

**CUBIC INTERPOLATED PSEUDO PARTICLE (CIP)-LATTICE
BOLTZMANN SCHEME FOR THE SIMULATION OF LID-DRIVEN CAVITY
FLOW**

DANY AK DOMINIC NENG

**MAY 2010
UNIVERSITI MALAYSIA PAHANG**

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: CUBIC INTERPOLATED PSEUDO PARTICLE (CIP)-LATTICE BOLTZMANN SCHEME
FOR THE SIMULATION OF LID DRIVEN CAVITY FLOW

SESI PENGAJIAN: 2009/2010

Saya DANY AK DOMINIC NENG

(HURUF BESAR)

mengaku membenarkan kertas projek ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut :

1. Hakmilik kertas projek adalah di bawah nama penulis melainkan penulis sebagai projek bersama dan dibiayai oleh UMP, hakmiliknya adalah kepunyaan UMP.
2. Naskah salinan di dalam bentuk kertas atau mikro hanya boleh dibuat dengan kebenaran bertulis daripada penulis.
3. Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan pengajian mereka.
4. Kertas projek hanya boleh diterbitkan dengan kebenaran penulis. Bayaran royalti adalah mengikut kadar yang dipersetujui kelak.
5. *Saya membenarkan/tidak membenarkan Perpustakaan membuat salinan kertas projek ini sebagai bahan pertukaran di antara institusi pengajian tinggi.
6. ** Sila tandakan (✓)

☐

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☐

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan).

☐

TIDAK TERHAD

Disahkan oleh

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap: LOT 2164, S/LOT 1872,
JLN BEREMBANG 2, PUJUT 7A,
98000 MIRI, SARAWAK

MUHAMAD ZUHAIRI BIN SULAIMAN

Nama Penyelia

Tarikh: MAY 2010

Tarikh: MAY 2010

CATATAN: * Potong yang tidak berkenaan.

** Jika Kertas Projek ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

♦ Tesis ini dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM)

UNIVERSITI MALAYSIA PAHANG
FACULTY OF MECHANICAL ENGINEERING

We certify that the project entitled “Cubic Interpolated Pseudo Particle (CIP)-Lattice Boltzmann Scheme for Simulation of Lid-Driven Cavity Flow” is written by Dany Ak Dominic Neng. We have examined the final copy of this project and in our opinion; it is fully adequate in terms of scope and quality for the award of the degree of Bachelor of Engineering. We herewith recommend that it be accepted in partial fulfillment of the requirement for the degree of Bachelor of Mechanical Engineering.

Examiner

Signature

CUBIC INTERPOLATED PSEUDO PARTICLE (CIP)-LATTICE BOLTZMANN
SCHEME FOR THE SIMULATION OF LID-DRIVEN CAVITY FLOW

DANY ANAK DOMINIC NENG

Report submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Mechanical Engineering

Faculty of Mechanical Engineering
UNIVERSITI MALAYSIA PAHANG

MAY 2010

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Mechanical Engineering.

Signature:

Name of Supervisor: MUHAMAD ZUHAIRI BIN SULAIMAN

Date:

STUDENT'S DECLARATION

I hereby declare that the work in this project is my own except for quotations and summaries which have been duly acknowledged. The project has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature:

Name: DANY AK DOMINIC NENG

Date:

To my family, a special thanks for your love,
Dear friends, thank you for being my helper and guide,
Last, but not least to my supervisor, thanks a lot for your support and advice.

ACKNOWLEDGEMENT

First of all, I would like to express my sincere gratitude my supervisor Mr. Muhamad Zuhairi Bin Sulaiman for his ideas, invaluable guidance, continuous encouragement and constant support in making this research possible. During under his supervision, I saw him as a professional and understanding person. His knowledge in his field of study also impresses me. I appreciate his support, understanding and tolerance for my naïve mistakes. I also would like to give special thanks to Mr. Mohd Rosdzimin Bin Abdul Rahman, who had supervised me previously for my research. Sincere thanks for his commitment and co-operation in helping me with my research.

My sincere thanks go to all friends and members of the staff of Mechanical Engineering Department, UMP, who helped me in many ways and made my stay at UMP pleasant and unforgettable. Many special thanks go to member of research group for their excellent co-operation, inspirations and supports during this study.

Last but not least, I acknowledge my sincere indebtedness and gratitude for my parents for their love, dream and sacrifice throughout my life. I would like to give thanks to my parents for their constant support and encouragement during my study in UMP. Their devotion, support, and faith in me are appreciated with more than just words as I will do my best in achieving my goal.

ABSTRACT

In this thesis, the Cubic Interpolated Pseudo Particle (CIP)-Lattice Boltzmann Model (LBM) scheme is applied for the simulation of lid driven flow for deep cavities. The thesis describe on how the advection term in the lattice Boltzmann equation is discretise and solved using CIP. The CIP which is similar to the form of finite difference method is known as numerical method for solving advection equations with low numerical diffusion. The obtained results feature the flow in deep lid driven cavity, in which the streamlines, the dynamics of primary vortices and corner vortices, and the velocity profile are investigated and compared with previous study from experiment and theory. By coupled the CIP and LBM, good agreement is obtained for the present study with those previous study using original LBM.

ABSTRAK

Dalam tesis ini, simulasi menggunakan ‘Cubic Interpolated Pseudo Particle (CIP)-lattice Boltzmann Model (LBM)’ dijalankan untuk bekas dengan penutup bergerak yang dalam. Tesis ini menjelaskan persamaan kerangka ‘Boltzmann’ yang dideskritisikan dan menyelesaikannya dengan menggunakan kaedah CIP. Kaedah CIP yang juga hampir dalam bentuk yang serupa dengan kaedah ‘finite difference’ dikenalpasti sebagai kaedah ‘numerical’ untuk menyelesaikan persamaan kerangka bagi difusi ‘numerical’ yang rendah. Keputusan yang diperoleh menunjukkan perolakan dalam bekas dengan penutup bergerak yang dalam, ‘streamlines’, dinamik pusaran utama dan pusaran tepi, dan profil halaju dikaji serta dibandingkan dengan kajian melalui teori dan eksperimen sebelum ini. Dengan menggabungkan kaedah CIP dan LBM, persamaan yang ketara diantara kaedah yang baru dengan kaedah yang lama iaitu kaedah LBM konvensional dapat dilihat.

TABLES OF CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT	ii
ABSTRAK	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	v
LIST OF SYMBOLS	vi

CHAPTER 1 INTRODUCTION

1.1	Project Background	1
1.2	Project Objective	2
1.3	Project Scope	3
1.4	Problem Statement	3

CHAPTER 2 LITERATURE REVIEW

2.1	Navier-Stokes Equation	4
2.2	Computational Fluid Dynamics	5
2.3	Lattice Gas Approach	5
2.4	Lattice Boltzmann Method	6
2.5	Lid Driven Cavity	9
2.6	Cubic Interpolated Pseudo Particle	9

CHAPTER 3 METHODOLOGY

3.1	Introduction	16
3.2	Flow Chart	17
3.3	Literature Study of Lattice Boltzmann	18
3.4	Study on The Theory of Lattice Boltzmann	18
3.5	Study of Lattice Boltzmann Isothermal Model	19
3.5.1	Flow in Pipe	19

3.5.2	Flow with Moving Plate	19
3.6	Study On Lid Driven Cavity Flow	19
3.6.1	Initial and Boundary Conditions	20
3.6.2	Stationary Wall Conditions	20
3.6.3	Moving Wall Conditions	21
3.7	Study of CIP-LBM	21
3.8	Project Code Validation	22
3.8.1	Original LBM Algorithm	22
3.8.2	CIP-LBM Algorithm	23
3.9	Project Simulation	24

CHAPTER 4 RESULT AND DISCUSSION

4.1	Introduction	25
4.2	Validation of Original LBM Code Result	25
4.2.1	Poiseuille Flow Analysis	26
4.2.2	Coutte Flow Analysis	27
4.3	Results for Deep Lid-Driven Cavity Flow	28
4.4	Structure of Vortices	33
4.4.1	Structure of Primary Vortices	33
4.4.2	Structure of Corner Vortices	34
4.4.3	Comparison of Physical Details	35
4.5	Velocity Profile	36

CHAPTER 5 CONCLUSION AND RECOMMENDATION

5.1	Conclusions	39
5.2	Recommendations for Future Studies	40

REFERENCES	41
-------------------	----

APPENDICES	44
-------------------	----

LIST OF FIGURES

Figure No.	Title	Page
2.1	Particle distributions before and after stream step	7
2.2	Square lid driven cavity with three static and one moving walls	9
3.1	The flow process of study	17
3.2	Deep lid driven cavity with aspect ratio $K = H/L$	20
3.3	Flow chart of the original LBM algorithm	22
3.4	Flow chart of CIP-LBM algorithm	23
3.5	The visualization software, AVS Express 7.2, application menu	24
4.1	Flow in pipe based on different dimensionless velocity value	26
4.2	Flow velocity of Couette flow using lattice Boltzmann method	27
4.3	Comparison of the x -component of velocity, u , at the mid-plane with result by Patil et al. (2006), for $K = 2$	29
4.4	Streamlines of deep lid-driven cavity for $K = 1.5$; $Re =$ (a) 100, (b) 400, and (c) 1000	31
4.5	Streamlines (top row) and vorticity (bottom row) contours for deep-cavity flow by Patil et al. (2006) with aspect ratio $K = 1.5$. $Re =$ (a) 50, (b) 400, (c) 1000, (d) 3200	36
4.6	Profile of x -component velocity, u , through mid-plane at $Re =$ 100, 400, and 1000	37
4.7	Profile of y -component velocity, v , through mid-plane at $Re =$ 100, 400, and 1000	38

LIST OF SYMBOLS

a	Acceleration
\mathbf{c}	Micro velocity vector
$f(\mathbf{x}, \mathbf{c}, t)$	Density distribution function
f_i	Discretized density distribution function
f_i^{eq}	Discretized equilibrium density distribution function
$F_{f,g}$	External force
g_i	Discretized internal energy distribution function
g_i^{eq}	Discretized equilibrium internal energy distribution function
ν	Kinematic viscosity
ρ	Density
t	Time
Ω	Collision operator
U	Horizontal velocity of top plate
τ	Time relaxation
u	Horizontal velocity
v	Vertical velocity

CHAPTER 1

INTRODUCTION

1.1 PROJECT BACKGROUND

Fluid dynamics is a branch of classical physics and it is an instance of continuum mechanics. Fluid dynamics influence the improvement and development of technologies in a variety of field in which its application involves industries such as automotive, aviation, marine, construction, biomedical, chemical processing, oil and gas, and many others. Wide range of problems from ventilations in building to the smallest scale in micro-pumps can be addressed because of fluid dynamic scalable nature.

The fundamentals of fluid dynamics are related to the conservation law that consist of conservation of mass, Newton's Second Law of Motion, and the First Law of Thermodynamics which are based on classical mechanics and are modified in quantum mechanics and general relativity. Subsequently produce the continuity equation and the Navier-Stoke equation, a mathematical relationship governing the fluid flow. The Navier-Stoke equations are nonlinear partial differential equations in many cases, and the nonlinearity makes most cases difficult or impossible to solve.

The rise and constant development of computer technologies however helps to solve fluid problems by using computational or numerical method. One of the essential tools that can provide solution to a complex equation such as the Navier-Stoke equation, the continuity equation, or any equations which derives from it is the Computational Fluid Dynamics (CFD).

In recent years, another class of computational technique is used to simulate fluid flow as oppose to conventional methods based on the Navier-Stoke equation. The method which is known as Lattice Boltzmann method (LBM) is a discrete computational method based upon the Boltzmann equation. A number of standard benchmark problems and other complex flow problems have been simulated. The LBM prove to be an efficient method with its ability to easily represent complex physical phenomena, ranging from single and multiphase flow in complex geometries to fluids with chemical reaction. One of the key advantages for LBM is the suitability for parallel computation. However, the framework of LBM is still lacking in spite of the great interest in the method. But with rapid development in the field, the LBM are constantly become one the subject for potential improvement.

One of the classical benchmark problem been used regularly with LBM simulation is the lid-driven cavity flow, notably a two dimensional square cavity. The lid-driven cavity had been used to test new numerical schemes and methods. For this research, the lid-driven cavity is used as the main benchmark problem because of its available analytical solution. Details of previous simulations that had been carried out for the lid-driven cavity flow are widely available to be used for the study. The lid-driven cavities are used to simulate, observe, and analyze the flow structure or vortex formation in lid-driven cavity. However, the shape of the cavity does not necessarily limited to a two dimensional square cavity. In this study, the deep lid-driven cavity is to be use.

One of the purposes of this study is to understand the theory of LBM as well as the Cubic Interpolated Pseudo Particle (CIP) method. Other than that, the developing of CIP-LBM will be use to analyze the flow for deep lid-driven cavity.

1.2 PROJECT OBJECTIVE

The objective of this study is to develop Cubic Interpolated Pseudo Particle-Lattice Boltzmann scheme for the simulation of deep lid driven cavity flow.

1.3 PROJECT SCOPE

Analysis of lid driven cavity flow using Reynolds number between 1×10^3 to 1×10^5 . To simulate deep lid driven cavity flow using CIP-LBM scheme. Detail characteristic numerical value of flow will be carrying out; isothermal line, streamlines, maximum horizontal velocity and maximum vertical velocity with position where they occur will be compared.

1.4 PROBLEM STATEMENT

One of the problems of LBM is that the conventional LBM were limited to uniform grid in which the microscopic velocity is constant. Uniform grid cannot perform when high resolution is needed at specific location. Another problem case would be the low order accuracy of conventional LBM as well as its second order accuracy in time and space. The Cubic Interpolated Pseudo Particle (CIP) method can be use in application to linear or multi-dimensional problem. Since the method approach is almost similar to finite difference form, a CIP-LBM scheme would be develop for the simulation of lid cavity flow in this project.

CHAPTER 2

LITERATURE REVIEW

2.1 NAVIER-STOKES EQUATION

The Navier-Stokes equations were named for the French engineer and scientist Claude Louis Henri Navier and the English mathematical physicist George Gabriel Stokes. The equations essential form was set forth by Navier in 1822; however, the origin of viscous stress was not properly treat. The latter was addressed by others, in particular Poisson and Saint-Venant, but independently developed by Stokes in 1845. Stokes constructed a number of solutions to the equations of viscous flow, which confirmed their ability to describe fluid dynamical phenomena.

The equation which describe the motion of fluid substances, that is substances which can flow, arise from applying Newton's second law to fluid motion, together with the assumption that the fluid stress is the sum of a diffusing viscous term (proportional to the gradient of velocity), plus a pressure term. The mathematical relationship which govern the fluid flow is the continuity equation and Navier-Stokes equation given by

$$\nabla \cdot \mathbf{u} = 0 \quad (2.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u} \quad (2.2)$$

with velocity \mathbf{u} , pressure P , and kinematic shear viscosity ν . The Navier–Stokes equations are a set of nonlinear partial differential equations which, unlike algebraic

equations, do not explicitly establish a relation among the variables of interest (e.g. velocity and pressure). Rather, they establish relations among the rates of change.

2.2 COMPUTATIONAL FLUID DYNAMICS

Computational Fluid Dynamics (CFD) is one of the branches of fluid mechanics that uses numerical methods and algorithms to solve and analyze problems that involve fluid flows. The CFD has become an essential tool in solving problems governing the Navier-Stokes equation and the continuity equation or any equation which derived from them.

The fundamental in CFD is how continuous fluid is treated in a discretized fashion on a computer. One method is to discretize the spatial domain into small cells to form a volume mesh or grid, and then apply a suitable algorithm to solve the equations of motion (Euler equations for inviscid, and Navier-Stokes equations for viscous flow). In many instances, other equations are solved simultaneously with the Navier-Stokes equations. These other equations can include those describing species concentration (mass transfer), chemical reactions, heat transfer, etc. More advanced codes allow the simulation of more complex cases involving multi-phase flows (e.g. liquid/gas, solid/gas, liquid/solid), non-Newtonian fluids (such as blood), or chemically reacting flows (such as combustion).

The basic approach in the use of CFD includes preprocessing, simulation, and postprocessing. In preprocessing, the geometry of the problem is defined and the volume occupied by fluid is divided into mesh. During this process, both physical modeling and boundary conditions are defined. Simulation begins after the process and the equations are solve iteratively. Postprocessing is where the postprocessor is used for the analysis and visualization of the resulting solution.

2.3 LATTICE GAS APPROACH

The lattice gas approach (LGA) is a particular type of cellular automaton used to simulate fluid flow. The approach which has been prove to be less popular for the

simulation of fluid flow consists of a lattice in which the sites can take a certain number of different states. In this approach, the various states are particle with certain velocities. The LGA evolution of the simulation is done in discrete time steps.

The processes that are carried out at each time steps consist of a collision step and a propagation step. In the collision step, collisions of particles can change their velocities' direction. Collision rules can be use to analyze or predict the outcome of multiple particles that reach the same site. The collision rules are required to conserve mass and total momentum. In propagation step, particles velocities will determine the direction sites the particles will move to.

The LGA has the advantages of having exact computing without any round-off error due to floating-point precision because of the Boolean logic operation, and that the method has much simple computing for simulation compare to other method. Despite its advantages, the LGA also has the disadvantages that include the lack of Galilean invariance and can be noisy. There is also the difficulty in expanding model for the three dimensional problems where it needs more dimensions for maintaining sufficient symmetric grid. However, the disadvantages are overcome by the lattice Boltzmann method, thus making it the precursor the new alternative method.

2.4 LATTICE BOLTZMANN METHOD

The lattice Boltzmann method (LBM) is a relatively new numerical method for simulating viscous fluid flow. However, the lattice Boltzmann (LB) approach has found success in a host of fluid dynamic related problems and has been attracting interest from the physic and engineering communities as possible alternative approach to solve complex fluid dynamic problems in recent years.

Basically the lattice Boltzmann approach is developed from the lattice gas, but it can also be derived from the Boltzmann equation. Instead of solving the Navier–Stokes equations, the discrete Boltzmann equation is solved to simulate the flow of a Newtonian fluid with collision models such as Bhatnagar-Gross-Krook (BGK). By simulating the interaction of a limited number of particles the viscous flow behavior

emerges automatically from the intrinsic particle streaming and collision processes. The Boltzmann equation is given by

$$f(x + c\Delta t, t + \Delta t) - f(x, t) = \Omega(f), \quad (2.3)$$

in which f is the density distribution function, c is the microscopic velocity, and $\Omega(f)$ is the collision integral. The Bhatnagar-Gross-Krook (BGK) collision model can be define as

$$\Omega(f) = \frac{1}{\tau} (f^{eq} - f), \quad (2.4)$$

with relaxation parameter τ , and equilibrium distribution f^{eq} . Thus, the equation obtain from Eq. (2.3) and Eq. (2.4) is expressed as

$$f(x + c\Delta t, t + \Delta t) - f(x, t) = -\frac{f - f^{eq}}{\tau}, \quad (2.5)$$

which represents the Lattice Boltzmann BGK model. Both equations represent the steaming step and the collision step that occur in the LBM. The illustration of the streaming and collision step is shown in Figure 2.1.

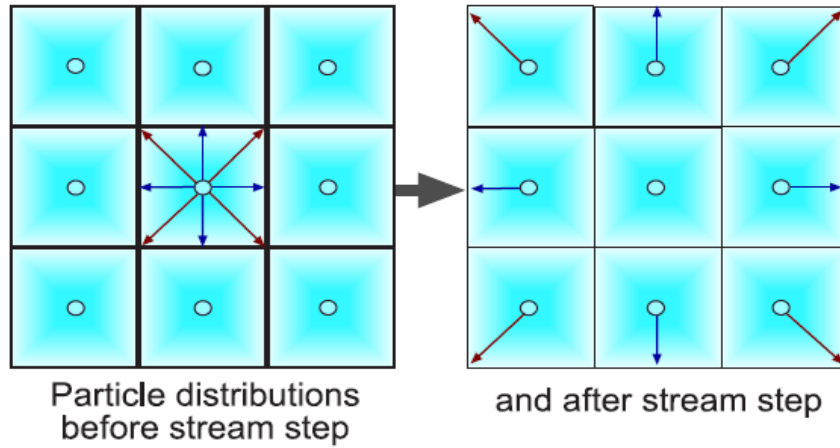


Figure 2.1: Particle distributions before and after stream step

While the LGA models are Boolean ones, the lattice Boltzmann equation models are floating-number counterpart of the corresponding LGA models-where a particle which is represented by a Boolean number in the LGA model is replaced by the single-particle distribution function which represented by real number. Theoretically the lattice Boltzmann equation is independent of the LGA. The lattice Boltzmann equation is a finite-difference form of the continuous Boltzmann equation.

The lattice Boltzmann method has several advantages over other method. One of the advantages is the stability of the approach. In other words, it is easy to implement and parallelize the computation of the basic LBM. Another advantage of a standard LBM is the use of uniform Cartesian grid. Besides, the rule for the approach is explicitly updated and particle distribution information is shown. The method however does have its limitation in terms of the high-mach number flows, where it would be difficult to be used in this method. Other than that, the method also lacks a consistent thermo-hydrodynamic scheme.

Despite its limitation, the LBM has become an ideal tool for fluid flow simulation. It has essentially become one of useful tool in few engineering application which requires fast and semi-automatic integration of complex geometries, grid generation process can be avoided, and short turn-around times in industrial process. It can be use to analyze heat transfer, and turbulence in fluid as well as chemical reaction modeling in chemical engineering. Other LBM application include in biomedical, nanotechnology, and weather prediction.

2.5 LID DRIVEN CAVITY

One of a widely used test case for benchmarking incompressible isothermal flow code can be represented by the lid driven cavity flow. The lid driven cavity problem has simple geometry and it is usually in two-dimensional. The boundary conditions for the problem are simple as well. A standard lid driven cavity problem case consist of fluid contained inside a square cavity with Dirichlet boundary conditions on all side, having

three static walls, and an upper wall that slides at constant speed which set the fluid into motion.

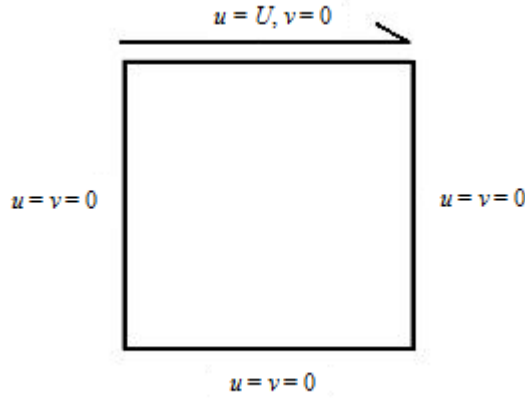


Figure 2.2: Square lid driven cavity with three static and one moving walls

2.6 CUBIC INTERPOLATED PSEUDO PARTICLE

The Cubic Interpolated Pseudo Particle (CIP) scheme was introduced by H. Takewaki, A. Nishiguchi, and T. Yabe et al. (1985) for solving hyperbolic-type equations which proved to be stable and less-diffusive scheme. The method has similar approach to the finite element method (FEM), the boundary element method (BEM), and the particle scheme. Accordingly the scheme may be a bridge between the FEM and the particle schemes. The CIP can be seen as a semi-Lagrangian method due to its advection scheme which employs a Lagrangian invariant solution (T. Yabe, 2001).

To review the CIP method, a simple model equation is used to show the advection equation such as

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = 0 \quad (2.6)$$

where velocity u is a constant value. Since the profile propagates at a velocity u , at constant velocity, the advection equation gives simple translation of function f .

$$f(x_i, t + \Delta t) = f(x_i + u\Delta t, t) \quad (2.7)$$

Given two values of $f(x)$ at two grid points, the profile between this point can be interpolated by the cubic polynomial $f(x) = a^3 + bx^2 + cx + d$. Since the conventional interpolation requires four unknown that need to be solve, the equation such as

$$F_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad (2.8)$$

is differentiate with the spatial variable x . Given an interval (x_{i-1}, x) , at point $x = x_i$, equation

$$F_i(x_i) = d_i = f_i \quad (2.9)$$

were obtained. The gradient at this point:

$$\frac{F_i(x_i)}{dx} = c_i = g_i \quad (2.10)$$

At point $x = x_{i-1}$, equation obtained:

$$F_i(x_{i-1}) = -a_i\Delta x^3 + b_i\Delta x^2 - c_i\Delta x + d_i = f_{i-1} \quad (2.11)$$

The gradient at this point:

$$\frac{dF_i(x_{i-1})}{dx} = 3a_i\Delta x^2 - 2b_i\Delta x + g_i = g_{i-1} \quad (2.12)$$

Substitute Eq. (2.9) and Eq. (2.10) into Eq. (2.12) gives;

$$a_i = \frac{f_i - f_{i-1} - b_i\Delta x^2 - g_i\Delta x}{\Delta x^3} \quad (2.13)$$

Substitute Eq. (2.10) and Eq. (2.13) into Eq. (2.12) gives;

$$b_i = \frac{3(f_{i-1} - f_i)}{\Delta x^2} + \frac{2g_i + g_{i-1}}{\Delta x} \quad (2.14)$$

Substitute Eq. (2.14) into Eq. (2.13) gives;

$$a_i = \frac{g_i + g_{i-1}}{\Delta x^2} + \frac{2(f_i - f_{i-1})}{\Delta x^3} \quad (2.15)$$

Once a_i and b_i are given in terms of f^n and g^n , the value f^{n+1} and g^{n+1} are simply given by shifting the polynomial as in Eq. (2.7). The superscript n is omitted in Eq. (2.14) and Eq. (2.15) because it will be use later as a quantity having a different meaning. Without confusion, a quantity with no superscript can be recognized as to be that at time step n . For a linear advection equation where velocity u is constant, the value at $n+1$ can be determined by substituting Eq. (2.14) and Eq. (2.15) into Eq. (2.11) and Eq. (2.12) which gives;

$$f_i^{n+1} = a_i \xi^3 + b_i \xi^2 + g_i^n \xi + f_i^n \quad (2.16)$$

And

$$g_i^{n+1} = 3a_i \xi^2 + 2b_i \xi + g_i^n \quad (2.17)$$

where,

$$f^{n+1} = F(x - u\Delta t),$$

$$g^{n+1} = \frac{dF(x - u\Delta t)}{dx},$$

$$\xi = -u\Delta t$$

By using the time splitting technique, the one-dimensional scheme can be extent into multidimensional scheme. The Boltzmann equations in the two dimensional can be expressed as

$$\frac{\partial f}{\partial t} + c_x \frac{\partial f}{\partial x} + c_y \frac{\partial f}{\partial y} = -\frac{1}{\tau}(f - f^{eq}) \quad (2.18)$$

By differentiating Eq. (2.18) with respect to x and y will gives

$$\frac{\partial}{\partial t} g_x + c_x \frac{\partial}{\partial x} g_x + c_y \frac{\partial}{\partial y} g_x = -\frac{1}{\tau}(g_x - \frac{\partial}{\partial x} f^{eq}) \quad (2.19)$$

$$\frac{\partial}{\partial t} g_y + c_x \frac{\partial}{\partial x} g_y + c_y \frac{\partial}{\partial y} g_y = -\frac{1}{\tau}(g_y - \frac{\partial}{\partial y} f^{eq}) \quad (2.20)$$

Where

$$\frac{\partial f}{\partial x} = g_x \quad (2.21)$$

$$\frac{\partial f}{\partial y} = g_y \quad (2.22)$$

In the case of CIP method, using cubic polynomial as in Eq. (2.23), the profile between lattice points is interpolated.

$$\begin{aligned} F_{i,j}(x, y) &= A1(y - y_i)^3 + A2(x - x_i)^2 + A3(x - x_i)^2(y - y_i) + A4(x - x_i)^3 \\ &+ B1(y - y_i)^2 + B2(x - x_i)(y - y_i) + B3(x - x_i)^2 \\ &+ C1(y - y_i) + C2(x - x_i) + C3 \end{aligned} \quad (2.23)$$

Eq. (2.23) can also be rewrite as

$$\begin{aligned}
& F_{i,j}(x, y) \\
& = [(A1_{i,j} X + A2_{i,j} Y + A3_{i,j})X + A4_{i,j} Y + \partial_x f_{i,j}]X \\
& + [(A5_{i,j} Y + A6_{i,j} X + A7_{i,j})Y + \partial_y f_{i,j}]Y \\
& + f_{i,j}
\end{aligned} \tag{2.24}$$

where $X=x-x_{i,j}$ and $Y=y-y_{i,j}$

In Eq. (2.24), the coefficient can be expressed as

$$A1_{i,j} = [-2d_i + \partial_x (f_{i+1,j} + f_{i,j})\Delta x] / \Delta x^3 \tag{2.25}$$

$$A2_{i,j} = [A8_{i,j} - \partial_x d_j \Delta x] / \Delta x^2 \Delta y \tag{2.26}$$

$$A3_{i,j} = [3d_i - \partial_x (f_{i+1,j} + 2f_{i,j})\Delta x] / \Delta x^2 \tag{2.27}$$

$$A4_{i,j} = [-A8_{i,j} + \partial_x d_j \Delta x + \partial_y d_i \Delta y] / \Delta x \Delta y \tag{2.28}$$

$$A5_{i,j} = [-2d_j + \partial_y (f_{i,j+1} + f_{i,j})\Delta y] / \Delta y^3 \tag{2.29}$$

$$A6_{i,j} = [A8_{i,j} - \partial_y d_i \Delta y] / \Delta x \Delta y^2 \tag{2.30}$$

$$A7_{i,j} = [3d_j - \partial_y (f_{i,j+1} + 2f_{i,j})\Delta y] / \Delta y^2 \tag{2.31}$$

$$A8_{i,j} = f_{i,j} - f_{i+1,j} - f_{i,j+1} + f_{i+1,j+1} \tag{2.32}$$

where $d_i=f_{i+1,j}-f_{i,j}$ and $d_j=f_{i,j+1}-f_{i,j}$

The advection terms for BGK-LB equation is solved using CIP method, the non-advection terms for BGK-LB is solved using Rungge Kutta Method.

Advection;

$$\frac{\partial f}{\partial t} + c_x \frac{\partial f}{\partial x} + c_y \frac{\partial f}{\partial y} = 0 \quad (2.34)$$

$$\frac{\partial g_x}{\partial t} + c_x \frac{\partial g_x}{\partial x} + c_y \frac{\partial g_x}{\partial y} = 0 \quad (2.35)$$

$$\frac{\partial g_y}{\partial t} + c_x \frac{\partial g_y}{\partial x} + c_y \frac{\partial g_y}{\partial y} = 0 \quad (2.36)$$

Non-advection;

$$\frac{\partial f}{\partial t} = -\frac{1}{\tau}(f - f^{eq}) \quad (2.37)$$

$$\frac{\partial}{\partial t} g_x = -\frac{1}{\tau}(g_x - \frac{\partial}{\partial x} f^{eq}) \quad (2.38)$$

$$\frac{\partial}{\partial t} g_y = -\frac{1}{\tau}(g_y - \frac{\partial}{\partial y} f^{eq}) \quad (2.39)$$

In CIP-LBM, the algorithm consists of advection process and collision process. The initial value of distribution f are specified at each grid point. The profile of distribution function of two neighboring nodes is constructed according to Eq. (2.24). The advection process is applied on advection terms to obtain the distribution value at new time step. The collision process is applied on non-advection term base on Eqs. (2.37-2.39).

Above equation is for negative value of u and v . All variables above must be change depending on the signs of u and v . Below shows the signs changes of the variables;

For $u \geq 0$

$$i + 1 \rightarrow i - 1$$

$$\Delta x \rightarrow -\Delta x$$

For $v \geq 0$

$$j + 1 \rightarrow j - 1$$

$$\Delta y \rightarrow -\Delta y$$

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

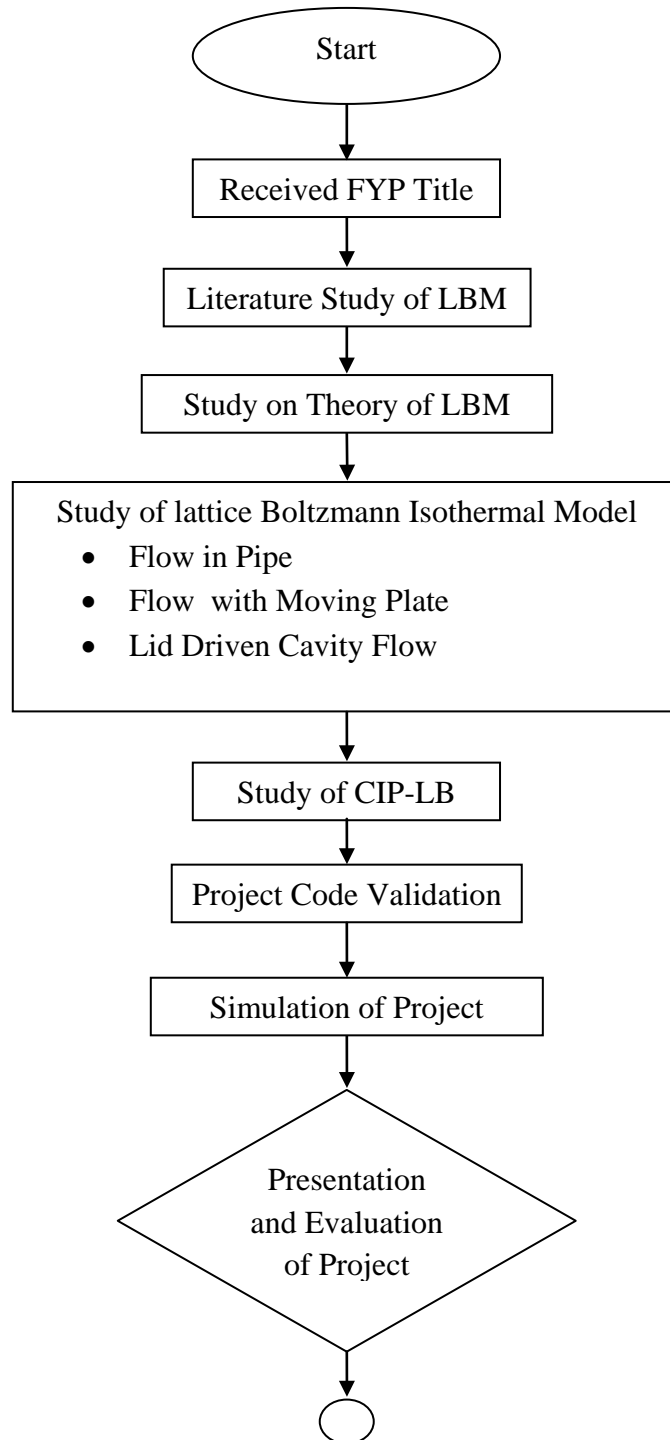
As mentioned in the first chapter, the objective of this project is to couple LBM with CIP to develop a CIP-LBM scheme for simulation of cavity flow. In this chapter, the project method is discussed in order to get the needed results. It is important to discuss the steps as the project requires good understanding about the topic and requires other knowledge as well. Validation of the conventional LBM code is done by using Poiseuille flow and lid driven cavity flow. This part of the paper is also crucial to ensure the flow of the project progress is smooth.

At the beginning of the project, meeting and discussion with project supervisor is carried out. The meeting is important to gain an insight about the project and to understand the needed task that should be done. Several meeting and discussions with project supervisor are also done once in a while throughout the time of project. The introduction about the project was done by project supervisor to make sure some needed information is given as a head start for the project.

Several steps and task such as literature study of LBM, the study on theory of LBM, study on lattice Boltzmann isothermal model, study of CIP-LBM and code validation of some cases. Since this project involves both LBM and CIP, thus literature study as well as theoretical study is important. As the project is about simulation, knowledge in programming is needed to run the simulations. The programming software, COMPACT VISUAL FORTRAN 6, is used for the code validation. The visualization software, AVS Express 7.2, was also to visualize the data from the

simulation. In term coding, applying the numerical method to the flow problem is simple and straightforward.

3.2 FLOW CHART



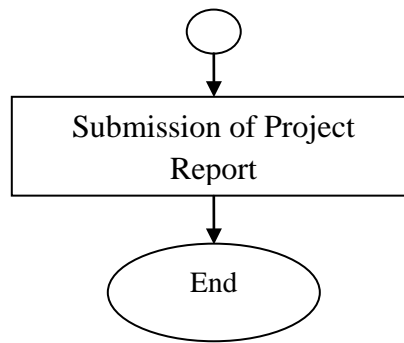


Figure 3.1: The flow process of study

3.3 LITERATURE STUDY OF LATTICE BOLTZMANN METHOD

One of the early steps to begin the project is to understand the concept of lattice Boltzmann method. In addition to gain information regarding to how it was introduced, functions, advantages and disadvantages, literature study on lattice Boltzmann are carried out. The lattice Boltzmann would be compared to other existing method such as CFD and its previous predecessor LGA. Thus, literature study about Navier-Stokes equation, CFD and other approach were also carried out. Meeting and discussions with project supervisor are done once in a while throughout the time of project.

3.4 STUDY ON THE THEORY OF LATTICE BOLTZMANN

In order to understand the derivatives of lattice Boltzmann equation, theory concerning lattice Boltzmann are studied which include isothermal LBM. Since the scope of the project does not involve heat properties, thermal LBM can be neglected. Boundary conditions of LBM for various test cases were also studied. Other test cases are used as research to theory of LBM. Various aspect of LBM can be look into such as discretization of the Boltzmann equation, simulation using LBM in fluids related cases, and properties of lattice Boltzmann.

3.5 STUDY OF LATTICE BOLTZMANN ISOTHERMAL MODEL

For this section, the lattice Boltzmann isothermal model is studied by investigating several test cases. In this project, the lattice Boltzmann method used is focus on isothermal model since it does not involve any temperature and heat properties or heat transfer. Thus, studies on lattice Boltzmann thermal model are neglected. The test case that will be investigated is the flow in pipe.

3.5.1 Flow in Pipe

For the isothermal model, the flow profile in pipe is investigated by using the effect of different value of Reynolds number. Using existing valid code, simulation of flow is done by adjusting some value to the code. The type of flow that is use to demonstrate the pipe flow is known as Poiseuille flow. The Poisuville flow program code was provided by project supervisor and certain value were edited.

3.5.2 Flow with Moving Plate

In this section, flow with moving plate is studied as part of the required knowledge to be use for the project simulation. The project is concerned to simulation of lid driven cavity flow, which boundary conditions is known to be three static wall and one moving upper wall. Thus, various cases which come to flow with moving plate are investigated. Such flow is to be investigated by validation of Coutte flow code.

3.6 STUDY ON LID-DRIVEN CAVITY FLOW

Various cases regarding to lid driven cavity flow is studied in this section. The reason for using lid driven cavity flow is to observe the vortex inside the lid driven cavity. The vortex observed is tested using different Reynolds number. The Reynolds numbers used are 100, 400, and 1000. Figure 3.2 shows the illustration for deep lid driven cavity flow. For lid-driven cavity, the aspect ratio is given as $K = (H/L)$, where cavity depth and cavity width are represented by H and L respectively. The aspect ratio for deep cavity is $K > 1$.

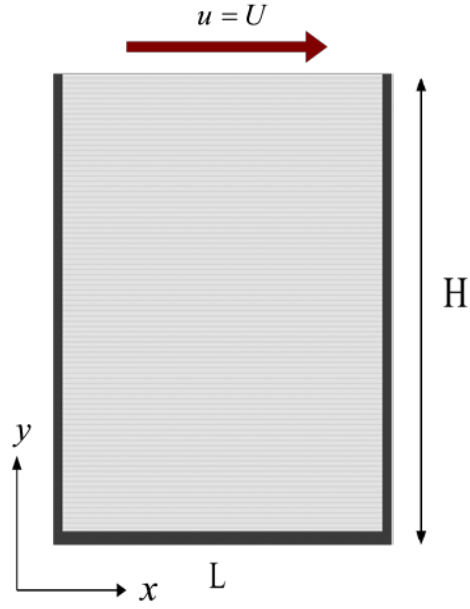


Figure 3.2: Deep lid driven cavity with aspect ratio $K = H/L$

3.6.1 Initial and Boundary Conditions

As been stated, the lid driven cavity problem used the Dirichlet boundary conditions in which left and right walls are static and having $u = v = 0$ at any given direction. The same also goes for the bottom wall which has $u = v = 0$. However, in previous study by Patil et al. (2006) for simulation of lid-driven flow in cavities using LBM, it mentions the physical boundary conditions may be implemented in some different way within the LBE code.

3.6.2 Stationary Wall Conditions

The ‘complete on-grid bounce back’ boundary condition model is implemented on the three stationary walls since it is numerically straightforward. On the flat boundary, the model gives a reasonable result. This however could not be said for an inclined flat walls or curved ones, as it could not shows a result accurate enough when involving such problems.

3.6.3 Moving Wall Conditions

For the moving lid, the lid velocity chosen is set to $U = 0.1$ for all Reynolds number in simulation of deep lid-driven cavity.

3.7 STUDY OF CIP-LBM

In chapter 2, the Cubic Interpolated Pseudo Particle (CIP) method was reviewed together with the lattice Boltzmann method (LBM). The CIP-LBM is studied to understand the modification that concerned to develop CIP-LBM scheme for the project simulation.

3.8 PROJECT CODE VALIDATION

3.8.1 Original LBM Algorithm

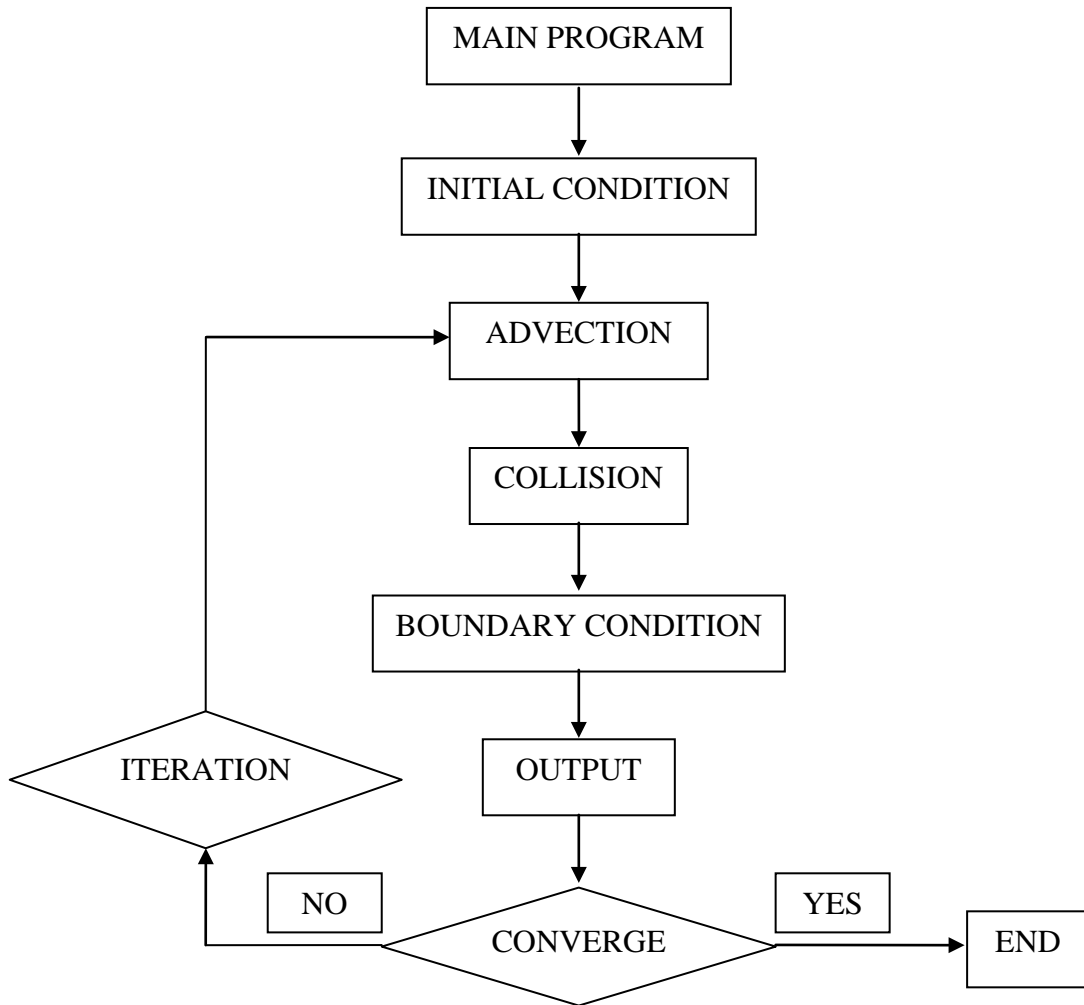


Figure 3.3: Flow chart for the original LBM algorithm

As shown in Figure 3.3, the LBM algorithm consists of two processes; advection process and collision process. Initial values of density distribution f are specified at each grid point. Then the system goes through the following steps:

- Advection term is solved by applying the density distribution function streaming process.
- Collision process is solved by BGK collision model.

- The boundary conditions are defined based on bounce back boundary conditions.

3.8.2 CIP-LBM Algorithm

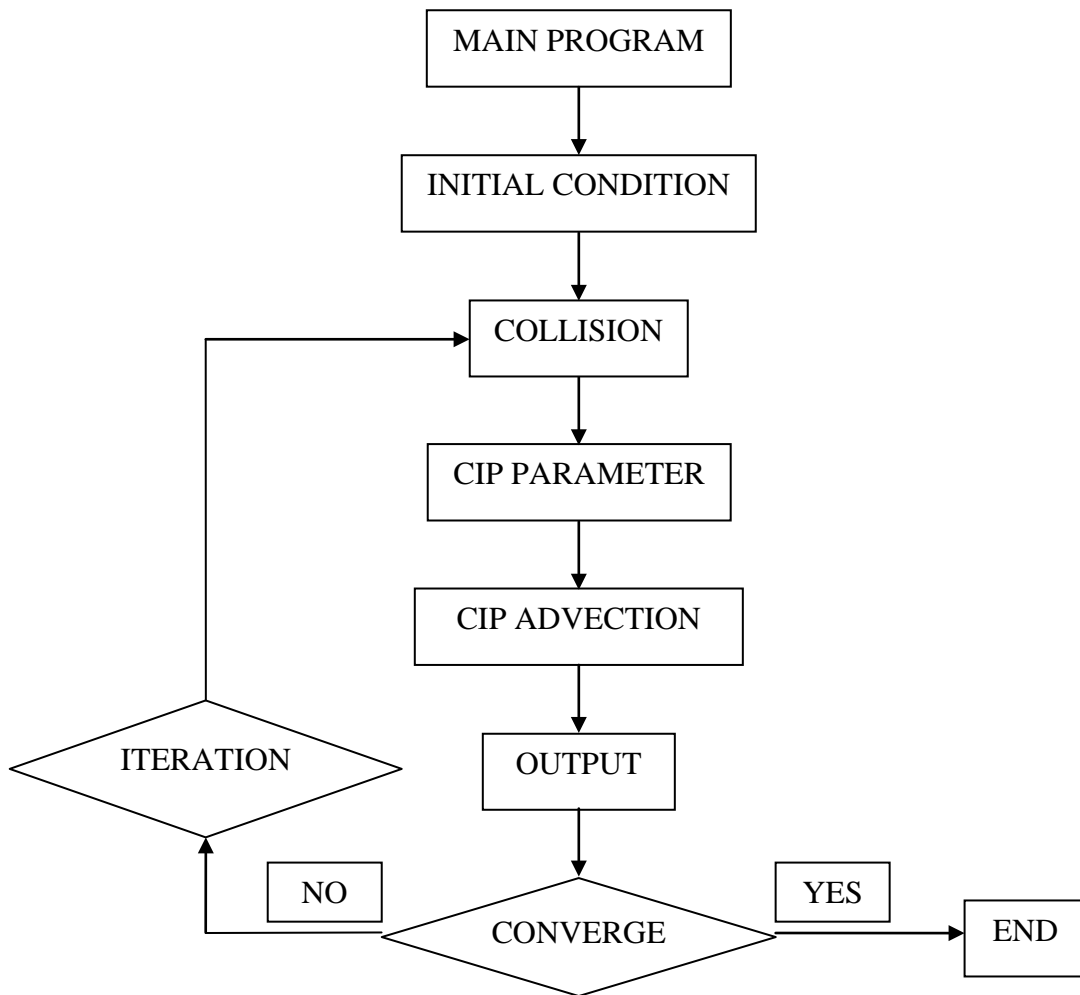


Figure 3.4: Flow chart for CIP-LBM algorithm

As shown in Figure 3.4, the CIP-LBM algorithm also consists of two processes; advection process and collision process. Initial values of density distribution f are specified at each grid point. However, the system is different from the original LBM algorithm in term of the steps.

- The non-advection term is solved by applying the collision process.

- The profile of distribution function between two neighboring nodes is constructed afterwards. Applying the advection process on advection terms, the distribution function value at new time step can be obtained.

3.9 PROJECT SIMULATION

The project is simulated using CIP-LBM code which was written by Azwadi (2007). The variables are also based on previous studies by Azwadi (2007) and Rosdzimin (2008). The project is run on COMPACT VISUAL FORTRAN 6 software.

The data for the simulation is visualized using AVS Express 7.2 software. Figure 3.5 shows the AVS Express 7.2 application that is used.

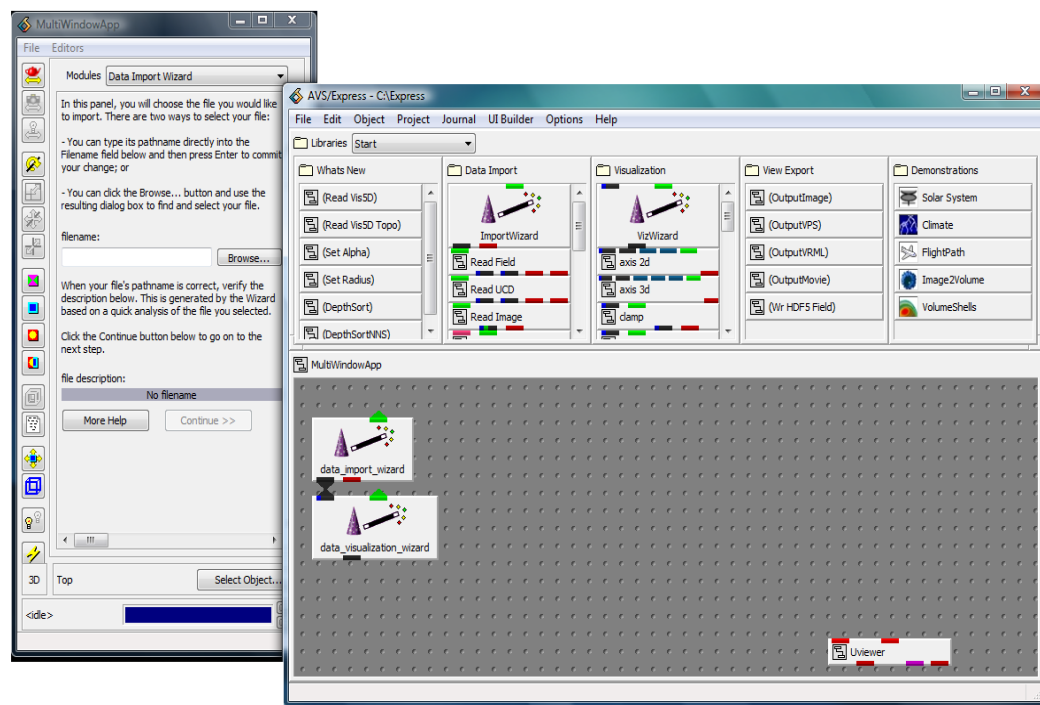


Figure 3.5: The visualization software, AVS Express 7.2, application menu.

CHAPTER 4

RESULT AND DISCUSSION

4.1 INTRODUCTION

In this chapter, result analysis for original isothermal LBM code validation will be discuss. The study for the original LBM code consist of Poiseuille flow analysis which similar to the flow inside a pipe or duct. The original was also done for Coutte flow. Later, the result for the main case study regarding deep lid-driven cavity flow will also be analyzed. The streamlines for the deep cavities will be shown. The structure for the vortices, effects of different Reynolds number and aspect ratio used in present simulation for deep cavities will be discussed. Result from the present deep lid-driven cavity flow are to be compared with previous simulation for deep cavities using the original lattice Boltzmann that were carried out by Patil et al.(2006). The physical details of streamlines of both studies will be investigated.

4.2 VALIDATION OF ORIGINAL LBM CODE RESULT

In this section, original LBM code validation is carried out for the study of Poiseuille flow to observe the dimensionless velocity profile in pipe and also Coutte flow for study of flow with moving plate using LBM. The Poiseuille flow and Coutte flow are chosen as one the benchmark for isothermal Lattice Boltzmann code validation due to the availability of its analytical solution. All variables used are based on previous study by Azwadi (2007).

4.2.1 Poiseuille Flow Analysis

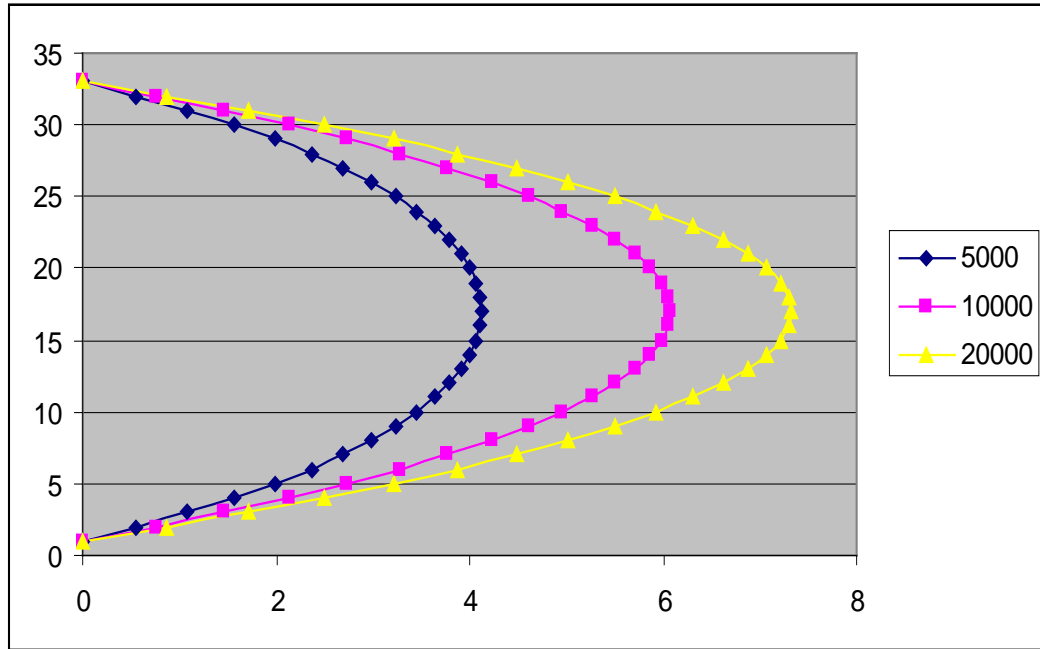


Figure 4.1: Flow in pipe based on different dimensionless velocity value

Figure 4.1 shows the representation of dimensionless velocity profile in pipe using LBM. The flow for this benchmark case is driven by the pressure gradient. In this case however, different dimensionless velocity number is used to validate the Lattice Boltzmann code for the Poiseuille flow. The velocity value are set to 5000, 10000, and 20000 in LBM unit. Based on Figure 4.1, the results are shown for the inlet of the pipe or duct velocity profile which increases according to the value used. The boundary conditions set are non-slip bounce back boundary condition for the solid top and bottom wall, as well the periodic boundary condition for the horizontal fluid flow.

4.2.2 Coutte Flow Analysis

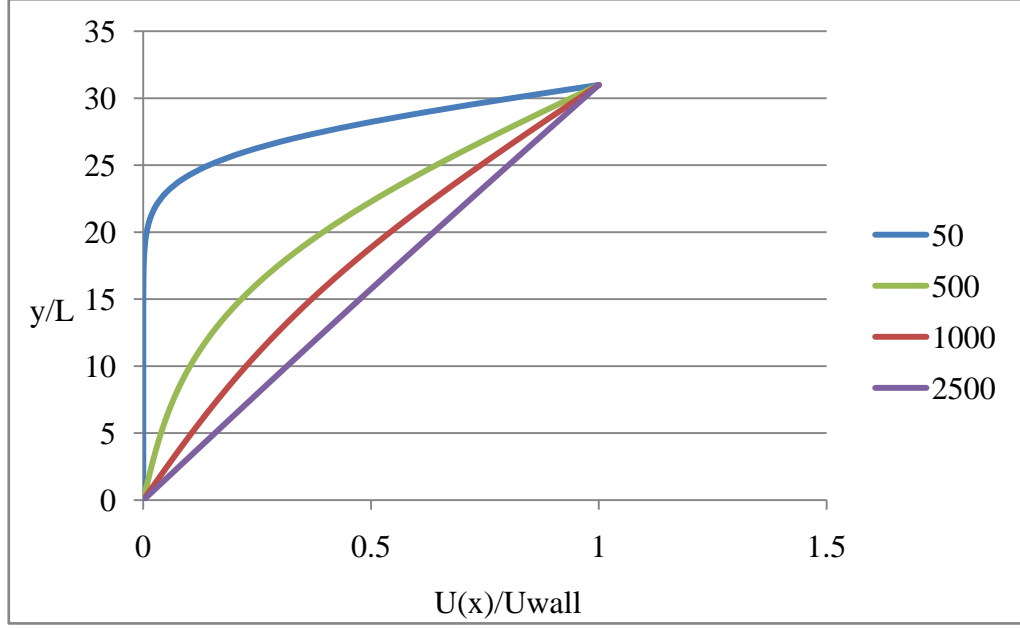


Figure 4.2: Flow velocity of Coutte flow using lattice Boltzmann method

The Coutte flow is used to demonstrate in the space between two parallel plates, specifically for laminar flow. The code validation for Coutte flow had been carried out regarding to flow with moving plate. In Coutte flow, the top plate is moved with constant velocity while the bottom plate is at stationary.

In Figure 4.2, the flow velocities of the Coutte flow are shown for various dimensionless numbers which can be represented by the Reynolds number. From the figure, the flow reaches the steady state at high Reynolds number with the velocity increase linearly zero from at bottom to U at the top plate.

For the initial conditions, velocity is set to zero everywhere except for the top boundary. For the boundary conditions, the non-slip bounce back is implemented for both solid top and bottom wall. The periodic boundary condition is implemented for the fluid flow in x -direction.

4.3 RESULTS FOR DEEP LID-DRIVEN CAVITY FLOW

In this section, the result of the main case study which is for rectangular, deep lid-driven cavities will be presented and discussed. The Reynolds number used for these series simulations are set to $Re = 100, 400, 1000$, together with various value of aspect ratio K . The Reynolds number can define by lid velocity U and the width of the cavity L given by;

$$Re = \frac{UL}{\nu}$$

With aspect ratio, K given by;

$$K = \frac{H}{L}$$

where H is the cavity depth and L is the length. Thus a rectangular cavity specifically main interest of this case which is for deep lid-driven cavities, the aspect ratio would be $K > 1$. The test run are all performed under lid-velocity $U = 0.1$.

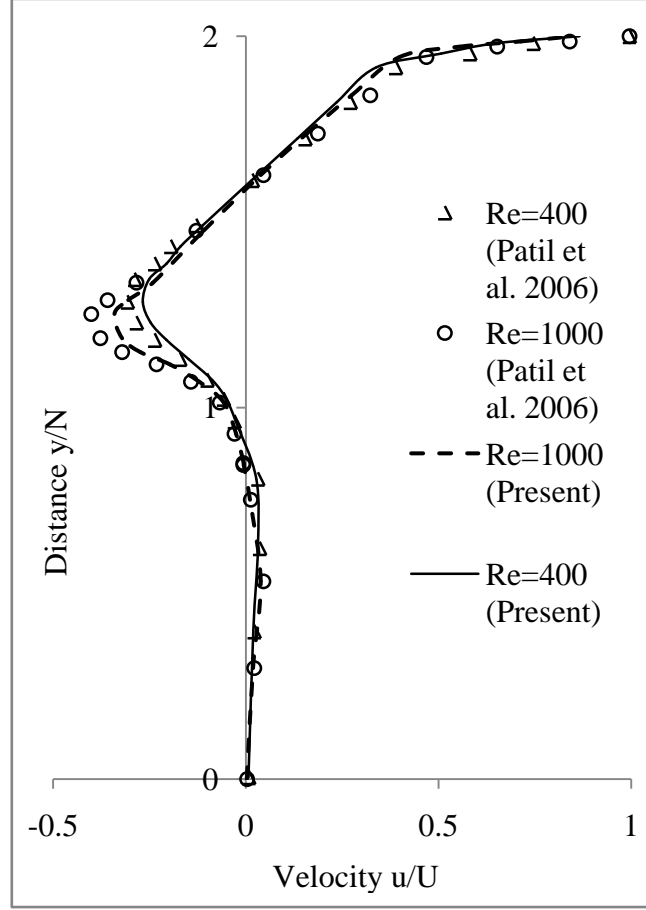


Figure 4.3: Comparison of the x -component of velocity, u , at the mid-plane with result by Patil et al. (2006), for $K = 2$.

Shown in Figure 4.3 is comparison at the mid-plane velocity profile from the present CIP-LBM with previous original LBM simulation results by Patil (2006) for the deep lid-driven cavities with aspect ratio $K = 2$. The selected value of $Re = 400$ and $Re = 1000$ are used for the simulations. Based on Figure 4.3, the present CIP-LBM results are in good agreement with previous study using original LBM by Patil et al. (2006) for both Reynolds numbers. However, by using CIP-LBM, the meshing used is less than compared to the meshing used in original LBM about 50%.

Simulations of deep cavities for aspect ratio $K = 1.5$ was done with $Re = 100$, 400, and 1000 been selected. For these test, the mesh of 40×60 is employed for every conditions. Shown in the three Figures 4.4(a)-(c) are streamlines of the deep cavities for $K = 1.5$, which correspond to $Re = 100$, 400, 1000, respectively. By observing the

figure, it is seen that a series of successive of counter-rotating vortices are formed below the moving lid. With increasing Reynolds number, the center of primary vortex begins to move downward and will remain at almost a constant depth below the top lid after it reach value of $Re = 1000$ and beyond. A previous study by Patil et al. (2006) using original LBM stated that the center of primary vortex remains at constant depth of $l_y \sim 0.43$ for both $K = 1.5$ and 4 . It is also noted that the flow for square cavity shows similar behavior where the center of the primary vortex reaches the geometric center of the cavity.

For deep cavity, the center of primary vortex does not reach the geometric center of the cavity due to the formation of second primary vortex at higher Reynolds number. At low Reynolds numbers, two stationary corner eddies which also known as “Moffatt eddies” can be seen at the bottom of the cavity as shown in Figure 4.4. The simulations from previous study using original LBM by Patil et al. (2006) is also included for comparison with present simulations using CIP-LBM which will be discuss later.

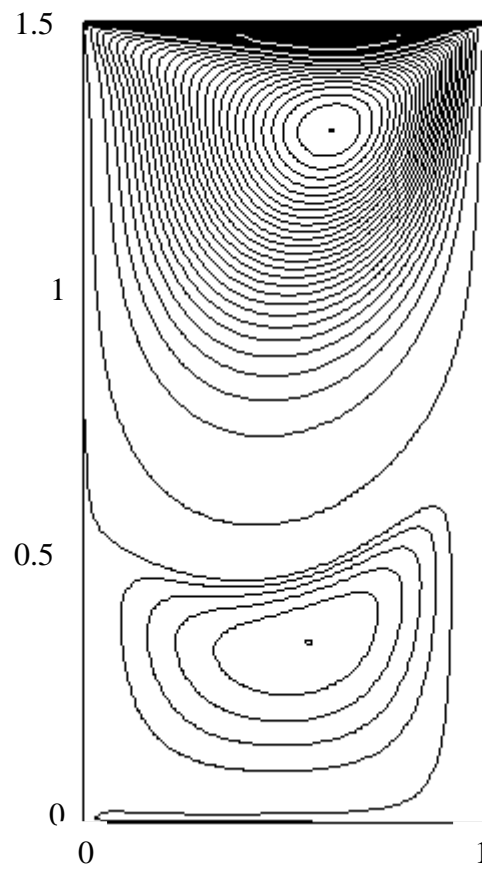


Figure 4.4(a): Streamline of deep lid-driven cavity for $K = 1.5$; $\text{Re} = 100$.

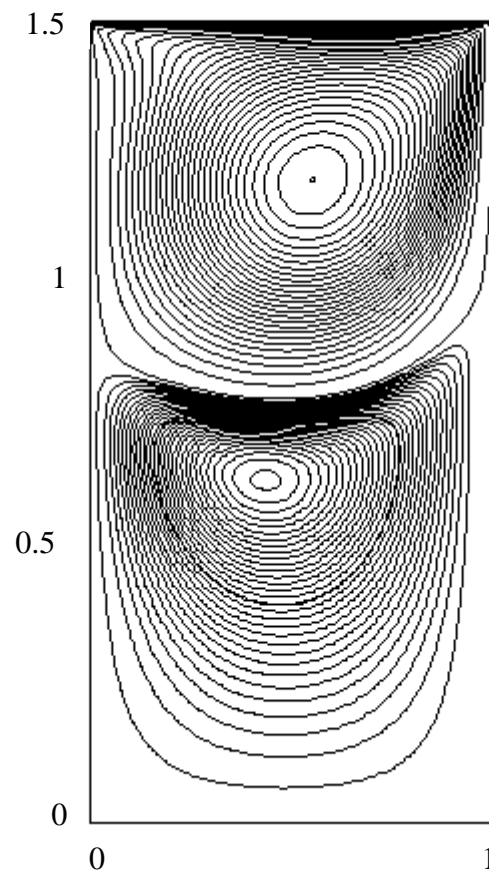


Figure 4.4(b): Streamline of deep-lid driven cavity for $K = 1.5$; $\text{Re} = 400$.

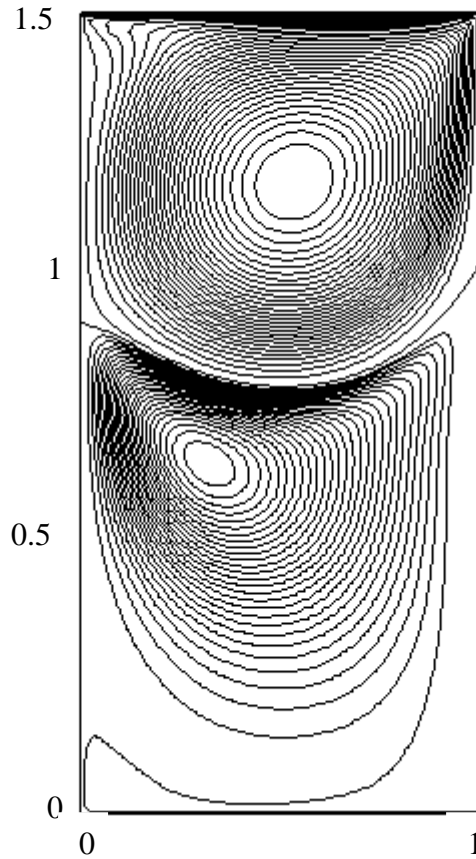


Figure 4.4(c): Streamline of deep lid-driven cavity for $K = 1.5$; $Re = 1000$.

4.4 STRUCTURE OF VORTICES

As discussed earlier, simulations involving the lid-driven cavity flow for any geometry have common similarity which is the formation of vortices. The vortices will produce two kinds of vortices in the terms of primary and corner vortices depending on the Reynolds number.

4.4.1 Structure of Primary Vortices

Primary vortices for lid-driven cavities are formed below the moving lid. The number of primary vortices formed is influenced by the aspect ratio of the cavity. For cavities having aspect ratios $K \leq 1$, single primary vortex is usually formed. In the case of deep cavities, where aspect ratio $K > K_{cr}$, a train of counter-rotating primary vortices

is formed as shown in Figure 4.4(a)-(c). It can be seen in Figure 4.4(a)-(c) that, two primary vortices are formed with the first primary vortex formed directly below the moving lid and the second primary vortex is located below the first primary vortex.

For $Re = 100$ as in Figure 4.4(a), the center of first primary vortex is located near the lid. The second primary vortex can be seen begin to develop. The second primary vortex is the result of two corner vortices that have coalesced together. As Reynolds number increases, the structure of primary vortices will also undergo changes. In Figure 4.4(b), the lid-driven cavity is under the effect of $Re = 400$. The first primary vortex appears to shift downward situated at one third of cavity from the top moving lid. This is caused by the decreased viscous forces at higher Reynolds number. The center of second primary vortex can be located very close to the mid-plane. Based on Figure 4.4(c) which is for $Re = 1000$, the center of the first primary vortex remains at almost constant depth. As for the second primary vortex, it can be seen to be drifting towards the left wall.

Another observation can also be made regarding to the size of the primary vortices. As Reynolds number increases, the size of the primary vortices can be seen to be shrinking. The size of the first primary vortex shrinks at Reynolds number of 100 to 400, whereas the second primary vortex shrinks at Reynolds number of 400 to 1000. The effect of shrinking is caused by the decreased viscous forces at higher Reynolds number.

4.4.2 Structure of Corner Vortices

The corner vortices are usually occurred at the bottom of the cavity. When aspect ratio attained certain critical value, the corner vortices would coalesce together resulting to the formation of the second primary as shown in Figure 4.4(a). Shown in Figure 4.5 are the streamlines and vorticity contours carried out by Patil et al. (2006) for deep cavity flow. From the Figure 4.5(a), it can be seen that there are two visible corner vortices formed at low Reynolds number which is $Re = 50$. When Reynolds number increases to $Re = 3200$, the corner vortices can be clearly seen again at the bottom of the

cavity. As for the corner vortices asymmetry, it can be observed that the right side corner vortex is much bigger compared to the left side of the corner vortex.

4.4.3 Comparison of Physical Details

The details of the streamlines for present simulations of deep cavity flow are compared with streamlines from previous simulations of deep cavity flow done by Patil et al. (2006). The streamlines however, are compared for deep cavities with aspect ratio $K = 1.5$ as shown in Figure 4.4 and 4.5. In the discussion earlier, comparison of the x -component of velocity, u , at the mid-plane with result by Patil et al. (2006) for $K = 2$ were discussed.

From the streamlines shown, the simulations using CIP-LBM for deep cavities in Figure 4.4(a)-(c) are qualitatively in good agreement with previous study in Figure 4.5(a)-(d). In Figure 4.5(a), a single primary vortex is formed accompanied by two corner vortices. The corner vortices can be clearly seen separated from each other. As Reynolds number increases, the two corner vortices merged to form the second primary vortex in Figure 4.4(a). In both Figure 4.4(b) and Figure 4.5(b), the center of primary vortex is seen to be in similar position and no corner visible. Figure 4.4(c) and Figure 4.5(c) also shows the center of second primary vortex drifts towards the left wall. At higher Reynolds number such as 3200, the corner vortices are clearly visible to have formed at the bottom of the cavity.

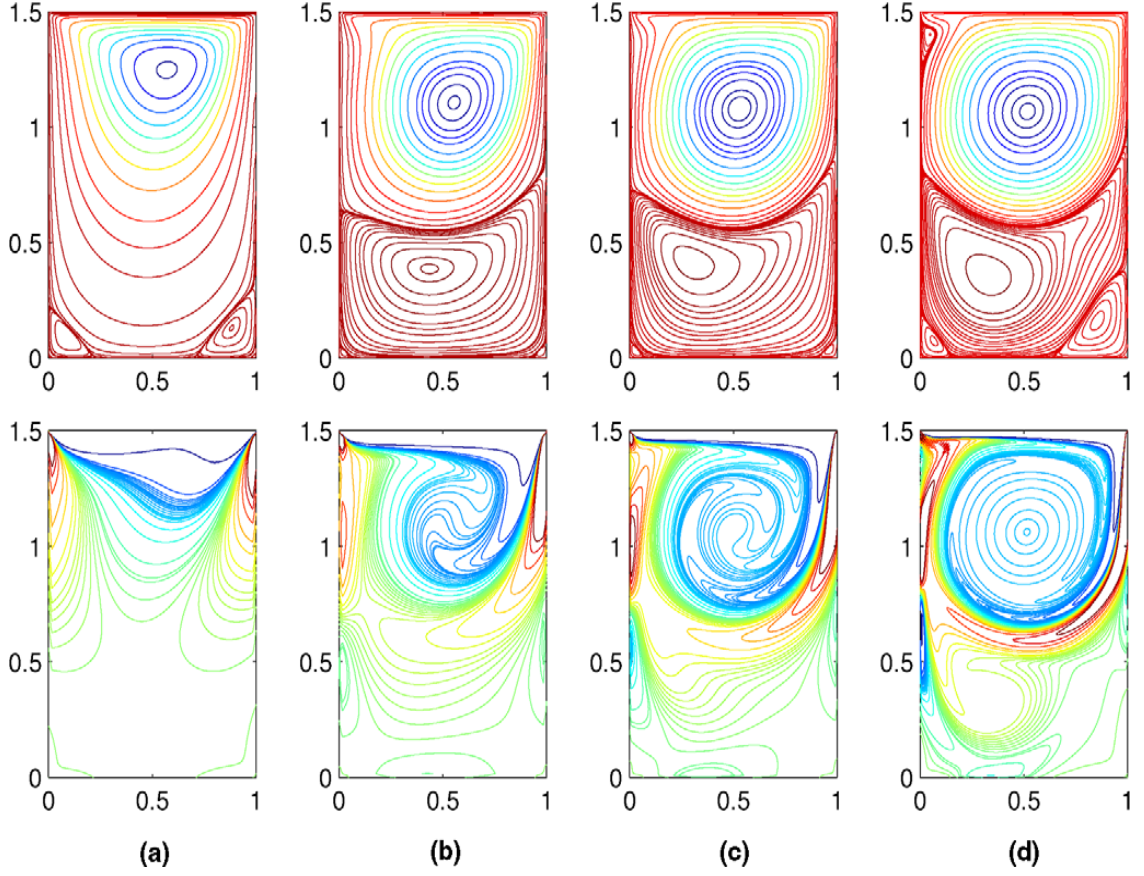


Figure 4.5: Streamlines (top row) and vorticity (bottom row) contours for deep-cavity flow by Patil et al. (2006) with aspect ratio $K = 1.5$. $Re =$ (a) 50, (b) 400, (c) 1000, (d) 3200.

4.5 VELOCITY PROFILE

Analysis of maximum horizontal velocity and maximum vertical velocity are done for the study. Details of the velocity profile in both axes are shown in Figure 4.6 and Figure 4.7.

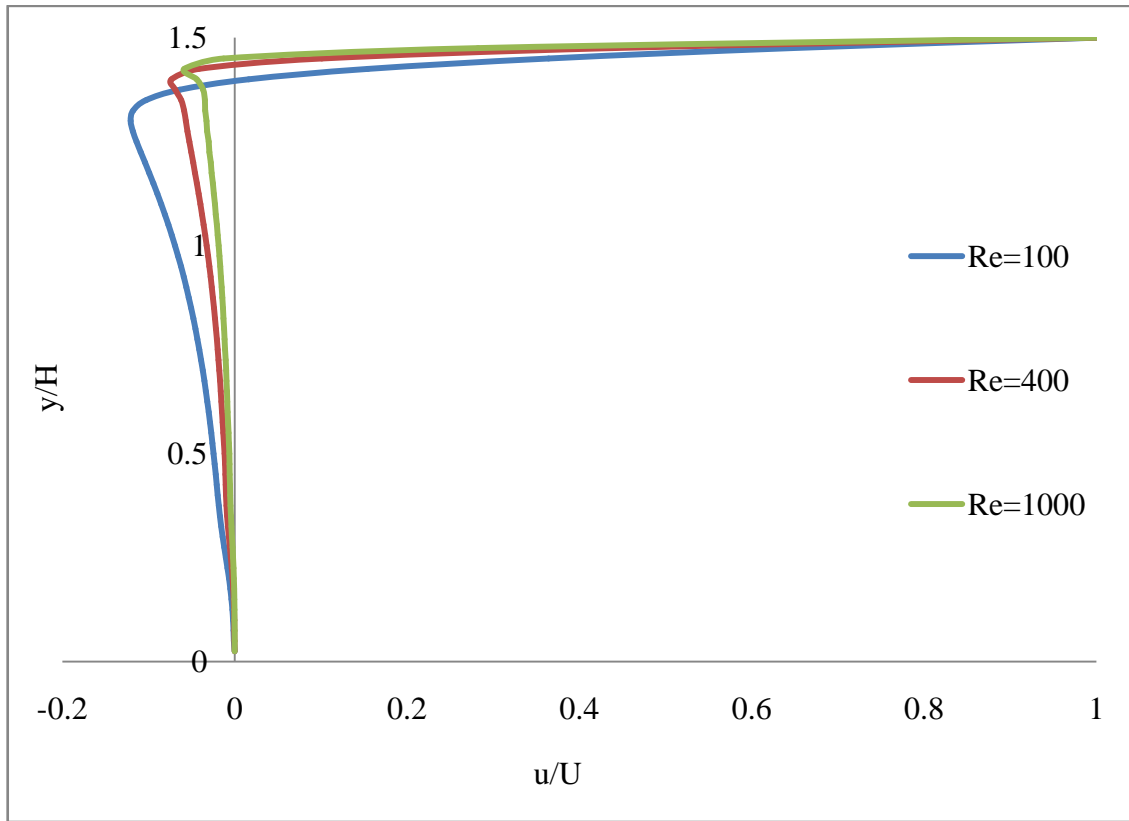


Figure 4.6: Profile of x -component velocity, u , through mid-plane at $Re = 100, 400$, and 1000 .

Figure 4.6 shown is for profile of u velocity at the vertical center of geometry. Based on the figure, the profile shows similar trend for x -component u velocity for each Reynolds number.

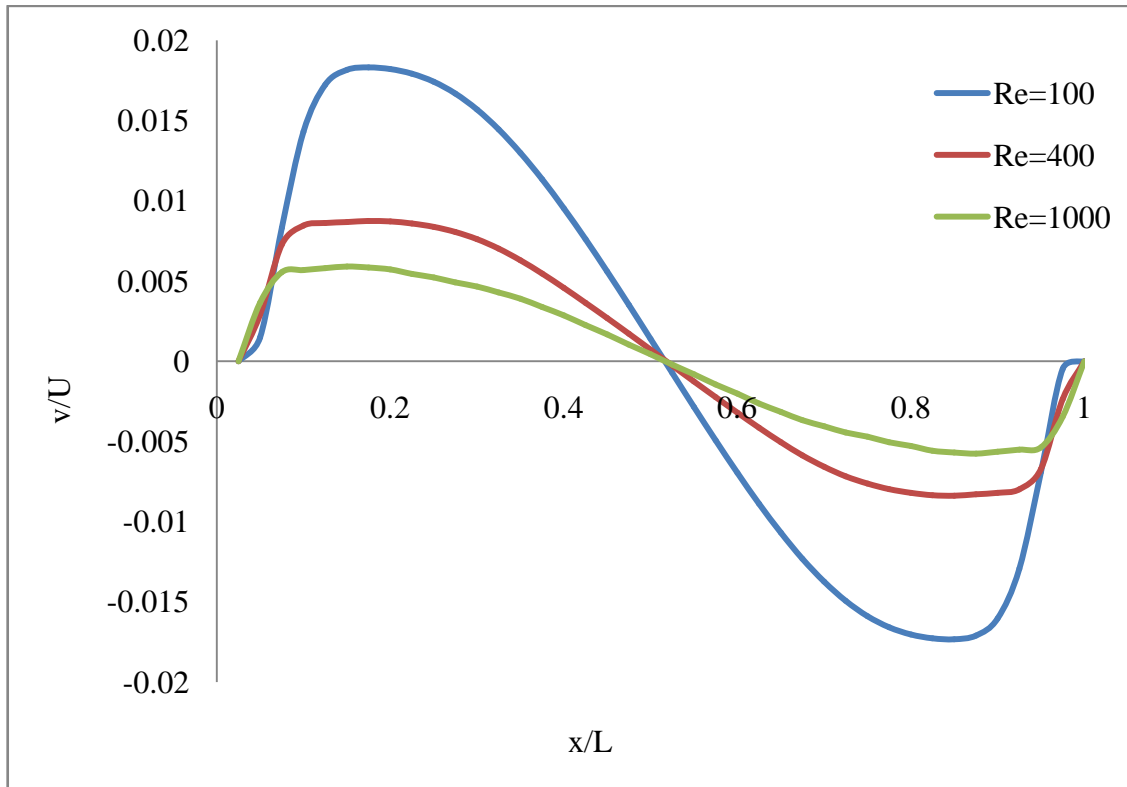


Figure 4.7: Profile of y-component velocity, v , through mid-plane at $Re = 100, 400$, and 1000 .

Figure 4.7 shown is for profile of v velocity at the horizontal center of geometry. Based on the figure, the profile also shows similar trend for the y-component v velocity for each Reynolds number.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSIONS

In the first chapter, background for the study is introduced regarding to fluid dynamics. Constant development of computer technologies has provided solution to fluid problems through computational or numerical method, with example of one of the essential tools is the Computational Fluid Dynamics (CFD). Another class of computational technique known as lattice Boltzmann method is also been introduced as well as the problem regarding to the conventional lattice Boltzmann method. Using the deep lid-driven cavity as the standard benchmark problem, the lid-driven flow for deep cavities was investigated by using CIP-LBM simulations.

The second chapter is introduces to the theory of LBM and its development from the literature review. In literature review, the advantages and disadvantages of several computational techniques and the development along with LBM are discussed. In this chapter, the lid-driven cavity was introduced. The initial condition and boundary condition for lid-driven cavity was discussed. The theory of CIP was also discussed in the same chapter. The method was reviewed and used to solve the one dimensional advection equation as well as the extended two dimensional advection equations.

Chapter three explained the methodology involved for the simulations. The algorithm used for the simulation is mentioned for CIP-LBM scheme. This chapter discussed the steps or flow of the study and the required tools used for the simulation.

Results for the simulation are shown in chapter four. The numerical simulation for Poiseuille flow using the isothermal lattice Boltzmann model was performed and the results are discussed. In the same chapter, the results for the main case study, which is simulation for deep lid-driven cavity using CIP-LBM scheme, are shown and discussed. The streamlines and velocity profiles are shown from the simulation.

Based on the results of the simulation, the physical detail of CIP-LBM scheme for the simulated deep lid shows qualitatively in good agreement with original LBM. The meshing used in CIP-LBM scheme for all Reynolds number is less than the original LBM because of the third order spatial accuracy. It is also capable to capture the velocity profile near the wall for high Reynolds number. The flow structure for the primary vortex below the top moving lid does not affected much by the cavity depth, but shows drastic changes with Reynolds number.

The simulations for the deep lid-driven cavity have been produced and were well captured. The results were obtained and simulation had been run on the specified scope. Overall, the objective of the study has been well achieved.

5.2 RECOMMENDATIONS FOR FUTURE STUDIES

Based on the finding in the present, here are some recommendations for improvement in future studies:

- a) In order to fully understand the present study, a simulation for shallow lid-driven cavity flow using CIP-LBM scheme should be done and compare with the CIP-LBM scheme simulation of deep lid-driven cavity.
- b) Extend the simulation of present study of using CIP-LBM scheme to three dimensional flows.

REFERENCES

1. Begum R, & Basit M.A (2008). Lattice Boltzmann Method and its Applications to Fluid Flow Problems. *European Journal of Scientific Research*, Vol.22 No.2 (2008), pp 216-231.
2. Durbin P.A, & Medic G (2007). *Fluid Dynamics with a Computational Perspective*. New York: Cambridge University Press.
3. Gongwen Peng, Haowen Xi, Comer Duncan, & So-Hsiang Chou (1999). Finite volume scheme for the lattice Boltzmann method on unstructured meshes. *Physical Review E*, Volume 59, Number 4.
4. Haowen Xi, Gongwen Peng, & So Hsiang-Chou (1999). Finite-volume lattice Boltzmann method. *Physical Review E*, Volume 59, Number 5.
5. Mercan H, & Atalik K (2009). Vortex formation in lid-driven arc-shape cavity flows at high Reynolds numbers. *European Journal of Mechanics B/Fluids* 28 (2009) 61-71.
6. Nor Azwadi Bin Che Sidik (2007). The development of simplified thermal lattice Boltzmann models for the simulation of thermal fluid flow problems. Doctor of philosophy. Keio University: Japan.
7. Mohd Rosdzimin Bin Abdul Rahman (2008). Cubic Interpolated Pseudo-Particle (CIP)-lattice Boltzmann scheme for the simulation of natural convection heat transfer. Master of Engineering (Mechanical). Universiti Teknologi Malaysia.
8. Patil D.V, Lakshmisha K.N, Rogg B (2006). Lattice Boltzmann simulation of lid-driven flow in deep cavities. *Computers & Fluids* 35 (2006) 1116-1125.
9. Takashi Abe (1997). Derivation of the Lattice Boltzmann Method by Means of the Discrete Ordinate Method for the Boltzmann Equation. *Journal of Computational Physics* 131, 241-246 (1997).

10. Takewaki H, Nishiguchi H, Yabe T (1985). Cubic Interpolated Pseudo-particle Method (CIP) for solving hyperbolic-type equations. *Journal of Computational Physics*, Vol. 61 (1985), 2.
11. Tolke J, Krafczyk M, & Rank E (2002). A Multigrid-Solver for the Discrete Boltzmann Equation. *Journal of Statistical Physics*, Vol. 107, Nos. 1/2.
12. Wagner W (1995). Approximation of the Boltzmann Equation by Discrete Velocity Models. *Journal of Statistical Physics*, Vol. 78, Nos. 5/6.
13. Xiaoyi He, & Li-Shi Luo (1997). A priori deviation of the lattice Boltzmann equation. *Physical Review E*, Volume 55, Number 6.
14. Xiaoyi He, & Li-Shi Luo (1997). Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. *Physical Review E*, Volume 56, Number 6.
15. Yabe T (1991). A universal cubic interpolation solver for compressible and incompressible fluids. *Shock Waves* (1991) 1: 187-195.
16. Zhaoli Guo & T.S. Zhao (2003). Explicit finite-difference lattice Boltzmann method for curvilinear coordinates. *Physical Review E* 67, 066709 (2003).
17. Ziegler D.P (1993). Boundary Conditions for Lattice Boltzmann Simulations. *Journal of Statistical Physics*, Vol. 71, Nos 5/6.
18. Lattice Boltzmann methods (n.d). Retrieved 19th March 2009, from Wikipedia: http://en.wikipedia.org/wiki/Lattice_Boltzmann_methods
19. Lattice Boltzmann Methods (n.d). Retrieved 30th March 2009, from Math, Statistics, and Computational Science at NIST: <http://math.nist.gov/mcsd/savg/parallel/lb/#id226247>

20. Lid-driven cavity problem (n.d.). Retrieved 20th July, 2009, from CFD-Online:
http://www.cfd-online.com/Wiki/Lid-driven_cavity_problem
21. Two dimensional lid-driven cavity (n.d.). Retrieved 11th August, 2009 from:
<http://www.unibas.it/utenti/bonfiglioli/node14.html>

APPENDICES

A. MAIN PROGRAM SOURCE CODE

```

|*****|
!      CIP lattice Boltzman                                !
!      Cavity flow                                          !
!      By: Nor Azwadi Bin Che Sidik                        !
!      PhD. Open environment system, Keio University, Japan !
!      24 jan 2005                                          !
!      Edited: 01 Aug 2008                                  !
|*****|

      program cavity

      implicit real*8 (a-h,o-z)
      parameter (ij = 100, kkk = 8)
      common /var1/ f(0:ij,0:ij,0:kkk),feq(0:ij,0:ij,0:kkk)
      common /var2/ cx(0:kkk),cy(0:kkk),w(0:kkk)
      common /var3/ u(ij,ij),v(ij,ij),u2(0:ij,0:ij),vel(0:ij,0:ij)
      common /var4/ rho(ij,ij),p(ij,ij)
      common /var5/ tmp(0:ij,0:ij,0:kkk),ff(0:ij,0:ij,0:kkk)
      common /var6/
      fn(0:ij,0:ij,0:kkk),gx(0:ij,0:ij,0:kkk),gy(0:ij,0:ij,0:kkk),gxn(0:ij,0:ij,0:kkk),gyn(0:ij,0:ij,0:kkk)

      common /con1/ uinit,re,delt
      common /con2/ lx,ly
      common /con3/ ntin,nstep
      common /con4/ pi,tau,rho,int,delx,dely,delx2,dely2,rtau,xnyu
      integer lx,ly,ntin,nstep,i,j,k,l
      integer ::unit, ierror
      character (len=6)::filename

      pi = atan(1.0d0)*4.0d0

      write (6,*) 'input time steps number for display'
      read (5,*) ntin
      write (6,*) ntin

      write (6,*) 'input lattice number for x-direction'
      read (5,*) lx
      write (6,*) lx

      write (6,*) 'input lattice number for y-direction'
      read (5,*) ly
      write (6,*) ly

      write (6,*) 'input delta t'
      read (5,*) delt
      write (6,*) delt

      write (6,*) 'input reynolds number'
      read (5,*) re
      write (6,*) re

      write (6,*) 'input initial density'
      read (5,*) rho,int
      write (6,*) rho,int

      write (6,*) 'input upper wall speed '
      read (5,*) uinit
      write (6,*) uinit

!      xnyu = lx*uinit/re
!      tau = !3*xnyu + 0.5

      write (6,*) 'calculation start'

      call initial

```

```

do nstep = 1, 1000000

call calculation

call output

if (mod(nstep,ntin) .eq. 0) then
    write(*,10)
10    format ('time step   u(x=1/2(lx))   v(y=1/2(ly))')
    write (*,20)
20    format ('=====')
    write(*,30) nstep, u((lx+1)/2,(ly+1)/2),v((lx+1)/2,(ly+1)/2)
30    format (I6,6X,E11.4,6X,E11.4,/)
    write(*,40) lx,ly
40    format ('mesh   = ',i6,' x ',i6)
    write(*,50) re
50    format ('Rey Nu = ',F6.1)
    write(*,70) uinit
70    format ('uinit = ',E8.2)
    !    write(*,*) velmaxn1 - velmaxn
    !    end if

!    if (mod(nstep,ntin) .eq.0) then
!        do i = 0,lx
!            do j = 0,ly
!                vel(i,j) = (u(i,j)**2 + v(i,j)**2)**0.5
!            end do
!        end do
!
!        velmaxn = abs(vel(1,1))
!!        do i = 0,lx
!            do j = 0,ly
!                if(abs(vel(i,j)) .gt. velmaxn) velmaxn = vel(i,j)
!            end do
!        end do
!
!    endif

!    if (mod(nstep,ntin) .eq.1) then
!        do i = 0,lx
!            do j = 0,ly
!                vel(i,j) = (u(i,j)**2 + v(i,j)**2)**0.5
!            end do
!        end do
!
!        velmaxn1 = abs(vel(1,1))
!        do i = 0,lx
!            do j = 0,ly
!                if(abs(vel(i,j)) .gt. velmaxn1) velmaxn1 = vel(i,j)
!            end do
!        end do
!
!    endif

!***** if converge*****:
!    if (abs(velmaxn1 - velmaxn) .le. 1.0e-8 ) then
!
!        write(*,*)'solution converge'
!
!        if (mod(nstep,ntin) .eq.0) then
!            do i = 0,lx
!                do j = 0,ly
!                    write (*,*) u(i,j)/uinit, v(i,j)/uinit
!                end do
!            end do
!        endif
!
!        unit =29
!        filename ='CIPLBM400'

```

```

open (unit=29,file='velocity',status='replace',action='write',iostat=ierror)
write(29,*) 'output result (end of iteration)'
write(29,*) 'Reynolds =',Re
write(29,*) 'mesh=',lx,' x ',ly
write(29,*) 'tau=',tau
write(29,*) 'time step=', nstep
write(29,*) 'v velocity at the horizontal center geometry'
do i = 1,lx
    write(29,*) v(i,(ly+1)/2)/uinit
end do
!*****
write(29,*)
write(29,*)
write(29,*) 'u velocity at the vertical center geometry'
do j = 1,ly
    write(29,*) u((lx+1)/2,j)/uinit
end do

! go to 1000
!     end if

        end do

write(*,*)'end of iteration'
unit =29
filename = 'CIPLBM400'
open (unit=29,file='velocity',status='replace',action='write',iostat=ierror)
write(29,*) 'output result (end of iteration)'
write(29,*) 'Reynolds =',Re
write(29,*) 'mesh=',lx,' x ',ly
write(29,*) 'tau=',tau
write(29,*) 'time step=', nstep
write(29,*) 'v velocity at the horizontal center geometry'
do i = 1,lx
    write(29,*) v(i,(ly+1)/2)/uinit
end do
!*****
write(29,*)
write(29,*)
write(29,*) 'u velocity at the vertical center geometry'
do j = 1,ly
    write(29,*) u((lx+1)/2,j)/uinit
end do

1000      stop

        end

        !!!

```

B. SUBROUTINE INITIAL SOURCE CODE

```

=====
      subroutine initial
=====
      implicit real*8 (a-h,o-z)
      parameter (ij = 100, kkk = 8)
      common /var1/ f(0:ij,0:ij,0:kkk),feq(0:ij,0:ij,0:kkk)
      common /var2/ cx(0:kkk),cy(0:kkk),w(0:kkk)
      common /var3/ u(ij,ij),v(ij,ij),u2(0:ij,0:ij),vel(0:ij,0:ij)
      common /var4/ rho(ij,ij),p(ij,ij)
      common /var5/ tmp(0:ij,0:ij,0:kkk),ff(0:ij,0:ij,0:kkk)
      common /var6/
      fn(0:ij,0:ij,0:kkk),gx(0:ij,0:ij,0:kkk),gy(0:ij,0:ij,0:kkk),gxn(0:ij,0:ij,0:kkk),gyn(0:ij,0:ij,0:kkk)

      common /con1/ uinit,re,delt
      common /con2/ lx,ly
      common /con3/ ntin,nstep
      common /con4/ pi,tau,rho,int,delx,dely,delx2,dely2,rtau,xnyu
      integer lx,ly,ntin,nstep,i,j,k,l
      integer ::unit, ierror
      character (len=6)::filename

      delx = 1.0
      dely = 1.0
      dely2 = dely*dely
      delx2 = delx*delx
      tau = 3.0*uinit*(lx-1)/re - delt*0.5
      rtau = 1.0/tau
      write (6,*) 'tau = ',tau , 'rtau = ',rtau

      !setup physical data!
      cx(0) = 0.0d0
      cy(0) = 0.0d0

      do k = 1,8
         w(k) = sqrt (2.0d0)
         if(mod(k,2) .eq. 1) w(k) = 1.0d0
         cx(k) = w(k)*cos((k-1)*pi/4.0d0)
         cy(k) = w(k)*sin((k-1)*pi/4.0d0)
      end do

      do i = 1,lx
         do j = 1,ly
            rho(i,j) = rho,int

            if (j.eq.ly) then
               u(i,j) = uinit
               v(i,j) = 0.0
            else
               u(i,j) = 0.0
               v(i,j) = 0.0
            end if
         end do
      end do

      !setup up function!

      call equilibrium

      do i = 0,lx+1
         do j = 0,ly+1
            do k = 0,8
               f(i,j,k) = feq(i,j,k)
               gx(i,j,k) = 0.0
               gy(i,j,k) = 0.0
            end do
         end do
      end do

```

```
return  
end  
  
!!!!
```

C. SUBROUTINE EQUILIBRIUM SOURCE CODE

```

=====
      subroutine equilibrium
=====
      implicit real*8 (a-h,o-z)
      parameter (ij = 100, kkk = 8)
      common /var1/ f(0:ij,0:ij,0:kkk),feq(0:ij,0:ij,0:kkk)
      common /var2/ cx(0:kkk),cy(0:kkk),w(0:kkk)
      common /var3/ u(ij,ij),v(ij,ij),u2(0:ij,0:ij),vel(0:ij,0:ij)
      common /var4/ rho(ij,ij),p(ij,ij)
      common /var5/ tmp(0:ij,0:ij,0:kkk),ff(0:ij,0:ij,0:kkk)
      common /var6/
fn(0:ij,0:ij,0:kkk),gx(0:ij,0:ij,0:kkk),gy(0:ij,0:ij,0:kkk),gxn(0:ij,0:ij,0:kkk),gyn(0:i
j,0:ij,0:kkk)
      common /con1/ uinit,re,delt
      common /con2/ lx,ly
      common /con3/ ntin,nstep
      common /con4/ pi,tau,rho,int,delx,dely,delx2,dely2,rtau,xnyu
      integer lx,ly,ntin,nstep,i,j,k,l
      integer ::unit, ierror
      character (len=6)::filename

      do i = 1,lx
        do j = 1,ly
          u2(i,j) = u(i,j)**2 + v(i,j)**2
          feq(i,j,0) = rho(i,j)*(1.0 - 3.0/2.0*u2(i,j))*4.0/9.0
          do l = 1,4
            k = l*2      ; dir = cx(k)*u(i,j) + cy(k)*v(i,j)
            feq(i,j,k) = rho(i,j)*(1. + 3.*dir + 9./2.*dir**2 -
3./2.*u2(i,j))/36.

                                k = l*2 - 1; dir = cx(k)*u(i,j) + cy(k)*v(i,j)
                                feq(i,j,k) = rho(i,j)*(1. + 3.*dir + 9./2.*dir**2 -
3./2.*u2(i,j))/9.

          end do
        end do
      end do

      return
      end

```

D. SUBROUTINE CALCULATION SOURCE CODE

```

=====
      subroutine calculation
=====
      implicit real*8 (a-h,o-z)
      parameter (ij = 100, kkk = 8)
      common /var1/ f(0:ij,0:ij,0:kkk),feq(0:ij,0:ij,0:kkk)
      common /var2/ cx(0:kkk),cy(0:kkk),w(0:kkk)
      common /var3/ u(ij,ij),v(ij,ij),u2(0:ij,0:ij),vel(0:ij,0:ij)
      common /var4/ rho(ij,ij),p(ij,ij)
      common /var5/ tmp(0:ij,0:ij,0:kkk),ff(0:ij,0:ij,0:kkk)
      common /var6/
fn(0:ij,0:ij,0:kkk),gx(0:ij,0:ij,0:kkk),gy(0:ij,0:ij,0:kkk),gxn(0:ij,0:ij,0:kkk),gyn(0:ij,0:ij,0:kkk)
      common /con1/ uinit,re,delt
      common /con2/ lx,ly
      common /con3/ ntin,nstep
      common /con4/ pi,tau,rhoint,dex,dely,dex2,dely2,rtau,xnyu
      integer lx,ly,ntin,nstep,i,j,k,l
      integer ::unit, ierror
      character (len=6)::filename

!      write (*,*) 'call calculation'

      call equilibrium

      call adv

!      call collision

      return
      end

```


E. SUBROUTINE ADVECTION SOURCE CODE

```

!=====
      subroutine adv
!=====
      implicit real*8 (a-h,o-z)
      parameter (ij = 100, kkk = 8)
      common /var1/ f(0:ij,0:ij,0:kkk),feq(0:ij,0:ij,0:kkk)
      common /var2/ cx(0:kkk),cy(0:kkk),w(0:kkk)
      common /var3/ u(ij,ij),v(ij,ij),u2(0:ij,0:ij),vel(0:ij,0:ij)
      common /var4/ rho(ij,ij),p(ij,ij)
      common /var5/ tmp(0:ij,0:ij,0:kkk),ff(0:ij,0:ij,0:kkk)
      common /var6/
      fn(0:ij,0:ij,0:kkk),gx(0:ij,0:ij,0:kkk),gy(0:ij,0:ij,0:kkk),gxn(0:ij,0:ij,0:kkk),gyn(0:ij,0:ij,0:kkk)

      common /con1/ uunit,re,delt
      common /con2/ lx,ly
      common /con3/ ntin,nstep
      common /con4/ pi,tau,rhoint,dex,dely,dex2,dely2,rtau,xnyu
      integer lx,ly,ntin,nstep,i,j,k,l
      integer ::unit, ierror
      character (len=6)::filename

!      write (*,*) 'call calculation2'

      total = 0.0
      if (nstep .ne. 1) then
      do j = 1,ly
      do k = 0,8

          f(0,j,k) = 2.0*f(1,j,k) - f(2,j,k)
          f(lx+1,j,k) = 2.0*f(lx,j,k) - f(lx-1,j,k)

      end do
      end do

      do i = 1,lx
      do k = 0,8
          f(i,0,k) = 2.0*f(i,1,k) - f(i,2,k)
          f(i,ly+1,k) = 2.0*f(i,ly,k) - f(i,ly-1,k)

      end do
      end do

      do k = 0,8
          f(0,0,k) = 2.0*f(1,1,k) - f(2,2,k)
          f(0,ly+1,k) = 2.0*f(1,ly,k) - f(2,ly-1,k)
          f(lx+1,0,k) = 2.0*f(lx,1,k) - f(lx-1,2,k)
          f(lx+1,ly+1,k) = 2.0*f(lx,ly,k) - f(lx-1,ly-1,k)

      end do
      end if

!      write (*,*) 'call calculation3'

      do i = 1,lx
      do j = 1,ly
      do k = 0,8
          xx = -cx(k)*delt
          yy = -cy(k)*delt

          zx = sign(1.0,cx(k))
          zy = sign(1.0,cy(k))

          iup = i-int(zx)
          jup = j-int(zy)

          al = ((gx(iup,j,k) + gx(i,j,k))*dex*zx -
      2.0*(f(i,j,k) - f(iup,j,k)))/(dex**3*zx)
          el = (3.0*(f(iup,j,k) - f(i,j,k)) + (gx(iup,j,k) +
      2.*gx(i,j,k))*dex*zx)/(dex*dex)
          bl = ((gy(i,jup,k) + gy(i,j,k))*dely*zy -
      2.0*(f(i,j,k) - f(i,jup,k)))/(dely**3*zy)
          fl = (3.0*(f(i,jup,k) - f(i,j,k)) + (gy(i,jup,k) +
      2.0*gy(i,j,k))*dely*zy)/dely**2

```

```

!                                     tmp = f(i,j,k) - f(i,jup,k) - f(iup,j,k) +
f(iup,jup,k) !!!! cuba check gx, gy, gxn, dan gyn sebab tak define dia punya initial
value

!                                     tmq = gy(iup,j,k) - gy(i,j,k)

                                     dl = ( - (f(i,j,k) - f(i,jup,k) - f(iup,j,k) +
f(iup,jup,k)) - (gy(iup,j,k) - gy(i,j,k))*dely*zy)/(delx*dely**2*zx)
                                     cl = ( - (f(i,j,k) - f(i,jup,k) - f(iup,j,k) +
f(iup,jup,k)) - (gx(i,jup,k) - gx(i,j,k))*delx*zx)/(delx**2*dely*zy)
                                     gl = ( - (gy(iup,j,k) - gy(i,j,k)) +
cl*delx*delx)/(delx*zx)

                                     fn(i,j,k) = ((a1*xx+cl*yy+el)*xx + gl*yy +
gx(i,j,k))*xx + ((b1*yy+dl*xx+f1)*yy + gy(i,j,k))*yy + f(i,j,k)
                                     gxn(i,j,k) = (3.0*a1*xx + 2.0*(cl*yy+el))*xx +
(dl*yy+gl)*yy+gx(i,j,k)
                                     gyn(i,j,k) = (3.0*b1*yy + 2.0*(dl*xx+f1))*yy +
(c1*xx+gl)*xx+gy(i,j,k)
                                     end do
                                     end do
                                     end do
do i = 1,lx
do j = 1,ly
do k = 0,8
f(i,j,k) = fn(i,j,k)
gx(i,j,k) = gxn(i,j,k)
gy(i,j,k) = gyn(i,j,k)
end do
end do
end do
do i = 1,lx
do j = 1,ly
do k = 0,8
feq(i,j,k) = f(i,j,k)-delt*rtau*(f(i,j,k)-
if (i .eq. lx) then
gxn(i,j,k) = gx(i,j,k)-delt*rtau*(gx(i,j,k)-
0.5*(3*feq(i,j,k)-4*feq(i-1,j,k)+feq(i-2,j,k)))
elseif (i .eq. 1) then
gxn(i,j,k) = gx(i,j,k)-delt*rtau*(gx(i,j,k)- 0.5*(-
3*feq(i,j,k)+4*feq(i+1,j,k)-feq(i+2,j,k)))
else
gxn(i,j,k) = gx(i,j,k)-delt*rtau*(gx(i,j,k)-
0.5*(feq(i+1,j,k)-feq(i-1,j,k)))
end if
if (j .eq. ly) then
gyn(i,j,k) = gy(i,j,k)-delt*rtau*(gy(i,j,k)-
0.5*(3*feq(i,j,k)-4*feq(i,j-1,k)+feq(i,j-2,k)))
elseif (j .eq. 1) then
gyn(i,j,k) = gy(i,j,k)-delt*rtau*(gy(i,j,k)- 0.5*(-
3*feq(i,j,k)+4*feq(i,j+1,k)-feq(i,j+2,k)))
else
gyn(i,j,k) = gy(i,j,k)-delt*rtau*(gy(i,j,k)-
0.5*(feq(i,j+1,k)-feq(i,j-1,k)))
end if
end do
end do
end do
!
write (*,*) 'call calculation5'
do i = 1,lx
do j = 1,ly
do k = 0,8
f(i,j,k) = fn(i,j,k)
gx(i,j,k) = gxn(i,j,k)
gy(i,j,k) = gyn(i,j,k)
if (f(i,j,k) <= 0 ) then
write (*,*) ' error'
end if
total = total + f(i,j,k)

```

```
                                end do
                                end do
                                end do
                                if ( mod(nstep,ntin) == 0) then
                                write (*,60) total
60                                format ('total  = ',F8.2,/)
                                end if
!                                write (*,*) 'call calculation4'

                                return
                                end
```

F. SUBROUTINE OUTPUT SOURCE CODE

```

=====
      subroutine output
=====
      implicit real*8 (a-h,o-z)
      parameter (ij = 100, kkk = 8)
      common /var1/ f(0:ij,0:ij,0:kkk),feq(0:ij,0:ij,0:kkk)
      common /var2/ cx(0:kkk),cy(0:kkk),w(0:kkk)
      common /var3/ u(ij,ij),v(ij,ij),u2(0:ij,0:ij),vel(0:ij,0:ij)
      common /var4/ rho(ij,ij),p(ij,ij)
      common /var5/ tmp(0:ij,0:ij,0:kkk),ff(0:ij,0:ij,0:kkk)
      common /var6/
fn(0:ij,0:ij,0:kkk),gx(0:ij,0:ij,0:kkk),gy(0:ij,0:ij,0:kkk),gxn(0:ij,0:ij,0:kkk),gyn(0:ij,0:ij,0:kkk)
      common /con1/ uinit,re,delt
      common /con2/ lx,ly
      common /con3/ ntin,nstep
      common /con4/ pi,tau,rhoint,delx,dely,delx2,dely2,rtau,xnyu
      integer lx,ly,ntin,nstep,i,j,k,l
      integer ::unit, ierror
      character (len=6)::filename

      do i = 1,lx
        do j = 1,ly
          rho(i,j) = 0.0
        end do
      end do

      do i = 2,lx-1
        do j = 2,ly-1
          do k = 0,8
            rho(i,j) = rho(i,j) + f(i,j,k)
          end do
        end do
      end do

      do i = 2,lx-1
        rho(i,1) = rho(i,2)
        rho(i,ly) = rho(i,ly-1)
      end do

      do j = 2,ly-1
        rho(1,j) = rho(2,j)
        rho(lx,j) = rho(lx-1,j)
      end do

      rho(1,1) = rho(2,2)
      rho(lx,1) = rho(lx-1,2)
      rho(lx,ly) = rho(lx-1,ly-1)
      rho(1,ly) = rho(2,ly-1)

      do i = 2,lx-1
        do j = 2,ly-1
          u(i,j) = 0.d0
          v(i,j) = 0.d0
        end do
      end do

      do i = 2, lx-1
        do j = 2, ly-1
          do k = 0,8
            u(i,j) = u(i,j) + f(i,j,k)*cx(k)/rho(i,j)
            v(i,j) = v(i,j) + f(i,j,k)*cy(k)/rho(i,j)
          end do
        end do
      end do

      return
      end

```

G. SUBROUTINE COLLISION SOURCE CODE

```

=====
      subroutine collision
=====
      implicit real*8 (a-h,o-z)
      parameter (ij = 100, kkk = 8)
      common /var1/ f(0:ij,0:ij,0:kkk),feq(0:ij,0:ij,0:kkk)
      common /var2/ cx(0:kkk),cy(0:kkk),w(0:kkk)
      common /var3/ u(ij,ij),v(ij,ij),u2(0:ij,0:ij),vel(0:ij,0:ij)
      common /var4/ rho(ij,ij),p(ij,ij)
      common /var5/ tmp(0:ij,0:ij,0:kkk),ff(0:ij,0:ij,0:kkk)
      common /var6/
fn(0:ij,0:ij,0:kkk),gx(0:ij,0:ij,0:kkk),gy(0:ij,0:ij,0:kkk),gxn(0:ij,0:ij,0:kkk),gyn(0:ij,0:ij,0:kkk)
      common /con1/ uinit,re,delt
      common /con2/ lx,ly
      common /con3/ ntin,nstep
      common /con4/ pi,tau,rhoint,delx,dely,delx2,dely2,rtau,xnyu
      integer lx,ly,ntin,nstep,i,j,k,l
      integer ::unit, ierror
      character (len=6)::filename

      total = 0.0
      do i = 1,lx
        do j = 1,ly
          do k = 0,8
            f(i,j,k) = f(i,j,k) - (f(i,j,k) - feq(i,j,k))/tau
!            if (f(i,j,k) <= 0 ) then
!              write (*,*) ' error'
!            end if
            total = total + f(i,j,k)
          end do
        end do
      end do
      if ( mod(nstep,ntin) == 0) then
        write (*,60) total
60      format ('total = ',F8.2,/)
      end if

=====

      return
      end

```