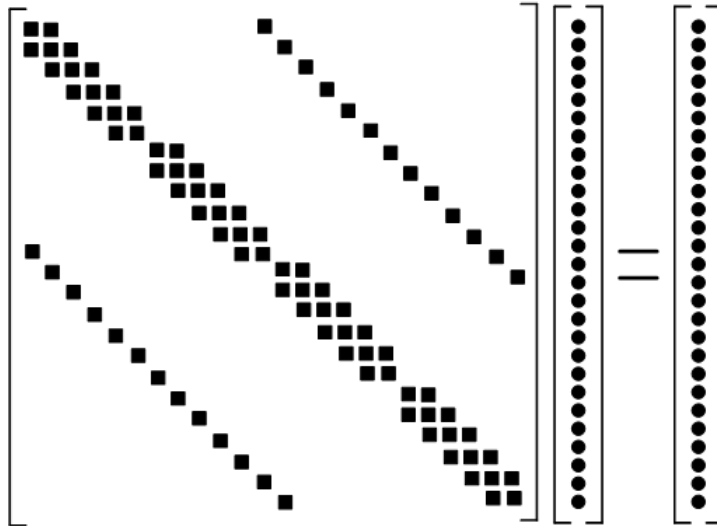


Figure 3.7: Implicit operator matrix

A similar analysis can be done for 3D domains with n, m, p elements in I, J, K directions.

$$[A]_{m \cdot n \cdot p, m \cdot n \cdot p} [X]_{m \cdot n \cdot p} = [B]_{m \cdot n \cdot p} \quad (3.60)$$

The equation system is solved by using the approximate factorization method presented in Appendix D. The solution is further accelerated using the local time step technique where each cell uses a different time step based on the local numerical stability criterion.

3.6 Local Time Stepping for Acceleration

The local time stepping technique is an approach to accelerate the convergence of steady state solutions. The idea of local time stepping is simple indeed. During iterations maximum allowable time step is used for each individual cell. The local time stepping technique incorporates with the explicit methods. Moreover, it is also possible to use the advantages of the technique for linearized implicit formulation. Local time stepping can be used in sub-iterations of dual time stepping schemes.

The local time stepping method invalidates the temporal accuracy for the transient part of the solution. For LBM, the CFL number is defined as:

$$CFL = \frac{\sqrt{3}\Delta t}{\Delta x} \quad (3.61)$$

where Δt is the nondimensional time and Δx is the nondimensional representative cell length. To use a global time step, minimum of Δt should be calculated for each control volume must be selected. A representative Δx is also selected as the smallest length for quadrilateral or hexagonal elements. More detailed explanation for the local time step method for LBM can be found in the work of Inamuro et al. [68] .

3.7 Dual Time Stepping for Transient/Unsteady Problems

Dual Time Stepping (DTS) approach can be used whether the base method is implicit or explicit. For the time-accurate solution of equation 3.36, DTS approach is used. First, the governing equation is put into a steady form in pseudo time, then the convergence acceleration techniques such as local time stepping or multigrid methods can be used for the pseudo time. In DTS scheme, the advancement in time is restricted by the desired temporal accuracy. The stability criteria for the high aspect ratio grid cells used for boundary layers and the stiffness coming from the small values of relaxation time does not have impact on DTS. A second order accurate base method for time integration is coupled with the DTS scheme following the method proposed by Jameson [69]. Equation 3.36 is modified by adding a pseudo time derivative to the Left Hand Side (LHS).

$$\frac{\partial f_{\alpha[i,j]}}{\partial t^*} + \frac{\partial f_{\alpha[i,j]}}{\partial t} + \frac{1}{V_{[i,j]}} F_{\alpha[i,j]} = -\frac{1}{\tau_c} (f_{\alpha[i,j]} - f_{\alpha[i,j]}^{eq}) \quad (3.62)$$

The term t^* is the pseudo time derivative of the distribution functions and must approach to zero during the DTS process. Equation 3.62 is discretized as follows:

$$\frac{f_{\alpha}^{(l+1)} - f_{\alpha}^{(l)}}{\Delta t^*} + \frac{3f_{\alpha}^{(n+1)} - 4f_{\alpha}^{(n)} + f_{\alpha}^{(n-1)}}{2\Delta t} = -\frac{1}{V} F_{\alpha}^{(l)} - \frac{1}{\tau_c} (f_{\alpha} - f_{\alpha}^{eq})^{(l)} \quad (3.63)$$

In equation 3.63, the superscripts (l) and (n) indicate the time level of the calculations to be performed. l is the pseudo time and n is the nondimensional lattice time. As indicated in the beginning of this section, DTS can be used for both implicit and explicit time integration methods. The time level of the RHS of the equation 3.63 defines the nature of the time integration. In this paper, the time level of the equation 3.63 is $(n+1)$. If sufficient number of integration in pseudo time is performed, the term $f_{\alpha}^{(l+1)} - f_{\alpha}^{(l)}$ will approach to 0 and the equation $f_{\alpha}^{(l+1)} = f_{\alpha}^{(n+1)}$ will hold. Then, the equation 3.63 can be modified as follows.

$$\frac{f_{\alpha}^{(l+1)} - f_{\alpha}^{(l)}}{\Delta t^*} + \frac{3f_{\alpha}^{(l+1)} - 4f_{\alpha}^{(n)} + f_{\alpha}^{(n-1)}}{2\Delta t} = -\frac{1}{V} F_{\alpha}^{(l+1)} - \frac{1}{\tau_c} (f_{\alpha} - f_{\alpha}^{eq})^{(l+1)} \quad (3.64)$$

Equation 3.64 can further be simplified by rearranging terms together and defining

$$\begin{aligned} \Delta f_{\alpha}^{(l)} &= f_{\alpha}^{(l+1)} - f_{\alpha}^{(l)} \\ \left(1 + \frac{3\Delta t^*}{2\Delta t}\right) \Delta f_{\alpha}^{(l)} &= -\frac{3\Delta t^*}{2\Delta t} f_{\alpha}^{(l)} - \Delta t^* \left[\frac{4f_{\alpha}^{(n)} - f_{\alpha}^{(n-1)}}{2\Delta t} - Q_{\alpha}^{(l+1)} \right] \\ Q_{\alpha}^{(l+1)} &= \frac{1}{V} F_{\alpha}^{(l+1)} + \frac{1}{\tau_c} (f_{\alpha} - f_{\alpha}^{eq})^{(l+1)} \end{aligned} \quad (3.65)$$

Equation 3.65 can be written in vector form and the terms in time level $(l+1)$ are linearized by a similar method described in section 3.5.

$$\begin{aligned} \left(1 + \frac{3\Delta t^*}{2\Delta t}\right) \Delta \vec{f}^{(l)} &= -\frac{3\Delta t^*}{2\Delta t} \vec{f}^{(l)} - \\ &\Delta t^* \left[\frac{4\vec{f}^{(n)} - \vec{f}^{(n-1)}}{2\Delta t} - \vec{Q}^{(n)} - \left(\frac{\partial}{\partial \vec{f}} \vec{Q}\right)^{(l)} \Delta \vec{f}^{(l)} \right] \end{aligned} \quad (3.66)$$

The resulting form of the equations will give exactly the similar systems of equations with different values of the coefficients as shown in Figure 3.7 . The sub iterations in the DTS can be solved using the same method for the steady state solutions.

3.8 Turbulence Modeling

The turbulence model used in this method is the Spalart-Allmaras (SA) one equation turbulence model based on the work of the Prandtl's [61] original one-equation model. The SA turbulence model [65] given in equation 3.67 is the transport of the kinematic eddy viscosity $\tilde{\nu}$. The model is widely used for external aerodynamics and turbo machinery. This model has been shown to give good results for adverse pressure gradients and boundary layers.

$$\frac{\partial \tilde{\nu}}{\partial t} = M(\tilde{\nu}) + P(\tilde{\nu}) - D(\tilde{\nu}) \quad (3.67)$$

Where

$$\begin{aligned} M(\tilde{\nu}) & \quad \text{advection diffusion term} \\ P(\tilde{\nu}) & \quad \text{production term} \\ D(\tilde{\nu}) & \quad \text{destruction term} \end{aligned}$$

The advection/diffusion term, production term and destruction terms are given in equation 3.68.

$$\begin{aligned} M(\tilde{\nu}) & = -(\vec{u} \cdot \vec{\nabla})\tilde{\nu} + \frac{1 + c_{b2}}{\sigma} \vec{\nabla} \cdot [(\nu + \tilde{\nu})\vec{\nabla}\tilde{\nu}] - \frac{c_{b2}}{\sigma}(\nu + \tilde{\nu})\nabla^2\tilde{\nu} \\ P(\tilde{\nu}) & = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} \\ D(\tilde{\nu}) & = [c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}[\frac{\tilde{\nu}}{d}]^2] \end{aligned} \quad (3.68)$$

The parameters that are used for calculating the advection/diffusion term, production term and destruction term are given as;

$$\begin{aligned}
\chi &= \frac{\tilde{\nu}}{\nu'}, & S &= |\omega|, \\
f_{t2} &= c_{t3} \exp(-c_{t4} \chi^2), & f_{\nu 2} &= 1 - \frac{\chi}{1 + \chi f_{\nu 1}}, \\
f_w &= g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right), & f_{\nu 1} &= \frac{\chi^3}{1 + \chi^3 c_{v1}^3}, \\
\tilde{S} &= S + \frac{\tilde{\nu}}{\kappa^2 d^2 f_{\nu 2}}, & g &= r + c_{w2} (r^{6-r}), \\
r &= \frac{\tilde{\nu}}{\tilde{S} \kappa^2 d^2},
\end{aligned}$$

where d is the minimum distance to the closest wall boundary, ω is the viscosity and c_{b1} , c_{b2} , σ , κ , c_{w1} , c_{w2} , c_{w3} , c_{v1} , c_{t3} , c_{t4} are the model constants.

The values of the constants are given in Table 3.2

Table 3.2: Spalarat Allmaras one equation model constants

constant	value
c_{b1}	0.1355
σ	$2/3$
c_{b2}	0.622
κ	0.41
c_{w2}	0.3
c_{w3}	2
c_{v1}	7.1
c_{t3}	1.2
c_{t4}	0.5
c_{w1}	$\frac{c_{b1}}{\kappa^2} + \frac{1+c_{b2}}{\sigma} = 3.2391$

The SA eddy viscosity parameter transport equation is solved implicitly. The same cell centered finite volume approach is used to discretize the transport equation. The RHS is linearized with respect to $\tilde{\nu}$. The time integration of LBE and the eddy viscosity parameter are decoupled. The calculated values of macroscopic quantities are

used directly in eddy viscosity parameter transport equation. That is the eddy viscosity parameter is solved after the time advancement is performed for the distribution functions. The convection terms are calculated by using 2^{nd} order accurate MUSCL scheme and the diffusion terms are calculated by using 2^{nd} order central difference scheme. The time advancement is done by the backward Euler scheme. The same approximate factorization method is used to solve the eddy viscosity parameter. The solution procedure is more simple since the eddy viscosity variable is scalar.

At the wall boundaries, the eddy-viscosity variable is set to zero. For the inflow boundaries, a ratio of laminar viscosity is used. And for the outflow boundaries the eddy viscosity variable is extrapolated using the neighbor cells inside the solution domain.

After the solution of eddy viscosity parameter is finished, the turbulent kinematic viscosity is calculated using the relation $\nu_t = f_{\nu 1} \tilde{\nu}$. Then, the relaxation time is updated with the total viscosity as;

$$\tau = \frac{\nu + \nu_t}{c_s^2}$$

3.9 Approximate Factorization

There are different ways to solve the system of equations shown in Figure 3.7. The first one is the direct inversion of matrix A , which takes a serious amount of computational time. On the other hand, there some approximate factorization methods like Alternating Direction Implicit (ADI), Diagonally Dominant Alternating Direction Implicit (DDADI) and Lower-Upper Symmetric Gauss Seidel (LUSGS) which can be applied with a lesser computational time.

ADI method is chosen for present implementation. Equation 3.56 is turned into equation 3.69.

$$\begin{aligned}
& [L^I + D^I + U^I] \cdot [L^J + D^J + U^J] \cdot \dots \\
& \left[L^K + D^K + U^K - \frac{\Delta t}{\tau_c} \frac{\partial}{\partial \vec{f}} \vec{f}^{eq} \right] \Delta \vec{f} = \vec{R}
\end{aligned} \tag{3.69}$$

The solution is obtained as follows:

$$[L^I + D^I + U^I] \Delta \vec{f}^{(1)} = \vec{R}$$

$$[L^J + D^J + U^J] \Delta \vec{f}^{(2)} = \Delta \vec{f}^{(1)}$$

$$\left[L^K + D^K + U^K - \frac{\Delta t}{\tau_c} \frac{\partial}{\partial \vec{f}} \vec{f}^{eq} \right] \Delta \vec{f} = \Delta \vec{f}^{(2)}$$

$$\Delta \vec{f}^{(1)} = [L^I + D^I + U^I]^{-1} \vec{R}$$

$$\Delta \vec{f}^{(2)} = [L^J + D^J + U^J]^{-1} \Delta \vec{f}^{(1)}$$

$$\Delta \vec{f} = \left[L^K + D^K + U^K - \frac{\Delta t}{\tau_c} \frac{\partial}{\partial \vec{f}} \vec{f}^{eq} \right]^{-1} \Delta \vec{f}^{(2)}$$

All the elements in the given matrices are matrix blocks. Hence, it is necessary to use an algorithm for block tri-diagonal systems. Three step sweeping is required to solve 3D problems. However, for the first two sweep, all the sub-matrices are diagonal. Then, the standard Thomas Algorithm to solve tridiagonal matrix system known as Tri-diagonal Matrix Algorithm (TDMA) is sufficient to get a solution. For the third sweep, the JEDF adds non-zero elements to the sub matrices of D matrix. Therefore, the third sweep is solved with a similar algorithm for the block tri-diagonal systems.

CHAPTER 4

2D MODEL VALIDATION

The set of equations and relations are derived in Chapter 3 To validate the model, an algorithm is developed and implemented. For the implementation process, FORTRAN coding language is selected which is originally developed for scientific computing. The FORTRAN language is fast and suitable for vector operations. 2D and 3D codes are implemented separately, although 3D code has the capability to solve 2D problems by utilizing symmetry condition on the 3rd axis. However since the size of the unknown vector \vec{f} is equal to 19 as compared 9 for 2D, the solution process takes more time.

The codes can be used to solve problems with structured grid on single block and multiple blocks. Various problems which have data in the literature are studied to validate the method presented in this thesis. The validation procedure is simply divided into four steps where 2D flows are used in the first three steps. The first step is to show the methods capability to solve the laminar cases for steady state. Second step is to show the capability of the method for the turbulent flows. Third step is to show the capability of the method for the unsteady flows. Then the last step is to demonstrate to capability of the method for the 3D problems and the results will be described in Chapter 5.

The residuals, for density and velocity, which are mentioned in the proceeding sections are calculated by the relations given in equations 4.1 and 4.2.

$$\rho_{res} = \sqrt{\frac{\sum_{cells} (\rho_{present} - \rho_{previous})^2}{\#cells}} \quad (4.1)$$

$$u_{res} = \sqrt{\frac{\sum_{cells} (\vec{u}_{present} - \vec{u}_{previous})^2}{\#cells}} \quad (4.2)$$

Simple Algorithm for model implementation is given in Algorithm 4.1.

1. Read Input Data
2. Read Grid Data
3. Calculate Grid Metrics (check metric error)
4. Initialize
 - (a) Case Steady, Calculate Δt , $\Delta t = CFL \cdot \frac{\Delta x}{\sqrt{3}}$
 - (b) Case Unsteady, Δt is given, Calculate Pseudo Time Δt^* , $\Delta t^* = CFL \cdot \frac{\Delta x}{\sqrt{3}}$
5. Solve for $\Delta \vec{f}$
6. Calculate $\vec{f}^{n+1} = \vec{f}^n + \Delta \vec{f}^n$
7. Calculate Macroscopic Quantities ρ , \vec{u} at new time level
8. Calculate Equilibrium Distribution Functions \vec{f}^{eq}
 - (a) *If Turbulent*
 - i. calculate ν_T and $\nu = \nu_L + \nu_T$
 - ii. calculate $\tau = \nu \cdot M_{ref}^2$
9. Check Convergence, GOTO (5.a) or Check Iteration Time, GOTO (5.a)
10. Output Solution Data

Figure 4.1: Simple algorithm for the codes

4.1 Laminar Steady Flow

4.1.1 Lid Driven Cavity Flow

Lid driven cavity is the one of the famous benchmarks for the CFD community. The most simple application is the flow defined in a square cavity whose top wall is moving. Although, it seems an easy problem, the intersection points of the stationary wall and moving wall may cause difficulties for some methods, yet must be handled carefully. For the present implementation, the calculations are performed for four different Reynolds numbers. The Re numbers set is defined as $Re = 400, 1000, 3200, 5000$. The results in the work of Ghia et al. [70] (high fidelity solution of Navier Stokes equations) are used for comparison purposes as done by many other researchers interested in CFD. The cavity has equal dimensions in width and height. The top edge is considered as a moving wall with a reference velocity " U_{ref} " from towards to positive x direction and no velocity component in y direction. The velocity is taken as $U_{ref} = 0.2$. The problem setup is given in Figure 4.2.

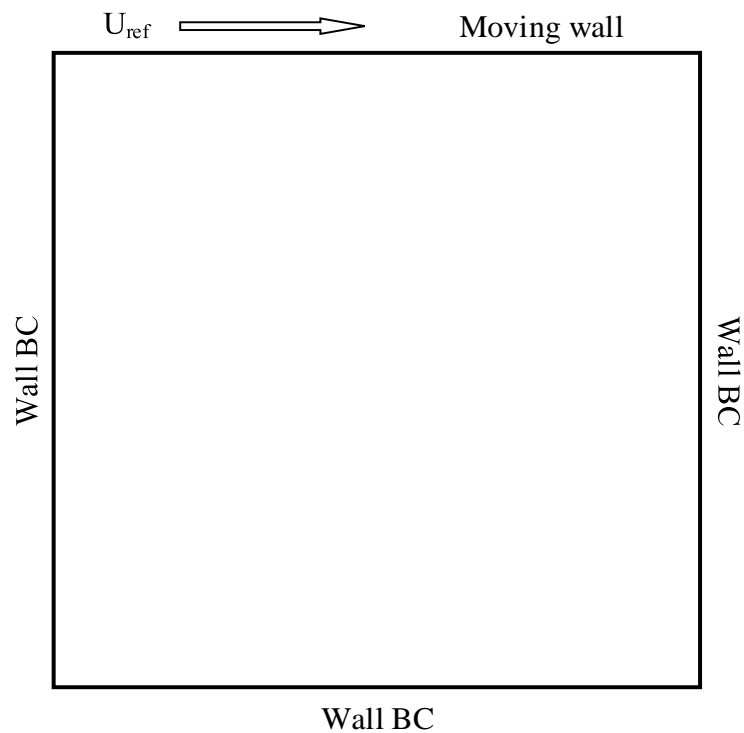


Figure 4.2: Case setup for cavity flow

The cavity flow for the considered ranges can be categorized as a incompressible, laminar flow. The nature of the flow for Reynolds numbers 400 and 1000 is visualized as a large vortex occupying almost all the flow field accompanied by 2 small vortexes at the lower corners. For Reynolds numbers 3200 and 5000 , an additional vortex at the upper left corner is formed.

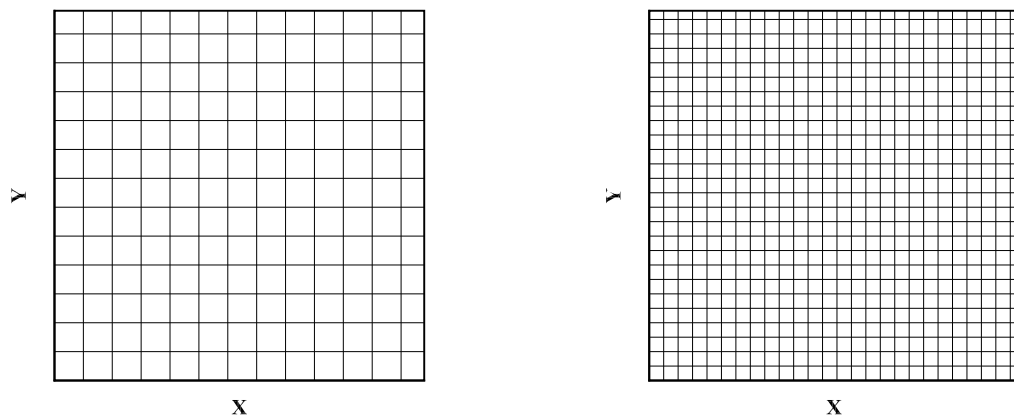
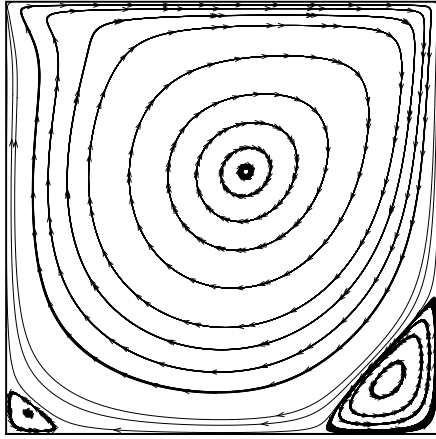
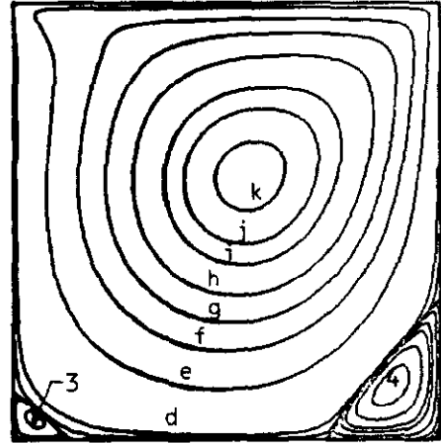


Figure 4.3: Grids for 129 x 129 and 257 x 257 (Every 10th gridline is shown)

The flow domain for $Re = 400$, 1000 and 3200 are composed of 128 x 128 uniform cells and for $Re = 5000$ 256 x 256 uniform cells which are exactly the same solution grid used by Ghia et al. [70]. The streamlines calculated using IFVLBM are presented in Figure 4.4, 4.5, 4.6 and 4.7. The general arrangement of the streamlines are in a good agreement with the work of Ghia et al..

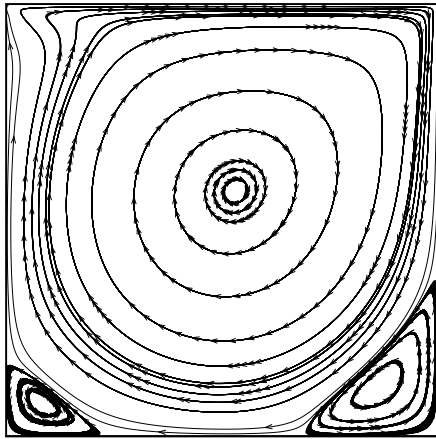


(a) $Re = 400$

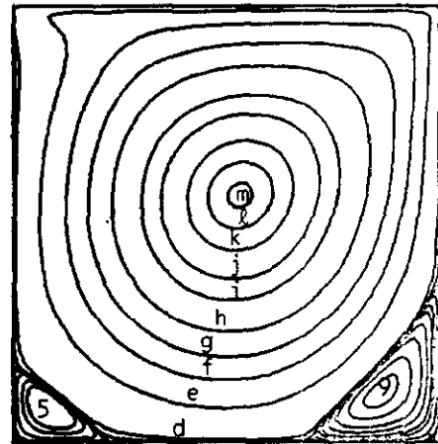


(b) $Re = 400$ @GHIA et.al

Figure 4.4: Streamlines inside the cavity by IFVLBM $Re = 400$



(a) $Re = 1000$



(b) $Re = 1000$ @GHIA et.al

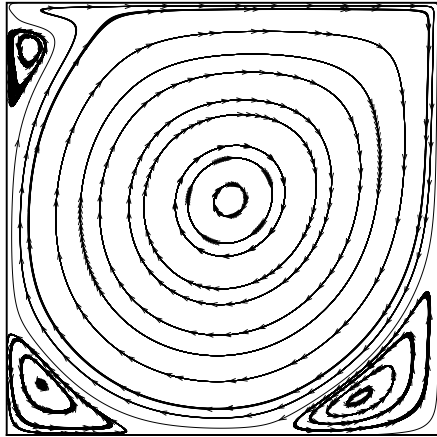
Figure 4.5: Streamlines IFVLBM and Ghia et.al $Re = 1000$

Furthermore, the center coordinates of the eddies are compared with the results of Ghia's in Table 4.1. The center coordinates of the eddies are also in good agreement with the Ghia's work. The error is defined as follows:

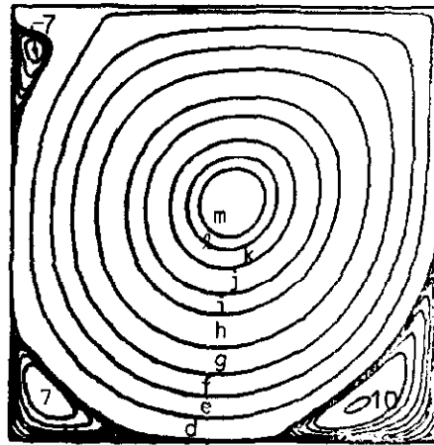
$$Error = \frac{|x_{GHIA} - x_{IFVLBM}|}{\Delta x}$$

which means that the error is normalized with respect to the cell length. If the error is smaller than 1, the center of the calculated eddy and the reference eddy is the vicinity less than 1 cell length. The secondary eddies for bottom right and bottom left

positions are captured for $Re = 5000$ with the present method.

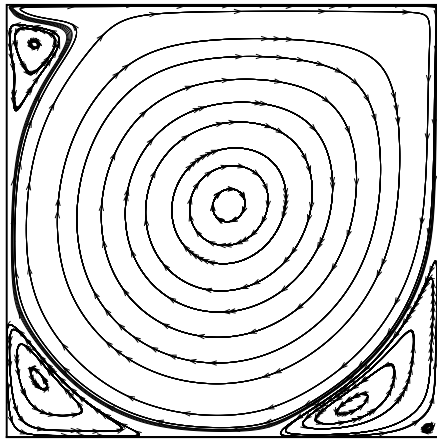


(a) $Re = 3200$

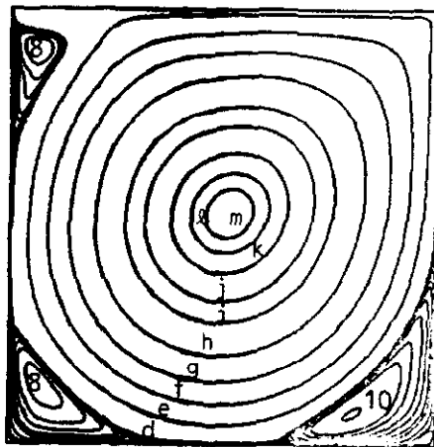


(b) $Re = 3200$ @GHIA et.al

Figure 4.6: Streamlines IFVLBM and Ghia et.al $Re = 3200$



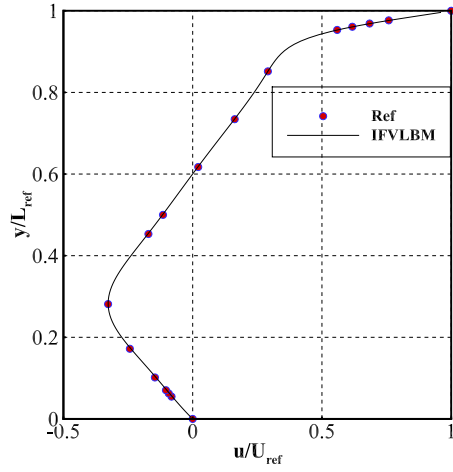
(a) $Re = 5000$



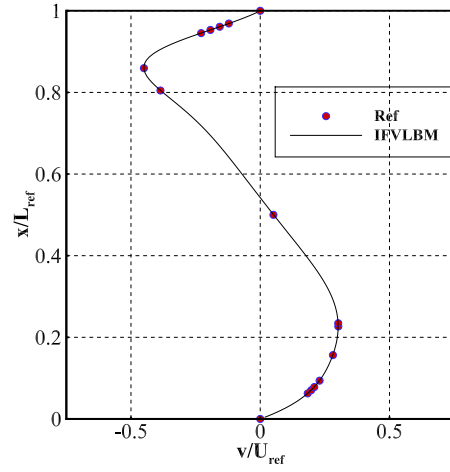
(b) $Re = 5000$ @GHIA et.al

Figure 4.7: Streamlines IFVLBM and Ghia et.al $Re = 5000$

Moreover, to complete the analysis, horizontal and vertical velocity profiles are plotted at the horizontal and vertical geometrical center of the cavity. As shown in Figures 4.8,4.9, 4.10 and 4.11 the velocity profiles are matched perfectly with the reference data of Ghia et al..

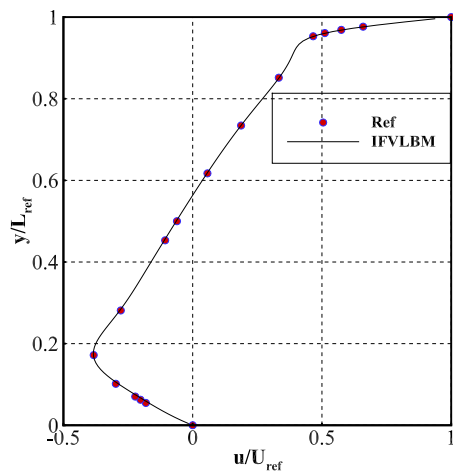


(a) $Re = 400$

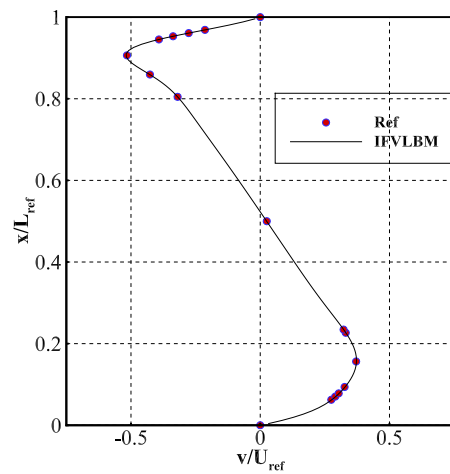


(b) $Re = 400$

Figure 4.8: Horizontal and vertical velocity distribution at vertical and horizontal geometric center-line

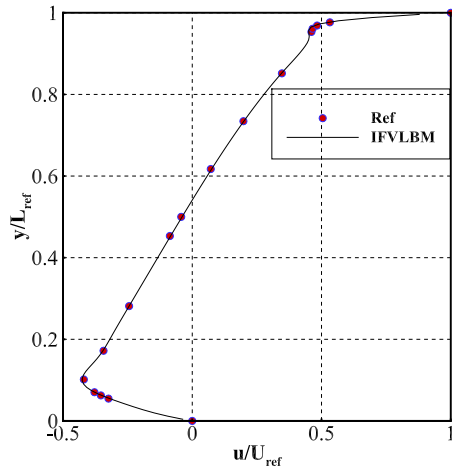


(a) $Re = 1000$

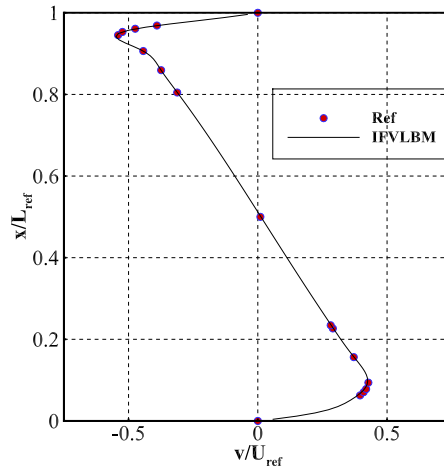


(b) $Re = 1000$

Figure 4.9: Horizontal and vertical velocity distribution at vertical and horizontal geometric center-line

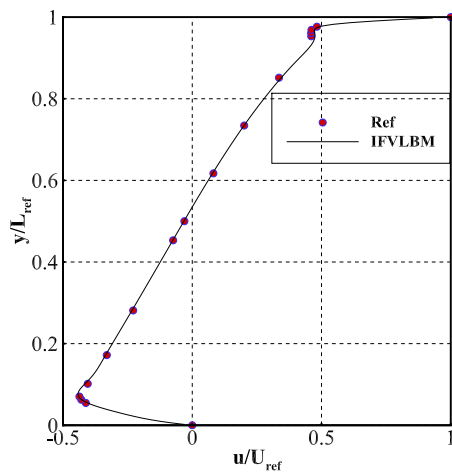


(a) $Re = 3200$

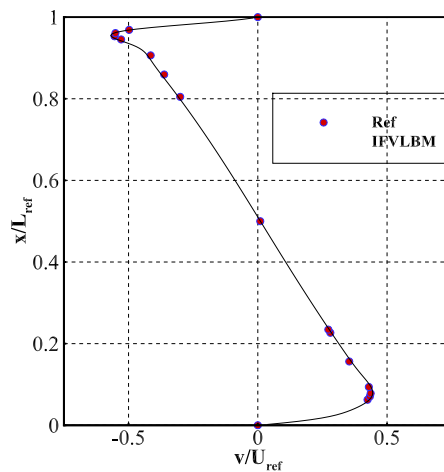


(b) $Re = 3200$

Figure 4.10: Horizontal and vertical velocity distribution at vertical and horizontal geometric center-line



(a) $Re = 5000$



(b) $Re = 5000$

Figure 4.11: Horizontal and vertical velocity distribution at vertical and horizontal geometric center-line

Table 4.1: Comparison of center coordinates of eddies

Re	Eddy	Ghia et al		IFV-LBM		% Error	
		x	y	x	y	x	y
400	Primary Eddy	0.5547	0.6055	0.5543	0.6068	0.05	0.17
	Bottom Right Eddy	0.8906	0.1250	0.8830	0.1244	0.97	0.08
	Bottom Left Eddy	0.0508	0.0469	0.0517	0.0474	0.12	0.06
1000	Primary Eddy	0.5313	0.5625	0.5299	0.5678	0.18	0.68
	Bottom Right Eddy	0.8594	0.1094	0.8601	0.1146	0.09	0.67
	Bottom Left Eddy	0.0859	0.0781	0.0827	0.0778	0.41	0.04
3200	Primary Eddy	0.5165	0.5469	0.5161	0.5435	0.05	0.44
	Bottom Right Eddy	0.8125	0.0859	0.8149	0.0865	0.31	0.08
	Bottom Left Eddy	0.0859	0.1094	0.0803	0.1175	0.72	1.04
	Top Left Eddy	0.0547	0.8984	0.0549	0.8983	0.03	0.01
5000	Primary Eddy	0.5117	0.5352	0.5134	0.5369	0.44	0.44
	Bottom Right Eddy 1	0.8086	0.0742	0.7980	0.0748	2.71	0.15
	Bottom Left Eddy 1	0.0703	0.1367	0.0720	0.1357	0.44	0.26
	Bottom Right Eddy 2	0.9805	0.0195	0.9742	0.0214	1.61	0.49
	Bottom Left Eddy 2	0.0117	0.0078	0.0056	0.0084	1.56	0.15
	Top Left Eddy	0.0625	0.9102	0.0644	0.9092	0.49	0.26

4.1.2 Steady Laminar Flow Over a Flat Plate

Flat plate with zero pressure gradient is one of the handy test cases for the CFD model and software developers. The problem was studied by Blassius and the solution of the nonlinear ordinary differential equation for the laminar boundary layer derived by him exists [71]. The results of Blassius' study are used to validate the present IFVLBM. The grid dependency study is not performed as the grid file is directly obtained from the NASA CFL3D page for turbulent flow studies. For CFL3D, the the solutions are obtained for grids 35×25 , 69×49 , 137×97 , 273×193 and 545×385 . Since, analytical solution for laminar case exists in the literature, 273×193 grid resolution is selected for current study. This problem has an also other importance that the aspect ratio of the cells are large compared to a uniform grid especially near wall. The same problem with the same first layer resolution and solution domain dimensions would require 2.5×10^{12} nodes for classical Lattice Boltzmann Methods.

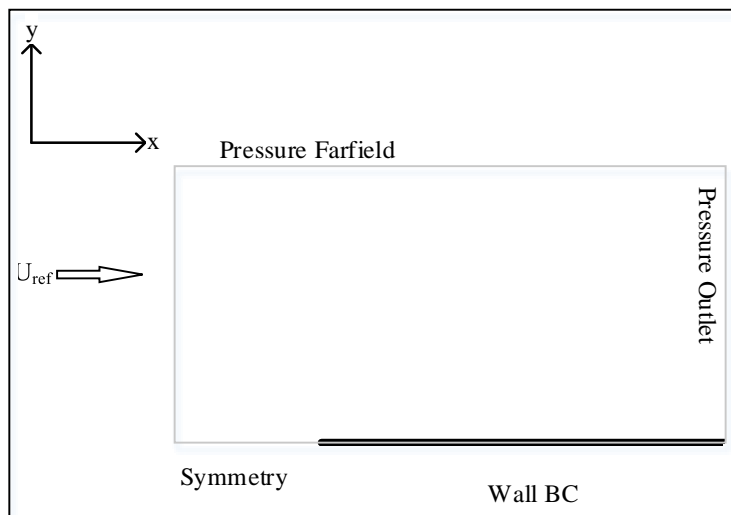


Figure 4.12: Flat plate problem setup

The problem given in Figure 4.12 is set by using a H type grid including 273×193 nodes obtained from reference [72]. The first layer of nodes is placed at $y = 1 \times 10^{-6}$ respectively. The length of the plate is 2 units and upper part of the solution domain is 1 unit. The studies are performed for $Re = 10000$ and $Re = 100000$ per unit length for the laminar case. Mach number is taken as $Ma = 0.2$. In Figure 4.13, the used 273×193 grid is given, every 5^{th} grid line is plotted for clarity. The horizontal mesh

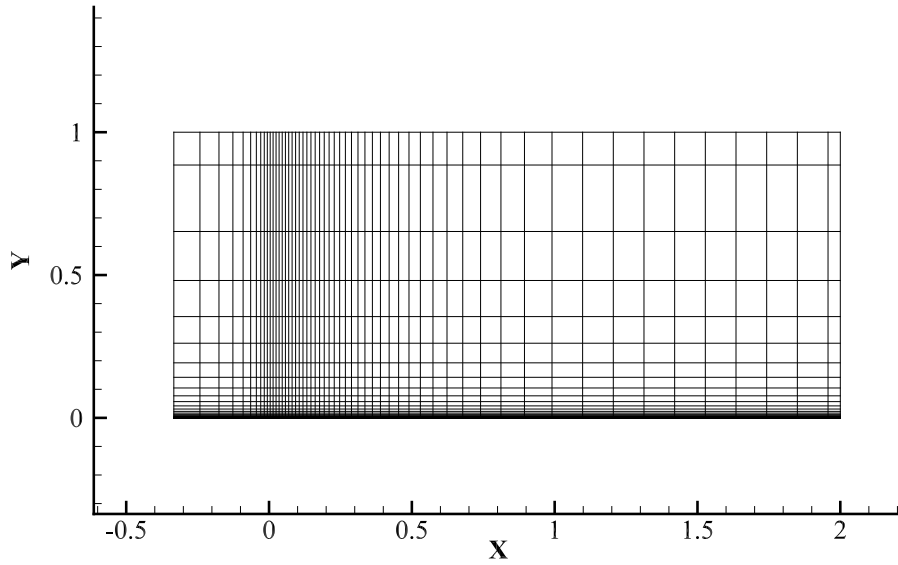


Figure 4.13: Flat plate 273×193 grid

density is increased near the leading edge of the flat plate.

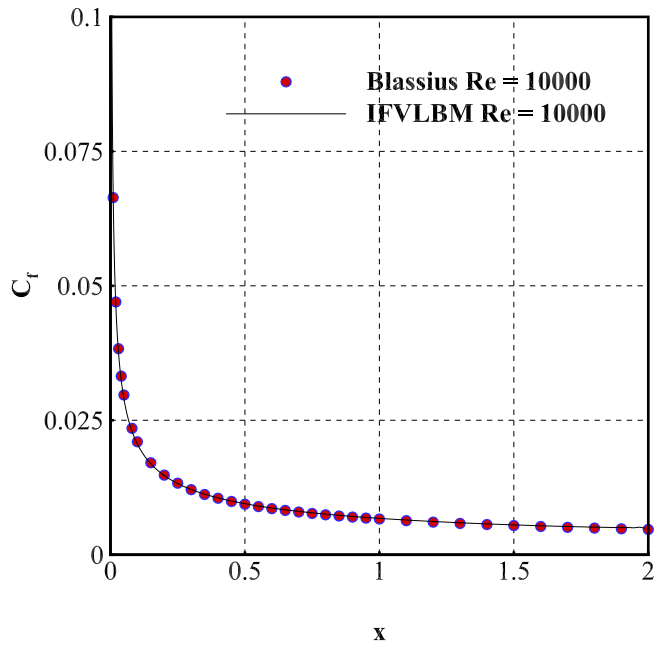
The non-dimensional velocity distribution in the boundary layer region and the calculated skin friction “ C_f ” values are compared with the solution of ODE. The velocity profiles are obtained at a section placed in the middle of the flat plate.

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho_\infty U_{ref}^2} \quad (4.3)$$

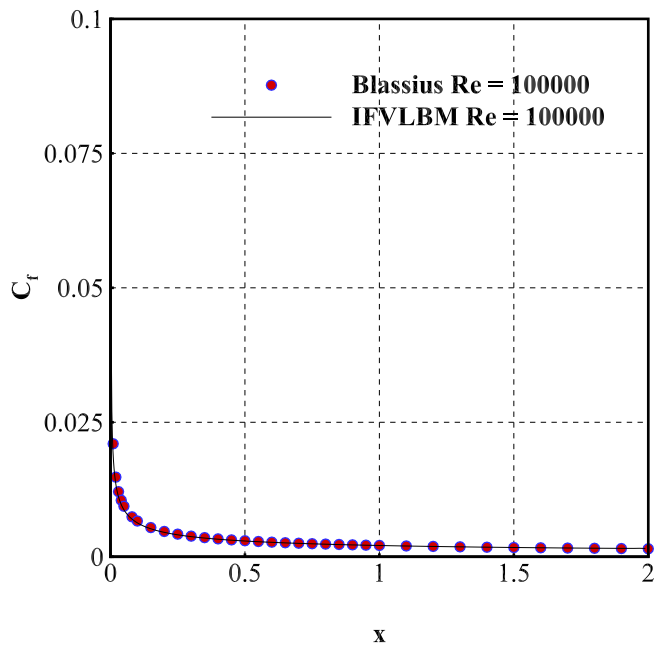
$$\tau_w = \mu \frac{\partial u}{\partial y} \quad (4.4)$$

where τ_w is the wall shear stress and μ is the dynamic viscosity, u is the velocity component parallel to the wall and y is the axis perpendicular to the wall. For the flat plate, the local skin friction coefficient depending on the local Reynolds number Re_x is given as:

$$C_f(x) = \frac{0.664}{\sqrt{Re_x}}$$



(a) $Re = 10000$

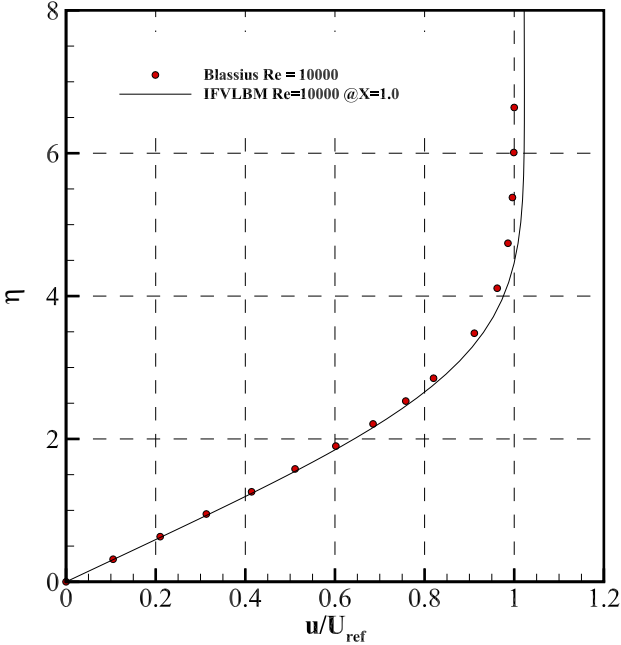


(b) $Re = 100000$

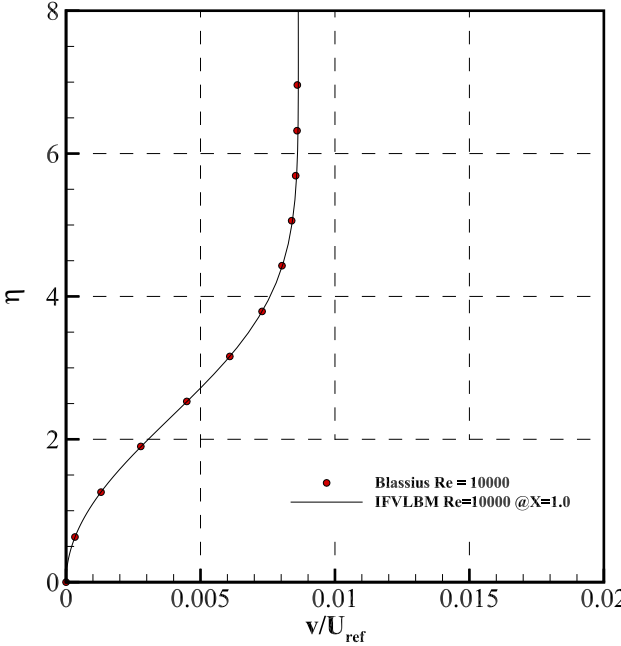
Figure 4.14: Local friction coefficients C_f along the flat plate

In Figure 4.14, the local skin friction coefficients are given as a function of downstream distance from the leading edge of the flat plate. The Blasius solution of flat plate is given as points filled with red circles, while the IFVLBM solution is given as

solid line.



(a) u velocity distribution in boundary layer

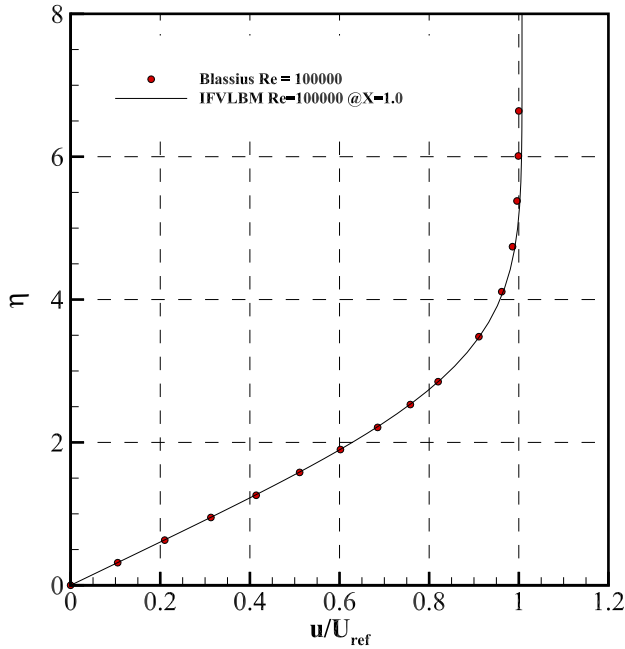


(b) v velocity distribution in boundary layer

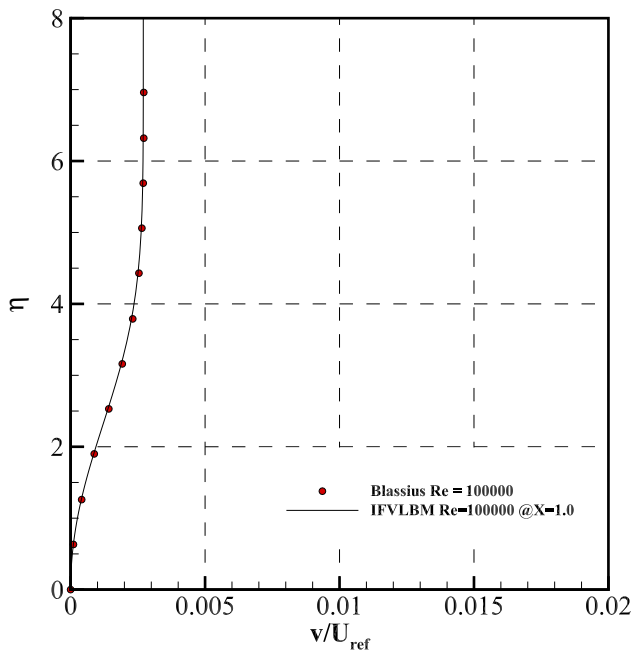
Figure 4.15: Velocity distributions in the boundary layer for $Re = 10000$

The vertical axis variable η is the non dimensional distance from the wall and calculated by the relation given in equation 4.5:

$$\eta = \frac{y}{x} \sqrt{Re_x} \quad (4.5)$$



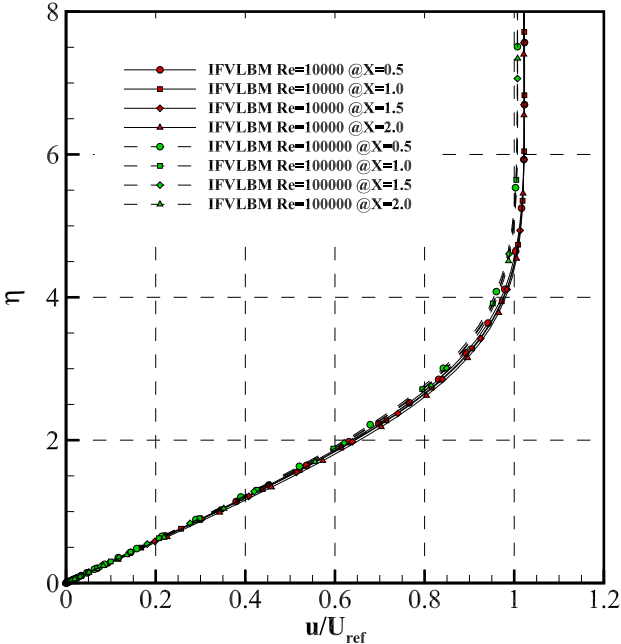
(a) u velocity distribution in boundary layer



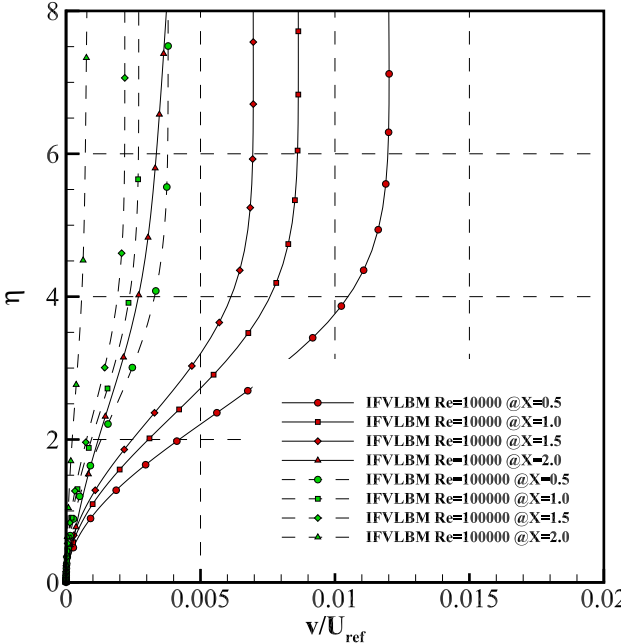
(b) v velocity distribution in boundary layer

Figure 4.16: Velocity distributions in the boundary layer for $Re = 100000$

The velocity distributions given in 4.15 and 4.16 are in good agreement with the reference solution.



(a) u velocity distribution in boundary layer at various sections



(b) v velocity distribution in boundary layer at various sections

Figure 4.17: Velocity distributions in the boundary layer for $Re = 10000$ and 100000

For a flow over a flat plate, there are some parameters which characterize the bound-

ary layer. Displacement thickness, momentum thickness and the shape factor which is the ratio of the previous two are the parameters. The higher the value of the shape factor, the stronger the adverse pressure gradient. Adverse pressure gradient is directly related to the transition phenomena from laminar to turbulent flow. For the laminar flows $H = 2.60$ is the typical value, while $H = 1.3 - 1.4$ is for turbulent flows. In Table 4.2, the calculated parameters are compared with the Blassius' solution. It can be seen that, as the Reynolds number increases, IFVLBM method predictions are getting better. Equations 4.6 and 4.7 are the formulas of displacement thickness and momentum thickness respectively.

Table 4.2: Boundary layer parameters for Re=10000, 100000

Reynolds Number	Blassius Solution			IFVLBM Solution		
	δ^*	θ	$H = \frac{\delta^*}{\theta}$	δ^*	θ	$H = \frac{\delta^*}{\theta}$
10000	1.7185	0.6563	2.6184	1.6508	0.6196	2.6643
100000				1.7123	0.6529	2.6226

$$\delta^* = \int_0^{\infty} \left(1 - \frac{u(y)}{U_{ref}} \right) dy \quad (4.6)$$

$$\theta = \int_0^{\infty} \frac{u(y)}{U_{ref}} \left(1 - \frac{u(y)}{U_{ref}} \right) dy \quad (4.7)$$

4.1.3 Steady Laminar Flow Over a Cylinder

To demonstrate the capability and performance of the IFVLBM for more general flows, it is applied to the flow past a circular cylinder. The flow past a circular cylinder is studied both experimentally and numerically by various scientists and researchers. The problem setup is given in Figure 4.18. The cylinder diameter is taken as unity. The distance between the outer boundary off the solution domain and the face of the cylinder is about 50 diameters.

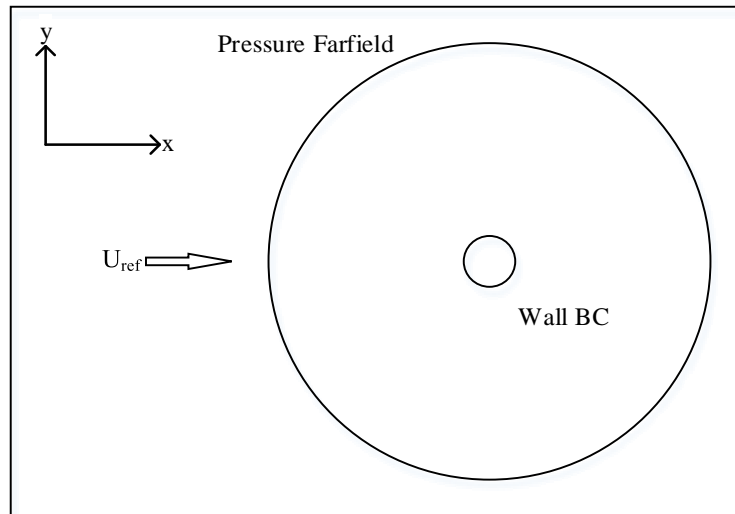
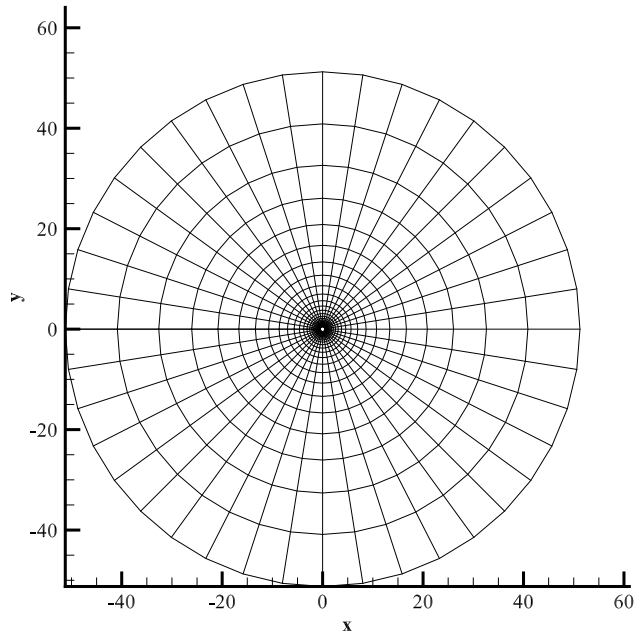


Figure 4.18: Cylinder problem setup

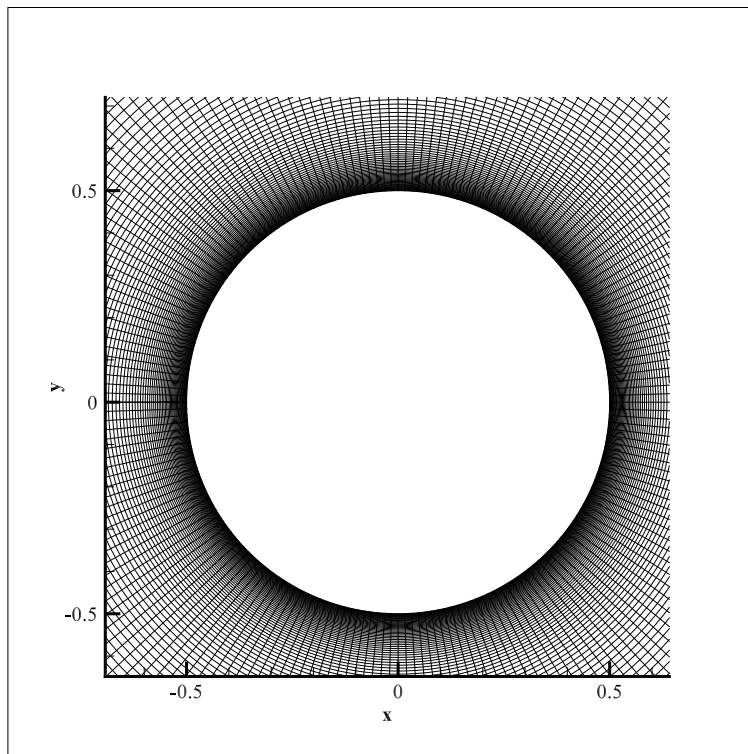
The grid generated for the problem is a 201 x 173 nodes “O” type grid given in Figure 4.19. The solutions are obtained for $Re = 10, 20, 40$. The first cell height from the surface of the cylinder is 0.001 diameter.

The separation angle is measured from the horizontal axis where the free stream velocity is parallel. The length of the eddies referenced to the radius of the cylinder.

To compare the results obtained from the IFVLBM following studies are selected from the literature. The experimental work of Coutanceau & Bouard [73], NS solutions of F. Nieuwstadt & Keller [74], LBM solutions of He & Doolen [75] and Mei & Shyy [29]. The characteristic of the flow around a cylinder for the specified Reynolds number is the length of the eddies occurred after the cylinder, the separation point of separating streamline and drag coefficient (C_d).

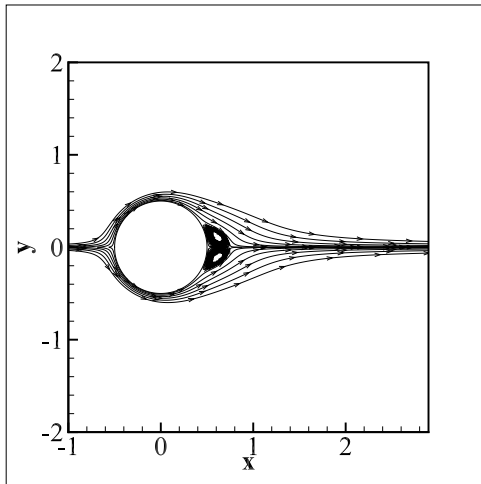


(a) solution domain grid (every 5th grid is shown for clarity)

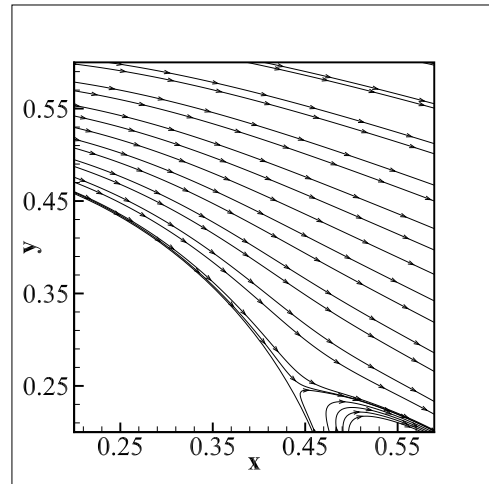


(b) Grid around the cylinder

Figure 4.19: Solution domain for steady cylinder problems.



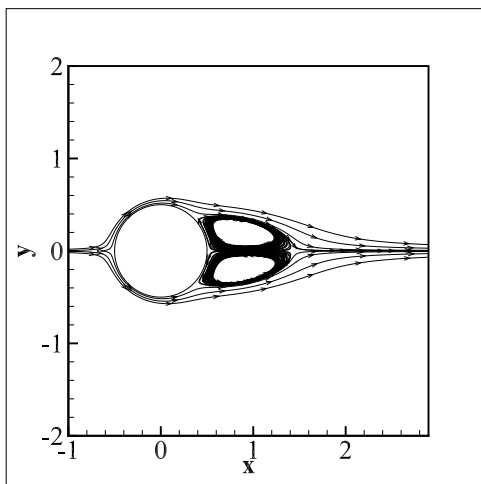
(a) Eddy formation behind the a cylinder



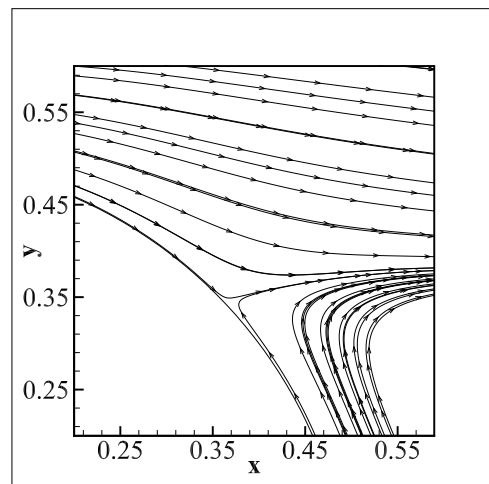
(b) Separating streamline

Figure 4.20: Eddy formations and separation streamline for the flow over cylinder for Reynolds numbers $Re = 10$

The length of the eddies are dependent to the Reynolds number and they are stretching up to some Reynolds number around $Re \gtrsim 50$. After that Reynolds number, the eddies are detached from the cylinder one after another periodically. The detached eddies travel to the downstream of the flow and they form a phenomena known as the Karman Vortex Street.

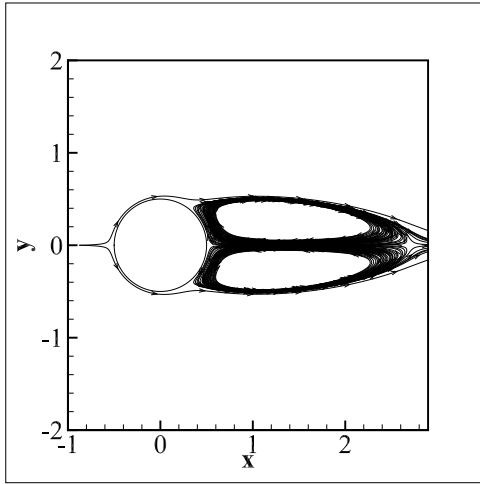


(a) Eddy formation behind the a cylinder

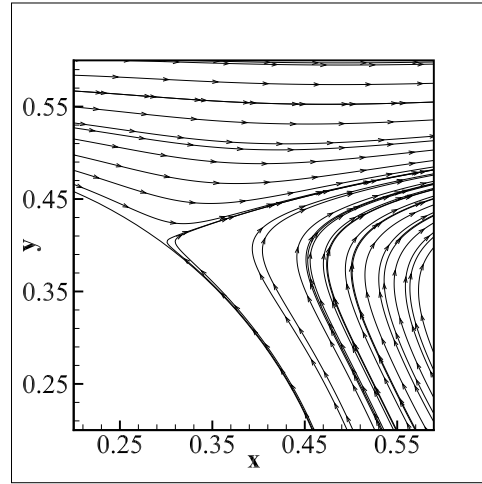


(b) Separating streamline

Figure 4.21: Eddy formations and separation streamline for the flow over cylinder for Reynolds numbers $Re = 20$



(a) Eddy formation behind the a cylinder



(b) Separating streamline

Figure 4.22: Eddy formations and separation streamline for the flow over cylinder for Reynolds numbers $Re = 40$

Table 4.3: Characteristic parameters for flow over a cylinder

	$Re = 10$			$Re = 20$			$Re = 40$		
	L/r	θ	C_d	L/r	θ	C_d	L/r	θ	C_d
[74]	0.434	27.96	2.828	1.786	43.37	2.053	4.357	53.34	1.550
[73]	0.680	32.50	-	1.860	44.80	-	4.260	53.50	-
[75]	0.474	26.89	3.170	1.842	42.96	2.152	4.490	52.84	1.499
[29]	0.498	30.00	-	1.804	42.10	-	4.380	50.12	-
IFVLBM	0.486	30.05	2.962	1.853	44.05	2.084	4.425	53.77	1.544

For laminar flow past a cylinder case, the shape of the eddies are symmetric for $Re = 10, 20, 40$. The calculated streamlines are shown in Figure 4.20, 4.21 and 4.22. The calculated values for $L/r, \theta$ and C_d are given in Table 4.3. The results are in very good agreement with the selected studies from the literature. Except for the separation angle for $Re = 40$, all results of IFVLBM are neither minimum nor maximum among the values obtained from the selected studies.

4.2 Turbulent Flow

As mentioned in Turbulence Modeling section 3.8, Spalart Allmaras one equation model is incorporated with the present method. To validate the turbulence model, two test cases are selected. The turbulent flow over a flat plate and flow over NACA0015 are simulated. The results are presented in the following sections.

4.2.1 Turbulent Flow Over Flat Plate

The problem is set by using a H type grid including 273×193 nodes which is obtained from the web page given in reference [72].

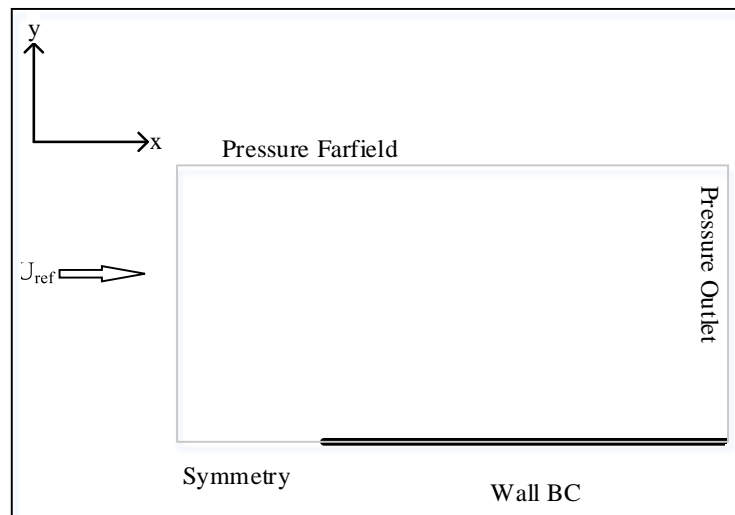
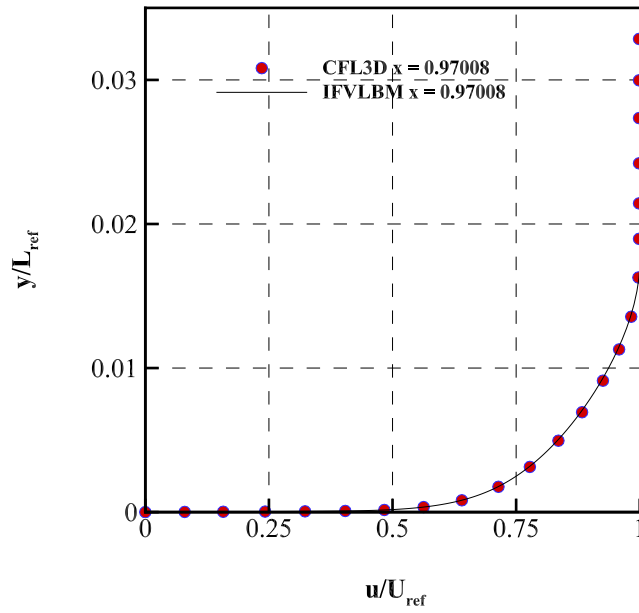
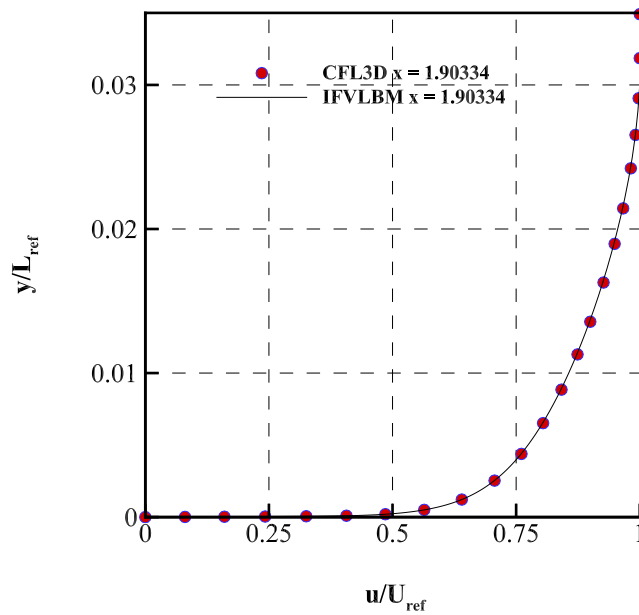


Figure 4.23: Flat plate problem setup

CFL3D is the structured grid, cell centered Finite Volume CFD code developed by NASA. CFL3D is a RANS solver. The grid is the same grid where the CFL3D test cases are insensitive to grid size [72] obtained from the grid dependence studies from the same reference. The first layer is placed at 1×10^{-6} and the length of the plate is 2 units. The study is performed for $Re = 5$ million per unit length. The non-dimensional velocity distribution in the boundary layer region and the calculated C_f values are compared with the results of CFL3D flat plate benchmark results.



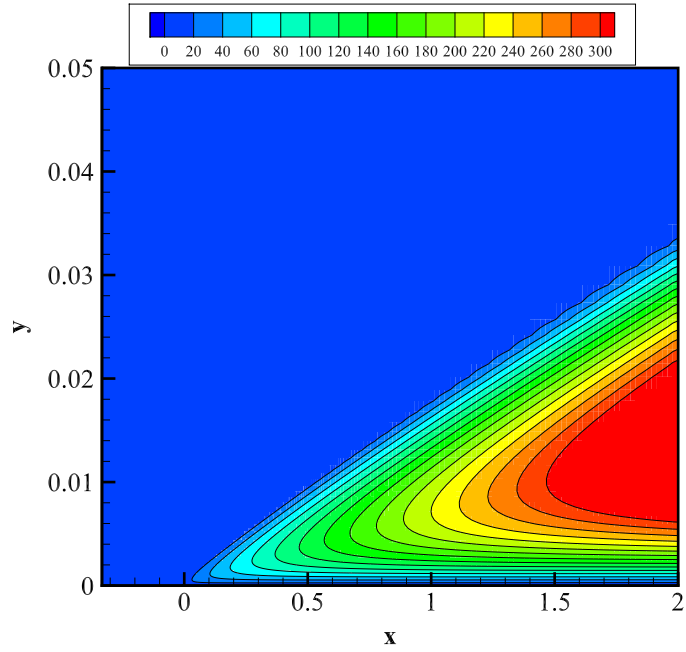
(a) $x = 0.97008$



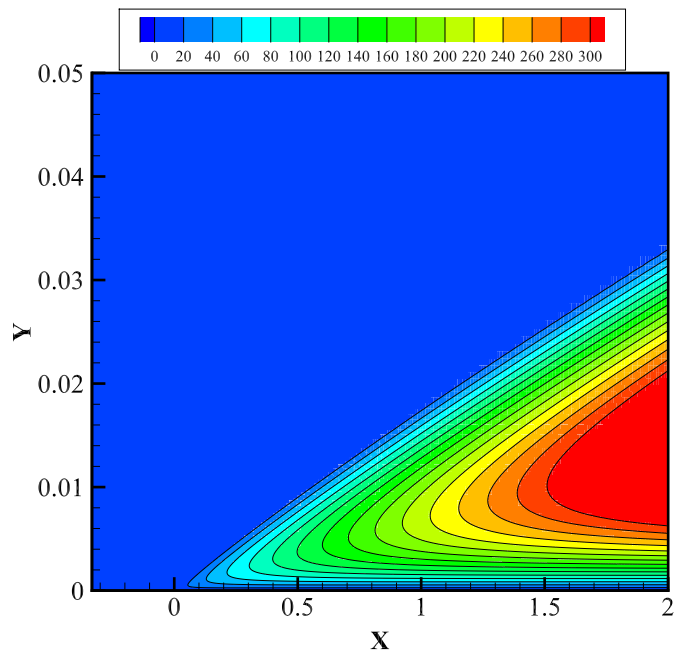
(b) $x = 1.90334$

Figure 4.24: Velocity distributions in the boundary layer at various locations

The boundary layer velocity distributions for different x locations are given in Figure 4.24. The results are in very good agreement with CFL3D code results. Actually, CFL3D data is dense but for clarity, some points are omitted.



(a) IFVLBM Results



(b) CFL3D Results

Figure 4.25: Contours of the ratio of turbulent viscosity to dynamic viscosity ratio

In Figure 4.25, the contours for the ratio of the turbulent and laminar viscosity is plotted with respect to same maximum and minimum values and the increment. Both contours are same. Also in Figure 4.26, the ratio of the turbulent viscosity is plotted

at $x = 0.97008$ location against y axis.

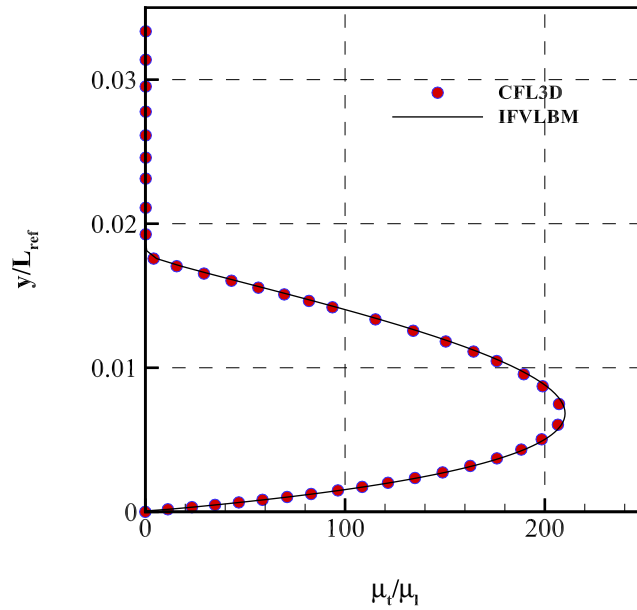


Figure 4.26: Ratio of turbulent viscosity to laminar viscosity at $x = 0.97008$

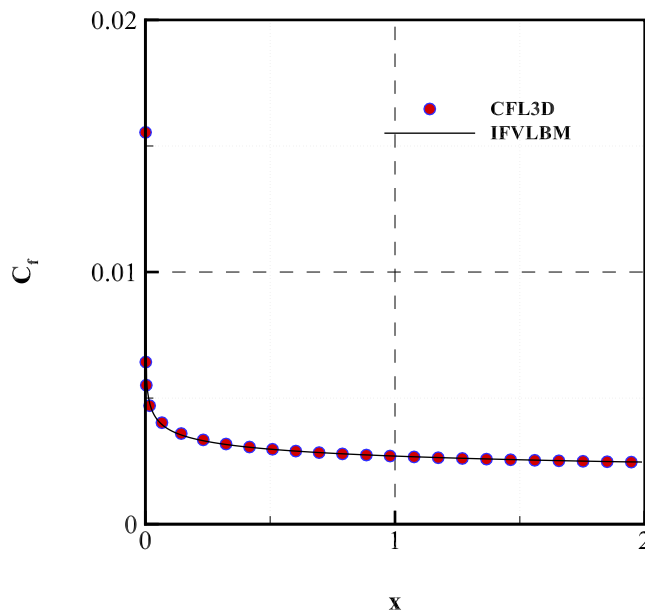


Figure 4.27: C_f values along the flat plate

The C_f values along the flat plate shown in Figure 4.27 are also in perfect agreement with CFL3D. The results for the flat plate for turbulent flow is satisfactory compared

with the NASA CFL3D code.

4.2.2 Flow Over NACA0015 Airfoil

NACA0015 airfoil is studied excessively by Piziali[76] and the technical report has been published in 1994. The experiments are conducted to identify some dynamic characteristics of the airfoil section and the 3D wing like stall. Especially the pitching motion is simulated to characterize a helicopter rotor blade. Moreover some steady state 2D data is also included in the document. A boundary layer trip is used at the leading edge during the tests.

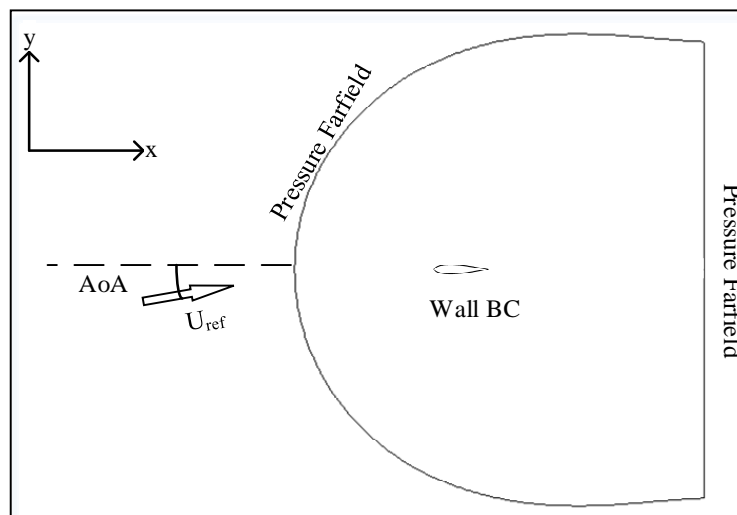


Figure 4.28: NACA0015 problem setup

For this study NACA0015 airfoil is presented by a 377 x 171 C-type grid. The solution domain is given in Figure 4.29. On the left part of Figure 4.29, whole domain is presented while on the part, the NACA0015 airfoil is focused. For the whole domain every 5th grid is shown for clarity. The flow is solved as fully turbulent at $Re = 1955000$. Moreover, $M_\infty = 0.29$ and free stream turbulence level is 1.341946 is taken for the simulations.

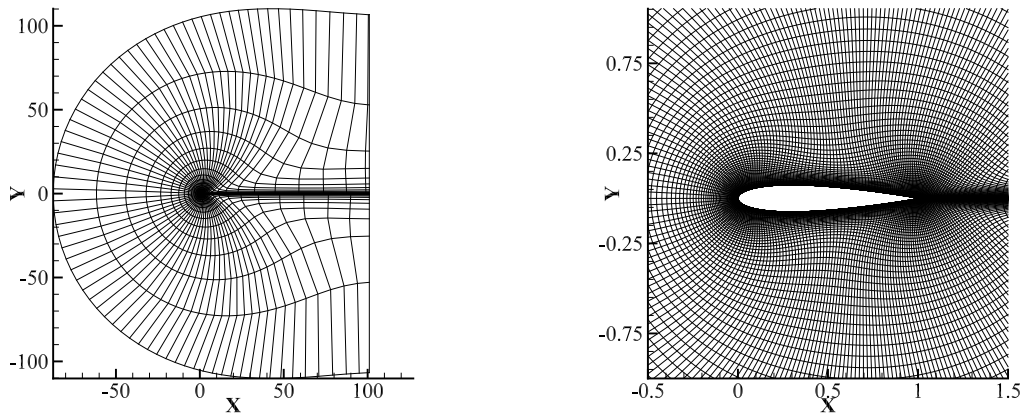


Figure 4.29: NACA0015 Solution domain

The problem is also solved with the commercial FLUENT software using the same grid to compare the results. The angle of attacks used for the simulations are 0.0, 2.5, 5.0, 7.5, 10.0, 12.5, 15.0, 16.0 and 18.0 degrees. The coefficients C_M , C_L and C_D are shown in Figure 4.30, 4.31 and 4.32 respectively.

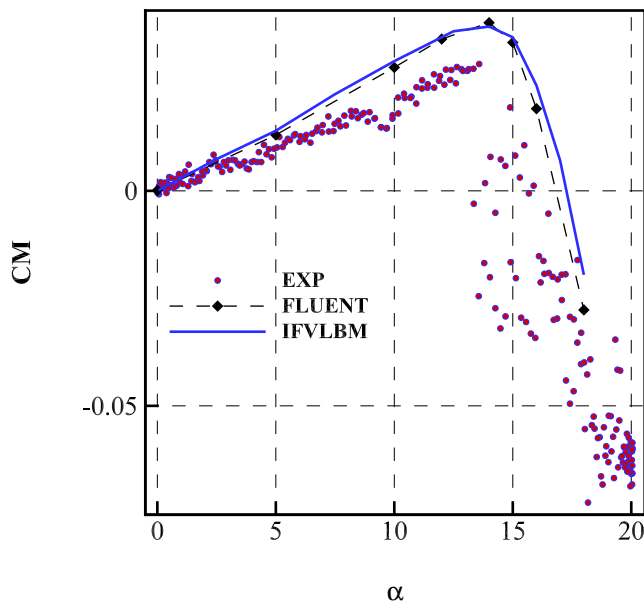


Figure 4.30: C_M vs α of NACA0015

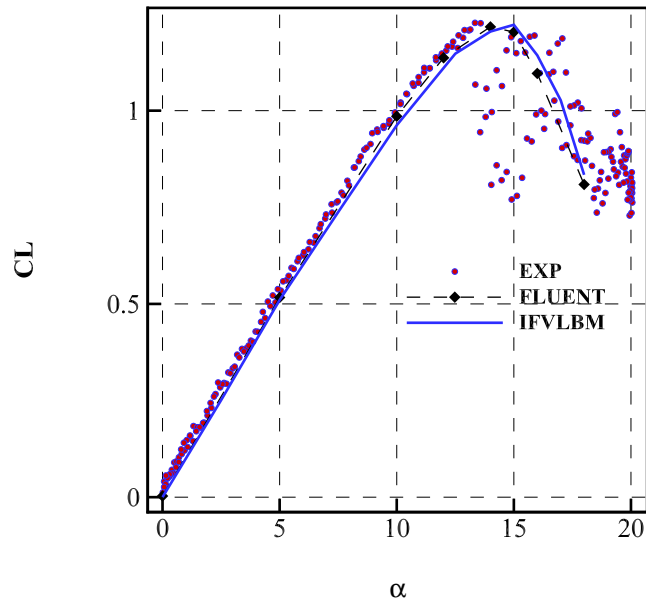


Figure 4.31: CL vs α of NACA0015

As seen from Figure 4.30 the results are in good agreement with the experimental results for the drag coefficient CD only the pressure drag from the simulation results are used since the drag coefficient of the experiment was calculated by using the pressure taps.

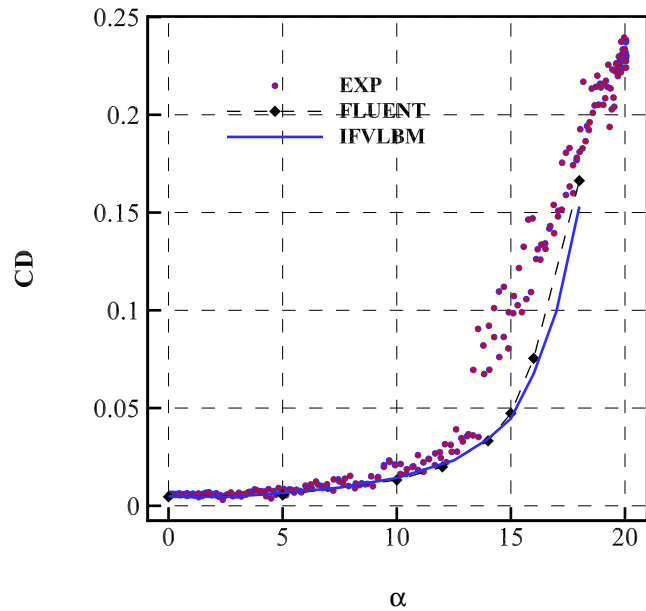


Figure 4.32: CD vs α of NACA0015

One interesting characteristic of the NACA0015 airfoil section is the Trailing Edge (TE) separation starts around 10 degrees.

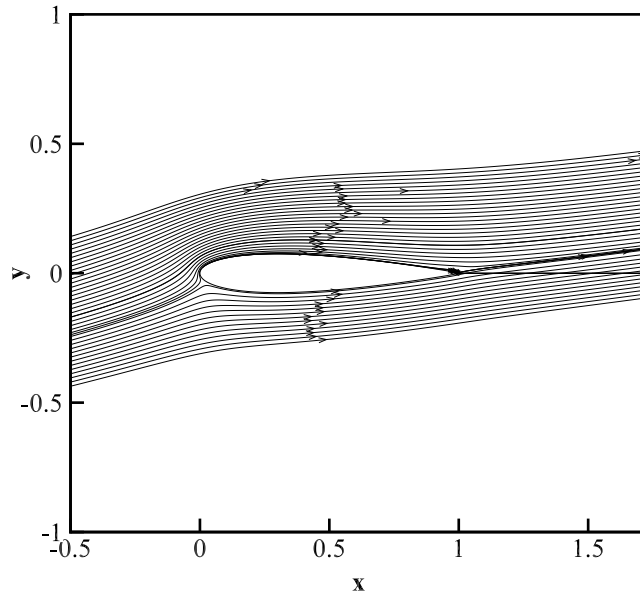


Figure 4.33: TE separation of NACA0015 airfoil $AoA = 10.0$ degrees

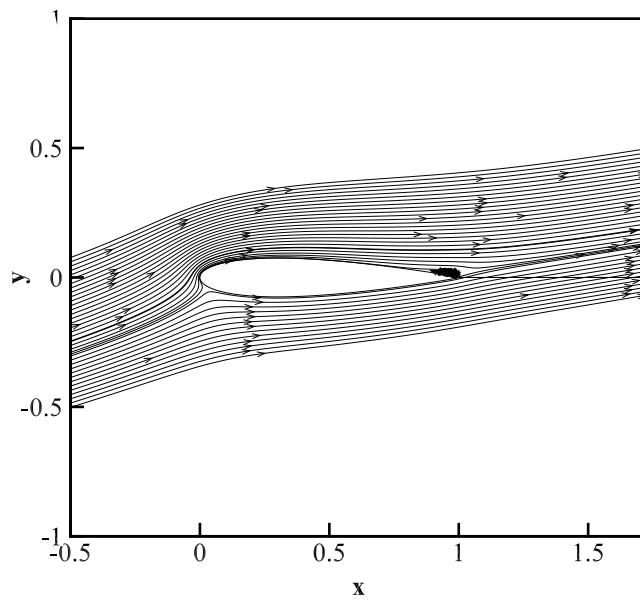


Figure 4.34: TE separation of NACA0015 airfoil $AoA = 12.5$ degrees

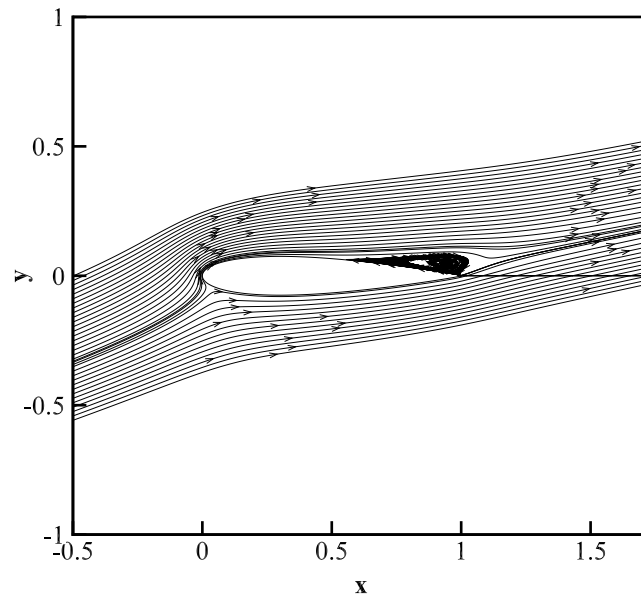


Figure 4.35: TE separation of NACA0015 airfoil $AoA = 15.0$ degrees

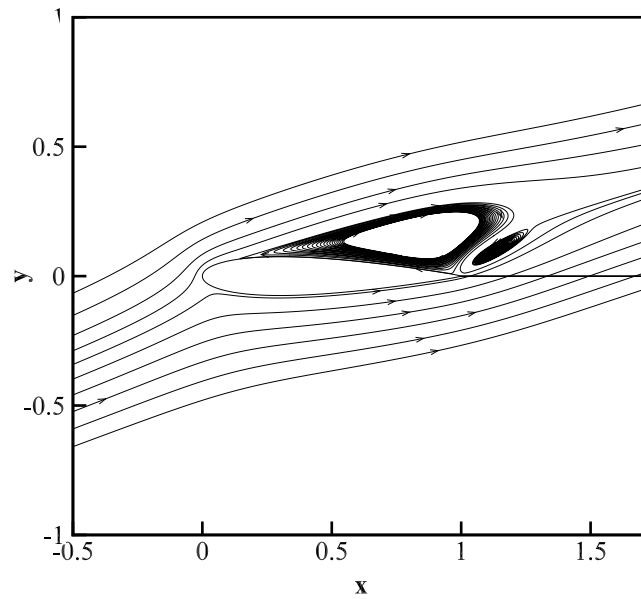


Figure 4.36: TE separation of NACA0015 airfoil $AoA = 18.0$ degrees

The evaluation of the TE separation is given in Figure 4.33, 4.34, 4.35 and 4.36. The solution is the steady state results of IFVLBM. It can be seen that the TE separation starts around $AoA = 10.0$ and it evolves as the AoA increases. At $AoA = 18.0$

degrees the airfoil is almost fully stalled. In Figure 4.37, the velocity vectors on the suction side of the airfoil is given. It can be seen that, the flow is reversed on the upper side at almost 90 percent of the airfoil.

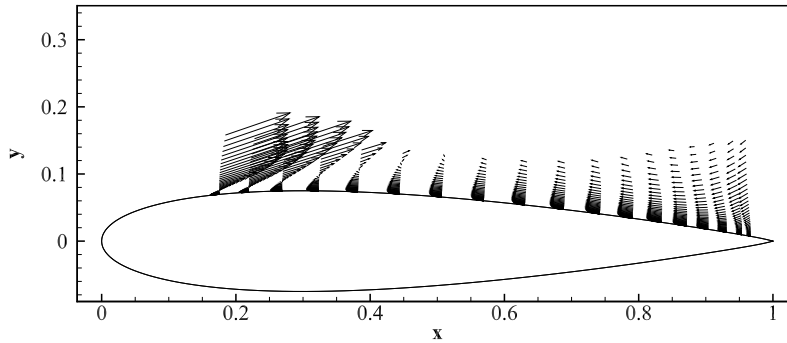


Figure 4.37: Velocity distribution on the surface and vicinity of airfoil at $AoA = 18$

4.3 Unsteady Flow

For the last test case, unsteady flow over a cylinder is solved. The time accurate solution capability of the IFVLBM is compared with the data obtained from the literature.

4.3.1 Flow Past a Circular Cylinder

Flow past a circular cylinder is one of the classical problems of the computational fluid mechanics discipline. Numerous experimental and numerical studies has been performed to investigate unsteady viscous flow past a circular cylinder. The problem is useful to study the vortex shedding phenomenon. For low Re numbers based on the diameter of the cylinder, the flow is symmetric, when the Re number increases the flow begins to separate.

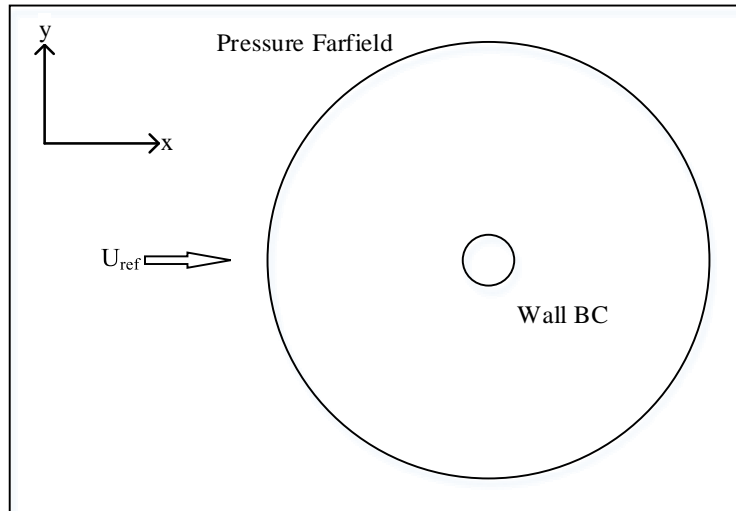


Figure 4.38: Cylinder problem setup

The laminar steady cases are presented in section 4.1.3. When Re number is between ~ 50 and ~ 200 , the upper and lower vortices are detached from the cylinder alternately and travel downstream. The phenomena is known as Karman Vortex Street. A parameter that describes the unsteady flow past a circular cylinder is the Strouhal number[77, p. 32], a relation between the vortex frequency and the free stream velocity.

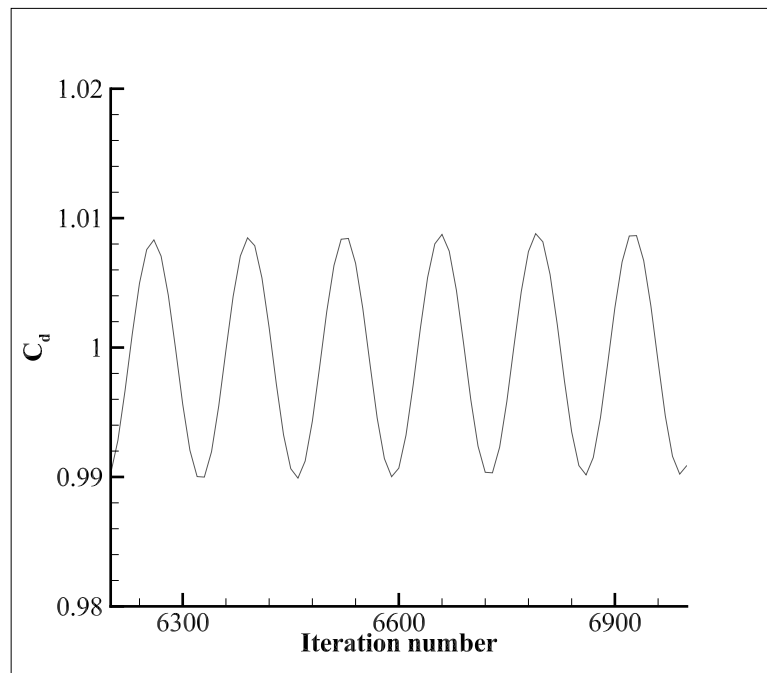


Figure 4.39: Drag coefficient change wrt iteration

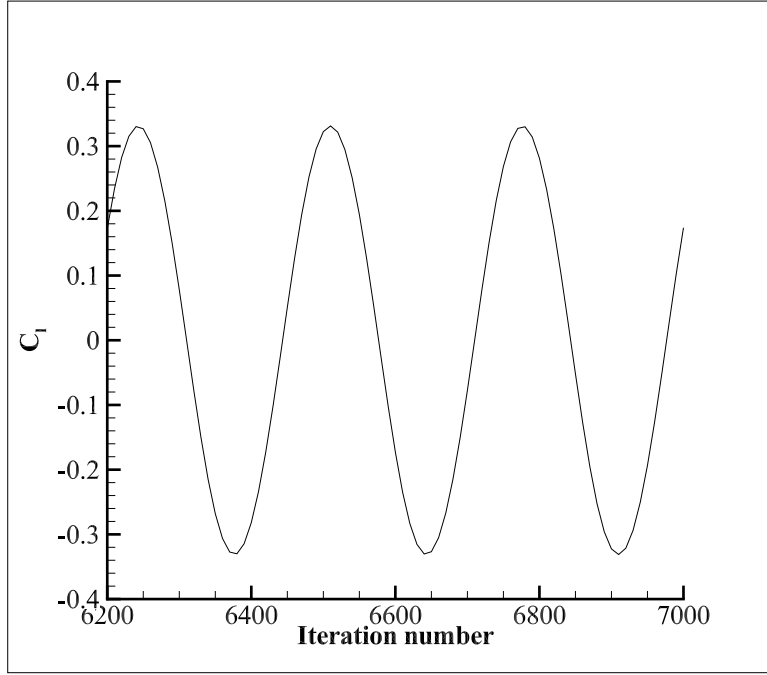


Figure 4.40: Lift coefficient change wrt iteration

The grid used for the problem is a 201 x 173 nodes “O” type grid same for the steady. The solutions are obtained for $Re = 100$. The cylinder diameter is 1 unit. The first cell height from the surface of the cylinder is 0.001 diameter. The nondimensional time increment is set to $\Delta t = 0.20$ and the number of sub iterations used for dual time stepping is 20. The average and RMS quantities given in Table 4.4 are calculated for 3 periods after the flow becomes totally periodic.

The frequency of the oscillation can be calculated from the C_d change or C_l change with respect to iteration from the figures. As expected, the frequency of change in drag is twice the frequency of change in lift.

The formula of Strouhal number is given in equation 4.8, where f is the vortex frequency, d is the diameter and V is the free stream velocity.

$$St = \frac{fd}{V} \quad (4.8)$$

Strouhal number can be expressed by lattice units as follows:

$$St = \frac{\sqrt{3}}{\Delta t * Ma_{\infty} * (\#peak3 - \#peak1)} \quad \text{for Cd} \quad (4.9)$$

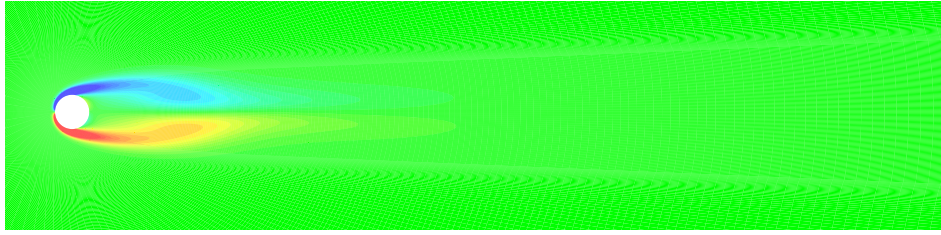
$$St = \frac{\sqrt{3}}{\Delta t * Ma_{\infty} * (\#peak2 - \#peak1)} \quad \text{for Cl} \quad (4.10)$$

The obtained results are given in Table 4.4.

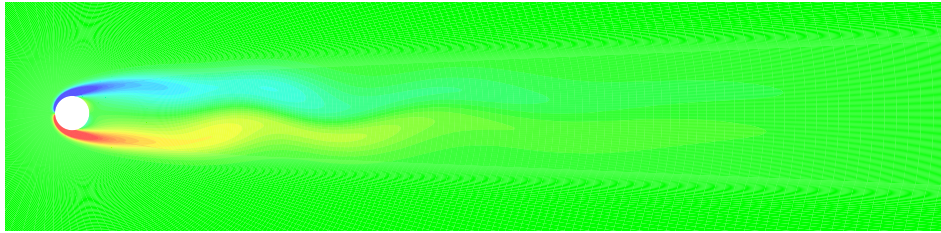
Table 4.4: Unsteady results comparison table for $Re = 100$

	$\overline{C_d}$	$C_{L,RMS}$	St
Golani[78]	1.3063	0.2181	0.1626
Rajani et al.[79]	1.3353	0.1792	0.1569
Posdziech and Grundmann [80]	1.3100	-	0.1630
Mittal [81]	1.3220	0.2256	0.1644
IFVLBM	1.3320	0.2137	0.1625

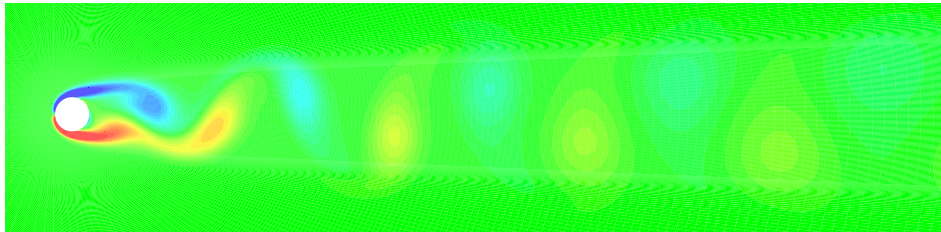
The calculated average drag coefficient is in good agreement with the literature. Some vorticity contours at different lattice times are shown in Figure 4.41. As seen from the figures, the flow is symmetric first, then the oscillation starts and after some time, it becomes completely periodic. The periodicity is observed also from the oscillation in drag and lift coefficients.



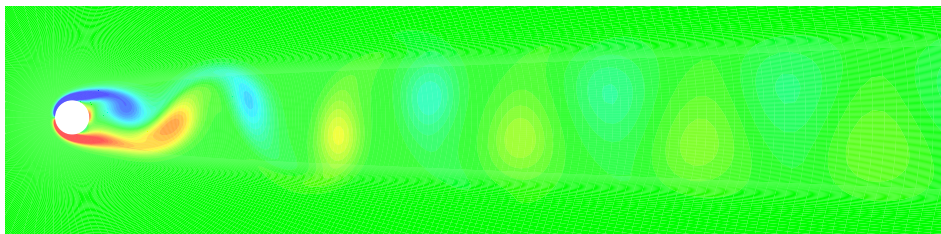
(a) lattice time $t = 125$



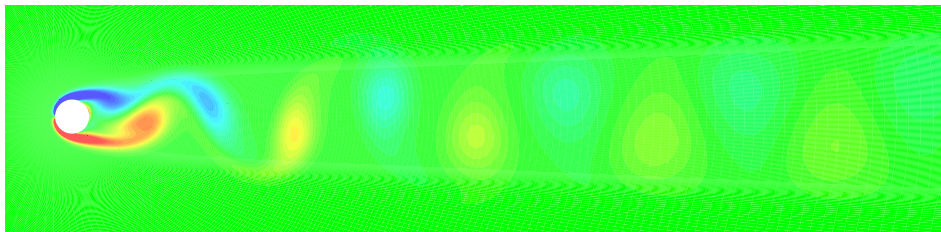
(b) lattice time $t = 250$



(c) lattice time $t = 500$



(d) lattice time $t = 750$



(e) lattice time $t = 1000$

Figure 4.41: Vorticity contours for $Re = 100$

4.4 Grid Convergence

The first aim to perform CFD simulation is to have accurate results. The results itself depend on several parameters. Using the proper size of the grids are one of the important parameter that the accuracy of the results depend on. It is a common approach to check the grid convergence for the CFD simulations. However, in this manuscript most of the validation cases have their grid available in the literature and the same grid is used for IFVLBM simulations. But, for the flow over a cylinder case and flow over NACA0015 case the grids must be checked for grid dependency.

It is assumed that, when the grid becomes finer and finer, the solution will asymptotically converge to the exact value. In general, 2 or 3 levels of grids are used to estimate the continuum value which is the theoretical value for zero grid spacing. Then the errors for the used grid spacings are calculated.

Richardson extrapolation is used to calculate the zero value of the function at zero grid spacing.

$$\phi_0 = \phi_1 + \frac{\phi_1 - \phi_2}{r^p - 1} \quad (4.11)$$

In equation 4.11, ϕ is the value of the function, index 1 represents the value belongs to the fine grid and index 2 represents the value belongs to the medium grid, r is the ratio of grid spacings and p is the order of convergence. If order of convergence is known p can directly used. If three level of grid is used p can be calculated as follows:

$$p = \frac{\ln\left(\frac{\phi_3 - \phi_2}{\phi_2 - \phi_1}\right)}{\ln(r)} \quad (4.12)$$

then the percent error for each level of grid spacing can be calculated by using the formula

$$\% \varepsilon_n = \frac{\phi_n - \phi_0}{\phi_0} \cdot 100 \quad (4.13)$$

Looking at the errors the desired grid level can be chosen.

4.4.1 Cylinder Grid

Three levels of grid is used to check the grid dependency. The drag coefficient is used as the function to estimate the errors. In Table 4.5, the drag coefficient values are given, the grid spacing is normalized with respect to fine grid and number of cells is normalized with respect to coarse grid.

Table 4.5: Grid dependency for cylinder

Grid Type	Grid Spacing	Grid Res.	# Cells	C_d	$\% \varepsilon$
Theoretical Value	0			1.5425	
Fine	1	401×341	16	1.5426	0.0068
Medium	2	201×171	4	1.5436	0.0716
Coarse	4	101×86	1	1.5643	1.4136

Equation 4.12 is used to calculate the convergence order. Then, by using the Richardson extrapolation, the C_d value for 0 grid spacing is calculated. The asymptotic behavior can be seen in Figure 4.42.

Looking at the errors medium grid is chosen for the flow over circular cylinder simulations.

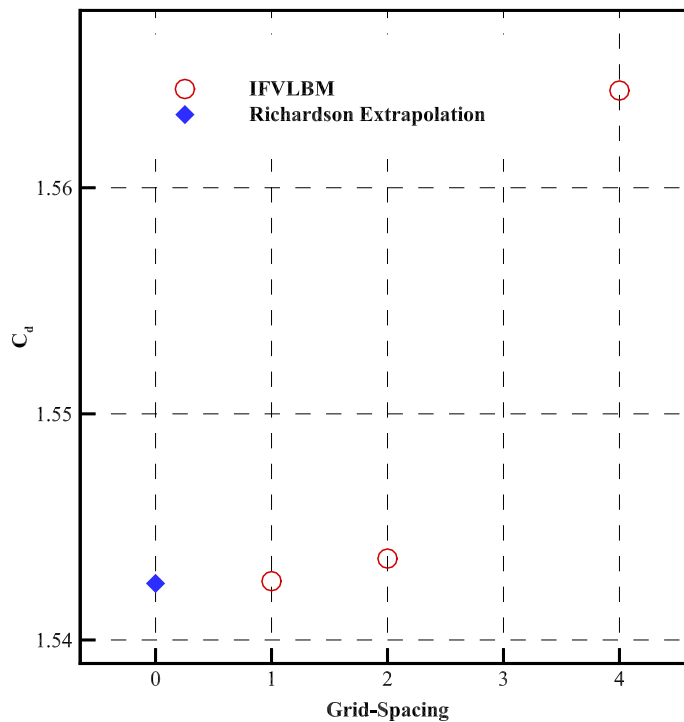


Figure 4.42: Asymptotic behavior of C_d for cylinder grids

4.4.2 NACA0015 Grid

A similar study is performed for NACA0015 grid convergence. For this case Lift coefficient is chosen as the function to be investigated for the grid convergence. The theoretical value is calculated by using the Richardson extrapolation. The calculation results and the errors are given in Table 4.6.

Table 4.6: Grid dependency for NACA0015

Grid Type	Grid Spacing	Grid Res.	# Cells	C_l	% ϵ
Theoretical Value	0			1.0256	
Fine	1	753×341	16	1.0194	0.6009
Medium	2	377×171	4	0.9981	2.6778
Coarse	4	189×86	1	0.8706	15.1100

The asymptotic behavior of the C_l value is shown in Figure 4.43. The medium case for NACA0015 airfoil is chosen for the validation runs as the error is in the acceptable region.

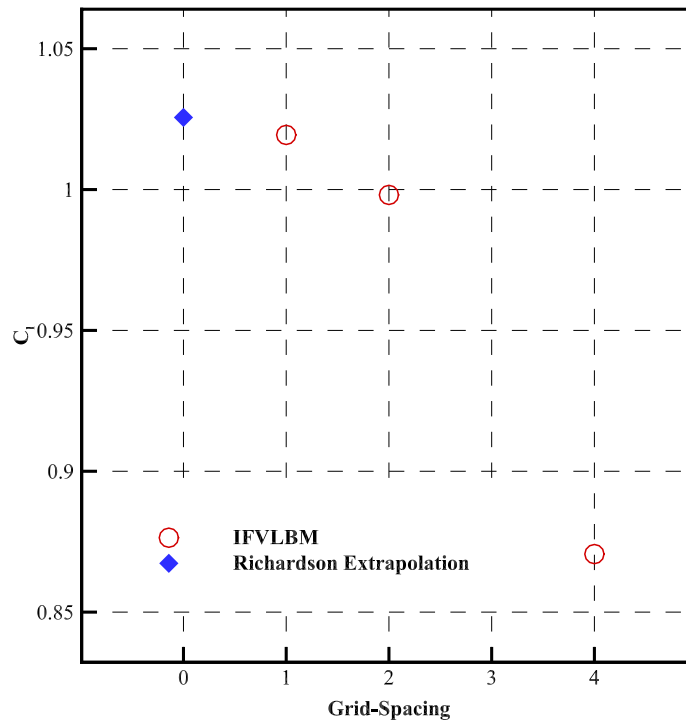


Figure 4.43: Asymptotic behavior of C_l for NACA0015 grids

The computational performance of IFVLBM will be discussed in Chapter 5.

CHAPTER 5

3D MODEL VALIDATION & MODEL ASSESSMENT

As stated in Chapter 4, a 3D code is written for IFVLBM method. The code strictly depends on the 2D code. Only the vector and matrix dimensions are increased from 9 to 19, since the solution stencils used in 2D and 3D problems are D2Q9 and D3Q19 respectively. One other difference is number of sweeps for the ADI method, 2 sweeps are performed in 2D whereas 3 sweeps are performed for 3D flow simulations.

Three examples are given for 3D codes. The first and second examples are the comparison for the 2D and 3D codes. Since it is just the comparison of the results are not compared with the literature. Both the aspect ratio of NACA0015 and confined cylinder is infinity. The examples are used to validate the implementation of 3D iteration algorithms in single block and multi block respectively. Third example is the laminar flow over a sphere. Fourth example is the comparison of the solution from the NASA-CFL3D code and IFVLBM code for a delta wing with 75 degrees sweep angle. The CFL3D code solution are extensively compared to the experimental data. In references [82] and [83], there are experimental data and the solution results of CFL3D code in the literature.

For each CFD code, the accuracy of the solution has to be known. The solution accuracy generally depends on the selected grid size and time interval. There are analytical and numerical methods to show the methods accuracy levels. In this manuscript, numerical methods are utilized for the study of temporal and spatial accuracy.

Then a comparison for convergence rate of an explicit FVLBM method and the present implicit method is performed.

A section for computational performance for the test cases presented in this thesis is also given in this chapter.

5.1 Flows in 3D

5.1.1 Flow over a Infinite AR NACA0015 3D Wing Section

The turbulent flow comparison case for 2D and 3D implementation of method described in this manuscript is given in this section. 2D problem setup is given in Figure 5.1. The airfoil section has 1 unit chord length. The solution domain is set by a C grid. The boundaries of free stream is taken about $x = -85$ to $x = 100$. The first cell height is 1×10^{-6} . The grid resolution is 377×171 where 201 point is on the airfoil. The Reynolds number used in the solution is $Re = 1955000$ and $M_\infty = 0.29$ and angle of attack $\alpha = 5$ degrees. For the 3D comparison the grid is replicated 5 times in the third direction as shown in Figure 5.2. Same conditions are used for the problem setup and the comparison is performed with the data taken from the mid section of the 3D solution.

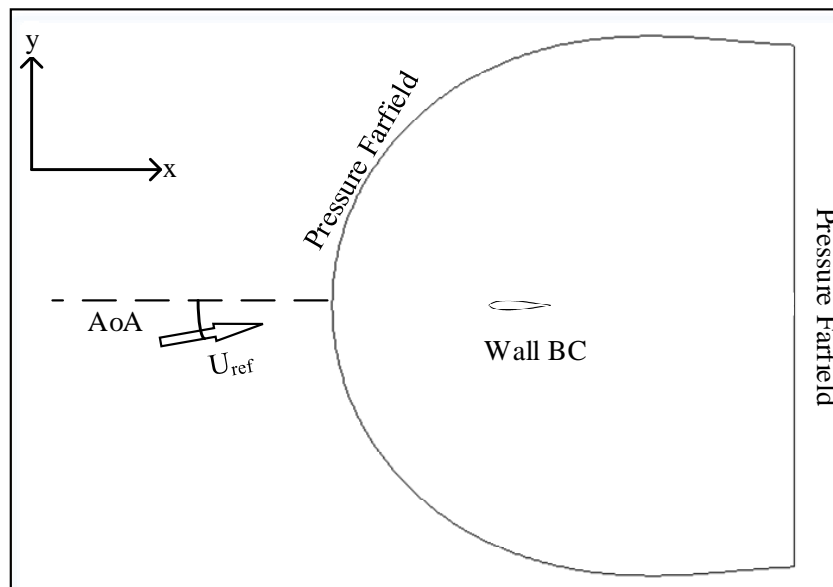


Figure 5.1: Problem setup for NACA0015

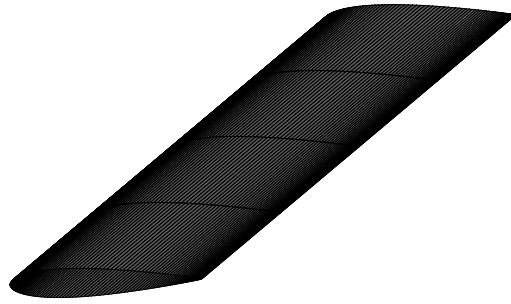


Figure 5.2: 3D NACA0015 airfoil sections

The solutions are performed for 10 thousand steps. The residuals for density and velocity is given Figure 5.3. In Table 5.1, the static coefficients are given. The difference between the C_l s are in about 2%. The pressure contours, velocity contours and some streamlines around the airfoil is given in Figures 5.4, 5.5 and 5.6.

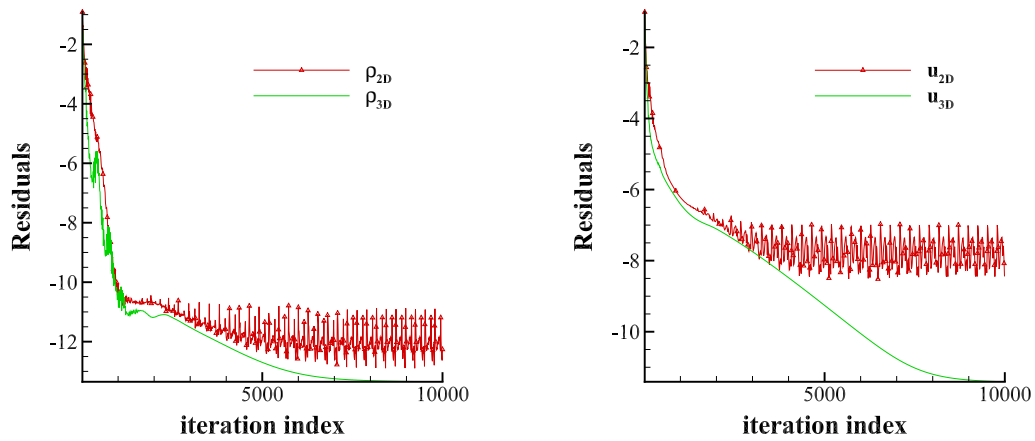
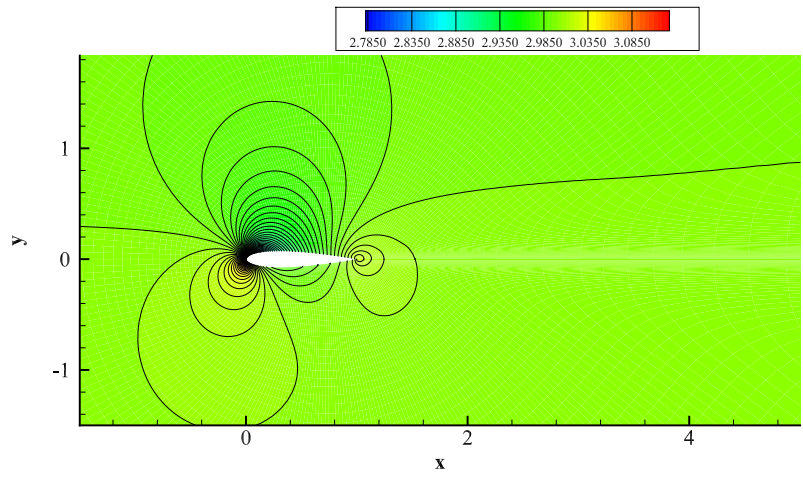


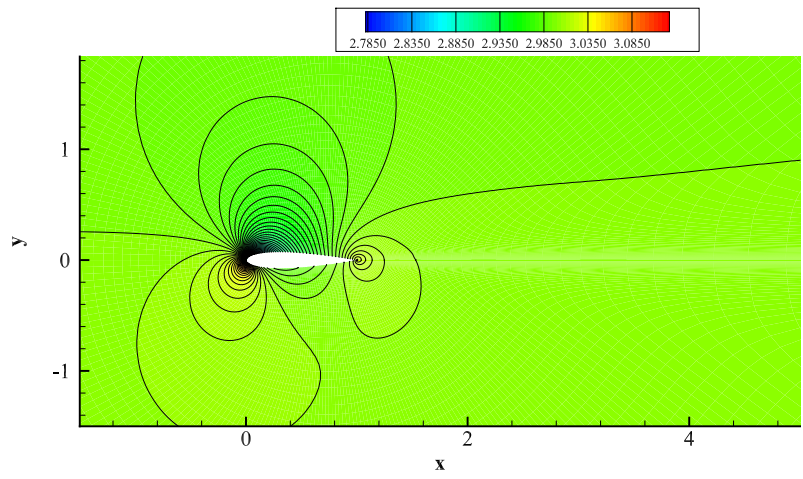
Figure 5.3: Residuals for NACA0015 solution

Table 5.1: Coefficient comparison for 2D and 3D solution

Dimension	C_l	$C_m@0.25$
2D	0.5258	0.01157
3D	0.5358	0.01008

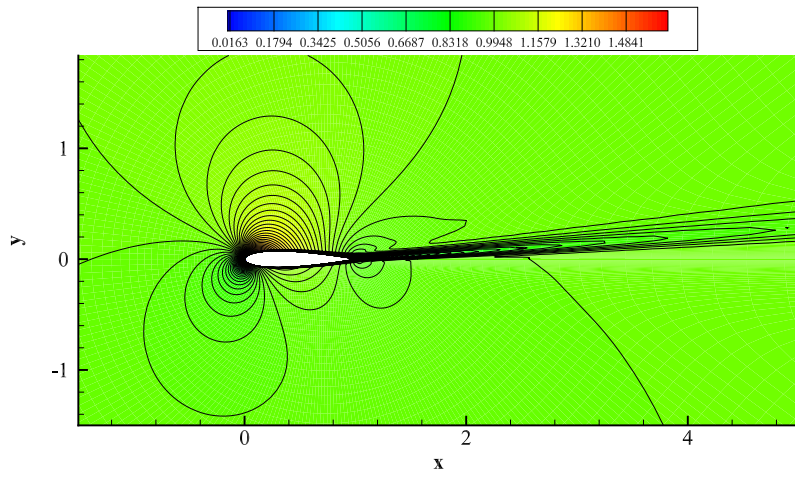


(a) 2D Solution

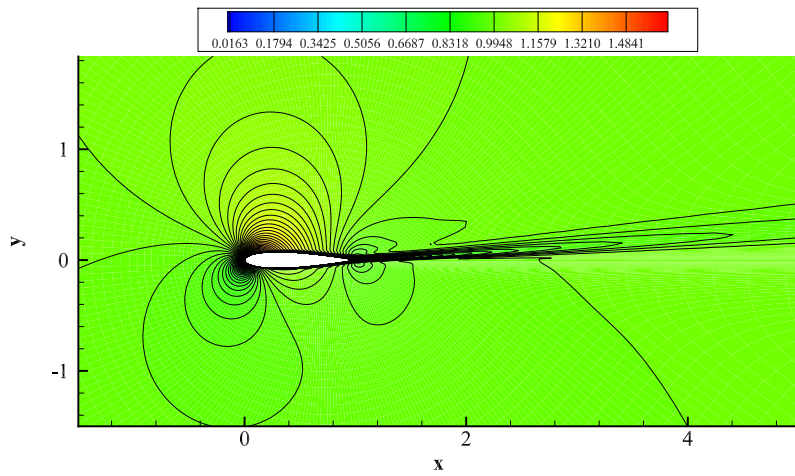


(b) 3D Solution

Figure 5.4: Pressure contours for NACA0015 solutions

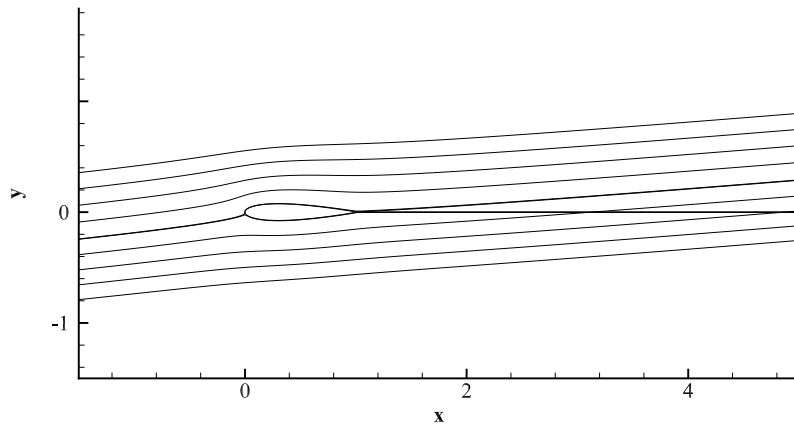


(a) 2D Solution

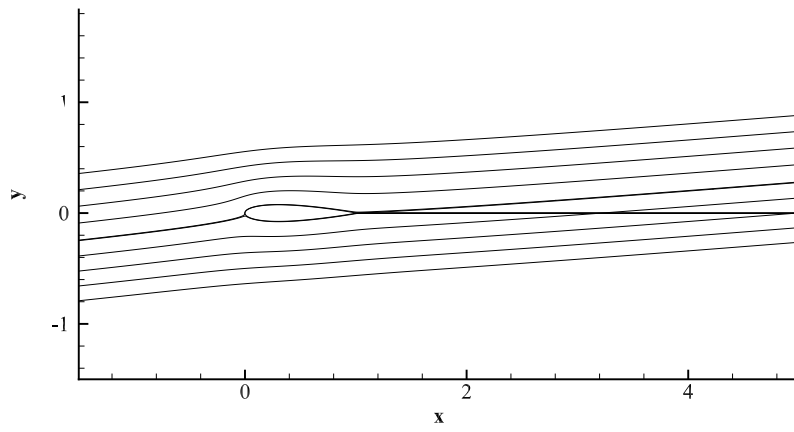


(b) 3D Solution

Figure 5.5: Velocity contours for NACA0015 solutions



(a) 2D Solution



(b) 3D Solution

Figure 5.6: Streamlines for NACA0015 solutions

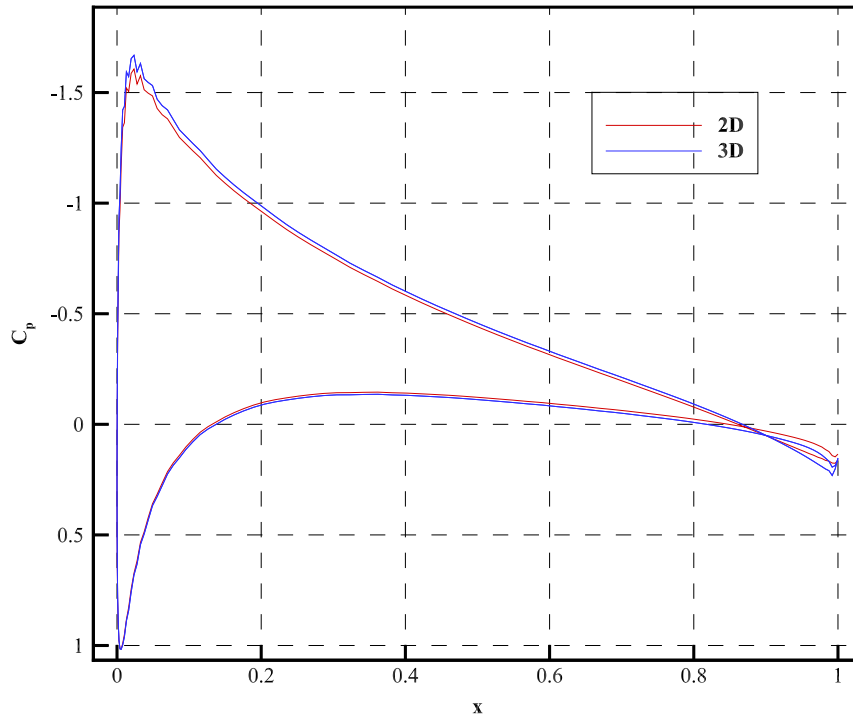


Figure 5.7: Pressure coefficient over 2D and 3D airfoil sections

The pressure coefficients are compared for 2D and 3D airfoil sections. The comparison is given in Figure 5.6.

5.1.2 Laminar Flow over a Infinite AR Confined Cylinder

The laminar comparison case for 2D and 3D implementation of the method described in this manuscript is given in this section. 2D problem setup is given in Figure 5.8. A single unit diameter cylinder is confined in a channel. The height of the channel is 6 units and length of the channel is 60 units. The cylinder is placed at the $1/3$ of the channel. The grid is divided into 5 blocks.

The Reynolds number is taken as 40, where the flow is not detached for a cylinder in free stream. Mach number is set to 0.2 and the solutions are iterated 5000. The grid resolutions of the blocks are given as 61×121 , 61×31 , 241×41 , 61×31 and 31×121 respectively. The residuals for density and velocity is calculated by equations 5.1 and 5.2.

$$\rho_{res} = \log \left(\sqrt{\frac{\sum_{Ncells} (\rho - \rho_{prev})^2}{Ncells}} \right) \quad (5.1)$$

$$v_{res} = \log \left(\sqrt{\frac{\sum_{Ncells} (\vec{u} - \vec{u}_{prev})^2}{Ncells}} \right) \quad (5.2)$$

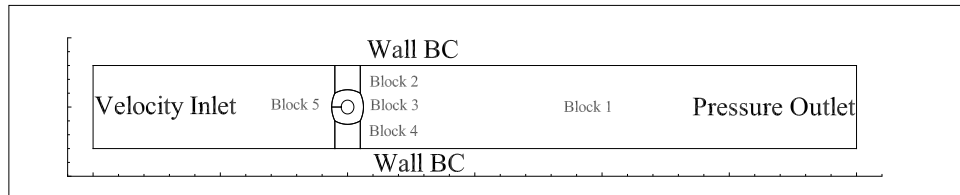


Figure 5.8: Problem setup for confined cylinder

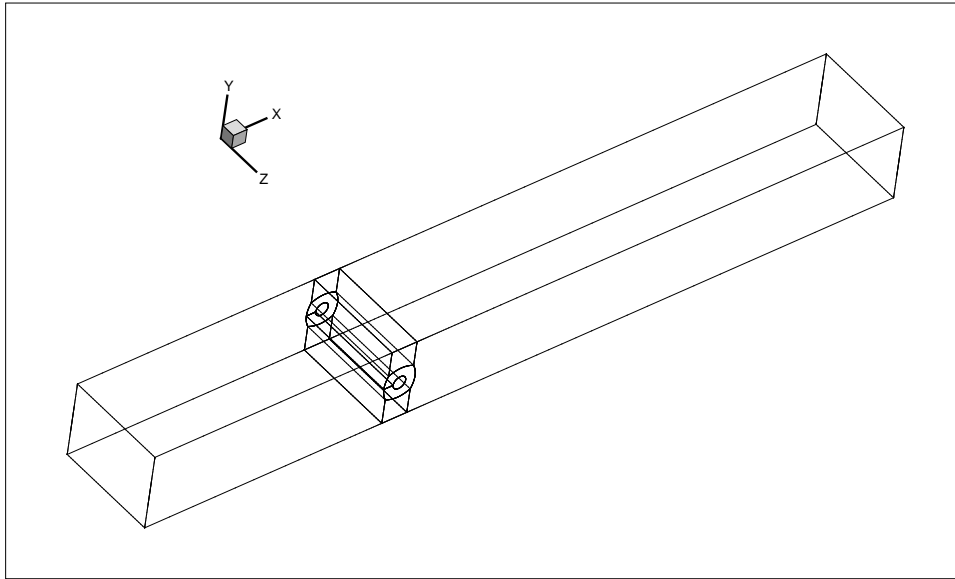


Figure 5.9: 3D problem setup for confined cylinder

For 3D problem setup the grids are replicated 41 times in out of plane direction. A schematic view of 3D problem is given in Figure 5.9. Symmetry boundary conditions are applied for the 3rd direction. The comparison performed between 2D and a slice taken from the 3D solution at $k=21$. The density and velocity residuals are given in Figure 5.10.

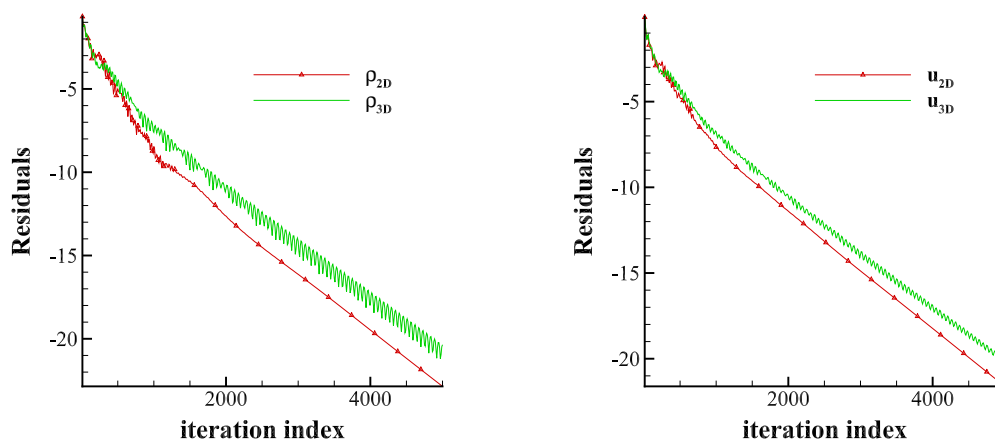


Figure 5.10: Residuals for confined cylinder solution

Table 5.2: Confined Cylinder Eddies

Dimension	L/r	θ°
2D	3.5345	34.25
3D	3.5424	34.30

The pressure coefficient is given as:

$$Cp = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2} \quad (5.3)$$

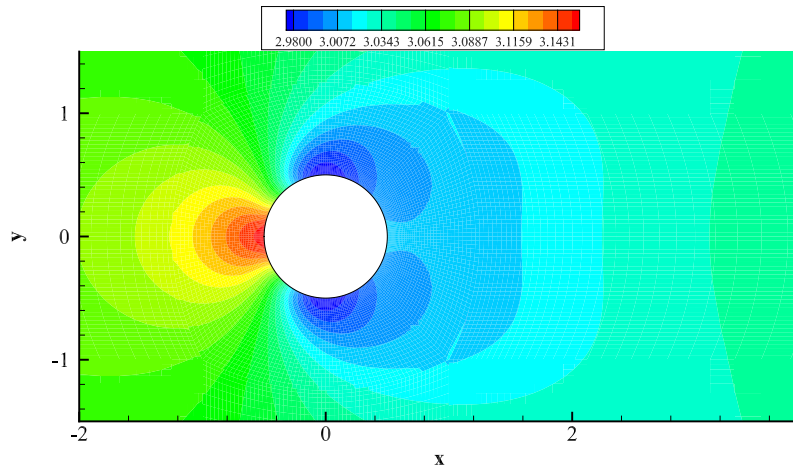
where p is the local pressure, p_∞ is the free stream pressure ρ_∞ is the free stream density and U_∞ is the free stream velocity. The same equation can be written by using the Lattice Boltzmann parameters as:

$$Cp = \frac{\rho - \rho_\infty}{\frac{1}{2}\rho_\infty M_\infty^2 M_r^2} = \frac{3\hat{\rho} - 1}{2 M_\infty^2} \quad (5.4)$$

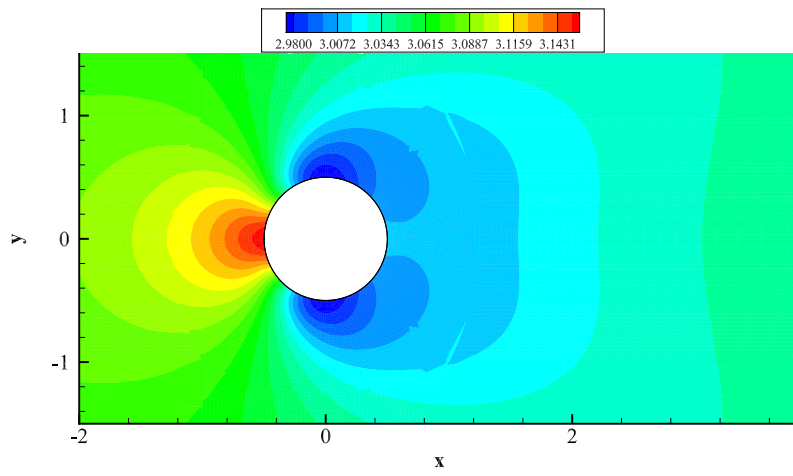
where $\hat{\rho} = \frac{\rho}{\rho_\infty}$ is the nondimensional density calculated by IFVLBM.

The contours of pressure coefficient for 2D and 3D solutions are given in Figure 5.11. The velocity contours for 2D and 3D solutions are given in Figure 5.12.

In Figure 5.13, streamlines in the wake of the cylinders are given. The length of the eddies that are formed after the cylinders are given in Table 5.2. Also in the same table the angle of separation point on the cylinder measured from the horizontal is given.

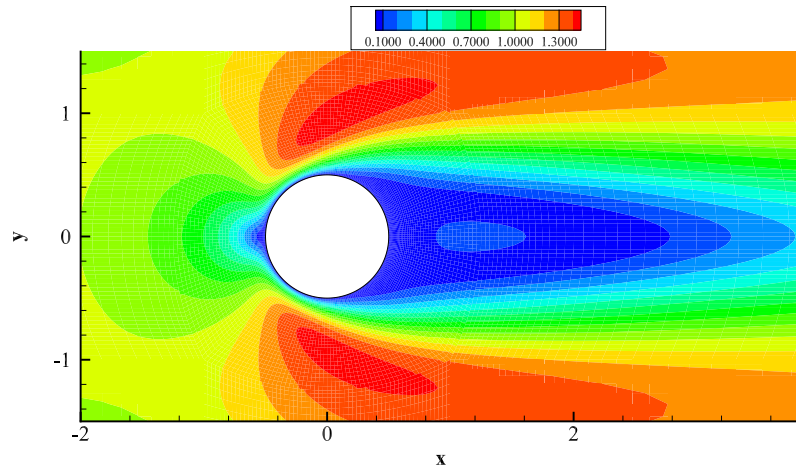


(a) C_p 2D Solution

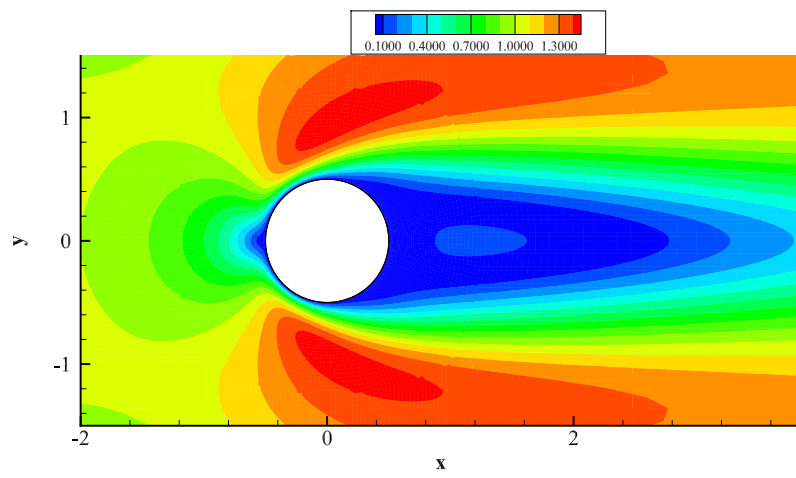


(b) C_p 3D Solution

Figure 5.11: Confined cylinder pressure coefficients

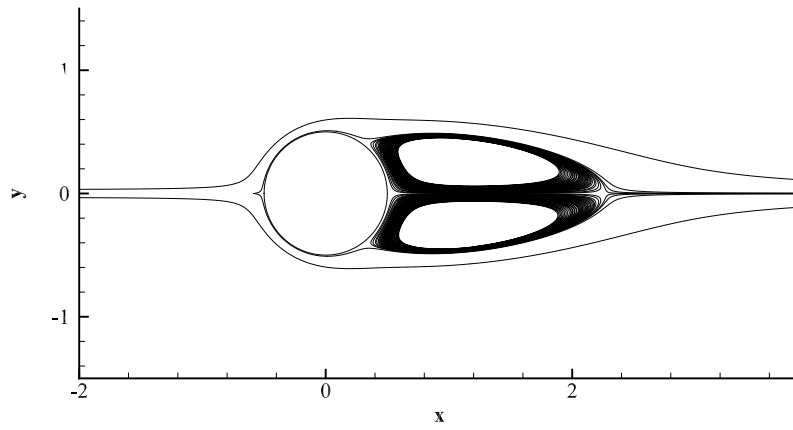


(a) 2D Solution

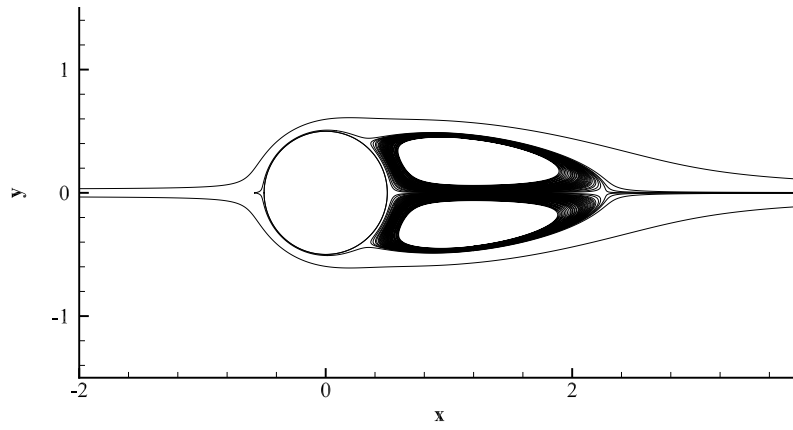


(b) 3D Solution

Figure 5.12: Confined cylinder velocity contours



(a) 2D Solution



(b) 3D Solution

Figure 5.13: Confined cylinder Streamlines

5.1.3 Uniform Flow Past a Sphere

To demonstrate the 3D capability of the model developed flow past a sphere is selected. There are numerical and experimental data available in the literature [84, 85, 86, 87] for comparison purposes. The geometry is simple but, it is very hard test case for numerical methods due to the 3D nature of the flow. The simulations are performed for $Re = 20, 60, 100, 150$ and 200 based on the free stream velocity and diameter of the sphere. The solution domain is selected as a outer sphere which has an 80 units diameter in terms of the sphere radius. The solution domain is shown in Figure 5.14.

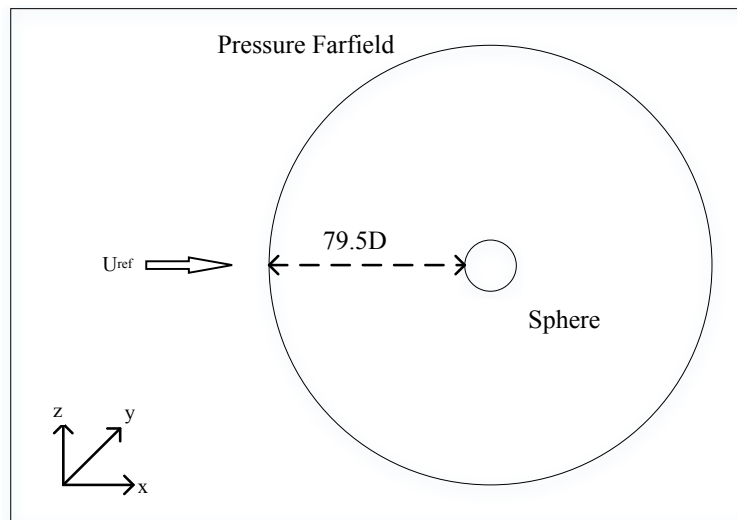


Figure 5.14: 3D problem setup for flow past a sphere

Three levels of grid resolution is used for grid convergence study. The C_d value of the sphere is used for the parameter to reach its final value asymptotically. Same extrapolation method described in section 4.4 is used to obtain the value for theoretical zero grid spacing.

Table 5.3: Grid dependency for 3D sphere

Grid Type	Grid Spacing	Grid Res.	# Cells	C_d	$\% \epsilon$
Theoretical Value	0			1.0742	
Fine	1	$105 \times 81 \times 101$	64	1.0727	0.1454
Medium	2	$53 \times 41 \times 51$	8	1.0674	0.6409
Coarse	4	$27 \times 21 \times 26$	1	1.0360	3.5589

The fine grid is used for the rest of the simulations. The solution domain and some mesh views are given in Figure 5.15. The first mesh height from the surface of the sphere is set to 5×10^{-4} .

The nature of the flow around a sphere shows different characteristics depending on the Reynolds number. Experimental investigations of Taneda [88] and Nakamura [89] shows that the wake behind the sphere is steady and axisymmetric, for low Reynolds numbers. The flow over a sphere has been investigated by other researchers both numerically or experimentally. The common consensus is that the flow is steady and axisymmetric for about $20 < Re < 200$. There is a stable vortex ring behind the sphere between the focused Reynolds numbers.

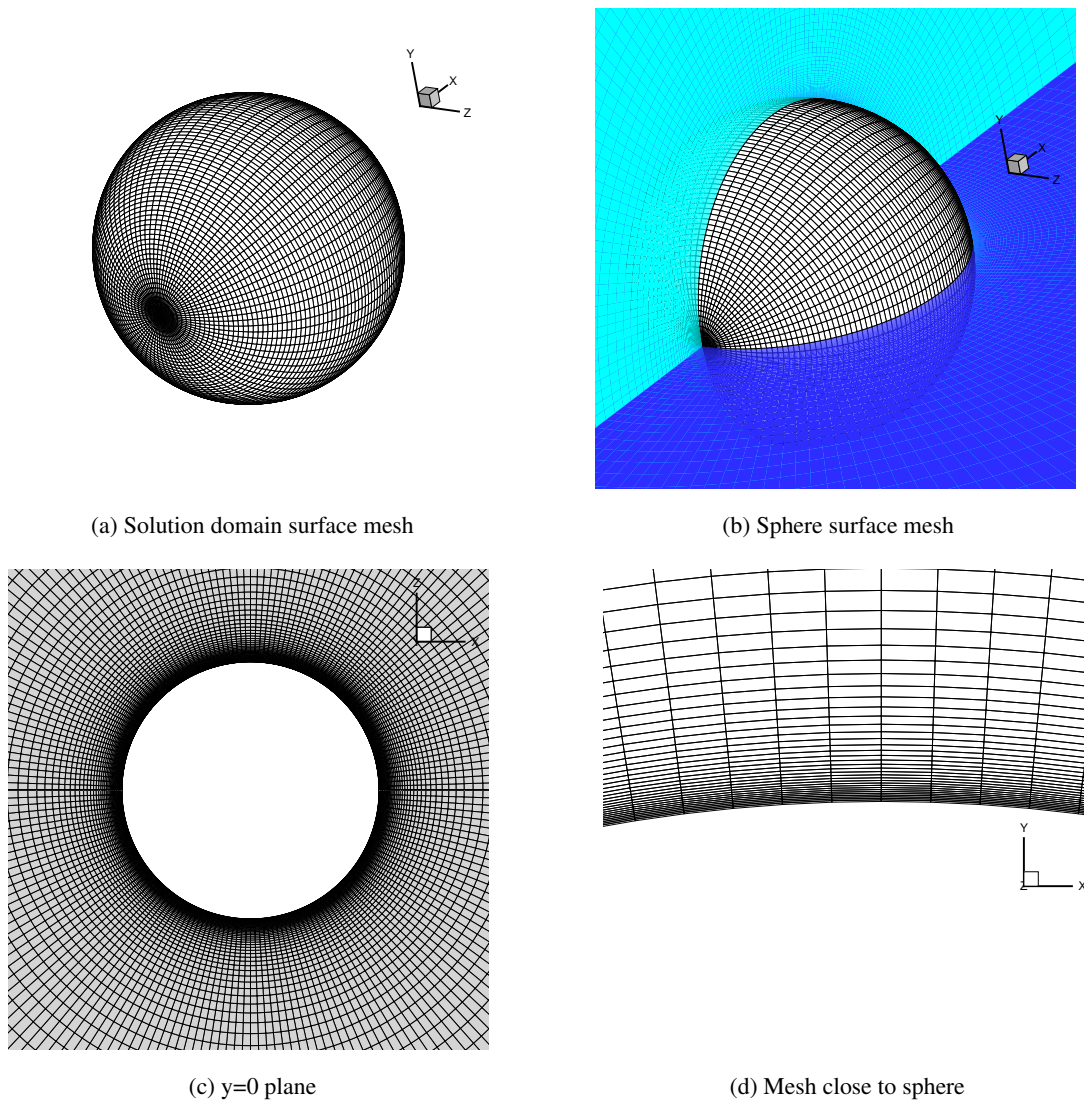


Figure 5.15: Sphere solution domain and surface meshes

The first comparison is the drag coefficient “ C_d ” with some literature data. The Figure gives a comparison of the drag coefficients from References [86, 90, 91, 92].

The pressure coefficient “ C_p ” is compared with the results of Gulcat et al. [87]. In his work, he has presented accurate calculations for the flow past a cylinder for $Re = 100$ by Finite Element Method. The comparison is given in Figure 5.17. The results are in good agreement with the reference values.

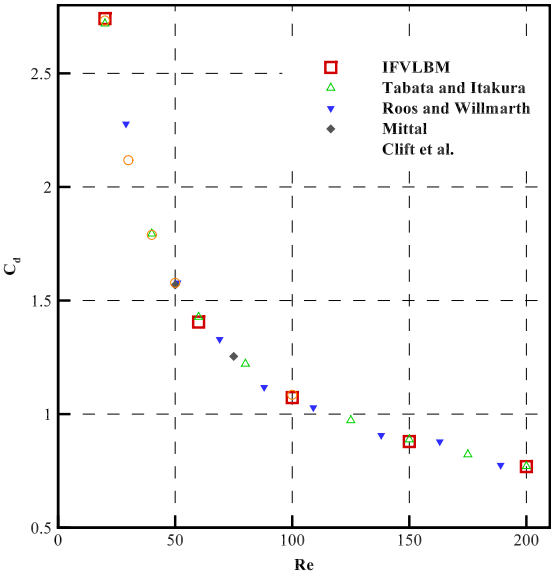


Figure 5.16: Drag coefficients of IFVLBM and literature data

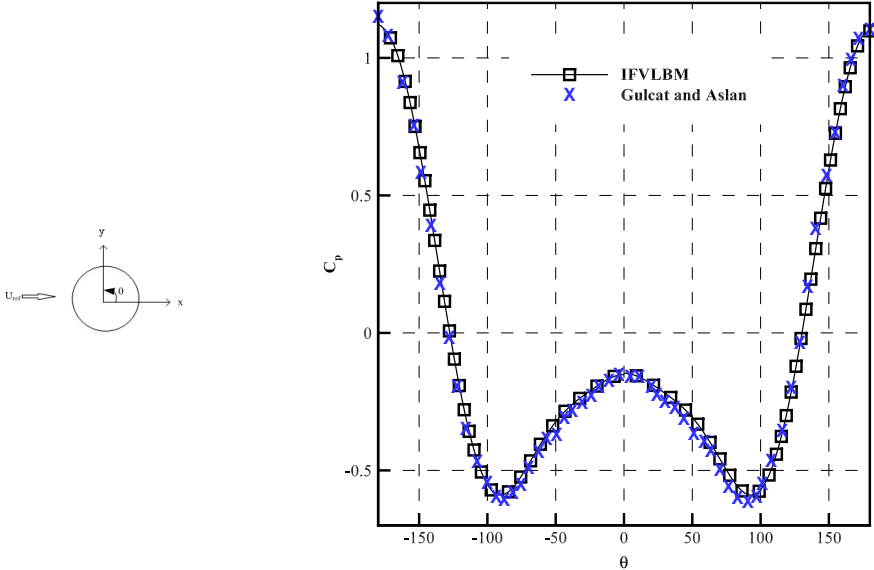
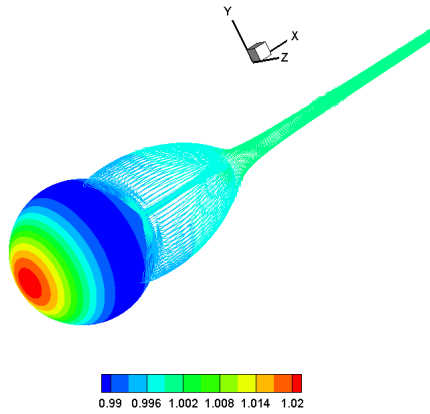
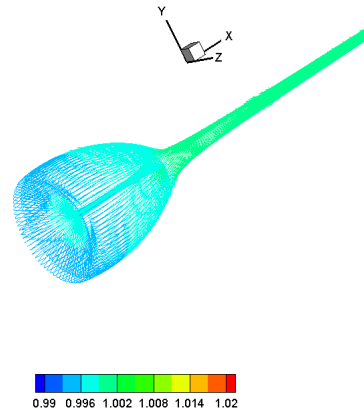


Figure 5.17: Pressure coefficient around the sphere at $z=0$ plane



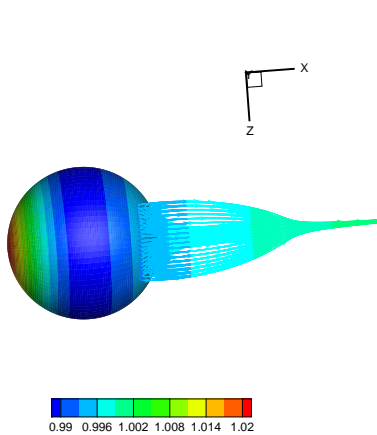
(a) Toroidal vortex with sphere



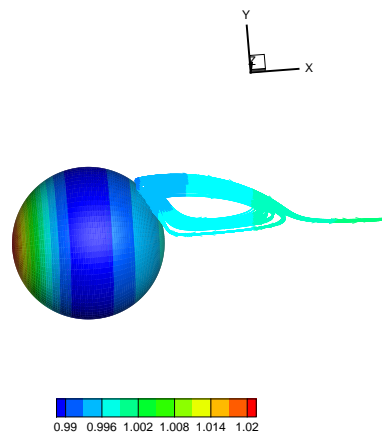
(b) Toroidal vortex without sphere

Figure 5.18: 360° toroidal vortex $Re = 100$

The flow past a sphere is axisymmetric and steady. In Figure 5.18, the toroidal streamlines are presented. In the left figure the sphere is in front of the streamlines, whereas in the right figure only streamlines are presented for clarity. The sphere has density contours and the streamlines are colored by density values.



(a) Top view

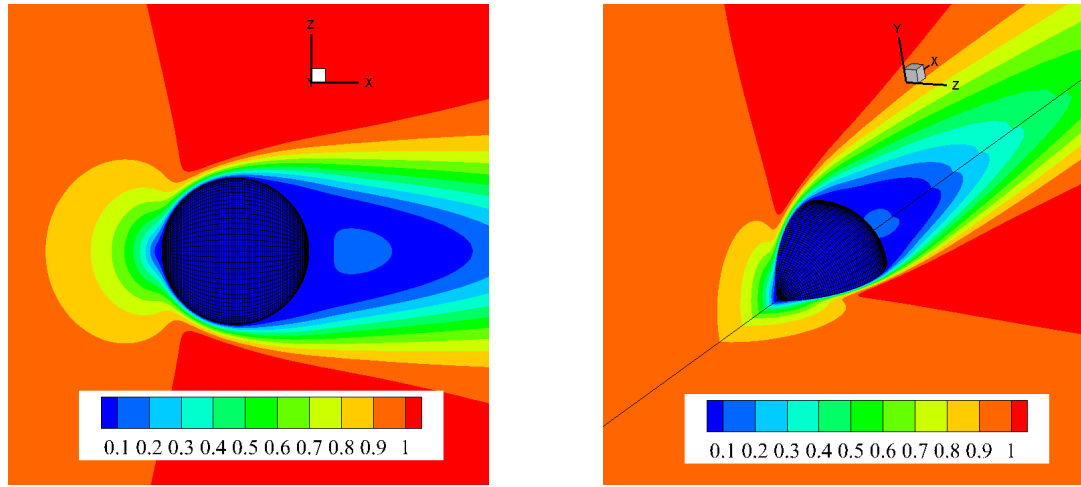


(b) Side view

Figure 5.19: A section of toroidal vortex $Re = 100$

In Figure 5.19, a section of toroidal streamlines are shown for a better visualization of the phenomenon. Again the sphere has density contours on and the streamlines are colored by density values.

In the Figure 5.20, the velocity contours are given. The left image of the figure, the contours at $y = 0$ section are given. The mesh is also shown on the sphere. At the right image of the figure, a quadrant between $y = 0$ and $z = 0$ planes are given.



(a) Velocity contours $y = 0$ plane

(b) Velocity contours $y=0, z=0$ plane

Figure 5.20: Velocity contours on some sections $Re = 100$

The length of the eddy behind the sphere, drag coefficient and the separation angle is can be used for a quantitative comparison with the data available in the literature. In Table 5.4 comparison of these parameters are given. The last column is the data of the IFVLBM simulation data. Except for the length of the eddy L_w behind the sphere, the data is in very good agreement with the literature.

Table 5.4: Comparison of computed data with literature data for $Re = 100$

	Ref. [84]	Ref. [85]	Ref. [86]	Ref. [87]	IFVLBM
C_d	1.05	1	1.09	1.07	1.073
θ	54.4	53.5	52.3	55	52.9
$\frac{L_w}{D}$	0.92	0.86	0.87	0.93	0.85

5.1.4 Flow over Delta Wing

As a benchmark case 3D delta wing is chosen. During late 80's, NASA has conducted experiments on delta wings. The 75 degrees swept delta wing is the test subject for some of the experiments [82, 83]. The experiments were carried out for the reason that, high performance aircrafts with delta wings generate substantial lift from the vortices generated by the sharp leading edges. The stability and the performance of the aircraft at high angles of attack are directly related to the stability, strength and the location of the vortex system.

During the experiments, it was shown that, up to $Re = 1 \times 10^6$ the flow was not turbulent. The main purpose of the experiments was summarized to supply validation data for computational methods. In reference [83], some of the produced data is compared with CFL3D code of NASA. For this manuscript, the results of IFVLBM will be compared to the solution data obtained from NASA web page [93], since it can be compared numerically.

The problem setup is given in Figure 5.21. The Reynolds number is taken as $Re = 0.5 \times 10^6$ and the free stream Mach number is $M_\infty = 0.3$. The grid resolution is $37 \times 65 \times 65$ in I, J, K directions respectively.

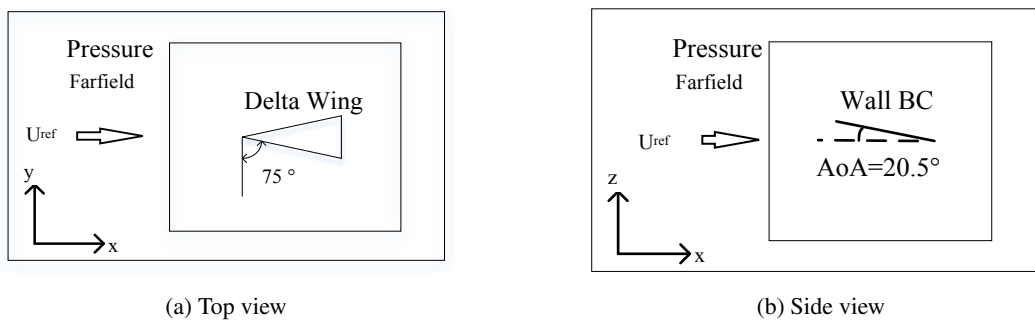
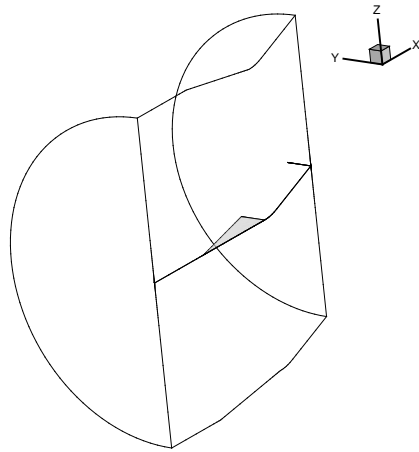
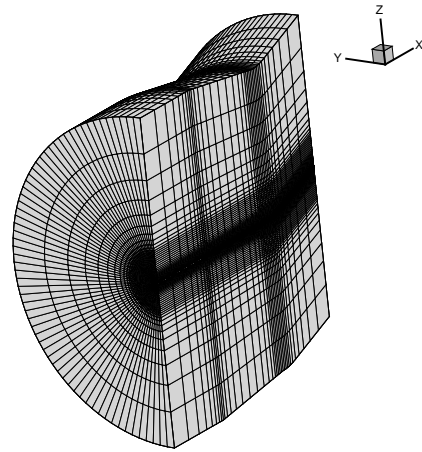


Figure 5.21: 3D delta wing problem setup

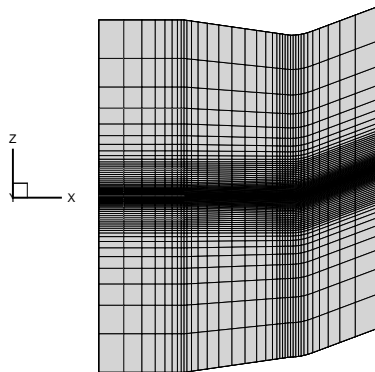
The problem is symmetric with respect to longitudinal axis, hence the solution domain is divided into 2 using the symmetry boundary condition.



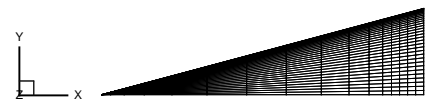
(a) Solution domain



(b) Solution domain surface mesh



(c) Side view mesh



(d) Delta wing surface mesh

Figure 5.22: Delta wing solution domain and surface meshes

The length of the delta wing on the symmetry axis is taken as unity. The solution domain, domain surface mesh, side view of the domain and the delta wing surface mesh are shown in Figures 5.22a, 5.22b, 5.22c and 5.22d respectively.

The comparison of IFVLBM and CFL3D is done by quantitative parameters first. The quantitative parameters are the static aerodynamic coefficients, pressure coefficients of some sections on the delta wing and the center of the vortices on same sections.

The over all static force coefficients calculated from both methods are given in Table 5.5. The results are in good agreement with the CFL3D NS solver.

Table 5.5: Static aerodynamic coefficients for CFL3D and IFVLBM

Dimension	C_l	C_d	$C_{m@0.25}$
CFL3D	0.795958	0.306602	-0.28840
IFVLBM	0.790416	0.304448	-0.28594

To make more reasonable comparison, some data are extracted at $x = 0.2$, $x = 0.4$, $x = 0.6$ and $x = 0.8$ locations. The pressure coefficient “ C_p ” values are plotted against the traverse direction on the delta wing for the upper surface. The plots are given in Figures 5.23, 5.24, 5.25 and 5.26 respectively.

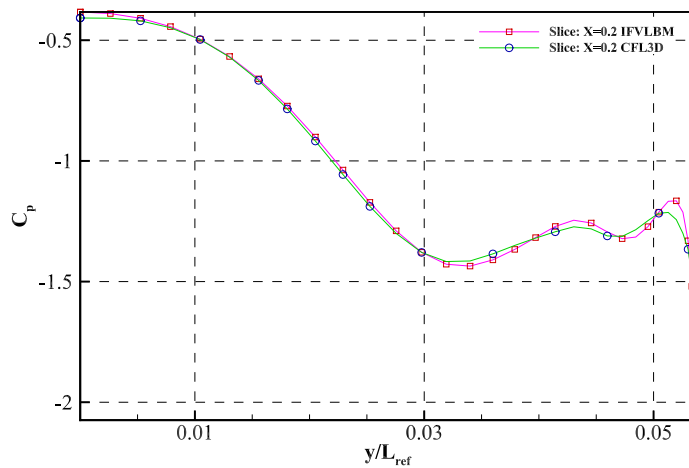


Figure 5.23: Pressure coefficient at upper surface $x = 0.20$

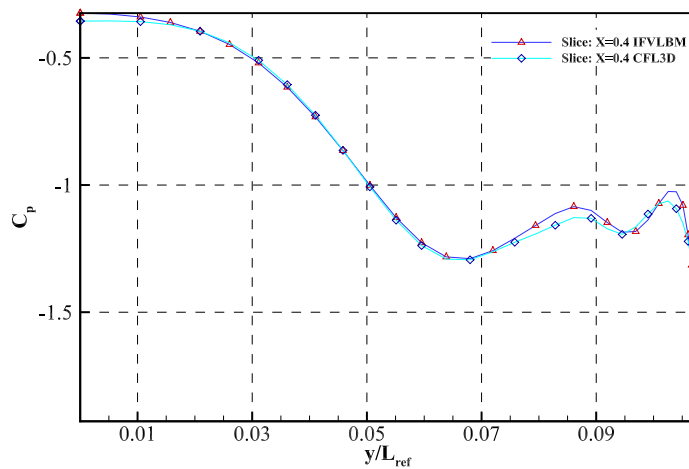


Figure 5.24: Pressure coefficient at upper surface $x = 0.40$

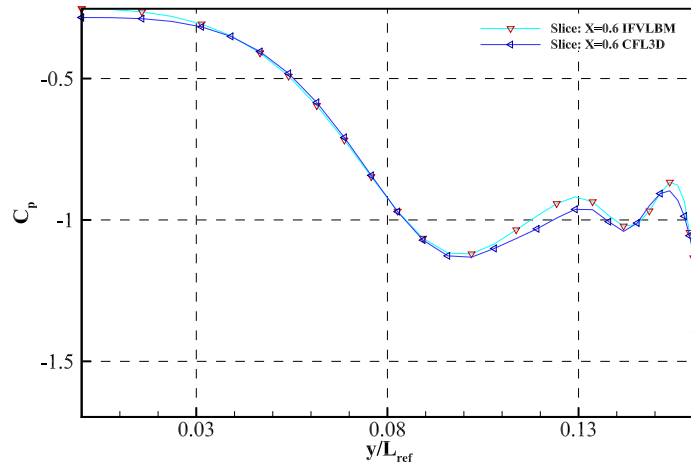


Figure 5.25: Pressure coefficient at upper surface $x = 0.60$

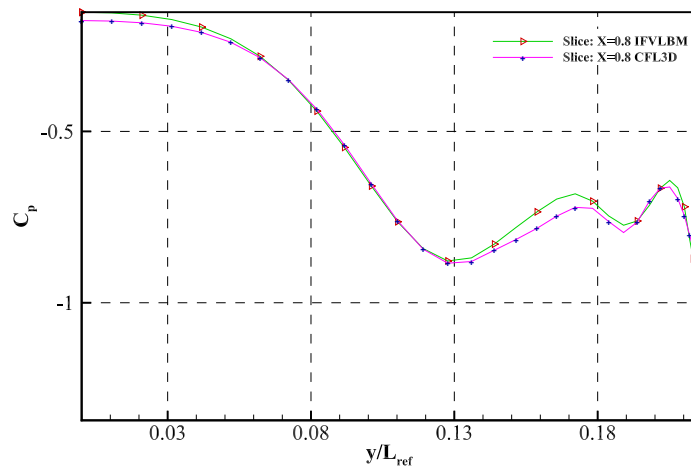


Figure 5.26: Pressure coefficient at upper surface $x = 0.80$

For each section the span length is difference so for comparison purposes all C_p are also plotted on the same graph shown in Figure 5.27. The upper surface C_p contours are given in Figure 5.28. The left figure is the result of the IFVLBM and the right figure is CFL3D result.

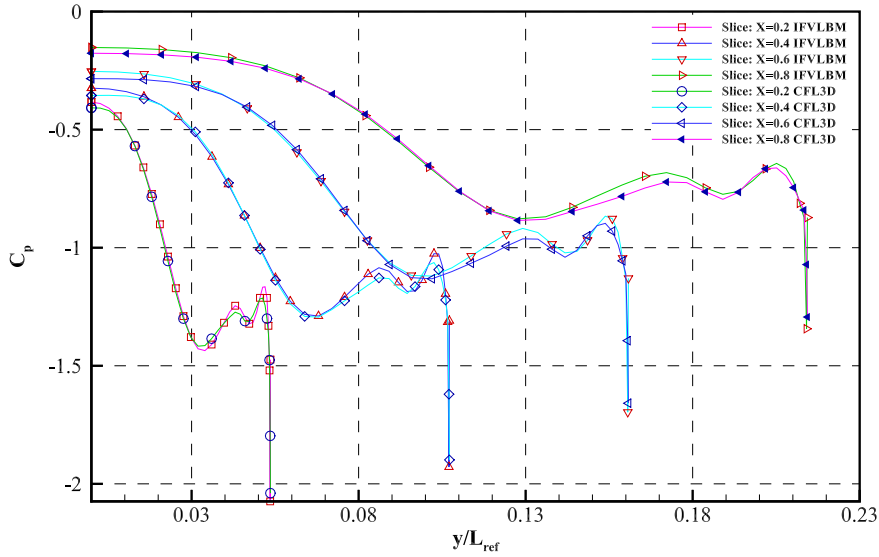


Figure 5.27: Pressure coefficient at upper surface for various chord locations

One other characteristic of the flow over the delta wing is the C_p distribution of the domain. The C_p contours are given in Figure 5.29. The domain symmetry is used to complement the solution domain, thus the plots are given for the whole delta wing.

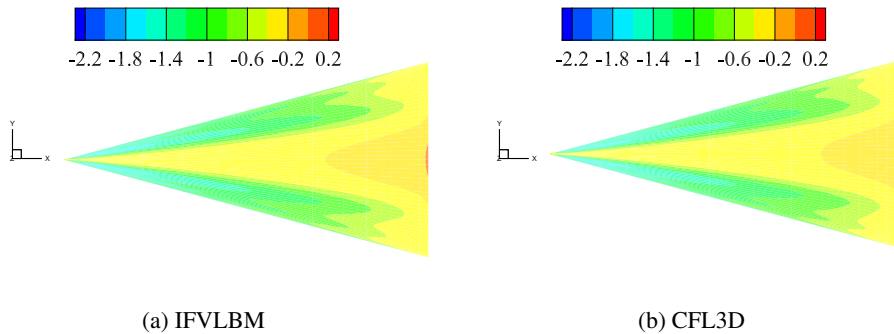


Figure 5.28: Pressure coefficient contours on upper surface

The pressure coefficient contours at different locations with respect to the main chord of the delta wing is plotted. One quantitative comparison can be done by using the center coordinates of the pressure coefficient eyes. The center coordinates are given in Table 5.6. The differences are given in domain units, where the major chord length of the delta wing is 1 unit. The differences are very small as the center of the contours are obtained from the graphics by hand.

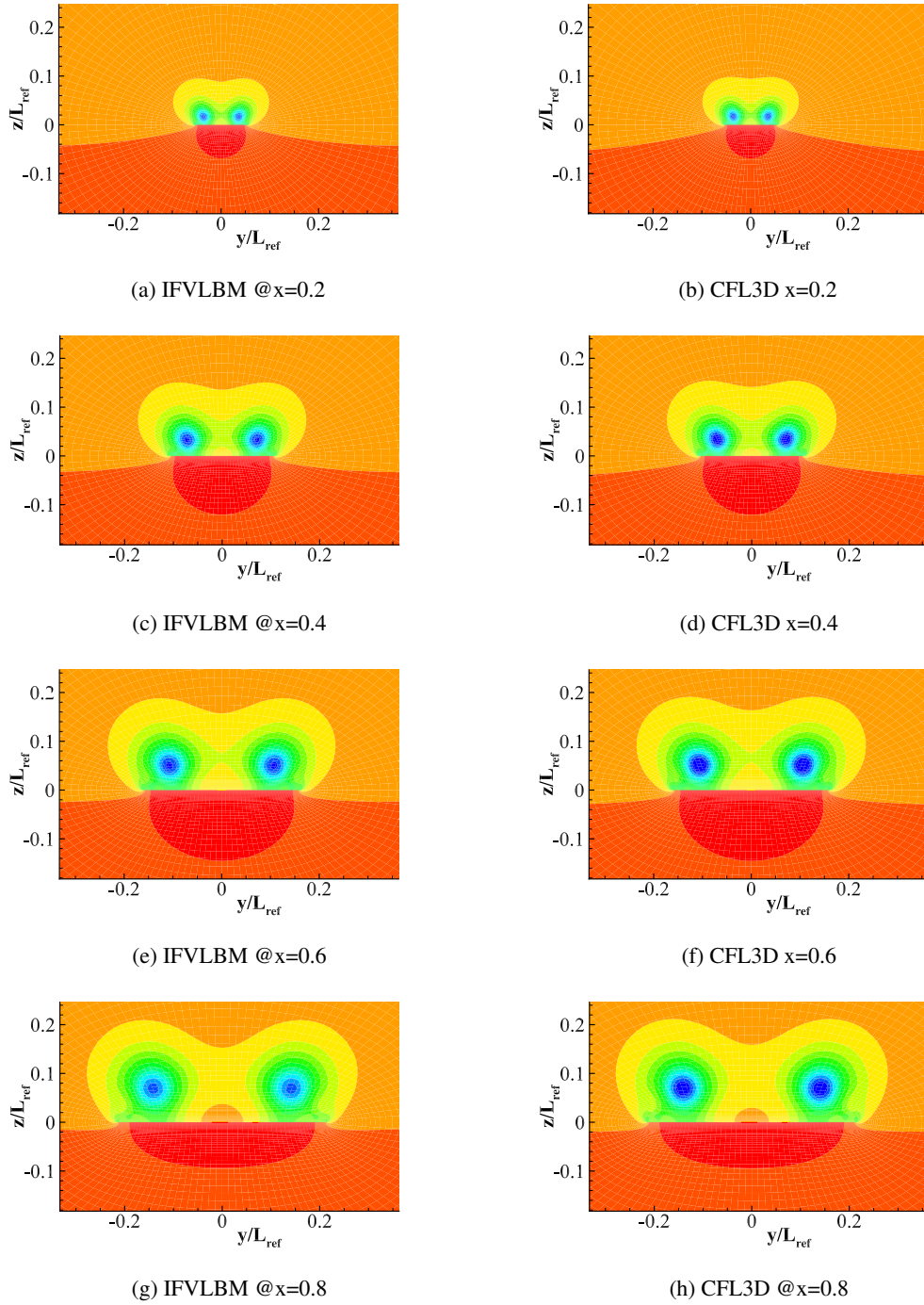
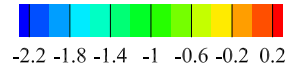


Figure 5.29: Pressure coefficient contours at various x locations

The flow field solution obtained from IFVLBM is used to draw the stream ribbons.

Table 5.6: Pressure coefficient center coordinates

IFVLBM			CFL3D			error in y	error in z
x	y	z	x	y	z		
0.20000	0.03585	0.01684	0.20000	0.03523	0.01731	0.0006	0.0005
0.40000	0.07132	0.03338	0.40000	0.07145	0.03355	0.0001	0.0002
0.60000	0.10647	0.05024	0.60000	0.10589	0.05074	0.0006	0.0005
0.80000	0.14237	0.06927	0.80000	0.14211	0.06977	0.0003	0.0005

The Figure 5.30 demonstrates the stream ribbons from different angles. It can be seen from the figures that IFVLBM has captured the vortex system generated by the leading edge of the delta wing.

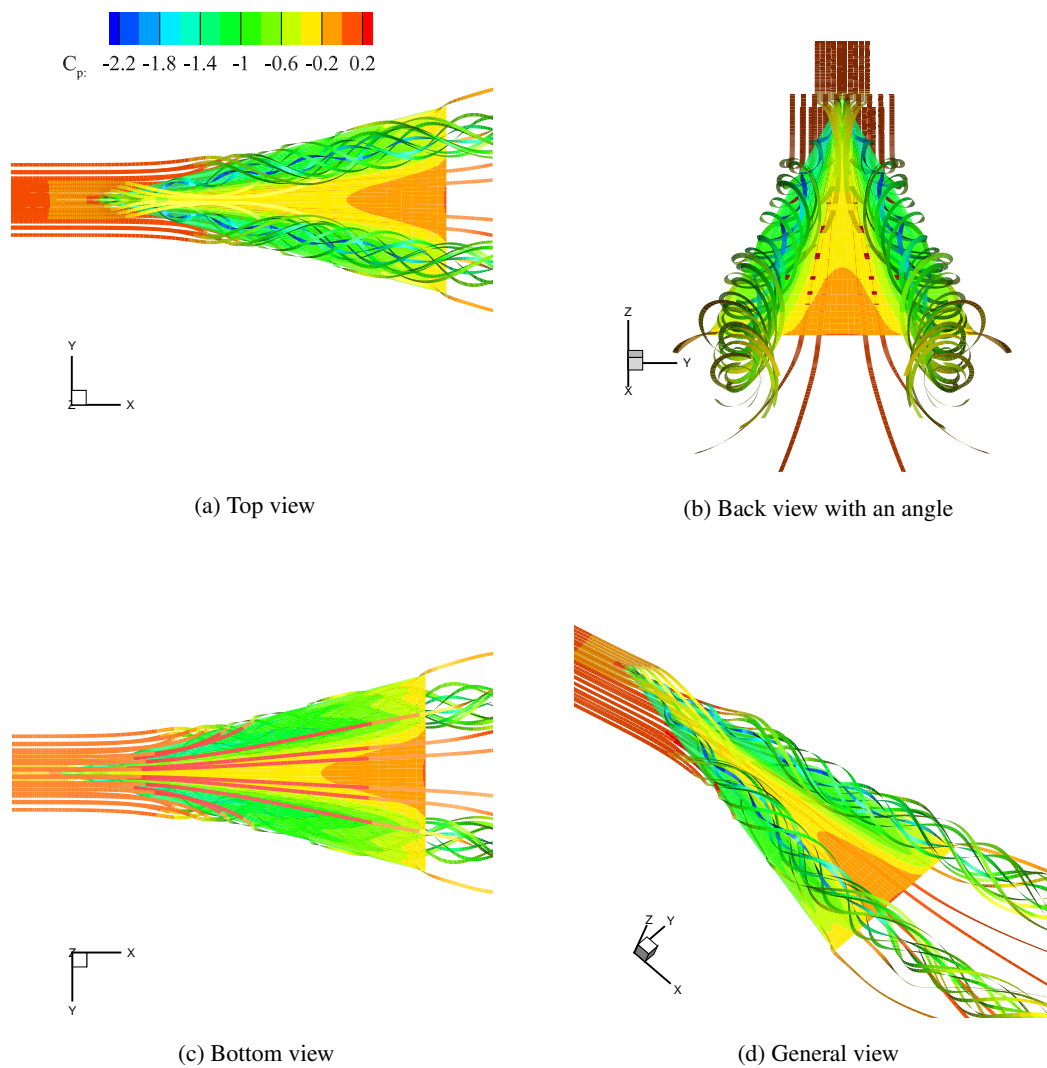
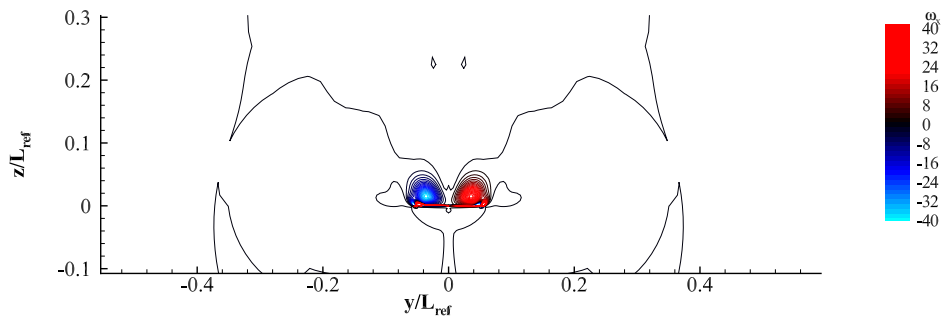


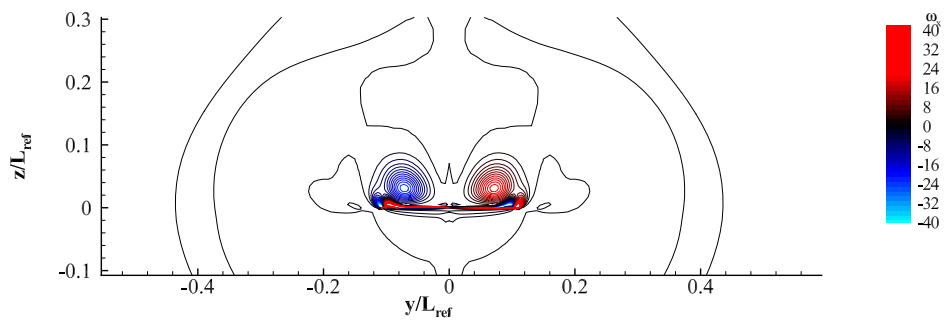
Figure 5.30: Stream ribbons on delta wing

The vorticity contours for the delta wing at various x locations are given in Figure

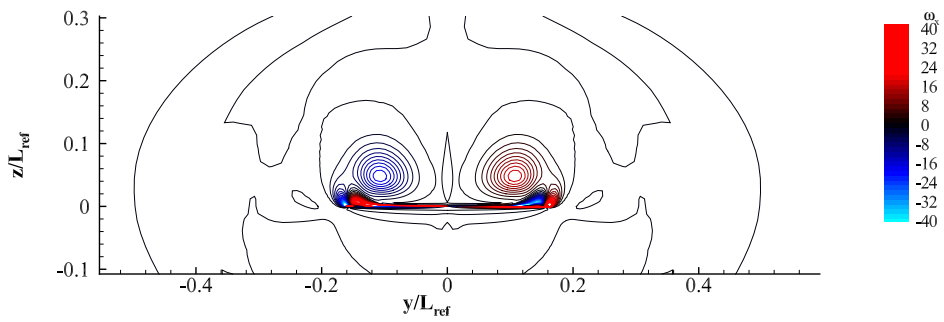
5.31.



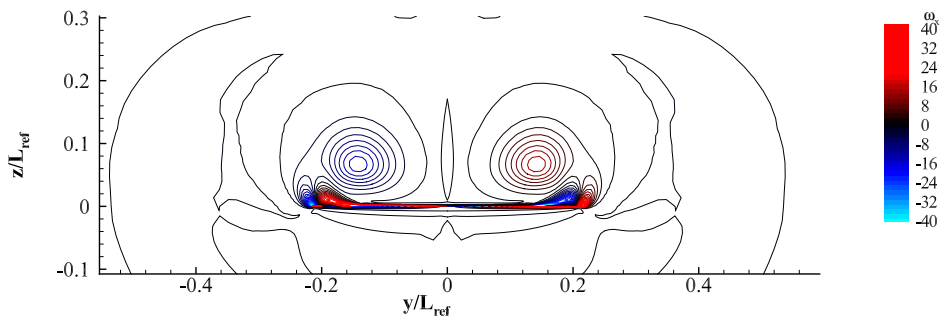
(a) ω_x @ $x=0.2$ section on solution domain



(b) ω_x @ $x=0.4$ section on solution domain



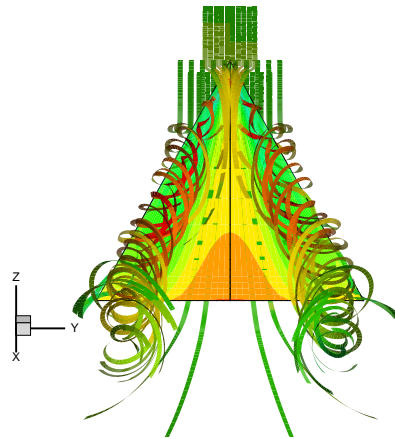
(c) ω_x @ $x=0.6$ section on solution domain



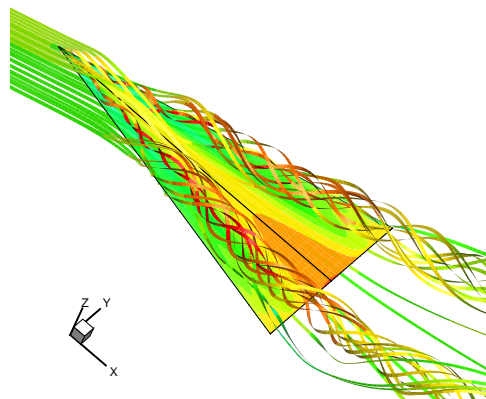
(d) ω_x @ $x=0.8$ section on solution domain

Figure 5.31: x component of Vorticity contours on delta wing

For the comparison purposes, 3D stream ribbons for IFVLBM and CFL3D are presented in the same figure. In Figure 5.32, the left of part of the Delta Wing solution is from CFL3D data and right part of the solution is from IFVLBM simulation. Stream ribbons are colored with respect to velocity magnitude.



(a) Stream ribbons back view



(b) Stream ribbons general view

Figure 5.32: Stream ribbons on delta wing

5.2 Model Assessment

A good CFD code must be accurate, that is the solutions obtained from the simulations must be close to the real phenomena. The accuracy of the code depends on the discretization schemes used for time integration and spatial derivatives. It is sometimes possible to perform analytical analysis, but there are also numerical approaches to assess the spatial and temporal accuracies of the CFD methods. In this section, numerical assessment of spatial accuracy and temporal accuracy will be performed. Then the convergence of the code will be compared with an explicit LBM code.

5.2.1 Spatial Accuracy

The spatial accuracy of the IFVLBM is checked with the following numerical approach. A known steady state solution is taken as the reference. Then, the flow field solutions from different size of grid resolutions are compared with the reference solution. As the grid becomes finer, the RMS error ϵ must be getting smaller. For this purpose, The solution of lid driven cavity for $Re = 400$ from Ghia et al. [70] is used as the reference. The grid is uniform so any unknown effect from the grid shape will be eliminated. The nondimensional u velocity values from the vertical section passing through the geometric center are used to calculate the RMS error. Calculation of the RMS error ϵ is given in equation 5.5. Where u is the horizontal velocity component and N is the number of spatial nodes. The steady state solution for 33×33 , 65×65 and 129×129 are calculated by IFVLBM and the residuals given in equations 5.1 and 5.2 for density and velocity are shown in Figure 5.34. Since 2^{nd} order MUSCL scheme is used for flux calculations, IFVLBM is expected to be 2^{nd} order accurate.

$$\epsilon = \sqrt{\frac{\sum (u - u_{Ghia})^2}{N^2}} \quad (5.5)$$

After the solutions are obtained, the nondimensional u velocities at the y coordinates given in Ghia's paper are extracted. The errors ϵ are calculated for each grid resolution and plotted against a hypothetical 2^{nd} order accurate method. By inspecting equation 5.5, it can be seen that doubling the spatial resolution, the logarithm of ϵ

will be halved. Then the results is plotted against the hypothetical 2^{nd} order method. As shown in Figure 5.33, logarithmic scale is used for ease of understanding. As expected IFVLBM method shows better accuracy characteristics than a 2^{nd} order accurate hypothetical method.

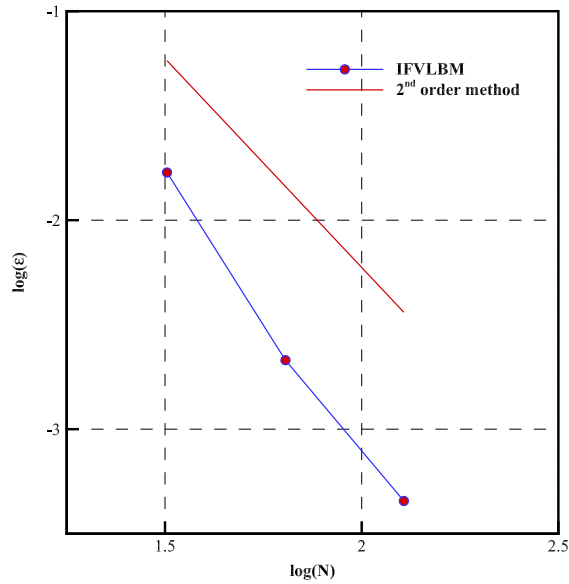


Figure 5.33: Numerical spatial accuracy of IFVLBM

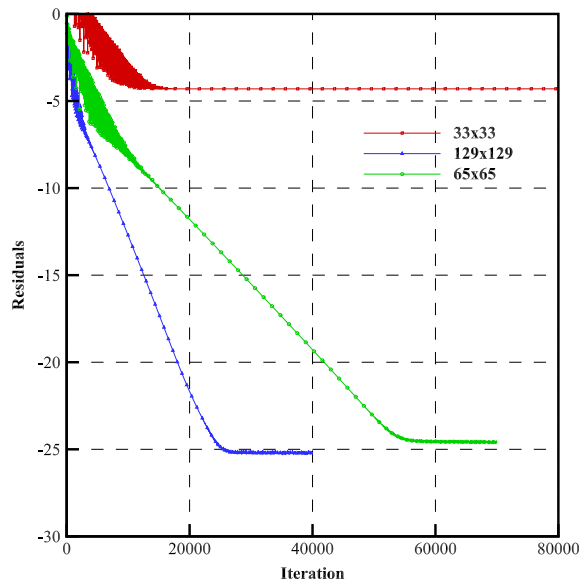


Figure 5.34: Density residuals for $Re = 400$ (cavity flow)

5.2.2 Temporal Accuracy

The approach to determine the temporal accuracy of IFVLBM is similar to the spatial accuracy case. However, in this case the time increment and the spatial resolution is changed together as proposed in reference [94, p. 80]. The unsteady flow over the cylinder with $Re = 100$ is selected as the benchmark case. The grid resolutions and Δt taken for the simulations are given in Table 5.7. In Table 5.7, the grid spacing is normalized with respect to fine grid and number of cells is normalized with respect to coarse grid. The simulations are performed with the same number of subiterations.

Table 5.7: Temporal accuracy simulation parameters

Grid Type	Grid Spacing	Grid Res.	Δt	# Cells
Fine	1	401×341	0.1	16
Medium	2	201×171	0.2	4
Coarse	4	101×86	0.4	1

The Strouhal number, related to the frequency of the eddies, is used as the test parameter. A similar approach used in grid convergence study is applied to define the theoretical asymptotic value. Richardson extrapolation given in equation 4.11 is used. The order of convergence p is calculated by the help of equation 4.12.

Table 5.8: Temporal accuracy simulation results

Grid Type	St	$\% \epsilon$
Theoretical Value	0.1650	
Fine	0.1645	0.2847
Medium	0.1625	1.4970
Coarse	0.1473	10.7108

The calculated theoretical value and the errors associated with the simulations are given in Table 5.8. The errors are about 0.3% percent for the fine grid and 1.5% for medium grid. The plot of the errors of IFVLBM is given in Figure 5.35. In the same Figure the errors of theoretical 2^{nd} and 3^{rd} order methods are demonstrated. The axes of the plot are in logarithmic scale to a clearer understanding.

The demonstrated temporal accuracy for IFVLBM lies between the 2^{nd} and 3^{rd} order accuracy. Since the time integration is second order accurate, IFVLBM method is 2^{nd} order accurate for unsteady problems.

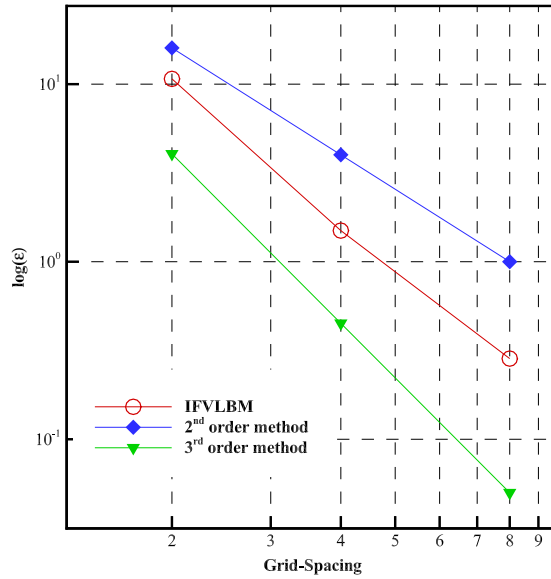


Figure 5.35: Temporal error of IFVLBM

5.2.3 Convergence Acceleration

Zarghami et al.[95] and Guzel et al. [44] showed that using Runge Kutta and IMEX Methods, the solution process becomes more stable and the process can be accelerated. To assess the convergence acceleration of the proposed method, the following approach is used. The solution of cavity flow for $Re = 1000$ by IMEX-LBM [44] with a $CFL = 1$ on a 129×129 nodes grid is used as reference. Then the same simulation is performed by seeking the maximum attainable CFL number for IMEX-LBM and IFVLBM.

$CFL = 2.5$ for IMEX-LBM and $CFL = 10$ for IFVLBM is reached without compromising the stability of the solution. The density and velocity residuals are calculated at every step and the simulation is stopped when the logarithm of largest of density residual and velocity residual is less than -10. All runs are performed on the same notebook computer, with the same compiler settings. Single iteration for IMEX-LBM takes ~0.015 seconds and single iteration for IFVLBM takes 0.115 seconds CPU time¹.

In Figure 5.36, it can be seen that, the proposed IFVLBM has the steepest density residual curve with a stable solution with $CFL = 10$. The IFVLBM is about ~4-5

¹ It should be noted that the codes are not optimized for speed during development phase

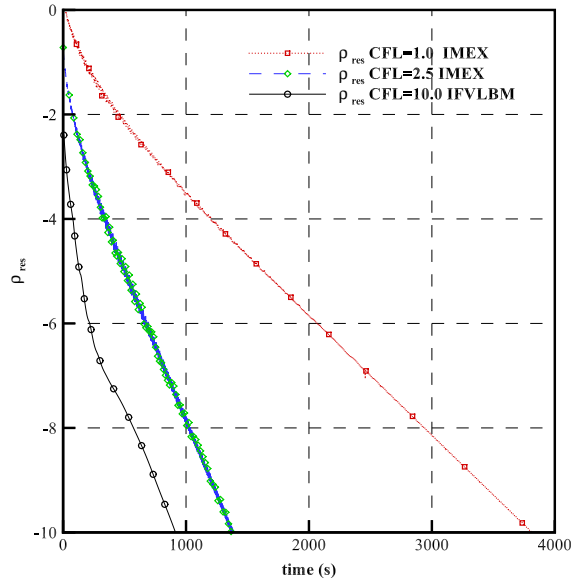


Figure 5.36: Convergence Acceleration of IFVLBM

times faster than the reference solution and about ~ 1.5 times faster than the IMEX solution with $CFL = 2.5$.

5.2.4 Computational Performance

The computational performance is an important aspect for the CFD codes. The solution time and how fast is the convergence rate are important parameters. Although, these parameters are strongly dependent on the computational hardware and optimization of the coding, some key data are presented to give information about the computational performance of IFVLBM code.

A workstation class notebook computer is used for obtaining the solutions for the test cases. The notebook has following configuration:

- Intel(R) Core(TM) i7-3720QM CPU @2.60 GHz.
- 16 GB RAM
- Windows 7 64 Bit Operating System
- NVIDIA Quadro K1000M GPU

In Table 5.9, The solution domain of the validation cases are described. The first column is the name of the case, the second column is the Reynolds number. If there is a “T” after the Reynolds number, it means that, the turbulence routine is incorporated with the calculation of the flow field. The columns under the “Solution Domain Elements” heading are the number of blocks and number of area/volume elements in each computational direction, respectively.

Table 5.9: Solution domains for the validation cases

Case	Re	Solution Domain Elements			
		# Blocks	I	J	K
Cavity	400	1	128	128	1
	1000	1	128	128	1
	3200	1	128	128	1
	5000	1	256	256	1
Cylinder	10	1	200	170	1
	20	1	200	170	1
	40	1	200	170	1
Flat Plate	10000	1	272	192	1
	100000	1	272	192	1
	5000000 T	1	272	192	1
NACA0015	1955000 T	1	376	170	1
Confined Cylinder	40	5	60	120	1
			60	30	1
			240	40	1
			60	30	1
			30	120	1
3D NACA0015	1955000 T	1	376	170	4
3D Delta Wing	500000	1	36	64	64
3D Confined Cylinder	40	5	60	120	40
			60	30	40
			240	40	40
			60	30	40
			30	120	40
3D Sphere	100	1	104	80	100

In Table 5.10, the computational performance results are presented. The third column represents the time for each iteration for the test cases. The fourth column is the number of iterations for the density residual which was set to 1×10^{-10} . The fifth column is the time in seconds to achieve the specified density residual level. The last

column is the time per iteration per volume element, which can give idea about the run time when the solution domain is constructed.

During the thesis study the correctness of the code is favored. Thus, the given figures for the performance can be enhanced with optimization in coding. Moreover, it is possible to use compilers speed optimization settings to have a better performance.

Table 5.10: Computational performance data for validation cases

Case	Re	t1(s)	$n\rho_{res} = 10^{-10}$	t2(s)	t3(s)
Cavity	400	0.14789	6800	1005.6	9.026E-06
	1000	0.14539	8430	1225.7	8.874E-06
	3200	0.14524	39370	5718.0	8.865E-06
	5000	0.49124	82270	40414.7	7.496E-06
Cylinder	10	0.27971	2240	626.5	8.227E-06
	20	0.27721	1980	548.9	8.153E-06
	40	0.28033	1620	454.1	8.245E-06
Flat Plate	10000	0.40934	8180	3348.4	7.838E-06
	100000	0.40950	7080	2899.3	7.841E-06
	5000000 T	0.41402	11500	4761.3	7.928E-06
N0015	1955000 T	0.50762	4610	2340.2	7.942E-06
Confined Cylinder	40	0.17097	1320	225.7	7.124E-06
3D N0015	1955000 T	4.35738	880	3834.5	1.704E-05
3D Delta Wing	500000	2.32846	950	2212.0	1.579E-05
3D Confined Cylinder	40	15.26828	1710	26108.8	1.590E-05
3D Sphere	100	13.24544	1660	21987.4	1.592E-5

CHAPTER 6

RESULTS & DISCUSSION

6.1 Summary

For the last two decades Lattice Boltzmann Methods are developed as an alternative application to numerical study of flows. It can be simply thought as a bridge to fill the gap between the microscopic world to macroscopic world. The first developed algorithms to solve the Lattice Boltzmann Equations are derived from the Cellular Gas Automata and called S&C stream and collide methods. The method has some restrictions, like uniform grid throughout the solution domain, and time step is strictly connected to $CFL = 1$ condition. Different techniques are studied to increase the convergence rate of S&C methods like multi-block techniques and nested grids.

Studies to solve LBE with different methods started almost at the same time period with the evolution of S&C method. Since the LBE discretized in velocity space is a partial differential equation, researches thought that it can be solved by different methods those are applicable to NS equations like finite difference, finite element finite volume etc. In the literature, it is possible to find a lot of scientific papers on the LBE solution methods. However, all these papers (up to the authors my knowledge) exerts explicit methods to solve the LBE to exploit some properties like parallel processing, easy coding etc.

Looking at the Lattice Boltzmann Equation, it is easy to conclude that, the equation can be solved independently for each cell and for each lattice direction in that cell. That is a huge advantage of S&C method and its derivatives as far as there are sufficient hardware resources. It is also true for other methods like FDM, FEM, FVM

applied on the LBE. But it is also known that implicit integration techniques will bring more stability to the solution scheme selected in price of more numerical effort.

In this manuscript, discrete LBE equation is solved by the Finite Volume Method with an implicit integration technique. Instead of solving for each unknown density distribution independently, the unknown distribution functions in each cell are collected in a vector form. Then all the unknowns for solution domains can be solved simultaneously. However, the obtained set of equations the coefficient matrix is a sparse matrix due to spatial discretization and difficult to solve and time consuming. The approximate factorization methods are the remedy to the standard banded sparse systems. The ADI method is utilized for the present method, in which the solution is performed in three steps. The coefficient matrices of each step becomes either a tridiagonal matrix or a tridiagonal block matrix. Furthermore, in each step the solution is independent for constant I, J, K lines or IJ, IK, JK surfaces and some operations can be performed in parallel.

The limitation of the stability of LBM is a known fact for high Reynolds number flows. The Spalarat Allmaras turbulence model based on Boussinesq eddy viscosity hypothesis is incorporated to increase the stability of the developed model. The LBM is considered pseudo compressible simulation model because the calculations are carried out the time rate of change of density. However, the developed model is still limited to the incompressible flows.

The derived method is coded in FORTRAN language for 2D and 3D separately. The validation of the codes are performed by the use of numerical and experimental data available in the literature. The model assessment is also performed by the use of numerical approaches given in the literature.

6.2 Discussion of the Results

The proposed model is coded for 2D and 3D and the validation is performed on different CFD problems used as benchmarking and validation test cases.

The first test case for 2D implementation is the lid driven cavity test case. The test

case is one of the most easy case for a simulation setup. The grid is uniform three faces are stationary walls and generally the top face is selected as the moving lid. The flow inside the cavity is laminar flow up to some Reynolds number. The problem is solved for four different Reynolds numbers. The results of the IFVLBM is discussed with the solution of Ghia et.al.. The streamlines are plotted and compared with the original streamline patterns of the referred material of Ghia's. Also the velocity components in horizontal and vertical geometric center of the cavity is compared to the Result of Ghia's and shown they are in perfect match.

The second test case is the flat plate, for laminar flow. The problem is interesting due to boundary layer phenomena and the analytical solution exists. The grid chosen to calculate the boundary layer properties has a very small first layer height. The aspect ratio of the grids are high compared to the uniform grid. The grid used for this simulation is obtained from the turbulent test case benchmarks used by NASA. The results are obtained for two different Reynolds number where the transition to turbulence does not occur on the flat plate. The friction coefficient along the flat plate and the velocity distributions in the mid section are compared to the analytical solutions. The solutions obtained from the IFVLBM are in good alignment with the analytical results.

The third case for the 2D laminar code is the flow over a circular cylinder. The purpose of this simulation case is to test the body fitted coordinates, which is an advantage over S&Ms, and the cells that are not rectangle and aligned with the lattice directions. The grid is selected after a short grid convergence study for three levels of grids named as fine normal and coarse grid. The simulation is performed for three Reynolds numbers. The characteristics of the flow around a cylinder for laminar case are the eddy formation after the cylinder, the detachment point of the flow (separating streamline) and the drag coefficient. The other coefficients are zero as the case is symmetrical with respect to an horizontal axis passing through the centerline of the cylinder. The results are compared to selected results experimental and numerical those appear in the literature. All the calculated values by IFVLBM is in very good agreement with the results given in references.

The fourth case is the turbulent flow over the flat plate. The results are compared

to the results of the finite volume NS solver of NASA. The grid used for flat plate laminar case is also used for the turbulent simulation case. Simulation is performed for one Reynolds number. The friction coefficient along the plate, and velocity profiles in the boundary layer for two different locations are plotted to compare the reference results. Also the ratio of the turbulence viscosity to laminar viscosity is plotted at mid section of the plate. The same ratio is also given as a contour plot to compare with the reference contours. The results of the IFVLBM are in good agreement with the reference benchmark solutions.

The fifth case is the turbulent flow over an airfoil designated as NACA0015. The airfoil is a symmetric NACA four digit series airfoil. It is commonly used for helicopter blades and extensive experimental data is available in the literature. The flow is tripped to start transition to turbulence at a point very close to the LE (0.5% of chord length) in the experiments. The simulations are performed for a single Reynolds number to validate turbulent flow on body fitted grids. The static flow coefficients are calculated by IFVLBM with angle of attack up to 18 degrees. The coefficients are plotted in the same graph to compare results of experimental study. The simulations for the same case are also done by using the FLUENT software, a commercial finite volume solver for NS equations. The flow separates from the upper surface of the airfoil about 10 degrees angle of attack. The evolution of the streamlines for the separation region are plotted for different angles of attack. The results are in good agreement with the experimental data and the FLUENT software.

The sixth and the last simulation case is the unsteady laminar flow around the circular cylinder. Same grid for the laminar steady simulations are used for the unsteady case. After some Reynolds number, the eddies after the cylinder are detached and they travel to the downstream. The detached eddies cause an instability of the flow. After some time the upper and lower eddies are detached periodically and move along with the flow. The situation known as the Karman vortex street and the main characteristic of unsteady flow around the cylinder. For the comparison of data to the data obtained from literature, the Strouhal number, which can be thought as the indicator of the frequency of vortices, the average C_d and RMS of C_l values were calculated by IFVLBM. Moreover, the vorticity contours at specific times are plotted to demonstrate the Karman vortex street phenomena. The numeric data obtained by IFVLBM

is in good agreement with the data obtained from the literature.

The next four examples are the validation cases for the 3D code. The two examples are just the comparison of 2D and 3D data to check the consistency of the model during transition of the code from 2D to 3D for laminar and turbulent cases respectively. The last two examples are the flow past a sphere and the 3D delta wing test case.

NACA0015 airfoil section is used for 2D and 3D simulation. For 3D simulation, the grid is extended in the third dimension. The results of the mid section from 3D simulation are used for comparison purposes. The flow is fully turbulent for both cases. The problems are iterated the same amount. The convergence of the 3D simulation was faster. The comparison is performed with the lift and pitching moment coefficients. The pressure coefficient contours and velocity contours are plotted for 2D and 3D simulations. The pressure coefficient is also plotted along the airfoil surface for both simulations. The results are satisfactory for 2D to 3D transition of the code for turbulent flows.

To investigate the 3D code more, the comparison for a cylinder confined in a duct for 2D and 3D is performed. The Reynolds number is chosen such that the flow is laminar. The solution domain is also divided into 5 blocks to investigate multi-block capability of the written code. The 3D grid is derived by replicating the grid in the third direction. The mid section of the 3D simulation is used to compare the 2D simulation. The eddy length, and separation angle are the numerical values used for comparison. Also some plots are given for comparison like the velocity contour and pressure coefficient contours. The length of the eddies and the separation angles from the 2D and 3D solutions are in harmony.

The 3D Flow past a sphere is studied for the real 3D validation case. The flow is 3D in nature and up to $Re = 200$ it is axisymmetric and steady. The sphere is a hard benchmark case for CFD simulations. The toroidal streamlines are presented behind the sphere to show the axisymmetric nature of the flow. Furthermore, the density contours and velocity contours on some planes are given. Moreover, the drag coefficient, C_p distribution on the sphere, the length of the eddy and separation angle are compared with data available in the literature. The IFVLBM simulations results are in good agreement.

The real 3D simulation is performed on 75 degrees sweep angle delta wing. There are experimental studies performed by NASA. The results obtained from the experiments are compared to the finite volume solver of NASA. The results obtained by NS solver are shared as a benchmark test case for CFD developers. The solution domain is prepared by half delta wing. The simulation is performed for one Reynolds number for the grid obtained from the benchmark case. The results are demonstrated as pressure contours around some sections perpendicular to the longitudinal axis of the delta wing. Also the pressure coefficient for both NS solver and IFVLBM is plotted at some sections. The overall static coefficients on the delta wing is also calculated from the simulation solution of IFVLBM. The results of the presented model are in good agreement with the NASA finite volume NS solver.

Any numerical solution to the partial differential equations must possess some properties. Some of the important properties of the numerical methods can be stated as the temporal accuracy and spatial accuracy. To assess the developed code for accuracy point of view some numerical studies are performed.

The spatial accuracy is studied on the cavity flow where the high fidelity solutions are available in the literature. Three levels of grid resolution with a factor of 2 in each dimension are used for the simulations. For each simulation, the errors are calculated and the rate of decrease for the errors is compared with error of hypothetical second order accurate method. The results showed better than second order accuracy. This result was expected as the spatial discretization is for the numerical flux term which is calculated by the second order accurate MUSCL scheme.

Temporal accuracy is studied on a time accurate solution for a cylinder in free flow. The Reynolds number chosen is high enough to have oscillation of eddies past the cylinder. Three simulations are performed. For each simulation the time step is halved whereas the spatial resolution in each direction is increased by a factor 2. Then the Strouhal numbers are used to estimate the rate of decrease of the errors. The results shows that the temporal accuracy is also better than second order accuracy. It was expected since, second order accurate backward differencing method is used for temporal discretization.

Finally the convergence rate is compared to an explicit higher order finite volume

method that utilizes Implicit Explicit Runge-Kutta time integration scheme. Although the name implies implicit, the method is actually an explicit method. The test simulation is selected as the cavity flow and the residuals are plotted against CPU time. Although the convergence rate per iteration for the IFVLBM method is much more better, single iteration takes more time compared to the IMEX-LBM. However, IFVLBM is a little bit faster than the IMEX with maximum attainable CFL and around five times faster when CFL is taken as 1. The simulations are performed on uniform grid, the advantage of the IFVLBM will be more clear, when nonuniform high aspect ratio grids are used where the explicit methods are more susceptible to stability issues.

6.3 Future Work

The method presented in this manuscript has potential to be extended for different types of fluid simulations. The extension can be done in many ways.

In the literature, it is stated that changing the relaxation time for lattice directions can increase the stability of the standard LBMs. The same improvement can be applied also for this method. There are methods called Multi Relaxation (MRT) and Dual Relaxation Time where the relaxation time for each lattice direction is recalculated and used in the collision term.

The present method is developed for the incompressible region. That is the NS equations can be recovered from the LBE with a small Mach number assumption. However, there are techniques, that can be applied to adopt the solution even for high Mach number flows in the compressible region. There are two methods used for this purpose; the first one uses a modified equilibrium distribution functions, basically and the second one uses an introduced energy particle to calculate energy and energy fluxes. It can be stated that the amount of work is much more than the amount of work for done for incompressible flows.

In IFVLBM, only Spalart Allmaras turbulence model is incorporated. For a future work, the affects of other turbulence models like $k - e$, $k - \omega$ or algebraic turbulence models can be adopted to solution method. Large Eddy Simulation (LBM-LES) in-

corporated with Lattice Boltzmann Methods is also a hot topic and there are applications for the standard LBMs. The same method can also be adopted to IFVLBM.

The mesh uses structured meshes. However, unstructured meshes are more flexible to define the solution domain on complex geometries. This solution method can be adopted to use unstructured meshes. A significant amount of work is required for the solution of the sparse matrix, since the matrix won't have regular bands.

6.4 Conclusion

Implicit solution techniques for finite volume methods for the differential equations are known in the literature. But it was not applied to discrete LBE as far as the authors' knowledge. In this thesis, an implicit solution scheme to discrete LBE is proposed. The method dictates that, instead of solving the LBE for each lattice direction independently, they are solved simultaneously as a vector combining EDFs in all lattice directions. The computational time can be optimized by using the parallel solution techniques for sweep directions.

The implicit implementation utilizes flexibility to use nonuniform grid, where the mesh density can be decreased for low gradients of the flow, which is an advantage over standard LBMs. The CFL number is also eased where much larger integration steps can be used to advance for the steady state solution. The IFVLBM method can be used in the incompressible region for any flow. Since it solves the LBE, the code can be used to solve micro channel flows where the Knudsen number is large enough so that, the continuum approach is not valid.

As a result, an implicit finite volume solution method is developed and validation cases are demonstrated in this manuscript. It is believed that the outcome of this thesis will be a good reference and used by other researchers studying on the Lattice Boltzmann Methods.

REFERENCES

- [1] Robert W. Fox, Philip J. Pritchard, and Alan T. McDonald. *Introduction to Fluid Mechanics*. John Wiley & Sons, Nov 2010.
- [2] Ya-Ling He and Wen-Quan Tao. Multiscale simulations of heat transfer and fluid flow problems. *Journal of heat transfer*, 134(3):031018, 2012.
- [3] GA Bird. Monte carlo simulation of gas flows. *Annual Review of Fluid Mechanics*, 10(1):11–31, 1978.
- [4] Dieter A. Wolf-Gladrow. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction*. Springer, Feb 2000.
- [5] Uriel Frisch, Dominique D’Humières, Brosl Hasslacher, Pierre Lallemand, Yves Pomeau, and Jean-Pierre Rivet. Lattice gas hydrodynamics in two and three dimensions. 1(4):649–707, 1987.
- [6] J. Hardy. Time evolution of a two-dimensional model system. i. invariant states and time correlation functions. *Journal of Mathematical Physics*, 14(12):1746, 1973.
- [7] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical Review Letters*, 56(14):1505–1508, apr 1986.
- [8] D D’Humières, P Lallemand, and U Frisch. Lattice gas models for 3d hydrodynamics. *Europhysics Letters (EPL)*, 2(4):291–297, 1986.
- [9] Ulf Daniel Schiller. *Thermal fluctuations and boundary conditions in the lattice Boltzmann method*. PhD thesis, 2008.
- [10] Wing Kam Liu, Harold S. Park, Dong Qian, Eduard G. Karpov, Hiroshi Kadowaki, and Gregory J. Wagner. Bridging scale methods for nanomechanics and materials. *Computer Methods in Applied Mechanics and Engineering*, 195(13–16):1407 – 1421, 2006. A Tribute to Thomas J.R. Hughes on the Occasion of his 60th Birthday.
- [11] John Edward Jones. On the determination of molecular fields. ii. from the equation of state of a gas. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 106, pages 463–477. The Royal Society, 1924.

- [12] A. A. Mohamad. *Lattice Boltzmann Method: Fundamentals and Engineering Applications With Computer Codes*. Springer, Apr 2011.
- [13] P. L. Bhatnagar, E. P. Gross, and M Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3):511–525, 1954.
- [14] Michael C. Sukop and Daniel T. Thorne. *Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers*. Springer, Dec 2006.
- [15] Guy R. McNamara and Gianluigi Zanetti. Use of the boltzmann equation to simulate lattice gas automata. *Physical review letters*, 61(20):2332–2335, 1988.
- [16] Zhaoli Guo and Chang Shu. *Lattice Boltzmann Method and Its Applications in Engineering*. World Scientific, Mar 2013.
- [17] O. Filippova and D. Hänel. Boundary-fitting and local grid refinement for lattice-bgk models. *International Journal of Modern Physics C*, 09(08):1271–1279, 1998.
- [18] Alexandre Dupuis and Bastien Chopard. Theory and applications of an alternative lattice boltzmann grid refinement algorithm. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 67(6 Pt 2):066707, 2003.
- [19] Dazhi Yu, Renwei Mei, and Wei Shyy. A multi-block lattice boltzmann method for viscous fluid flows. *International Journal for Numerical Methods in Fluids*, 39(October 2001):99–120, 2002.
- [20] Zhaoli Guo and T S Zhao. Explicit finite-difference lattice boltzmann method for curvilinear coordinates. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 67(6 Pt 2):066709, 2003.
- [21] Bernd Crouse, Ernst Rank, Manfred Krafczyk, and Jonas Tölke. a lb-based approach for adaptive flow simulations. *International Journal of Modern Physics B*, 17(01n02):109–112, 2003.
- [22] Jonas Tölke, Sören Freudiger, and Manfred Krafczyk. An adaptive scheme using hierarchical grids for lattice boltzmann multi-phase flow simulations. *Computers & Fluids*, 35(8-9):820–830, 2006.
- [23] H Ding and C Shu. A stencil adaptive algorithm for finite difference solution of incompressible viscous flows. *Journal of Computational Physics*, 214(1):397–420, 2006.
- [24] J. Wu and C. Shu. A solution-adaptive lattice boltzmann method for two-dimensional incompressible viscous flows. *Journal of Computational Physics*, 230(6):2246–2269, mar 2011.

- [25] Zhao Yu and Liang-shih Fan. An interaction potential based lattice boltzmann method with adaptive mesh refinement (amr) for two-phase flow simulation. *Journal of Computational Physics*, 228(17):6456–6478, 2009.
- [26] Xiaoyi He, Li-Shi Luo, and Micah Dembo. Some progress in lattice boltzmann method. part i. nonuniform mesh grids. *Journal of Computational Physics*, 129(2):357–363, 1996.
- [27] Xiaoyi He, Li-Shi Luo, and Micah Dembo. Some progress in the lattice boltzmann method: Reynolds number enhancement in simulations. *Physica A*, 239:276–285, 1997.
- [28] Nianzheng Cao, Shiyi Chen, Shi Jin, and Daniel Martínez. Physical symmetry and lattice symmetry in the lattice boltzmann method. *Physical Review E*, 55(1):R21–R24, 1997.
- [29] Renwei Mei and Wei Shyy. On the finite difference-based lattice boltzmann method in curvilinear coordinates. *Journal of Computational Physics*, 143(2):426–448, 1998.
- [30] Taehun Lee and Ching-Long Lin. A characteristic galerkin method for discrete boltzmann equation. *Journal of Computational Physics*, 171(1):336–356, 2001.
- [31] Yusong Li, Eugene J. LeBoeuf, and P. K. Basu. Least-squares finite-element lattice boltzmann method. *Physical Review E*, 69(6):065701, 2004.
- [32] Misun Min and Taehun Lee. A spectral-element discontinuous galerkin lattice boltzmann method for nearly incompressible flows. *Journal of Computational Physics*, 230(1):245–259, 2011.
- [33] YT Chew, C Shu, and XD Niu. A new differential lattice boltzmann equation and its application to simulate incompressible flows on non-uniform grids. *Journal of statistical physics*, 107(April):329–342, 2002.
- [34] C. Shu, Y. Chew, and X. Niu. Least-squares-based lattice boltzmann method: A meshless approach for simulation of flows with complex geometry. *Physical Review E*, 64:1–4, 2001.
- [35] X D Niu, Y T Chew, and C Shu. Simulation of flows, around an impulsively started circular cylinder by taylor series expansion- and least squares-based lattice boltzmann method. *Journal of Computational Physics*, 188(1):176–193, 2003.
- [36] C Shu, X D Niu, and Y T Chew. Taylor-series expansion and least-squares-based lattice boltzmann method: Two-dimensional formulation and its applications. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 65(3 Pt 2B):036708, 2002.

- [37] Francesca Nannelli and Sauro Succi. The lattice boltzmann equation on irregular lattices. *Journal of Statistical Physics*, 68(3-4):401–407, 1992.
- [38] Gongwen Peng, Haowen Xi, Comer Duncan, and So-Hsiang Chou. Finite volume scheme for the lattice Boltzmann method on unstructured meshes. *Phys. Rev. E*, 59(4):4675–4682, 1999.
- [39] G Peng, H Xi, C Duncan, and S-H Chou. Lattice boltzmann method on irregular meshes. *Physical Review E*, 58(4):R4124—R4127, 1998.
- [40] H Xi, G Peng, and S.-H. Chou. Finite volume lattice boltzmann scheme in two and three dimensions. *Physical Review E*, 60(3):3380–3388, 1999.
- [41] Stefano Ubertini and Sauro Succi. Recent advances of lattice boltzmann techniques on unstructured grids. *Progress in Computational Fluid Dynamics, an International Journal*, 5(1):85–96, January 2005.
- [42] Maik Stiebler, Jonas Tölke, and Manfred Krafczyk. An upwind discretization scheme for the finite volume lattice boltzmann method. *Computers and Fluids*, 35(8-9):814–819, 2006.
- [43] Dhiraj V. Patil and K.N. Lakshmisha. Finite volume tvd formulation of lattice boltzmann simulation on unstructured mesh. *Journal of Computational Physics*, 228(14):5262–5279, 2009.
- [44] Goktan Guzel and Ilteris Koc. Simulation of turbulent flows using a finite-volume based lattice boltzmann flow solver. *Communications in Computational Physics*, 17:213–232, 1 2015.
- [45] Uri M Ascher, Steven J Ruuth, and Raymond J Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2-3):151–167, November 1997.
- [46] Y. Wang, Y. L. He, T. S. Zhao, G. H. Tang, and W. Q. Tao. Implicit-explicit finite-difference lattice boltzmann method for compressible flows. *International Journal of Modern Physics C*, 18(12):1961–1983, 2007.
- [47] Sauro Succi. *The Lattice Boltzmann Equation: For Fluid Dynamics and Beyond*. Clarendon Press, Jun 2001.
- [48] Donald P Ziegler. Boundary conditions for lattice boltzmann simulations. *Journal of Statistical Physics*, 71:1171–1177, 1993.
- [49] Takaji Inamuro, Masato Yoshino, and Fumimaru Ogino. A non-slip boundary condition for lattice boltzmann simulations. 1995.
- [50] Qisu Zou and Xiaoyi He. On pressure and velocity boundary conditions for the lattice boltzmann bgk model. *Physics of Fluids*, 9(6):1591, 1997.

- [51] Jonas Lätt. *Hydrodynamic limit of lattice Boltzmann equations*. PhD thesis, University of Geneva, 2007.
- [52] P. A. Skordos. Initial and boundary conditions for the lattice boltzmann method. *Physical Review E*, 48(6):4823–4842, 1993.
- [53] Zhaoli Guo, Chuguang Zheng, and Baochang Shi. An extrapolation method for boundary conditions in lattice boltzmann method. *Physics of Fluids (1994-present)*, 14(6):2007–2010, 2002.
- [54] Renwei Mei, Li-Shi Luo, and Wei Shyy. An accurate curved boundary treatment in the lattice boltzmann method. *Journal of Computational Physics*, 155(2):307–330, 1999.
- [55] M’hamed Bouzidi, Dominique D’Humières, Pierre Lallemand, and Li-Shi Luo. Lattice boltzmann equation on a two-dimensional rectangular grid. *Journal of Computational Physics*, 172(2):704–717, 2001.
- [56] Dazhi Yu, Renwei Mei, Li-shi Luo, and Wei Shyy. Viscous flow computations with the method of lattice boltzmann equation. 39:329–367, 2003.
- [57] Joris CG Verschaeve and Bernhard Müller. A curved no-slip boundary condition for the lattice boltzmann method. *Journal of Computational Physics*, 229(19):6781–6803, 2010.
- [58] Th Von Karman. The fundamentals of the statistical theory of turbulence. *Journal of the Aeronautical Sciences*, 4(4):131–138, 1937.
- [59] Hendrik Tennekes and John Leask Lumley. *A First Course in Turbulence*. MIT Press, Jan 1972.
- [60] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, Sep 2005.
- [61] David C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Jan 2006.
- [62] R K Freitas, M Meinke, Lattice-boltzmann Method, and Grid Refinement. Turbulence simulation via the lattice-boltzmann method on hierarchically refined meshes. *European Conference on Computational Fluid Dynamics*, pages 1–12, 2006.
- [63] Farhad Ghaffari. Turbulent vortex-flow simulation over a 65 degrees sharp and blunt leading-edge delta wing at subsonic speeds. *Delta*, (July), 2005.
- [64] Teymour Javaherchi. Review of spalart-allmaras turbulence model and its modifications spalart-allmaras governing equation. pages 1–14, 2010.
- [65] S.R. Spalart , P.R., Allmaras. A one equation turbulence model for aerodynamic flows. *AIAA Journal*, (0439), 1992.

- [66] Ahad Zarghami, Stefano Ubertini, and Sauro Succi. Finite volume formulation of thermal lattice boltzmann method. *International Journal of Numerical Methods for Heat & Fluid Flow*, 24(2):270–289, 2014.
- [67] Zhaoli Guo and T. S. Zhao. Lattice boltzmann model for incompressible flows through porous media. *Physical Review E*, 66(3):1–9, 2002.
- [68] T Imamura, K Suzuki, T Nakamura, and M Yoshida. Acceleration of steady-state lattice boltzmann simulations on non-uniform mesh using local time step method. *Journal of Computational Physics*, 202(2):645–663, 2005.
- [69] a. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. 1596:1991, 1991.
- [70] U Ghia, K.N Ghia, and C.T Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, 48:387–411, 1982.
- [71] Joshua A. White, Ronaldo I. Borja, and Joanne T. Fredrich. Calculating the effective permeability of sandstone with multiscale lattice boltzmann/finite element simulations. *Acta Geotechnica*, 1(4):195–209, 2006.
- [72] 2d zero pressure gradient flat plate verification case. <http://turbmodels.larc.nasa.gov/flatplate.html>. Accessed: 2015.02.20.
- [73] Madeleine Coutanceau and Roger Bouard. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. part 2. unsteady flow. *Journal of Fluid Mechanics*, 79(02):257, 1977.
- [74] F. Nieuwstadt and H.B. Keller. Viscous flow past circular cylinders. *Computers & Fluids*, 1(1):59 – 71, 1973.
- [75] X He. Lattice boltzmann method on curvilinear coordinates system: Flow around a circular cylinder. *Journal of Computational Physics*, 134:306–315, 1997.
- [76] R A Piziali. R. a. piziali. *Nasa Technical Memorandum*, 1994.
- [77] Hermann T. Schlichting and Joseph Kestin. *Boundary-Layer Theory*. McGraw-Hill, Feb 1979.
- [78] Ravi Golani and A K Dhiman. Fluid flow and heat transfer across a circular cylinder in the unsteady flow regime. *International Journal of Engineering and Science*, 3(3):8–19, 2014.
- [79] B. N. Rajani, a. Kandasamy, and Sekhar Majumdar. Numerical simulation of laminar flow past a circular cylinder. *Applied Mathematical Modelling*, 33(3):1228–1247, 2009.

- [80] O. Posdziech and R. Grundmann. A systematic approach to the numerical calculation of fundamental quantities of the two-dimensional flow over a circular cylinder. *Journal of Fluids and Structures*, 23(3):479–499, 2007.
- [81] S Mittal. Excitation of shear layer instability in ow past a cylinder at low reynolds number. (August):1147–1167, 2005.
- [82] SO Kjelgaard, WL Sellers III, and RP Weston. Flowfield survey over a 75 deg swept delta wing at an angle of attack of 20.5 deg. 1986.
- [83] Scott O. Kjelgaard. Detailed flow-field measurements over a 75 deg swept delta wing. Technical report, 1990.
- [84] N. Rossi, Stefano Ubertini, C. Bella, and S. Succi. Unstructured lattice boltzmann method in three dimensions. *International Journal for Numerical Methods in Fluids*, 49(6):619–633, 2005.
- [85] Y Rimon and SI Cheng. Numerical solution of a uniform flow over a sphere at intermediate reynolds numbers. *Physics of Fluids (1958-1988)*, 12(5):949–959, 1969.
- [86] Rajat Mittal. A fourier-chebyshev spectral collocation method for simulating flow past spheres and spheroids. *International journal for numerical methods in fluids*, 30(7):921–937, 1999.
- [87] Ü Gülat and AR Aslan. Accurate 3d viscous incompressible flow calculations with the fem. *International Journal for Numerical Methods in Fluids*, 25(9):985–1001, 1997.
- [88] Sadatoshi Taneda. Experimental investigation of the wake behind a sphere at low reynolds numbers. *Journal of the Physical Society of Japan*, 11(10):1104–1108, 1956.
- [89] Isao Nakamura. Steady wake behind a sphere. *Physics of Fluids (1958-1988)*, 19(1):5–8, 1976.
- [90] Masahisa Tabata and Kazuhiro Itakura. A precise computation of drag coefficients of a sphere. *International Journal of Computational Fluid Dynamics*, 9(3-4):303–311, 1998.
- [91] Frederick W Roos and William W Willmarth. Some experimental results on sphere and disk drag. *AIAA Journal*, 9(2):285–291, 1971.
- [92] Roland Clift, John R Grace, and Martin E Weber. *Bubbles, drops, and particles*. Courier Corporation, 2005.
- [93] 3d delta wing cfl3d validation case. http://cfl3d.larc.nasa.gov/Cfl3dv6/Cfl3dv6_testcases.html. Accessed: 2015.02.20.

- [94] Patrick Knupp and Kambiz Salari. *Verification of computer codes in computational science and engineering*. CRC Press, 2002.
- [95] A Zarghami, M J Maghrebi, J Ghasemi, and S Ubertini. Lattice boltzmann finite volume formulation with improved stability. *Communications in Computational Physics*, 12(1):42–64, 2012.
- [96] Akira Satoh. *Introduction to Practice of Molecular Simulation: Molecular Dynamics, Monte Carlo, Brownian Dynamics, Lattice Boltzmann, Dissipative Particle Dynamics*. Elsevier, Dec 2010.
- [97] Sydney Chapman and Thomas George Cowling. *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press, 1970.
- [98] Moawwad El-Mikkawy. A generalized symbolic thomas algorithm. *Applied Mathematics*, 3:342–345, 2012.
- [99] Thomas H. Pulliam and David W. Zingg. *Fundamental Algorithms in Computational Fluid Dynamics*. Springer, Apr 2014.
- [100] P J Van Der Houwen and B P Sommeijer. Approximate factorization for time-dependent partial differential equations. 128:447–466, 2001.
- [101] Robert W. MacCormack. Iterative modified approximate factorization. *Computers & Fluids*, 30(7–8):917–925, 2001.
- [102] Robert W. MacCormack. Implicit methods for fluid dynamics. *Computers & Fluids*, 41(1):72–81, 2011.
- [103] P. E O Buelow, S. Venkateswaran, and C. L. Merkle. Stability and convergence analysis of implicit upwind schemes. *Computers and Fluids*, 30:961–988, 2001.
- [104] Antony Jameson and Timothy Baker. Solution of the euler equations for complex configurations. In *6th Computational Fluid Dynamics Conference Danvers*, page 1929, 1983.

APPENDIX A

INCOMPRESSIBLE NS AND LBE

A.1 From LBE to Incompressible NSE

It is possible to derive the Navier Stokes Equations, starting from the LBE. In this section of Appendix A, the derivation will be carried out as given in [96, p. 296] with the method described by Chapman and Cowling [97].

A.1.1 Chapman Enskog Expansion

To start the derivation the basic equations are listed as follows:

$$\rho(\vec{r}, t) = \sum_{\alpha} f_{\alpha}(\vec{r}, t) \quad (\text{A.1})$$

$$\rho(\vec{r}, t)\vec{u}(\vec{r}, t) = \sum_{\alpha} \vec{e}_{\alpha} f_{\alpha}(\vec{r}, t) \quad (\text{A.2})$$

$$\Pi_{ij} = \sum_{\alpha} \vec{e}_{\alpha i} \vec{e}_{\alpha j} f_{\alpha}(\vec{r}, t) \quad (\text{A.3})$$

$$f_{\alpha}(\vec{r} + \vec{e}\Delta t, t + \Delta t) = f_{\alpha}(\vec{r}, t) + \Omega_{\alpha}(\vec{r}, t) \quad (\text{A.4})$$

$$\Omega_{\alpha}(\vec{r}, t) = -\frac{1}{\tau} (f_{\alpha}(\vec{r}, t) - f_{\alpha}^{eq}(\vec{r}, t)) \quad (\text{A.5})$$

In the above equations, f is the distribution function, Ω is the collision operator, ρ is the density, \vec{u} is the velocity vector, Π is the momentum flux, f^{eq} is the equilib-

rium distribution function, \vec{e} is the lattice velocity and τ is the relaxation parameter. Equations A.1-A.5 are valid for D2Q9 stencil for 2D and D3Q19 stencil for 3D.

If a Taylor Series expansion is applied to left hand side of equation A.4, the following equation is obtained.

$$\begin{aligned} \Delta t \frac{\partial f_\alpha}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 f_\alpha}{\partial t^2} + \Delta t \left(\vec{e}_\alpha \cdot \vec{\nabla} \right) f_\alpha + \frac{(\Delta t)^2}{2} \left(\vec{e}_\alpha \cdot \vec{\nabla} \right) \left(\vec{e}_\alpha \cdot \vec{\nabla} \right) f_\alpha \\ + (\Delta t)^2 \left(\vec{e}_\alpha \cdot \vec{\nabla} \right) \frac{\partial f_\alpha}{\partial t} = -\frac{1}{\tau} (f_\alpha(\vec{r}, t) - f_\alpha^{eq}(\vec{r}, t)) \end{aligned} \quad (\text{A.6})$$

The particle distribution function is expanded as :

$$f_\alpha = f_\alpha^{eq} + \varepsilon f_\alpha^{(1)} + \varepsilon^2 f_\alpha^{(2)} + \dots \quad (\text{A.7})$$

where ε is small quantity.

Using equation A.7, equations A.1 and A.2 can be rewritten:

$$\rho = \sum_\alpha f_\alpha^{eq} \quad \rho \vec{u} = \sum_\alpha \vec{e}_\alpha f_\alpha^{eq} \quad (\text{A.8})$$

$$\sum_\alpha f_\alpha^{(n)} = 0 \quad \sum_\alpha \vec{e}_\alpha f_\alpha^{(n)} = 0 \quad \text{for } n=1,2,3 \quad (\text{A.9})$$

There are two characteristic times employed in characterizing fluid problems for Chapman-Enskog expansion. T1 and T2 relating to the fluid velocity, and the viscous dissipation respectively. Generally T2 is much greater than T1. Hence, if ε and Δt are chosen such that, $\Delta t/T_1 = O(\varepsilon)$ and $\Delta t/T_2 = O(\varepsilon^2)$, time derivations can be written with the summation of time derivations with the characteristics of T1 and T2.

$$\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2} \quad (\text{A.10})$$

Similarly the position derivative can be expressed as follows since the characteristic length is in the same order for the characteristic time for fluid velocity.

$$\frac{\partial}{\partial \vec{r}_i} = \varepsilon \frac{\partial}{\partial r_{1i}} \text{ where } (i=x,y,z) \quad (\text{A.11})$$

The collision term of the LBE has the following characteristics

$$\sum_{\alpha} \Omega_{\alpha} = 0, \sum_{\alpha} \vec{e}_{\alpha} \Omega_{\alpha} = 0 \quad (\text{A.12})$$

Using equations A.10 and A.11, equation A.6 can be rewritten as:

$$\begin{aligned} \Delta t \left[\varepsilon \frac{\partial f_{\alpha}}{\partial t_1} + \varepsilon^2 \frac{\partial f_{\alpha}}{\partial t_2} \right] + \frac{(\Delta t)^2}{2} \varepsilon^2 \frac{\partial^2 f_{\alpha}}{\partial t_1^2} + \Delta t \varepsilon \left(\vec{e}_{\alpha} \cdot \vec{\nabla}_1 \right) f_{\alpha} \\ + \frac{(\Delta t)^2}{2} \varepsilon^2 \left(\vec{e}_{\alpha} \cdot \vec{\nabla}_1 \right) \left(\vec{e}_{\alpha} \cdot \vec{\nabla}_1 \right) f_{\alpha} \\ + (\Delta t)^2 \varepsilon^2 \left(\vec{e}_{\alpha} \cdot \vec{\nabla}_1 \right) \frac{\partial f_{\alpha}}{\partial t} + O(\varepsilon^3) = \Omega_{\alpha} \quad (\text{A.13}) \end{aligned}$$

By multiplying both sides with \vec{e}_{α}

$$\begin{aligned} \Delta t \left[\varepsilon \vec{e}_{\alpha} \frac{\partial f_{\alpha}}{\partial t_1} + \varepsilon^2 \vec{e}_{\alpha} \frac{\partial f_{\alpha}}{\partial t_2} \right] + \frac{(\Delta t)^2}{2} \varepsilon^2 \vec{e}_{\alpha} \frac{\partial^2 f_{\alpha}}{\partial t_1^2} + \Delta t \varepsilon \vec{e}_{\alpha} \left(\vec{e}_{\alpha} \cdot \vec{\nabla}_1 \right) f_{\alpha} \\ + \frac{(\Delta t)^2}{2} \varepsilon^2 \vec{e}_{\alpha} \left(\vec{e}_{\alpha} \cdot \vec{\nabla}_1 \right) \left(\vec{e}_{\alpha} \cdot \vec{\nabla}_1 \right) f_{\alpha} \\ + (\Delta t)^2 \varepsilon^2 \vec{e}_{\alpha} \left(\vec{e}_{\alpha} \cdot \vec{\nabla}_1 \right) \frac{\partial f_{\alpha}}{\partial t} + O(\varepsilon^3) = \vec{e}_{\alpha} \Omega_{\alpha} \quad (\text{A.14}) \end{aligned}$$

By summing over α and using equation A.7, equations A.13 and A.14 can be rewritten as:

$$\sum_{\alpha} \left[\Delta t \frac{\partial f_{\alpha}^{eq}}{\partial t_1} + \Delta t \left(\vec{e}_{\alpha} \cdot \vec{\nabla}_1 \right) f_{\alpha}^{eq} \right] = 0 \quad (\text{A.15})$$

$$\sum_{\alpha} \left[\Delta t \frac{\partial}{\partial t_1} \left(\vec{e}_{\alpha i} f_{\alpha}^{eq} \right) + \Delta t \sum_j \vec{e}_{\alpha i} \vec{e}_{\alpha j} \frac{\partial}{\partial r_{1j}} f_{\alpha}^{eq} \right] = 0 \quad (\text{A.16})$$

Again using equations A.3 and A.8 equations A.15 and A.16 become:

$$\frac{\partial \rho}{\partial t_1} + \vec{\nabla}_1 \cdot (\rho \vec{u}) = 0 \quad (\text{A.17})$$

$$\frac{\partial}{\partial t_1} (\rho \vec{u}_i) + \sum_j e_{\alpha j} \frac{\partial}{\partial r_{1j}} (\Pi_{ij}^{(0)}) = 0 \quad (\text{A.18})$$

where $\Pi_{ij}^{(0)} = \sum_j e_{\alpha i} e_{\alpha j} f_{\alpha}^{eq}$.

The following equation is derived by substituting equation A.7 into equation A.13 and collecting the terms of ε^2 equal to 0.

$$\frac{\partial \rho}{\partial t_2} + \frac{\Delta t}{2} \frac{\partial^2 \rho}{\partial t_1^2} + \frac{\Delta t}{2} \sum_i \sum_j \frac{\partial}{\partial r_{1i}} \frac{\partial}{\partial r_{1j}} \Pi_{ij}^{(0)} + \Delta t \sum_i \frac{\partial}{\partial t_1} \frac{\partial}{\partial r_{1i}} (\rho \vec{u}_i) = 0 \quad (\text{A.19})$$

After applying a similar rule for derivation of equation A.16, equation A.20 is obtained.

$$\begin{aligned} \frac{\partial}{\partial t_2} (\rho \vec{u}_i) + \frac{\Delta t}{2} \frac{\partial^2}{\partial t_1^2} (\rho \vec{u}_i) + \sum_j \frac{\partial}{\partial r_{1j}} \Pi_{ij}^{(1)} \\ + \frac{\Delta t}{2} \sum_j \sum_k \frac{\partial}{\partial r_{1j}} \frac{\partial}{\partial r_{1k}} S_{ijk}^{(0)} + \Delta t \sum_j \frac{\partial}{\partial t_1} \frac{\partial}{\partial r_{1i}} \Pi_{ij}^{(0)} = 0 \end{aligned} \quad (\text{A.20})$$

where $\Pi_{ij}^{(1)} = \sum_j e_{\alpha i} e_{\alpha j} f_{\alpha}^{(1)}$ and $S_{ijk}^{(0)} = \sum_{\alpha} e_{\alpha i} e_{\alpha j} e_{\alpha k} f_{\alpha}^{eq}$.

Differentiating equation A.17 with respect to t_1 :

$$\frac{\partial^2 \rho}{\partial t_1^2} = \frac{\partial}{\partial t_1} \left[-\vec{\nabla}_1 \cdot (\rho \vec{u}) \right] = \frac{\partial}{\partial t_1} \left[\sum_i \frac{\partial}{\partial r_{1i}} (\rho \vec{u}_i) \right] \quad (\text{A.21})$$

Using equation A.21 and reordering equation A.19, the following equation is obtained:

$$\frac{\partial \rho}{\partial t_2} + \frac{\Delta t}{2} \sum_i \frac{\partial}{\partial r_{1i}} \left[\frac{\partial}{\partial t_1} (\rho \vec{u}_i) + \sum_j \frac{\partial}{\partial r_{1j}} \Pi_{ij}^{(0)} \right] = 0 \quad (\text{A.22})$$

With some algebra using equations A.18 and A.22, equation A.23 is obtained.

$$\frac{\partial \rho}{\partial t_2} = 0 \quad (\text{A.23})$$

Again differentiating equation A.18 with respect to t_1 yields:

$$\frac{\partial^2}{\partial t_1^2}(\rho \vec{u}_i) = \frac{\partial}{\partial t_1} \left[-\sum_j \frac{\partial}{\partial r_{1j}} \Pi_{ij}^{(0)} \right] = \left[\sum_i \frac{\partial}{\partial t_1} \frac{\partial}{\partial r_{1i}} \Pi_{ij}^{(0)} \right] \quad (\text{A.24})$$

Using this result and substituting it into equation A.20, the following relation is obtained.

$$\frac{\partial}{\partial t_2}(\rho \vec{u}_i) + \sum_j \frac{\partial}{\partial r_{1j}} \left[\Pi_{ij}^{(1)} + \frac{\Delta t}{2} \left[\frac{\partial}{\partial t_1} \Pi_{ij}^{(0)} + \sum_k \frac{\partial}{\partial r_{1k}} S_{ijk}^{(0)} \right] \right] = 0 \quad (\text{A.25})$$

Keeping in mind that the relation $\frac{\partial}{\partial t} = \varepsilon \frac{\partial}{\partial t_1} + \varepsilon^2 \frac{\partial}{\partial t_2}$ A.17 and A.23 can be added up after multiplying the equations with ε and ε^2 respectively. The obtained equation is the continuity equation.

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u}_i) = 0 \quad (\text{A.26})$$

A.1.2 Derivation of NS from LBE

A similar work can be done for the conservation of momentum equations.

$$\frac{\partial}{\partial t}(\rho \vec{u}_i) + \sum_j \frac{\partial}{\partial r_j} \Pi_{ij} + \sum_j \frac{\Delta t}{2} \frac{\partial}{\partial r_j} \left[\varepsilon \frac{\partial}{\partial t_1} \Pi_{ij}^{(0)} + \sum_k \frac{\partial}{\partial r_k} S_{ijk}^{(0)} \right] = 0 \quad (\text{A.27})$$

where $\Pi_{ij} \approx \Pi_{ij}^{(0)} + \varepsilon \Pi_{ij}^{(1)}$.

Using equations A.10 and A.11 and a variable transformation on equation A.6 the following equation is obtained.

$$\begin{aligned} \Delta t \left[\varepsilon \frac{\partial f_\alpha}{\partial t_1} + \varepsilon^2 \frac{\partial f_\alpha}{\partial t_2} \right] + \frac{(\Delta t)^2}{2} \varepsilon^2 \frac{\partial^2 f_\alpha}{\partial t_1^2} + \Delta t \varepsilon \left(\vec{e}_\alpha \cdot \vec{\nabla}_1 \right) f_\alpha \\ + \frac{(\Delta t)^2}{2} \varepsilon^2 \left(\vec{e}_\alpha \cdot \vec{\nabla}_1 \right) \left(\vec{e}_\alpha \cdot \vec{\nabla}_1 \right) f_\alpha + (\Delta t)^2 \varepsilon^2 \left(\vec{e}_\alpha \cdot \vec{\nabla}_1 \right) \frac{\partial f_\alpha}{\partial t_1} \\ = -\frac{1}{\tau} (f_\alpha - f_\alpha^{eq}) \quad (\text{A.28}) \end{aligned}$$

Substituting equation A.17 into the equation A.28 and setting the terms collected by ε zero the following equation is obtained.

$$-\frac{1}{\tau\Delta t}f_\alpha^{(1)} = \frac{\partial f_\alpha^{eq}}{\partial t_1} + \sum_i \frac{\partial}{\partial r_{1i}}(\vec{e}_{\alpha i}f_\alpha^{eq}) \quad (\text{A.29})$$

The following equation is obtained by substituting the macroscopic quantities into equations A.17 and A.18:

$$\begin{aligned} \frac{\partial f_\alpha^{eq}}{\partial t_1} &= \frac{\partial f_\alpha}{\partial \rho} \frac{\partial \rho}{\partial t_1} + \sum_i \frac{\partial f_\alpha^{eq}}{\partial(\rho\vec{u}_i)} \frac{\partial(\rho\vec{u}_i)}{\partial t_1} \\ &= \frac{\partial f_\alpha}{\partial \rho} \frac{\partial}{\partial \vec{r}_1}(\rho\vec{u}_i) - \sum_i \sum_j \frac{\partial f_\alpha^{eq}}{\partial(\rho\vec{u}_i)} \frac{\partial}{\partial r_{1j}} \Pi_{ij}^{(0)} \end{aligned} \quad (\text{A.30})$$

Substituting the macroscopic quantities into equation A.29 the following relation is obtained:

$$-\frac{1}{\tau\Delta t}f_\alpha^{(1)} = -\frac{\partial f_\alpha^{eq}}{\partial \rho} \frac{\partial f_\alpha}{\partial \vec{r}_1}(\rho\vec{u}_i) - \sum_i \sum_j \frac{\partial f_\alpha^{eq}}{\partial(\rho\vec{u}_i)} \frac{\partial}{\partial r_{1j}} \Pi_{ij}^{(0)} + \sum_i \frac{\partial}{\partial r_{1i}}(e_{\alpha i}f_\alpha^{eq}) \quad (\text{A.31})$$

Using equation A.31, the solution for $f_\alpha^{(1)}$ is needed to be obtained. Neglecting the higher order terms following equations are obtained:

$$f_\alpha^{(eq)} = f_\alpha^{(0)} = \rho w_\alpha \left[1 + 3 \frac{\vec{e}_\alpha \cdot \vec{u}}{e^2} \right] \quad (\text{A.32})$$

$$\Pi_{ij}^{(0)} = \frac{\rho}{3} e^2 \delta_{ij} + \rho u_i u_j \quad (\text{A.33})$$

$$\frac{\partial f_\alpha^{(0)}}{\partial \rho} = w_\alpha \quad (\text{A.34})$$

$$\frac{\partial f_\alpha^{(0)}}{\partial(\rho u_i)} = w_\alpha \frac{3}{e^2} e_{\alpha i} \quad (\text{A.35})$$

$$\frac{\partial}{\partial r_{1j}} \Pi_{ij}^{(0)} = \frac{e^2}{3} \frac{\partial \rho}{\partial r_{1j}} \delta_{ij} + \frac{\partial}{\partial r_{1j}} (\rho u_i u_j) \quad (\text{A.36})$$

$$\frac{\partial}{\partial r_{1i}} (e_{\alpha i} f_{\alpha}^{(0)}) = w_{\alpha} \frac{\partial}{\partial r_{1i}} (\rho e_{\alpha i}) + 3w_{\alpha} \frac{1}{e^2} \frac{\partial}{\partial r_{1j}} \left[\sum_j \rho e_{\alpha i} e_{\alpha j} u_j \right] \quad (\text{A.37})$$

Using the above relations in equation A.31, $f_{\alpha}^{(1)}$ is solved as

$$f_{\alpha}^{(1)} = -3w_{\alpha} \Delta t \tau \frac{1}{e^2} \sum_i \sum_j \left(e_{\alpha i} e_{\alpha j} - \frac{1}{3} e^2 \delta_{ij} \right) \frac{\partial}{\partial r_{1j}} (\rho u_i) \quad (\text{A.38})$$

Using the solution $f_{\alpha}^{(1)}$ and after some algebra, following relation is obtained:

$$\varepsilon \Pi_{ij}^{(1)} = -\frac{\Delta t \tau e^2}{3} \left[\frac{\partial}{\partial r_j} (\rho u_i) + \frac{\partial}{\partial r_i} (\rho u_j) \right] \quad (\text{A.39})$$

By using above equation following equation is obtained:

$$\sum_j \frac{\partial}{\partial r_j} (\varepsilon \Pi_{ij}^{(1)}) = -\frac{\Delta t \tau e^2}{3} \left[\frac{\partial^2}{\partial \vec{r}^2} (\rho u_i) + \frac{\partial}{\partial r_i} \left(\frac{\partial}{\partial \vec{r}} (\rho \vec{u}) \right) \right] \quad (\text{A.40})$$

$$\sum_j \frac{\partial}{\partial r_j} (\Pi_{ij}^{(0)}) = \sum_j \frac{\partial}{\partial r_j} [p \delta_{ij} + \rho u_i u_j] \quad (\text{A.41})$$

By using the relation $\Pi_{ij} \approx \Pi_{ij}^{(0)} + \varepsilon \Pi_{ij}^{(1)}$ and equations A.40, A.41 equation A.42 is obtained.

$$\begin{aligned} \sum_j \frac{\partial}{\partial r_{1j}} \Pi_{ij} = \\ \sum_j \frac{\partial}{\partial r_{1j}} \left[\frac{\rho}{3} e^2 \delta_{ij} + \rho u_i u_j \right] - \frac{\Delta t \tau e^2}{3} \left[\frac{\partial^2}{\partial \vec{r}^2} (\rho u_i) + \frac{\partial}{\partial r_i} \left(\frac{\partial}{\partial \vec{r}} (\rho \vec{u}) \right) \right] \end{aligned} \quad (\text{A.42})$$

Using equations A.11 and A.17 equation A.43 is obtained. Then after with some algebraic manipulation, equation A.43 yields to equation A.44

$$\frac{\Delta t}{2}\varepsilon\frac{\partial}{\partial t_1}\Pi_{ij}^{(0)} = -\frac{1}{6}\Delta te^2\varepsilon\delta_{ij}\frac{\partial\rho}{\partial t_1} = -\frac{1}{6}\Delta te^2\delta_{ij}\frac{\partial}{\partial\vec{r}}\cdot(\rho\vec{u}) \quad (\text{A.43})$$

$$\frac{\Delta t}{2}\varepsilon\sum_j\frac{\partial}{\partial r_{1j}}\left(\Pi_{ij}^{(0)}\right) = -\frac{1}{6}\Delta te^2\frac{\partial}{\partial r_i}\left(\frac{\partial}{\partial\vec{r}}(\rho\vec{u})\right) \quad (\text{A.44})$$

Then $S_{ijk}^{(0)}$ is evaluated and it can be put into the summation to evaluate the necessary terms.

$$S_{ijk}^{(0)} = \sum_{\alpha} e_{\alpha i}^{\rightarrow} e_{\alpha j}^{\rightarrow} e_{\alpha k}^{\rightarrow} f_{\alpha}^{(0)} = \sum_j e_{\alpha i}^{\rightarrow} e_{\alpha j}^{\rightarrow} e_{\alpha k}^{\rightarrow} \rho w_{\alpha} \left[1 + 3\frac{\vec{e}_{\alpha}\cdot\vec{u}}{e^2}\right] \quad (\text{A.45})$$

$$\sum_j\sum_k\frac{\partial}{\partial r_j}\cdot\frac{\partial}{\partial r_k}S_{ijk}^{(0)} = \frac{2}{3}e^2\left[\frac{\partial}{\partial r_i}\left(\frac{\partial}{\partial\vec{r}}(\rho\vec{u})\right) + \frac{1}{2}\frac{\partial^2}{\partial r^2}(\rho u_i)\right] \quad (\text{A.46})$$

Now it is easy to substitute the equations A.42, A.44, A.46 into equation A.27 and keeping in mind that $\frac{\partial}{\partial\vec{r}}\cdot(\rho\vec{u}) = 0$ for incompressible flows. The following NS equation is obtained

$$\rho\left\{\frac{\partial\vec{u}}{\partial t} + (\vec{u}\cdot\vec{\nabla})\vec{u}\right\} = -\vec{\nabla}p + \mu\vec{\nabla}^2\vec{u} \quad (\text{A.47})$$

where the viscosity μ is related to lattice Boltzmann parameters with the following equation:

$$\mu = \frac{\rho\Delta te^2}{3}\left(\tau - \frac{1}{2}\right) \quad (\text{A.48})$$

APPENDIX B

LATTICE BOLTZMANN STENCILS

In this Appendix, LBE models widely used for isothermal flows are presented. BGK models are the most popular ones. D2Q9 and D3Q19 models are utilized in this thesis. For all stencils, the equilibrium distribution function is given as:

$$f_{\alpha}^{eq} = \rho w_{\alpha} \left[1 + \frac{\vec{e}_{\alpha} \cdot \vec{u}}{c_s^2} + \frac{1}{2} \frac{(\vec{e}_{\alpha} \cdot \vec{u})^2}{c_s^4} - \frac{1}{2} \frac{(\vec{u} \cdot \vec{u})}{c_s^2} \right] \quad (\text{B.1})$$

where α is the lattice direction, ρ is the density, \vec{e}_{α} is the lattice velocity and \vec{u} is the physical macroscopic velocity. w_{α} can be considered as the weighting factor for each lattice direction. The lattice velocity vector is given as:

$$\vec{e}_{\alpha} = c \left(e_{\alpha x} \vec{i} + e_{\alpha y} \vec{j} + e_{\alpha z} \vec{k} \right) \quad (\text{B.2})$$

c is an arbitrary constant. Following sections will present the lattice velocity vectors, weighting factors for different stencils. Also the relation between c and c_s presented for each stencil.

B.1 D1Q3 Model

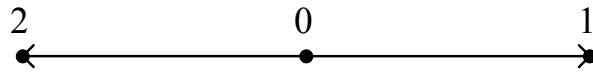


Figure B.1: 1 dimensional D1Q3 stencil

Table B.1: Parameters of D1Q3 model

alpha	lattice velocity $\frac{\vec{e}}{c}$	Weighting w_α	c/c_s
0	0	$\frac{2}{3}$	$\sqrt{3}$
1, 2	± 1	$\frac{1}{6}$	

B.2 D1Q5 Model

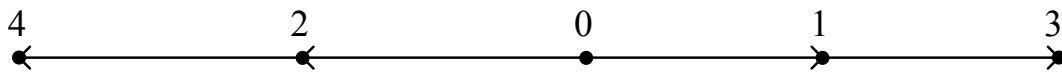


Figure B.2: 1 dimensional D1Q5 stencil

Table B.2: Parameters of D1Q5 model

alpha	lattice velocity $\frac{\vec{e}}{c}$	Weighting w_α	c/c_s
0	0	$\frac{1}{2}$	1
1, 2	± 1	$\frac{1}{6}$	
3, 4	± 2	$\frac{1}{12}$	

B.3 D2Q5 Model

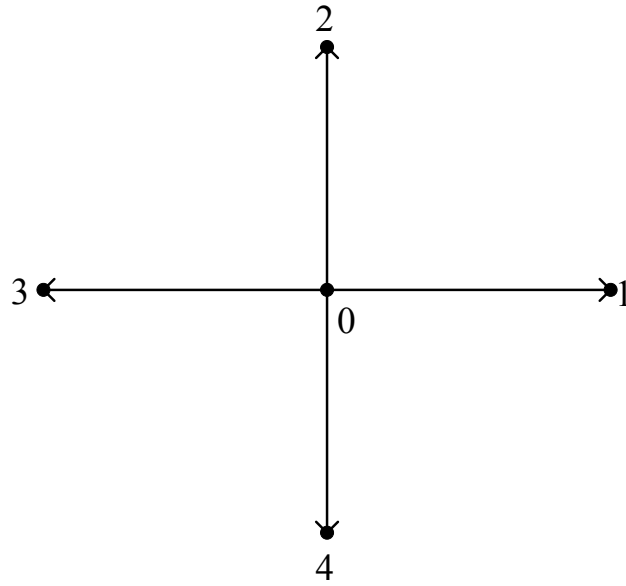


Figure B.3: 1 dimensional D2Q5 stencil

Table B.3: Parameters of D2Q5 model

alpha	lattice velocity $\frac{\vec{e}}{c}$	Weighting w_α	c/c_s
0	(0, 0)	$\frac{1}{3}$	$\sqrt{3}$
1, 3	($\pm 1, 0$)	$\frac{1}{6}$	
2, 4	(0, ± 1)	$\frac{1}{6}$	

B.4 D2Q9 Model

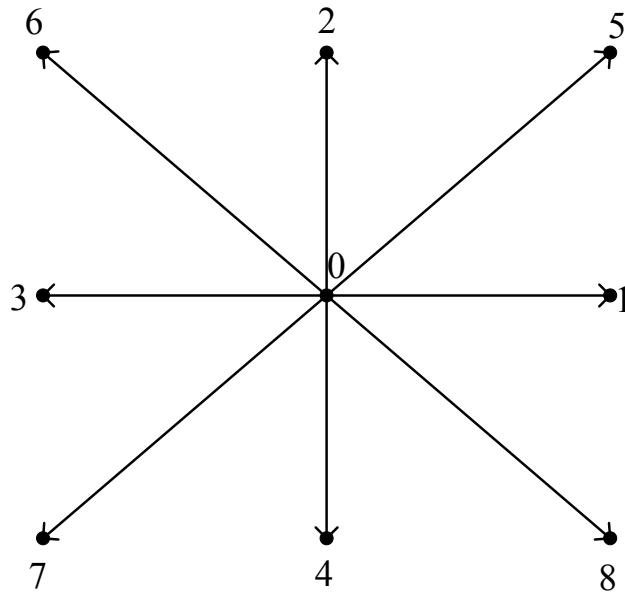


Figure B.4: 2 dimensional D2Q9 stencil

Table B.4: Parameters of some D2Q9 model

alpha	lattice velocity $\frac{\vec{c}}{c}$	Weighting w_α	c/c_s
0	(0, 0)	$\frac{4}{9}$	$\sqrt{3}$
1, 3	($\pm 1, 0$)	$\frac{1}{9}$	
2, 4	(0, ± 1)	$\frac{1}{9}$	
5, 6, 7, 8	($\pm 1, \pm 1$)	$\frac{1}{36}$	

B.5 D3Q15 Model

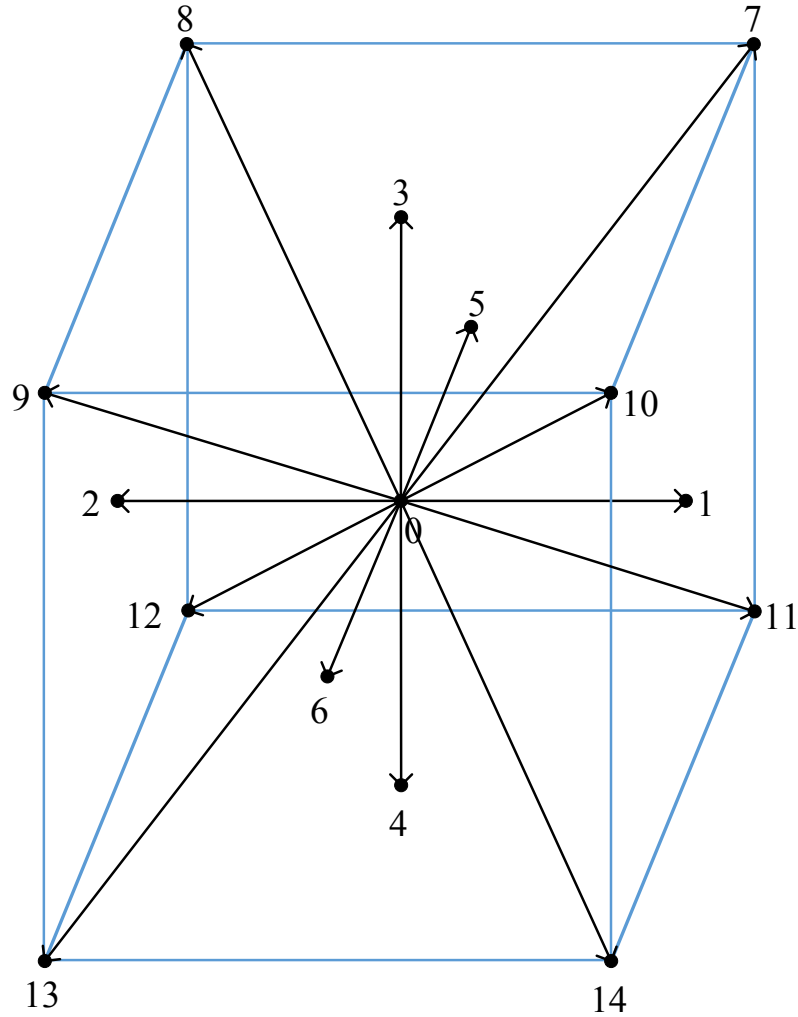


Figure B.5: 3 dimensional D3Q15 stencil

Table B.5: Parameters of some D3Q15 model

alpha	lattice velocity \vec{e}_α	Weighting w_α	c/c_s
0	(0, 0, 0)	$\frac{2}{9}$	$\sqrt{3}$
1, 2	($\pm 1, 0, 0$)	$\frac{1}{9}$	
3, 4	(0, $\pm 1, 0$)	$\frac{1}{9}$	
5, 6	(0, 0, ± 1)	$\frac{1}{9}$	
7 - 14	($\pm 1, \pm 1, \pm 1$)	$\frac{1}{72}$	

B.6 D3Q19 Model

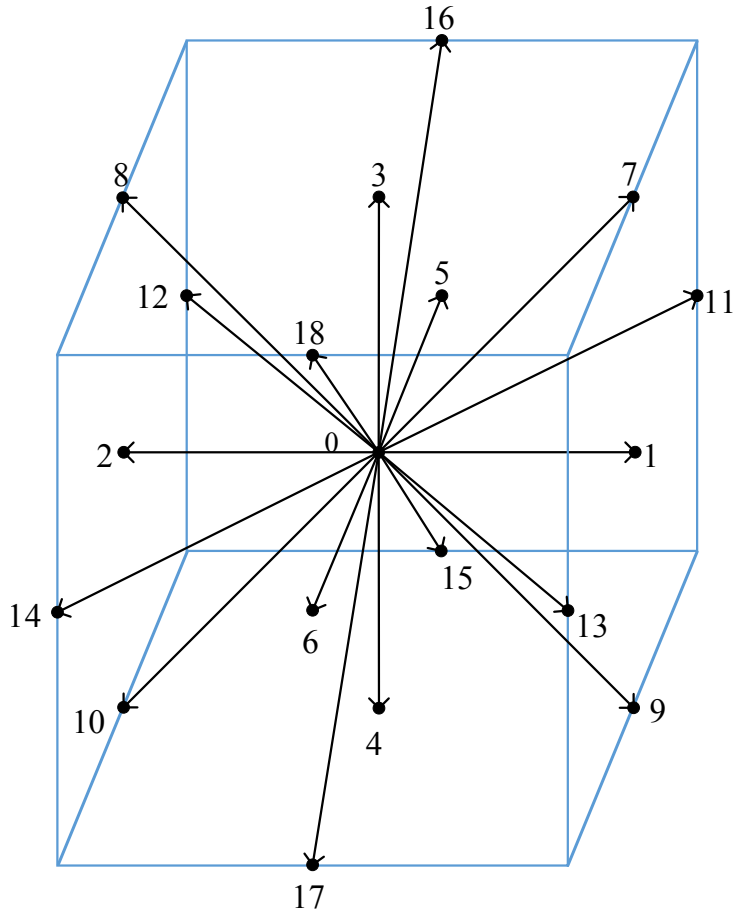


Figure B.6: 3 dimensional D3Q19 stencil

Table B.6: Parameters of some D3Q19 model

alpha	lattice velocity $\frac{\vec{e}}{c}$	Weighting w_α	c/c_s
0	(0, 0, 0)	$\frac{1}{3}$	$\sqrt{3}$
1, 2	($\pm 1, 0, 0$)	$\frac{1}{18}$	
3, 4	(0, $\pm 1, 0$)	$\frac{1}{18}$	
5, 6	(0, 0, ± 1)	$\frac{1}{18}$	
7 – 10	($\pm 1, \pm 1, 0$)	$\frac{1}{36}$	
11 – 14	($\pm 1, 0, \pm 1$)	$\frac{1}{36}$	
15 – 18	(0, $\pm 1, \pm 1$)	$\frac{1}{36}$	

B.7 D3Q27 Model

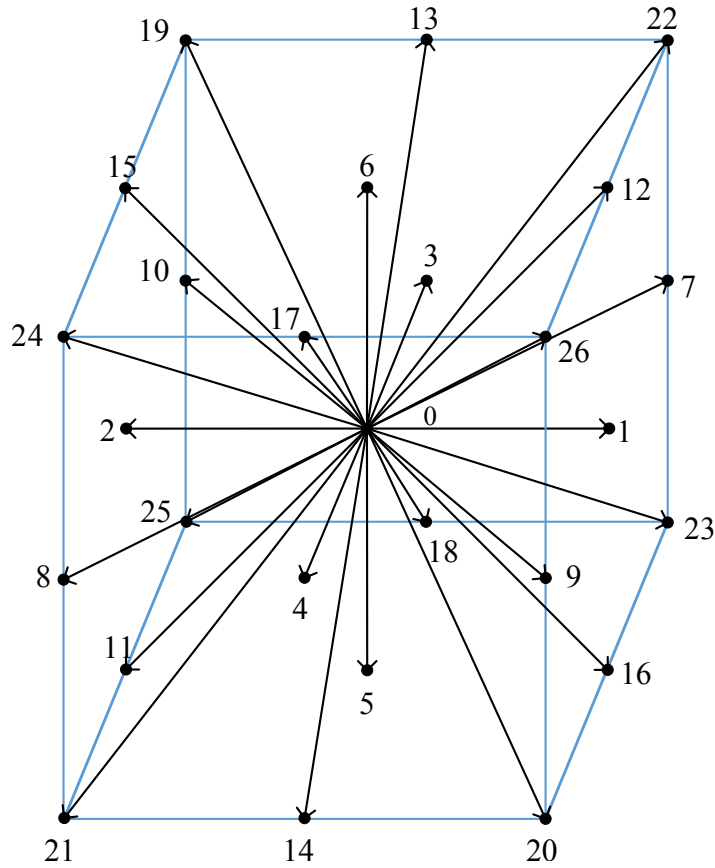


Figure B.7: 3 dimensional D3Q27 stencil

Table B.7: Parameters of some D3Q27 model

alpha	lattice velocity $\frac{\vec{e}}{c}$	Weighting w_α	c/c_s
0	(0, 0, 0)	$\frac{8}{27}$	$\sqrt{3}$
1, 2	($\pm 1, 0, 0$)	$\frac{2}{27}$	
3, 4	(0, $\pm 1, 0$)	$\frac{2}{27}$	
5, 6	(0, 0, ± 1)	$\frac{2}{27}$	
7 – 10	($\pm 1, \pm 1, 0$)	$\frac{1}{54}$	
11 – 14	($\pm 1, 0, \pm 1$)	$\frac{1}{54}$	
15 – 18	(0, $\pm 1, \pm 1$)	$\frac{1}{54}$	
19 – 27	($\pm 1, \pm 1, \pm 1$)	$\frac{1}{216}$	

APPENDIX C

THOMAS ALGORITHM

In numerical solution of some partial differential equations in CFD, a special form of algebraic equations are encountered. Then, Thomas Algorithm is used for solving these types of equations. Thomas Algorithm is basically a gauss elimination method optimized for tridiagonal system of equations. The solution algorithm of tridiagonal matrix is fairly easy. Sometimes the solution procedure yields an algebraic systems of equations such that the matrix is composed of small matrices, unknowns and constants are group of vectors. Such system is called block tridiagonal system and can be solved by adapting the Thomas Algorithm to the block tridiagonal system. The solution approaches are given in the proceeding sections of this Appendix. There are more documents about the properties of systems that can be solved by Thomas Algorithm for tridiagonal and block tridiagonal systems. The details of the methods can be found in the literature. In reference [98], a symbolic approach for generalized Thomas Algorithm is presented. The methods given in the Appendix assumes that the system wield enough properties to be solved by Thomas Algorithm. A more detailed information on solution procedures of tridiagonal and block tridiagonal systems are give in [99, ch 4]

C.1 Tridiagonal System Solution

A system of linear algebraic equations can be represented in a form given in equation C.1 . In the given equation A is a $n \times n$ matrix and \vec{x} and \vec{y} are vectors (or single column vectors) with n elements.

$$A\vec{x} = \vec{y} \quad (\text{C.1})$$

The matrix A has nonzero coefficients only in diagonal and the neighbors of diagonal elements. The equation can be written more explicitly:

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{n-2} & b_{n-2} & c_{n-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{n-1} & b_{n-1} & c_{n-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_n & b_n & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ . \\ . \\ . \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix}$$

The solution algorithm can be placed as follows:

- A forward sweep to calculate modified coefficients as denoted by primes,

$$c'_i = \begin{cases} \frac{c_i}{b_i} & i = 1 \\ \frac{c_i}{b_i - a_i c'_{i-1}} & i = 2, 3, \dots, n-1 \end{cases}$$

$$b'_i = 1 \quad i = 1, 2, \dots, n$$

$$y'_i = \begin{cases} \frac{y_i}{b_i} & i = 1 \\ \frac{y_i - a_i y'_{i-1}}{b_i - a_i c'_{i-1}} & i = 2, 3, \dots, n \end{cases}$$

- A backward sweep is used to calculate the unknowns:

$$x_i = \begin{cases} y'_i & i = n \\ y'_i - c'_i x_{i+1} & n-1, n-2, \dots, 1 \end{cases}$$

C.2 Block Tridiagonal System Solution

Let us consider a block tridiagonal system of linear equations with n blocks given by

$$TX = Y$$

Where T is a $n \times n$ block tridiagonal matrix, X and Y are vectors. Assume that each element of T (i.e. A, B, C) is $m \times m$ sub-matrix and each element of X and Y is $m \times 1$ vector. The system can be written more explicitly:

$$\begin{bmatrix} B_1 & C_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_2 & B_2 & C_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_3 & B_3 & C_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{n-2} & B_{n-2} & C_{n-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{n-1} & B_{n-1} & C_{n-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_n & B_n & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ . \\ . \\ . \\ X_{n-2} \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ . \\ . \\ . \\ Y_{n-2} \\ Y_{n-1} \\ Y_n \end{bmatrix}$$

The solution steps very similar to the one presented in section C.1 . The algorithm can be summarized as follows.

- A forward sweep to calculate modified coefficients as denoted by primes,

$$C'_i = \begin{cases} B_i^{-1} C_i & i = 1 \\ (A_i^{-1} B_i - C'_{i-1})^{-1} C_i & i = 2, 3, \dots, n-1 \end{cases}$$

$$B'_i = I \quad i = 1, 2, \dots, n$$

$$Y'_i = \begin{cases} B_i^{-1}Y_i & i = 1 \\ (A_i^{-1}B_i - C'_{i-1})^{-1} (A_i^{-1}Y_i - Y'_{i-1})^{-1} & i = 2, 3, \dots, n \end{cases}$$

- A backward sweep is used to calculate the unknowns:

$$X_i = \begin{cases} Y'_i & i = n \\ Y'_i - C'_i Y_{i+1} & i = n - 1, n - 2, \dots, 1 \end{cases}$$

APPENDIX D

APPROXIMATE FACTORIZATION METHOD

D.1 General Convection Equation

A general convection equation without a source term can be given as:

$$Q_t = -F_x - G_y - H_z \quad (\text{D.1})$$

The above equation can be solved by applying either an explicit or implicit methods. The discretization of the above equation in a more general way is given in equation D.2.

$$\left(\frac{\Delta Q}{\Delta t}\right)^n = -\frac{\beta}{1+\omega} R^{n+1} - \frac{1-\beta}{1+\omega} R^n + \frac{\omega}{1+\omega} \left(\frac{\Delta Q}{\Delta t}\right)^{n-1} \quad (\text{D.2})$$

where R is the residual term $R = F_x + H_y + G_z + S$. Setting $\beta = 1$ and $\omega = 0$ and defining the ΔQ^n operator as $Q^{n+1} - Q^n$ we have backward Euler method.

$$\Delta Q^n = -R^{n+1} \Delta t \quad (\text{D.3})$$

The equation can be linearized as follows:

$$\Delta Q^n = -\left(R^n + \left(\frac{\partial R}{\partial Q}\right)^n (\Delta Q)^n\right) \Delta t \quad (\text{D.4})$$

more explicitly

$$\begin{aligned} \Delta Q^n = & -[(F_x + G_y + H_z)^n + \left(\frac{\partial F_x}{\partial Q} \Delta Q\right)^n \dots \\ & + \left(\frac{\partial G_y}{\partial Q} \Delta Q\right)^n + \left(\frac{\partial H_z}{\partial Q} \Delta Q\right)^n] \Delta t \end{aligned} \quad (\text{D.5})$$

the equation can be rearranged

$$\Delta Q^n + \Delta t (F_Q \Delta Q)_x^n + \Delta t (G_Q \Delta Q)_y^n + \Delta t (H_Q \Delta Q)_z^n = -R^n \quad (\text{D.6})$$

The LHS matrix is sparse matrix. The bandwidth is very large due to the derivatives in x, y, and z direction. The exact solution is possible but it is very time consuming. Since there are already errors introduced by linearization of the equation and some lower order derivatives some approximate methods that can be used to ease the burden of calculation with an acceptable loss in the accuracy of the solution.

Equation D.6 can be rewritten in the following form:

$$[I + L_x + D_x + U_x + L_y + D_y + U_y + L_z + D_z + U_z] \Delta Q^n = -R \Delta t \quad (\text{D.7})$$

L, D, U are the lower, diagonal and upper terms obtained from the spatial discretization of the space derivatives in x, y and z directions. R is the residual which must approach to zero in either for steady state problems or sub-iteration in a particular time step for unsteady problems. More detailed literature, about the approximate factorization methods can be found in references[100, 101, 102, 103].

For the 3 approximate factorization methods, approximation error is introduced. The solution methods can be combined with Newton-Raphson sub-iterations or with dual time stepping to eliminate the errors coming from linearization and factorization.

D.2 Alternating Direction Implicit Method

The equation D.7 can be split in the spatial directions as follows:

$$[I + A][I + B][I + C] \Delta Q^n = R \Delta t \quad (\text{D.8})$$

where the A , B , C contain the terms due to spatial discretization.

$$\begin{aligned} A &= L_x + D_x + U_x \\ B &= L_y + D_y + U_y \\ C &= L_z + D_z + U_z \end{aligned} \quad (\text{D.9})$$

Then the equation D.8 can be solved as follows:

$$\begin{aligned} [I + A] \overline{\Delta Q} &= -\Delta t R \\ [I + B] \overline{\overline{\Delta Q}} &= \overline{\Delta Q} \\ [I + C] \Delta Q^n &= \overline{\overline{\overline{\Delta Q}}} \end{aligned} \quad (\text{D.10})$$

The factorization error can be defined by comparing equation D.7 and equation D.8.

The matrix multiplication of equation D.8 is written as follows:

$$[I + A][I + B][I + C] = [I + A + B + C + AB + AC + BC + ABC] \quad (\text{D.11})$$

It can be seen that the extra terms AB , AC , BC , ABC are the errors due to factorization. Each of the terms A , B , C are in the order of $\Delta t/\Delta x$, then we have the order of magnitude of the error such as

$$\begin{aligned} \epsilon &\approx O((\Delta t/\Delta x)^2) \quad \Delta t/\Delta x \ll 1 \\ \epsilon &\approx O((\Delta t/\Delta x)^3) \quad \Delta t/\Delta x \gg 1 \end{aligned}$$

Inspecting the errors it can be easily seen that ADI method is not suitable for large time steps, however the method is second order accurate for small time steps provided that the baseline method second order accurate.

For 2D problems, the error is given as AB , since there is no gradients in the 3rd axis. Then the order of magnitude of error is not changing with $\Delta t/\Delta x$. The ADI method

performs better in 2D problems but it is also applicable to 3D problems with small time steps.

$$\begin{aligned}\epsilon &\approx O((\Delta t/\Delta x)^2) \quad \Delta t/\Delta x \ll 1 \\ \epsilon &\approx O((\Delta t/\Delta x)^2) \quad \Delta t/\Delta x \gg 1\end{aligned}$$

D.3 Diagonally Dominant Alternating Direction Implicit Method

Diagonally Dominant Alternating Direction Implicit method can be written as the follows :

$$[D + \bar{A}] D^{-1} [D + \bar{B}] D^{-1} [D + \bar{C}] \Delta Q^n = -\Delta t R \quad (\text{D.12})$$

where the terms are grouped differently

$$\begin{aligned}D &\equiv [I + D_x + D_y + D_z] \\ \bar{A} &\equiv [L_x + U_x] \\ \bar{B} &\equiv [L_y + U_y] \\ \bar{C} &\equiv [L_z + U_z]\end{aligned} \quad (\text{D.13})$$

The D term is composed of all the diagonal terms, and \bar{A} , \bar{B} , \bar{C} are the off diagonal terms coming from the spatial directions. The solution algorithm is very similar to the ADI method.

$$\begin{aligned}[D + \bar{A}] \bar{\Delta Q} &= -\Delta t R \\ [D + \bar{B}] \bar{\bar{\Delta Q}} &= D \bar{\Delta Q} \\ [D + \bar{C}] \Delta Q^n &= D \bar{\bar{\Delta Q}}\end{aligned} \quad (\text{D.14})$$

However a very similar error analysis shows that the order of magnitude of the error is a little bit different for the 3D case.

$$[D + \bar{A}] D^{-1} [D + \bar{B}] D^{-1} [D + \bar{C}] = [D + \bar{A} + \bar{B} + \bar{C} + \bar{A}D^{-1}\bar{B} + \bar{A}D^{-1}\bar{C} + \bar{B}D^{-1}\bar{C} + \bar{A}D^{-1}\bar{B}D^{-1}\bar{C}] \dots$$

Comparing the above equation with equation D.7, the error terms are given as:

$$[\bar{A}D^{-1}\bar{C} + \bar{B}D^{-1}\bar{C} + \bar{A}D^{-1}\bar{B}D^{-1}\bar{C}] \quad (\text{D.15})$$

The terms \bar{A} , \bar{B} and \bar{C} are order of magnitude $\Delta t/\Delta x$ and the term D is order of magnitude $(1 + \Delta t/\Delta x)$. Then the factorization error can be written as:

$$\begin{aligned} \epsilon &\approx O((\Delta t/\Delta x)^2) \quad \Delta t/\Delta x \ll 1 \\ \epsilon &\approx O(\Delta t/\Delta x) \quad \Delta t/\Delta x \gg 1 \end{aligned}$$

DDADI method is better than the ADI method for large time steps. Also convergence for 2D and 3D is similar unlikely to ADI method.

D.4 Lower Upper Symmetric Gauss-Seidel Method

LU-SGS is an alternative approximation method and the splitting scheme is grouping the lower, upper and diagonal terms together.

$$[D + L] D^{-1} [D + U] \Delta Q^n = -\Delta t R \quad (\text{D.16})$$

The terms are defined as collecting the diagonal, lower and upper terms.

$$\begin{aligned} D &\equiv [I + D_x + D_y + D_z] \\ L &\equiv [L_x + L_y + L_z] \\ U &\equiv [U_x + U_y + U_z] \end{aligned} \quad (\text{D.17})$$

The LU-SGS is a two factor factorization method even for 3D problems. The solution algorithm is given as:

$$\begin{aligned} [D + L] \overline{\Delta Q} &= -\Delta t R \\ [D + U] \Delta Q^n &= D \overline{\Delta Q} \end{aligned} \quad (\text{D.18})$$

Furthermore, from the equation D.18 it can easily be seen that each factor contains more than 1 spatial direction. It is a problem during the calculation for parallel solvers. However, it is possible to define hyper planes where the solution is independent of the other hyper planes. Other than two step solution algorithms, there are other sweep techniques which would yield in the reduction of the factorization error. Some other approximate factorization methods can be found on reference [103].

Two-sweep LU-SGS method has the factorization error when compared with equation D.7:

$$[D + L] D^{-1} [D + U] \Delta Q^n = [D + L + U + LD^{-1}U] \quad (\text{D.19})$$

$$LD^{-1}U$$

Each of the terms L, U is of the order, $\Delta t/\Delta x$, and D is of the order, $1 + \Delta t/\Delta x$, then the order of magnitude of factorization error is given as:

$$\begin{aligned} \epsilon &\approx O((\Delta t/\Delta x)^2) \quad \Delta t/\Delta x \ll 1 \\ \epsilon &\approx O((\Delta t/\Delta x)) \quad \Delta t/\Delta x \gg 1 \end{aligned}$$

Which is why LU-SGS is very nice for large time steps! The convergence in 2-D is essentially similar to that in 3-D for LU-SGS.

APPENDIX E

CONTROL VOLUME AND GEOMETRIC QUANTITIES

In this thesis, finite volume formulation of the Lattice Boltzmann equation is coded on structured grids. This Appendix gives information about the metrics and parameters of the grid cell (control volume) for 2D and 3D. The magnitude of the surface area for face m is denoted by A_m , the unit normal vector for the same face is \vec{n}_m . The face vector is $\vec{S}_m = A_m \vec{n}_m$ and the volume of the cell in focus is denoted by Ω .

E.1 2D Control Volume

For the 2D application it is assumed that the volume elements has a unit thickness in z direction. By this, the correct physical units can be obtained. In Figure E.1, 2D control volume is presented. The I and J indexes are the index of the control volume.

The area (actually it is the volume where the depth is unity) of the quadrilateral can be calculated from the coordinates of the corner points. It is generally assumed that the control volume lies in the $x - y$ plane and the z is the symmetry axis.

$$\Omega_{I,J} = \frac{1}{2} [(x_1 - x_3)(y_2 - y_4) + (x_4 - x_2)(y_1 - y_3)] \quad (\text{E.1})$$

In 2D, the edges of the faces of the control volume in $x - y$ plane are straight lines, so it is a very clear procedure to calculate the area A of any face and the associated unit normal vector \vec{n} .

First we calculate the surface vectors:

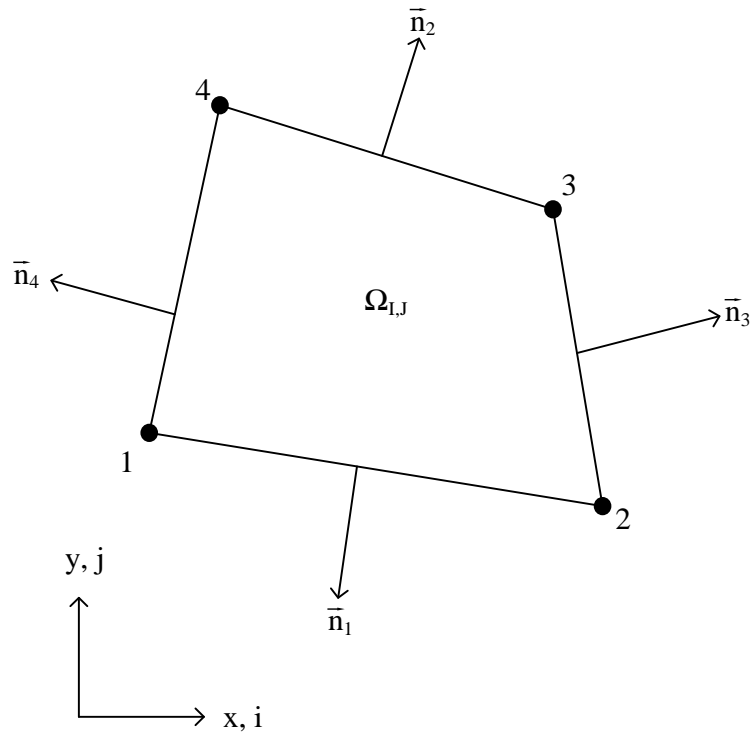


Figure E.1: 2D control volume element

$$\begin{aligned}
 \vec{S}_1 &= \begin{bmatrix} y_2 - y_1 \\ x_1 - x_2 \end{bmatrix} \\
 \vec{S}_2 &= \begin{bmatrix} y_3 - y_2 \\ x_2 - x_3 \end{bmatrix} \\
 \vec{S}_3 &= \begin{bmatrix} y_4 - y_3 \\ x_3 - x_4 \end{bmatrix} \\
 \vec{S}_4 &= \begin{bmatrix} y_1 - y_4 \\ x_4 - x_1 \end{bmatrix}
 \end{aligned} \tag{E.2}$$

The area of each face is the magnitude of the surface vector \vec{S} and the unit normal

vector \vec{n} is given as:

$$\begin{aligned}
A_1 = \left| \vec{S}_1 \right| &= \sqrt{(y_2 - y_1)^2 + (x_1 - x_2)^2} \Rightarrow \vec{n}_1 = \frac{\vec{S}_1}{A_1} \\
A_2 = \left| \vec{S}_2 \right| &= \sqrt{(y_3 - y_2)^2 + (x_2 - x_3)^2} \Rightarrow \vec{n}_2 = \frac{\vec{S}_2}{A_2} \\
A_3 = \left| \vec{S}_3 \right| &= \sqrt{(y_4 - y_3)^2 + (x_3 - x_4)^2} \Rightarrow \vec{n}_3 = \frac{\vec{S}_3}{A_3} \\
A_4 = \left| \vec{S}_4 \right| &= \sqrt{(y_1 - y_4)^2 + (x_4 - x_1)^2} \Rightarrow \vec{n}_4 = \frac{\vec{S}_4}{A_4}
\end{aligned} \tag{E.3}$$

The mid point coordinate of the cell element is the arithmetical average of the corner point coordinates.

$$x_{I,J} = \frac{1}{n_{vertices}} \sum_{m=1}^{n_{vertices}} x_m \tag{E.4}$$

$$y_{I,J} = \frac{1}{n_{vertices}} \sum_{m=1}^{n_{vertices}} y_m \tag{E.5}$$

E.2 3D Control Volume

To calculate the metrics and geometric properties of a 3D control volume shown in Figure E.2 is not strict forward as in 2D. The main reason is that in space 3 points define a plane, so the 4th point may be out of plane, which brings the problem that the normal vector is not constant on the face of the hexagonal. This situation can be solved in two ways the face can be divided into triangles and the fluxes can be calculated by using each triangle element on that particular face. The second way is to use average representative quantities. For smooth grids, using the average area and a representative normal vectors gives accurate results without dividing the faces into triangles as given in reference [104][60, p. 82].

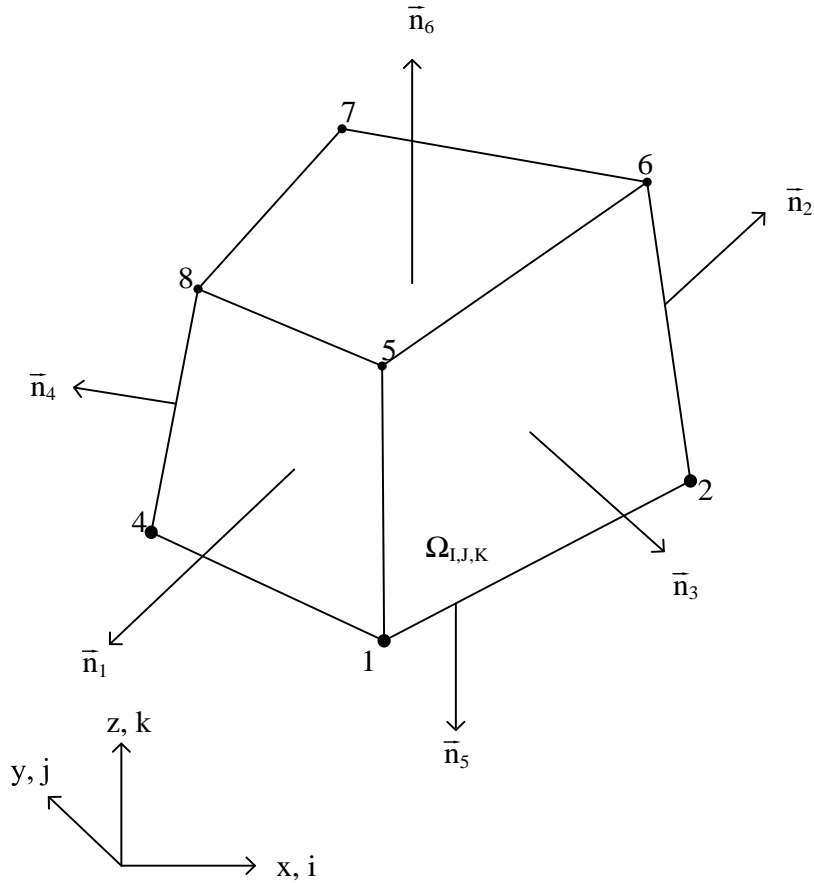


Figure E.2: 3D Control Volume Element

To calculate the surface vectors of faces: we use the given formulas:

Face 1 , related corner points 1, 5, 8, 4

$$x_A = x_8 - x_1 \quad x_B = x_5 - x_4 \quad (E.6)$$

$$y_A = y_8 - y_1 \quad y_B = y_5 - y_4 \quad (E.7)$$

$$z_A = z_8 - z_1 \quad z_B = z_5 - z_4 \quad (E.8)$$

$$\vec{S}_1 = \frac{1}{2} \begin{bmatrix} z_A y_B - y_A z_B \\ x_A z_B - z_A x_B \\ y_A x_B - x_A y_B \end{bmatrix} \quad (E.9)$$

Face 2 , related corner points 2, 3, 7, 6

$$x_A = x_7 - x_2 \quad x_B = x_3 - x_6 \quad (\text{E.10})$$

$$y_A = y_7 - y_2 \quad y_B = y_3 - y_6 \quad (\text{E.11})$$

$$z_A = z_7 - z_2 \quad z_B = z_3 - z_6 \quad (\text{E.12})$$

$$\vec{S}_2 = \frac{1}{2} \begin{bmatrix} z_A y_B - y_A z_B \\ x_A z_B - z_A x_B \\ y_A x_B - x_A y_B \end{bmatrix} \quad (\text{E.13})$$

Face 3, related corner points 1, 2, 6, 5

$$x_A = x_6 - x_1 \quad x_B = x_2 - x_5 \quad (\text{E.14})$$

$$y_A = y_6 - y_1 \quad y_B = y_2 - y_5 \quad (\text{E.15})$$

$$z_A = z_6 - z_1 \quad z_B = z_2 - z_5 \quad (\text{E.16})$$

$$\vec{S}_3 = \frac{1}{2} \begin{bmatrix} z_A y_B - y_A z_B \\ x_A z_B - z_A x_B \\ y_A x_B - x_A y_B \end{bmatrix} \quad (\text{E.17})$$

Face 4, related corner points 4, 8, 7, 3

$$x_A = x_7 - x_4 \quad x_B = x_8 - x_3 \quad (\text{E.18})$$

$$y_A = y_7 - y_4 \quad y_B = y_8 - y_3 \quad (\text{E.19})$$

$$z_A = z_7 - z_4 \quad z_B = z_8 - z_3 \quad (\text{E.20})$$

$$\vec{S}_4 = \frac{1}{2} \begin{bmatrix} z_A y_B - y_A z_B \\ x_A z_B - z_A x_B \\ y_A x_B - x_A y_B \end{bmatrix} \quad (\text{E.21})$$

Face 5 , related corner points 4, 3, 2, 1

$$x_A = x_2 - x_4 \quad x_B = x_3 - x_1 \quad (\text{E.22})$$

$$y_A = y_2 - y_4 \quad y_B = y_3 - y_1 \quad (\text{E.23})$$

$$z_A = z_2 - z_4 \quad z_B = z_3 - z_1 \quad (\text{E.24})$$

$$\vec{S}_5 = \frac{1}{2} \begin{bmatrix} z_A y_B - y_A z_B \\ x_A z_B - z_A x_B \\ y_A x_B - x_A y_B \end{bmatrix} \quad (\text{E.25})$$

Face 6 , related corner points 5, 6, 7, 8

$$x_A = x_7 - x_5 \quad x_B = x_6 - x_8 \quad (\text{E.26})$$

$$y_A = y_7 - y_5 \quad y_B = y_6 - y_8 \quad (\text{E.27})$$

$$z_A = z_7 - z_5 \quad z_B = z_6 - z_8 \quad (\text{E.28})$$

$$\vec{S}_6 = \frac{1}{2} \begin{bmatrix} z_A y_B - y_A z_B \\ x_A z_B - z_A x_B \\ y_A x_B - x_A y_B \end{bmatrix} \quad (\text{E.29})$$

The areas of the faces and normal vectors can be calculated as follows:

$$\begin{aligned}
A_1 &= \left| \vec{S}_1 \right| = \sqrt{S_{x,1}^2 + S_{y,1}^2 + S_{z,1}^2} \Rightarrow \vec{n}_1 = \frac{\vec{S}_1}{A_1} \\
A_2 &= \left| \vec{S}_2 \right| = \sqrt{S_{x,2}^2 + S_{y,2}^2 + S_{z,2}^2} \Rightarrow \vec{n}_2 = \frac{\vec{S}_2}{A_2} \\
A_3 &= \left| \vec{S}_3 \right| = \sqrt{S_{x,3}^2 + S_{y,3}^2 + S_{z,3}^2} \Rightarrow \vec{n}_3 = \frac{\vec{S}_3}{A_3} \\
A_4 &= \left| \vec{S}_4 \right| = \sqrt{S_{x,4}^2 + S_{y,4}^2 + S_{z,4}^2} \Rightarrow \vec{n}_4 = \frac{\vec{S}_4}{A_4} \\
A_5 &= \left| \vec{S}_5 \right| = \sqrt{S_{x,5}^2 + S_{y,5}^2 + S_{z,5}^2} \Rightarrow \vec{n}_4 = \frac{\vec{S}_4}{A_4} \\
A_6 &= \left| \vec{S}_6 \right| = \sqrt{S_{x,6}^2 + S_{y,6}^2 + S_{z,6}^2} \Rightarrow \vec{n}_4 = \frac{\vec{S}_4}{A_4}
\end{aligned} \tag{E.30}$$

To calculate the volume of the cell, it is better to start with the Gauss divergence theorem:

$$\int_{\Omega} (\vec{\nabla} \cdot \vec{r}) d\Omega = \int_S (\vec{r} \cdot \vec{n}) dS \tag{E.31}$$

Assume that the vector \vec{r} is an arbitrary vector drawn from the origin to any point that is positioned inside the cell and defined as $\vec{r} = x\vec{i} + y\vec{j} + z\vec{k}$. With these definitions it is easy to calculate the left hand side of equation E.31.

$$\int_{\Omega} (\vec{\nabla} \cdot \vec{r}) d\Omega = \int_{\Omega} \left(\frac{\partial x}{\partial x} + \frac{\partial y}{\partial y} + \frac{\partial z}{\partial z} \right) d\Omega = 3\Omega \tag{E.32}$$

The right side of equation E.31 can be approximated follows since \vec{n} is constant throughout each face .

$$\int_S (\vec{r} \cdot \vec{n}) dS \approx \sum_{m=1}^6 (\vec{r}_{mid} \cdot \vec{n})_m A_m \tag{E.33}$$

where \vec{r}_{mid} is the vector pointing through the midpoint of each face:

$$\begin{aligned}
r_{mid,1} &= \frac{1}{4} (\vec{r}_1 + \vec{r}_5 + \vec{r}_8 + \vec{r}_4) \\
r_{mid,2} &= \frac{1}{4} (\vec{r}_2 + \vec{r}_3 + \vec{r}_6 + \vec{r}_7) \\
r_{mid,3} &= \frac{1}{4} (\vec{r}_1 + \vec{r}_2 + \vec{r}_6 + \vec{r}_5) \\
r_{mid,4} &= \frac{1}{4} (\vec{r}_4 + \vec{r}_8 + \vec{r}_7 + \vec{r}_3) \\
r_{mid,5} &= \frac{1}{4} (\vec{r}_4 + \vec{r}_3 + \vec{r}_2 + \vec{r}_1) \\
r_{mid,6} &= \frac{1}{4} (\vec{r}_5 + \vec{r}_6 + \vec{r}_7 + \vec{r}_8)
\end{aligned} \tag{E.34}$$

Using equations E.32, E.33 the volume of the cell can be calculated for the cell .

$$\Omega_{I,J,K} = \frac{1}{3} \sum_{m=1}^6 (\vec{r}_{mid} \cdot \vec{S})_m = \frac{1}{3} \sum_{m=1}^6 (\vec{r}_{mid} \cdot \vec{n})_m A_m \tag{E.35}$$

If the control volumes are enclosed by planar faces, the calculated volume by equation E.35 is exact.

The midpoint coordinates of the cell is calculated as follows:

$$x_{I,J,K} = \frac{1}{n_{vertices}} \sum_{m=1}^{n_{vertices}} x_m \tag{E.36}$$

$$y_{I,J,K} = \frac{1}{n_{vertices}} \sum_{m=1}^{n_{vertices}} y_m \tag{E.37}$$

$$z_{I,J,K} = \frac{1}{n_{vertices}} \sum_{m=1}^{n_{vertices}} z_m \tag{E.38}$$

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Çevik, Fatih

Nationality: Turkish (TC)

Date and Place of Birth: 10.10.1970, Ereğli

Marital Status: Married

Phone: +90 532 646 2926

Fax: +90 312 213 82 17

EDUCATION

Degree	Institution	Year of Graduation
M.S.	METU, Graduate School of Natural and Applied Sciences, Mechanical Engineering Department	2010
B.S.	METU, Faculty of Engineering, Aeronautical Engineering Department	1994
High School	Ankara Science High School	1988

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2010 - Present	Aselsan Inc	Lead Design Engineer
2006 - 2010	STM Inc.	Lead Certification Engineer
1994 - 2006	Roketsan Inc.	Chief Engineer

PUBLICATIONS

International Conference Publications

1. Guzel, G., Cetiner, E. and Cevik, F., “An Alternative Method For Calculation Of Missile Steady Pitch Damping”, AIAC-2015-024, Ankara International Aerospace Conference, Turkey
2. Cevik, F., Guzel, G., Yagiz, B., Aksel, H., Albayrak., K., “Convergence Acceleration Of Lattice Boltzmann Method Via Finite Difference Implementation”, AIAC-2013-020, Ankara International Aerospace Conference, Turkey
3. Cevik, F., Guzel, G., Yagiz, B., Aksel, H., Albayrak., K., “Finite-Difference Implementation Of Lattice Boltzmann Method For Use With Non-Uniform Grids”, AIAC-2013-143, , Ankara International Aerospace Conference, Turkey

National Conference Publications

1. Ozgur, C., Cevik, F., “Ucus Testleri ” Ulusal Havacılık ve Uzay Kongresi (2008:ITU)
2. Ozdamar,H., Ucer, A. S., Oguztuzun, H., Cevik, F., "Kimyasal Silahlara Karsi Savunmada Etkinlik Modellemesi ve Simulasyonu" SAVTEK 2002:Savunma Teknolojileri Kongresi (2002:ODTU),Ankara, yayin.Kara Harp Okulu,Savunma Bilimleri Enstitusu;ODTU-Biltir Merkezi, 177-185.ss.

PATENTS

1. 2009-G-286764, “Dikey ruzgar tuneli hava degisim menfezi”, Fatih Çevik, Hakan İşçi, Mustafa Oğuz, 2012/07/23
2. 2010-G-2559, “Yuzey aciklilarinda akis kaynakli rezonans sonumleyici kabin”, Fatih Çevik, Hakan İşçi, Mustafa Oğuz, in progress