SIMULATION OF RAYLEIGH BERNARD CONVECTION USING LATTICE BOLTZMANN METHOD

MUHAMMAD AIMAN BIN ABDUL MOIN

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

Faculty of Mechanical Engineering
UNIVERSITI MALAYSIA PAHANG

NOVEMBER 2009

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: MOHAMAD MAZWAN BIN MAHAT
Position: LECTURER
Date: 17 NOVEMBER 2009

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: MUHAMMAD AIMAN BIN ABDUL MOIN
ID Number: MA06090
Date:

Dedicated to my beloved parents

ACKNOWLEDGEMENTS

I am grateful and would like to express my sincere gratitude to my supervisor En. Mohamad Mazwan Bin Mahat for his germinal ideas, invaluable guidance, continueous encouragement and constant support in making this research possible. He has always impressed me with his outstanding professional conduct, his strong conviction for science, and his belief that a Degree program is only a start of a life-long learning experience. I appreciate his consistent support from the first day I applied to graduate program to these concluding moments. I am truly grateful for his progressive vision about my training in science, his tolerance of my naïve mistakes, and his commitment to my future career

My sincere thanks go to all my and members of the staff of the Mechanical Engineering Department, UMP, who helped me in many ways and made my stay at UMP pleasant and unforgettable.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals. Special thanks should be given to my committee members. I would like to acknowledge their comments and suggestions, which was crucial for the successful completion of this study.

Above this all, my highness praises and thanks to Almighty Allah subhanahu waalla, the most gracious the most merciful, who gave me the knowledge, courage and patience to accomplish this thesis. May the peace and blessings of Allah be upon Prophet Muhammad Sallallahu alaihi wasallam.

ABSTRACT

In this thesis, a method of lattice Boltzmann is introduced. Lattice Boltzmann Method is to build a bridge between the microscopic and macroscopic dynamics, rather than to deal with macroscopic dynamics directly. In other words, LBM is to derive macroscopic equations from microscopic dynamics by means of statistic, rather than to solve macroscopic equations. Then, the methodology and general concepts of the lattice Boltzmann method are introduced. Next, a thermal lattice Boltzmann model is developed to simulate incompressible thermal flow. This report describes the flow pattern of Rayleigh Bernard Convection. This project will be focusing at low Rayleigh number and discretization of microscopic velocity using 9-discrete velocity model (D2Q9) and 4-discrete velocity model (D2Q4). This two discrete velocity model is applying the Gauss-Hermitte quadrate procedure. Rayleigh Bernard Convection and Lattice Boltzmann Method have been found to be an efficient and numerical approach to solve the natural convection heat transfer problem. Good Rayleigh Bernard Convection flow pattern agreement was obtained with benchmark (previous study).

ABSTRAK

Di dalam tesis ini, kaedah kekisi Boltzmann diperkenalkan. Kaedah kekisi Bolzmann ialah untuk membuat hubungan di antara mikroskopik dan makroskopik. Dalam erti kata lain, kaedah kekisi Boltzmann ialah menerbitkan persamaan makroskopik daripada pergerakan mikoskopik oleh statistik daripada menyelesaikan persamaan makroskopik. Kemudian, methodogi dan konsep umum kaedah kekisi Boltzmann diperkenalkan. Kemudian, model terma kekisi Boltzmann adalah untuk membina simulasi di dalam aliran terma yg tidak mampat. Laporan ini memperihalkan corak aliran arus perolakan Rayleigh Bernard. Projek ini menumpukan pada nombor Rayleigh yang rendah dan arah pergerakan halaju mikoskopik menggunakan model 9-diskrit halaju dan model 4-diskrit halaju. Kedua-dua model diskrit ini mengadaptasikan prosedur kuadrat Gauss-Hermitte. Pemanasan di antara dua plat dan kaedah kekisi Boltzmann merupakan satu cara yang berkesan untuk menyelesaikan pemanasan pemindahan haba. Bentuk aliran di antara dua plat mempunyai persamaan dengan bentuk aliran kajian sebelum.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

u	Velocity of the fluid parcel
P	Pressure
ρ	Fluid density
$\tau_{f\ g}$	Time relaxation
υ	Viscosity of fluid
χ	Thermal diffusivity
$\Omega_{(x\ t)}$	Single relaxation time
$f(\mathbf{x},t)$	Current distribution of particles
$f^{eq}(\mathbf{x},t)$	Equilibrium function
Ω	Collision integral
Δt	Value of unity
T_C	Cool plate temperature
T_H	Hot plate temperature
α	Thermal diffusivity
k	Thermal conductivity
c_p	Specific heat
x	Characteristic length
Ra_x	Rayleigh number at position x

Pr	Prandtl number
g	Acceleration due to gravity
β	Thermal expansion coefficient
V	Mean fluid velocity
Q	Volumetric flow rate
A	Pipe cross-sectional area
d	Depth of the chamber
δ	Thickness
L	Length
Gr_x	Grashof number at position x

LIST OF ABBREVIATIONS

BGK	Bhatnagar Groos Krook
CFD	Computational fluid dynamics
D2Q4	Two dimensionals four velocities model
D2Q9	Two dimensionals nine velocities model
LB	Lattice Boltzmann
LBE	Lattice Boltzmann equation
LBM	Lattice Boltzmann method
LGA	Lattice gas approach
MD	Molecular dynamics

CHAPTER 1

INTRODUCTION

1.1 PROJECT BACKGROUND

Navier-Stokes Equation

$$\frac{\partial u}{\partial t} + u\nabla \bullet u = -\nabla P + \left(\frac{2\tau_f - 1}{6} \right) \nabla^2 u \tag{1.1}$$

$$\nabla . u = 0 \tag{1.2}$$

Equation 1.1 shows the Navier-Stokes Equation that explains the flow of incompressible fluids together with the continuity equation show in equation 1.2 where ν is the kinematics viscosity, $\mathbf{u}$ is the velocity of the fluid parcel, P is the pressure, and ρ is the fluid density.

Lattice Gas Approach

Lattice Boltzmann models were first based on Lattice Gas Approach (LGA) in that they used the same lattice and applied the same collision. Instead of particles, Lattice Boltzmann (LBM) deal with continuous distribution functions which interact locally (only distributions at a single node are involved) and which propagate after collision to the next neighbor node. This is the main advantage of LBM compare to LGA. The next step in the development was the simplification of the collision and the choice of different distribution functions. This gives LBM is more flexible than LGA.

Molecular Dynamics

In molecular dynamics (MD), one tries to simulate macroscopic behavior of real fluids by setting up the model which described the microscopic interaction as good as possible. This leads to realistic equation of states whereas LGA or LBM posses only isothermal relations between mass density and pressure. The complexity of the interactions in MD restricts the number of particles and the time of integration. A method somewhat in between MD and LGA is maximally discretized molecular dynamics proposed by Colvin, Ladd and Alder (1988).

Lattice Boltzmann Method

The lattice Boltzmann method (LBM) has developed into an alternative and promising numerical scheme for simulating fluid flows and modeling physics in fluids. Historically, the lattice Boltzmann approach developed from improvement of lattice gases, although it can also be derived directly from the simplified Bhatnagar-Gross-Krook (BGK) equation.

The lattice Boltzmann method is based on microscopic models and macroscopic kinetic equations. The kinetic nature of the LBM introduces important features that distinguish it from other numerical methods. First, the streaming process of the LBM in velocity space is linear. Second, the incompressible Navier-Stokes (NS) equations can be obtained in the nearly incompressible limit of the LBM. The LBM originated from lattice gas approach (LGA), a discrete particle kinetics utilizing a discrete lattice and discrete time. The primary goal of LBM is to build a bridge between the microscopic and macroscopic dynamics rather than to deal with macroscopic dynamics directly. In other words, the goal is to derive macroscopic equations from microscopic dynamics by means of statistics rather than to solve macroscopic equation in Figure 1.1 below.

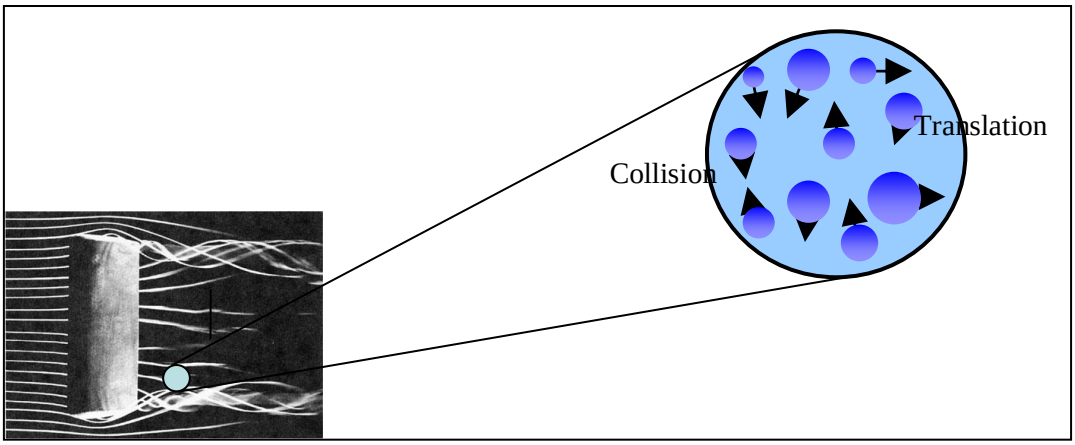


Figure 1.1: The relationship between macroscopic and microscopic.

Rayleigh-Bernard Convection

Rayleigh-Bernard convection is the instability of a fluid layer which is confined between two thermally conducting plates, and is heated from below to produce a fixed temperature difference. For small temperature differences between the plates there is no flow and heat is transported by conduction only. Above a certain temperature differences, convection sets in against the downward pointing gravitational acceleration, and a regular convection pattern is formed. At even higher temperature differences this pattern breaks down, eventually leading to plume dominated convection turbulence. Since liquids typically have positive thermal expansion coefficient, the hot liquid at the bottom of the cell expands and produces an unstable density gradient in the fluid layer. If the density gradient is sufficiently strong, the hot fluid will rise, causing a convective flow which results in enhanced transport of heat between the two plates.

1.2 PROBLEM STATEMENT

Simulate Rayleigh Bernard Convection using Lattice Boltzmann Method to study the flow pattern of Rayleigh Bernard Convection.

1.3 OBJECTIVE

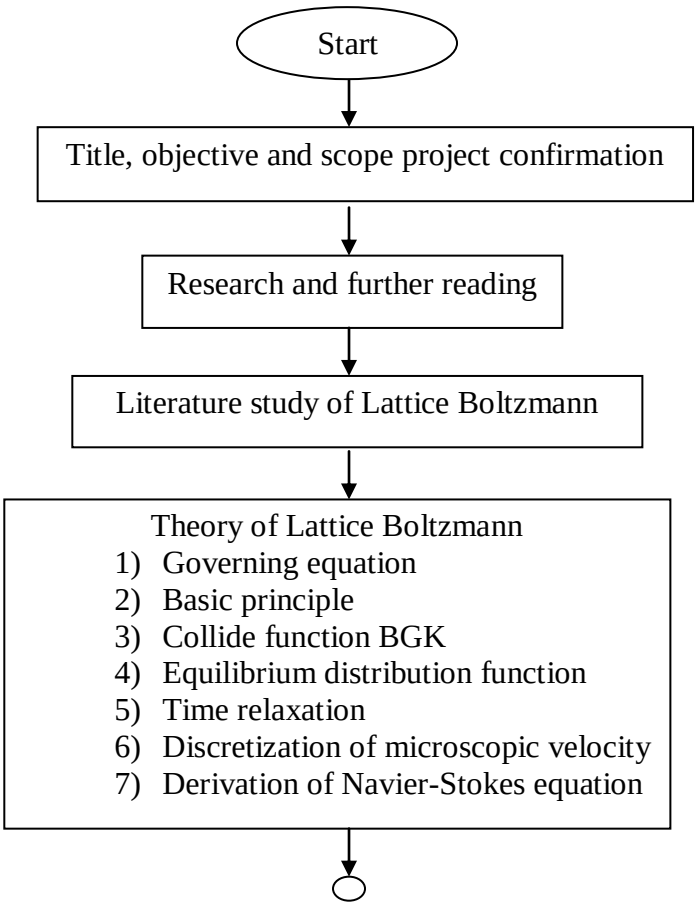
For this thesis the objective is to study the flow pattern of Rayleigh Bernard Convection.

1.4 SCOPE OF WORK

This project is focusing on the pattern of the Rayleigh Bernard Convection flow pattern at low Rayleigh number which is at laminar flow. This project also focusing on using D2Q9 and D2Q4 microscopic velocity model for discretization of microscopic velocity.

1.5 PROCESS FLOW CHART

The Figure 1.2 shows the separation of information or processes in a step-by-step flow and easy to understand diagrams showing how steps in a process fit together. This makes useful tools for communicating how processes work and for clarity due to time limitation how a particular job is done in Final Year Project.



`

Figure 1.2: Flowchart of PSM

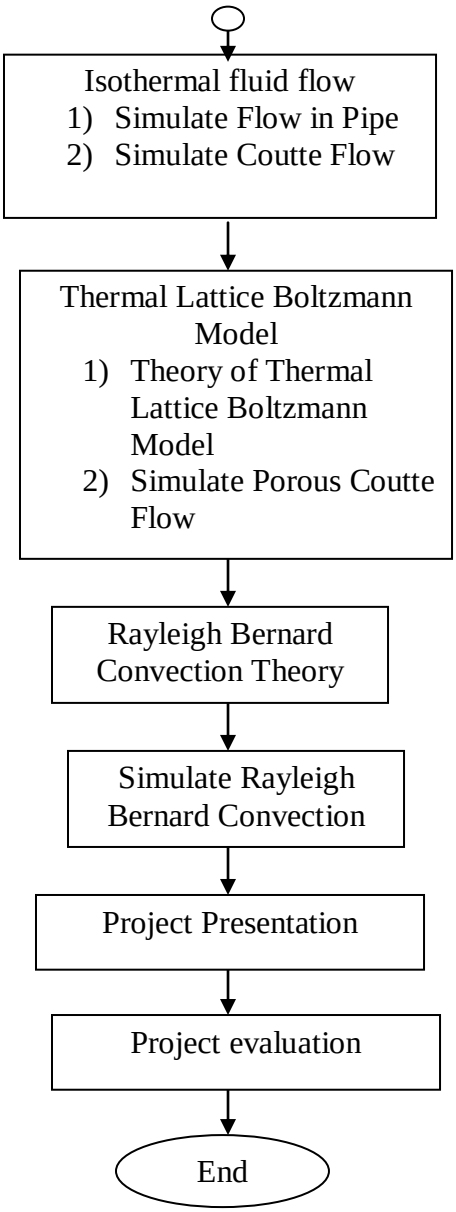


Figure 1.2: continue

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, the previous study for Bhatnagar–Gross-Krook (BGK) collision equation and model, Lattice Boltzmann Equation (LBE) will be discussed in this chapter. From BGK and LBE, discretization of microscopic velocity will be discussed. Discretization of microscopic velocity consist two models which are 9-discrete velocity model (D2Q9) and are 4-discrete velocity model (D2Q4). Furthermore, the dimensionless number (Prandlt number, Rayleigh number, Reynolds number and Nusselt number) also will be discussed in this chapter. Lastly, Rayleigh Bernard Convection and Rayleigh Bernard Convection problem will be discussed in this chapter. Rayleigh Bernard Convection problem is the boundary condition that will be used for the simulation.

2.2 NAVIER-STOKES EQUATION

$$\rho \frac{Du}{Dt} = -\nabla p + \rho f \tag{2.1}$$

$$\nabla . u = 0 \tag{2.2}$$

Sources: S. Chen and G.D. Doolen, Annu 1998.

Equation 2.1 shows the fluid velocity u of an inviscid (ideal) fluid of density ρ under the action of a body force ρf is determined and equation 2.2 shows as Euler's equation. The scalar p is the pressure. This equation is supplemented by an equation describing the conservation of mass. For an incompressible fluid this is simply to get the continuity equation.

Real fluids however are never truly inviscid. We must therefore see how Euler's equation is changed by the inclusion of viscous forces.

$$\frac{\partial u}{\partial t} + u\nabla \bullet u = -\nabla P + \left(\frac{2\tau_f - 1}{6} \right) \nabla^2 u \qquad (2.3)$$

Sources: C. S. Nor Azwadi 2007.

Equation 2.3 shows the flow of incompressible fluids can be described by the momentum equation. Derivation of continuity and momentum equation, we will get the Navier-Stokes equation by using Chapmann-Enskog expansion procedure. Chapmann-Enskog procedure is a method for obtaining an approximate solution. Chapmann-Enskog also manages to extract from the kinetic equation for the density distribution function, *F* a set of hydrodynamic equations for the particle number, the momentum and the energy per unit volume.

2.2.1 Macroscopic Equation for Isothermal

$$\upsilon = \frac{2\tau - 1}{6} \qquad (2.4)$$

Sources:M. Rohde, D. Kandhai, J. J. Derksen, and H. E. A. Van den Akker 2003.

Using Chapmann-Enskog expansion procedure, relation between the time relaxation τ , in microscopic level and viscosity of fluid υ, in macroscopic level is shown above.

2.2.2 Macroscopic Equation for Thermal

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \left(\tau_g - \frac{1}{2} \right) \nabla^2 T \tag{2.5}$$

Sources: C. S. Nor Azwadi 2007.

In thermal model, the energy equation is added. The energy equation is shown above.

$$\tau_f = 3\upsilon + \frac{1}{2} \tag{2.6}$$

$$\tau_g = \chi + \frac{1}{2} \tag{2.7}$$

M. Rohde, D. Kandhai, J. J. Derksen, and H. E. A. Van den Akker 2003.

Viscosity and thermal diffusivity will produce when using the Chapmann-Enskog expansion procedure where υ is viscosity and χ is thermal diffusivity show in equation 2.6 and equation 2.7.

2.3 BHATNAGAR-GROSS-KROOK (BGK)

The integral-differential Boltzmann equation (proposed by Boltzmann in 1872) can be solving by the model of Bhatnagar–Gross-Krook that had been proposed in 1954. The derivation of the transport equations for macroscopic variable becomes easier when BGK model replaced the nonlinear collision term of Boltzmann equation by a simpler term. The term is the relaxation of a state of a fluid to equilibrium state. Derivation of numerical schemes (kinetic schemes) is one of the important new applications by BGK model to solve hyperbolic conservation of laws (Bhatnagar, P.L., Gross, E.P and Krook, M., 1954).

Fermi-Dirac or Maxwellian distributions construct the collision process of the Lattice Boltzmann models. The single relaxation time BGK operator can solve these distribution functions as example microscopic equations are satisfied and Navier-Stokes equations are recovered.

$$f(x + c\Delta t, t + \Delta t) - f(x,t) = \Omega(f) \tag{2.8}$$

Sources: C. S. Nor Azwadi 2007.

The BGK equation was derived from the Boltzmann equation shown in equation 2.8.

$$\frac{1}{\tau} f^{neq}(\mathbf{x},t) = \frac{1}{\tau}\left[f(\mathbf{x},t) - f^{eq}(\mathbf{x},t)\right] = \Omega(f) \tag{2.9}$$

Bhatnagar, P.L., Gross, E.P and Krook, M. 1954.

Equation 2.9 show Bhatnagar-Gross-Krook equation where, $\Omega(f)$, is the single relaxation time BGK operator and $f(\mathbf{x},t)$ is the current distribution of particles on a lattice node. The τ symbol *is* the time relaxation parameter and $f^{eq}(\mathbf{x},t)$ is the equilibrium function.

2.4 LATTICE BOLTZMANN EQUATION

$$f(x + c\Delta t, t + \Delta t) - f(x,t) = \Omega(f) \tag{2.10}$$

Sources: S. Chen and G.D. Doolen, Annu 1998.

The Boltzmann equation for any lattice model is an equation for the time evolution of $f(x, t)$, the single-particle distribution at lattice site **x** where f is density distribution function, c is microscopic velocity, Ω is collision integral and Δt is a value of unity shown in equation 2.10.

$$\Omega(f(x,t)) = -\frac{f(x,t) - f^{eq}(x,t)}{\tau} \tag{2.11}$$

Sources: S. Chen and G.D. Doolen, Annu 1998.

Since the usual aim of lattice methods is to model macroscopic dynamics, the "exact" collision operator is unnecessarily complex and therefore numerically inefficient (S. Chen and G.D. Doolen, Annu, 1998). The collision operator is approximated by a single-time-relaxation process in which relaxation to some appropriately chosen equilibrium distribution occurs at some constant rate. In particular the collision term, $\Omega(f)$ is replaced by the single-time-relaxation approximation shown in equation 2.11.

The appropriately chosen equilibrium distribution, denoted by f^{eq}, depends on the local fluid variables, and $1/\tau$ is the rate of approach to this equilibrium. The relation $\sum\Omega = 0$ and $\sum c\Delta T\Omega = 0$ must be true to conserve mass and momentum, respectively.

Here $\frac{1}{\tau}$ is the relaxation constant, the rate of change towards the equilibrium.

τ is relaxation time :

when τ = 1: The distribution functions are exactly set to the equilibrium distribution.

when τ = 2: The distribution functions are midway between incoming distribution and the equilibrium distribution.

$$\Sigma f = \Sigma f^{eq} = \rho \quad \text{or} \quad \int f dc = \int f^{eq} dc = \rho \tag{2.12}$$

$$\Sigma cf = \Sigma cf^{eq} = \rho u \quad \text{or} \quad \int cf dc = \int cf^{eq} dc = \rho u \tag{2.13}$$

Sources: Xiaoyi He and Li-Shi Luo 1997.

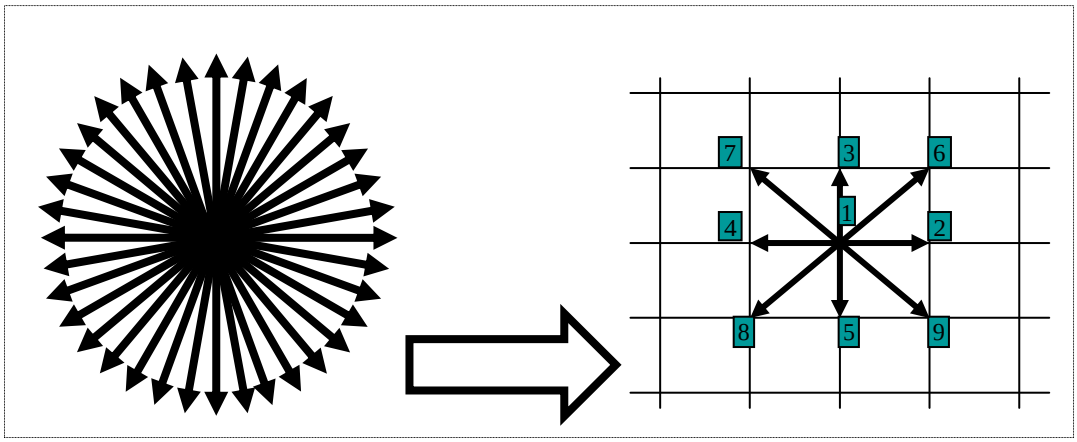
To derive the macroscopic equations obeyed by this model, one performs a Taylor expansion (C. Shu, Y. Peng and Y.T. Chew, 2002) in time (t) and space (x) and takes the long-wavelength and low-frequency limit of the lattice-Boltzmann equation for the single-particle distribution. The result is a continuum form of the Boltzmann equation correct to second order in the lattice spacing and the time step. The macroscopic variables, such as the density, ρ and flow velocity, u can be evaluated as the moment to the distribution function as equation 2.12 and equation 2.13.

2.5 DISCRETIZATION OF MICROSCOPIC VELOCITY

Discretizing the Boltzmann-BGK equation at a set of velocities that correspond to the nodes of a Gauss-Hermite quadrature in velocity space, we effectively solve the Boltzmann equation in a subspace spanned by the leading Hermite polynomials (C. S. Nor Azwadi 2007).

2.5.1 Lattice Boltzmann Isothermal Model

From the continuous velocity, use the Gauss-Hermite integration to get the 9-discrete velocity model. Figure below will shown how from continuous velocity to get 9-discrete velocity model by using Gauss-Hermite integration.



Using Gauss-Hermite integration

Figure 2.1: 9-discrete velocity model

$$f(x+c\Delta t, t+\Delta t) - f(x,t) = -\frac{f - f^{eq}}{\tau} \qquad (2.14)$$

Sources: C. S. Nor Azwadi 2007.

9-discrete velocity model means there are 9 directions that the particles will go after the collisions. The equation for 9-discrete velocity is shown in equation 2.14 where $f(x+c\Delta t, t+\Delta t) - f(x,t)$ is streaming process and $-\frac{f - f^{eq}}{\tau}$ is collision process.

There are only 9-discrete velocity for isothermal model because if use less then 9, we will not get the continuity and momentum equation. If use more then 9, we will take long time to simulate.

2.5.1.1 Example flow of Isothermal Model

2.5.1.1.1 Poisuelle Flow

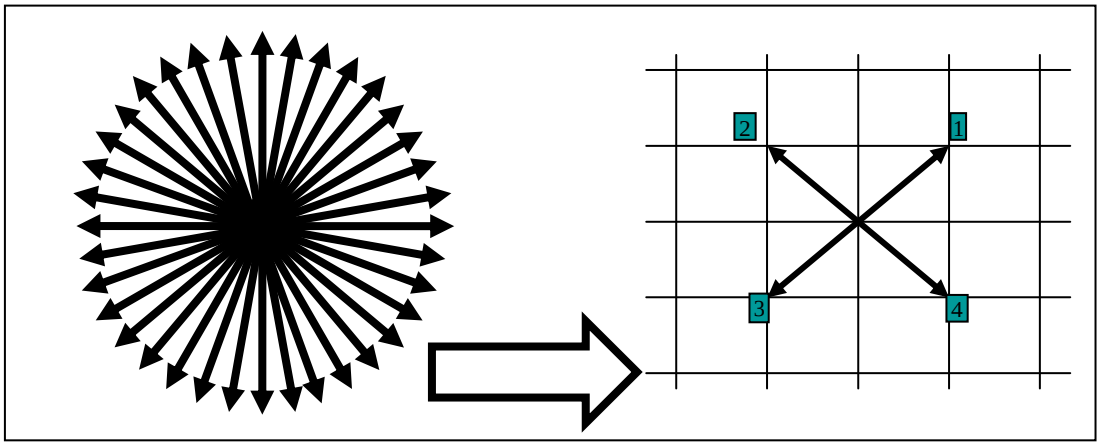
In this numerical simulation, the velocity is not set because the Poiseulle flow is driven by a pressure gradient between inlet and outlet end of the channel. In this case the pressure gradient is set between the inlet and outlet end of the channel. Setting the density to get the different values between inlet and outlet end of the channel. The bounce back boundary conditions are applied at the top and bottom walls.

2.5.1.1.2 Couette Flow

In this numerical simulation, the constant velocity is set at the top plate while the bottom plate is fixed. The initial conditions is related or similar to a null velocity everywhere expect on the top boundary, where the velocity is u = (U, 0). In this case, there is no pressure gradient between inlet and outlet end of the channel because the velocity is set to get the flow for this system.

2.5.2 Lattice Boltzmann Thermal Model

From the continuous velocity, use the Gauss-Hermite integration to get the 4-discrete velocity model. Figure below will shown how from continuous velocity to get 4-discrete velocity model by using Gauss-Hermite integration.



Using Gauss-Hermite integration

Figure 2.2: 4-discrete velocity model

$$g(x + c\Delta t, t + \Delta t) - g(x,t) = -\frac{g - g^{eq}}{\tau} \qquad (2.15)$$

Sources: C. S. Nor Azwadi 2007.

4-discrete velocity model means there are 4 directions that the particles will go after the collisions. The equation for 4-discrete velocity is shown in equation 2.15 where $g(x+c\Delta t, t+\Delta t) - g(x,t)$ is streaming process and $-\frac{g - g^{eq}}{\tau}$ is collision process.

There are only 4-discrete velocity for isothermal model because if use less then 4, we will not get the continuity and momentum equation. If use more then 4, we will take long time to simulate.

2.5.2.1 Example flow of Thermal Model

2.5.2.1.1 Porous Couette Flow

This numerical flow has two finite parallel flat plates which the upper plate is cool, T_C and moves at speed U while bottom plate is hot, T_H and it is fixed. There is space by a distance of L between the two parallel plates. A constant normal flow of fluid is injected through the bottom hot plate and withdrawn at the same rate from the upper plate. Periodic boundary conditions are used at the inlet and outlet of the channel. For velocity, the non-equilibrium bounce back boundary condition is used. The non-equilibrium bounce back boundary condition is also used for the temperature.

2.6 BOUNCE-BACK BOUNDARY CONDITION

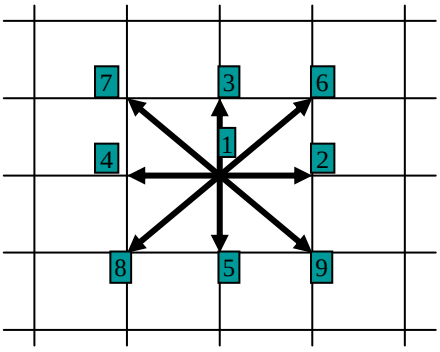


Figure 2.3: D2Q9 model

Sources: C. S. Nor Azwadi 2007.

The lattice gas Approach was the methods to simulate the initial approach of the boundary back. D2Q9 and D2Q4 velocity model are using the principle of the bounceback boundary conditions. In figure 2.4 shown the simple boundary condition where all the distribution functions at the boundaries back along to the initial position in lattice. This type of boundary condition can be shown by using 3x3 matrixes shown in figure 2.5.

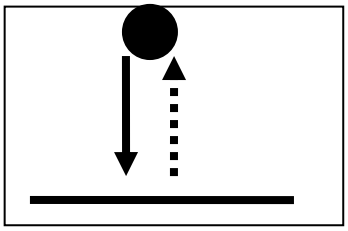


Figure 2.4: Bounceback boundary condition

Sources: C. S. Nor Azwadi 2007.

$$\begin{bmatrix} f_3 \\ f_6 \\ f_7 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} f_5 \\ f_8 \\ f_9 \end{bmatrix}$$

Figure 2.5: 3x3 matrixes boundary condition

Sources: C. S. Nor Azwadi 2007.

In bounceback boundary condition, velocity at wall defines zero by averaging the velocity at the boundary before and after the collision.

2.7 PRANDTL NUMBER

$$\mathrm{Pr} = \frac{\nu}{\alpha} = \frac{\text{viscous diffusion rate}}{\text{thermal diffusion rate}} = \frac{c_p \mu}{k} \qquad (2.16)$$

The Prandtl number, Pr is a dimensionless number, which describes the correlation between momentum diffusivity and thermal diffusivity. The ratio of momentum diffusivity (kinematic viscosity) and thermal diffusivity shown in equation 2.16 where:

- ν: kinematic viscosity, $\nu = \mu / \rho$, (SI units : m^2/s)
- α: thermal diffusivity, $\alpha = k / (\rho c_p)$, (SI units : m^2/s)
- μ: viscosity, (SI units : Pa s)

- *k*: thermal conductivity, (SI units : W/(m K))
- c_p: specific heat, (SI units : J/(kg K))
- ρ: density, (SI units : kg/m^3)

2.8 RAYLEIGH NUMBER

In fluid mechanics, the Rayleigh number for a fluid is a dimensionless number which connected with buoyancy driven flow. When the Rayleigh number is lower than critical value, the heat is transported by conduction only. Heat convection established when the Rayleigh number it exceeds the critical value.

The Rayleigh number is named after Lord Rayleigh and is defined as the product of the Grashof number, which describes the correlation between buoyancy and viscosity within a fluid, and the Prandtl number, which describes the correlation between momentum diffusivity and thermal diffusivity. Hence the Rayleigh number itself may also be viewed as the ratio of buoyancy forces and (the product of) thermal and momentum diffusivities.

$$\mathrm{Ra}_x = \mathrm{Gr}_x \mathrm{Pr} = \frac{g\beta}{\nu\alpha}(T_s - T_\infty)x^3 \tag{2.17}$$

For free convection near a vertical wall, this number is:

- x = Characteristic length (in this case, the distance from the leading edge)
- Ra_x = Rayleigh number at position x
- Gr_x = Grashof number at position x
- Pr = Prandtl number
- g = acceleration due to gravity
- T_s = Surface temperature (temperature of the wall)
- T_∞ = Quiescent temperature (fluid temperature far from the surface of the object)
- ν = Kinematic viscosity
- α = Thermal diffusivity
- β = Thermal expansion coefficient

2.9 REYNOLDS NUMBER

In fluid mechanics and heat transfer, the Reynolds number Re is a dimensionless number which described the correlation between the inertial forces ($\mathrm{V}\rho$) with the viscous forces (μ / L). Reynolds numbers can be used to differentiate the differences between laminar flow (occur at low Reynolds number) and turbulent flow (occur at high Reynolds number). The differences are in laminar flow, the viscous forces are dominant, and is characterized by smooth, constant fluid motion, while in turbulent flow, it is dominated by inertial forces (tend to produce random eddies, vortices and other flow fluctuations).

2.9.1 Flow in Pipe

$$\mathrm{Re} = \frac{\rho \mathrm{V} D}{\mu} = \frac{\mathrm{V} D}{\nu} = \frac{Q D}{\nu A} \tag{2.18}$$

For flow in a pipe or tube, the Reynolds number is generally defined as shown above, where:

- V is the mean fluid velocity in (SI units: m/s)
- D is the diameter (m)
- μ is the dynamic viscosity of the fluid (Pa·s or N·s/m²)
- ν is the kinematic viscosity (ν = μ / ρ) (m²/s)
- ρ is the density of the fluid (kg/m³)
- Q is the volumetric flow rate (m³/s)
- A is the pipe cross-sectional area (m²)

2.10 NUSSELT NUMBER

Nusselt number is a dimensionless number relating to the Rayleigh number. It corresponds to 1 for initial convective motion and increases with an increased probability of convective motions occurring. The Nusselt number is the ratio of convective to conductive heat transfer. These heat transfers are measured across a surface of a test object. The conductive heat transfer is measured under the same conditions as the heat convection transfer except that the test object is within a stagnant

fluid. The Nusselt may be calculated from a characteristic length of the test object, the convective heat transfer coefficient of the test material and the thermal conductivity of the fluid.

$$\mathrm{Nu}_L = \frac{hL}{k_f} = \frac{\text{Convective heat transfer coefficient}}{\text{Conductive heat transfer coefficient}} \tag{2.19}$$

where:

- h = convective heat transfer coefficient
- L = characteristic length
- k_f = thermal conductivity of the fluid

2.11 RAYLEIGH BERNARD CONVECTION

Rayleigh Bernard system is an instability fluid that contain in two horizontal parallel plates. The bottom plate is heated randomly to produce a fixed temperature different while the upper plate is still in room temperature. When the temperature different between the upper and bottom plate is lower than critical value, the heat is transported by conduction only. Heat convection established when the temperature different between upper and lower plate is exceed the critical value. In Figure 2.6 and Figure 2.7 shows the relationships between top and bottom plates.

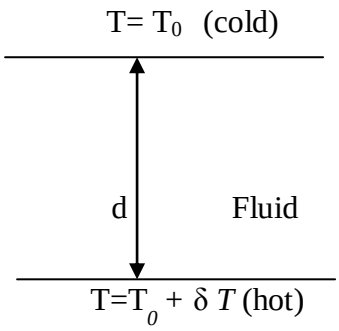


Figure 2.6: Boundary condition for Rayleigh Bernard Convection

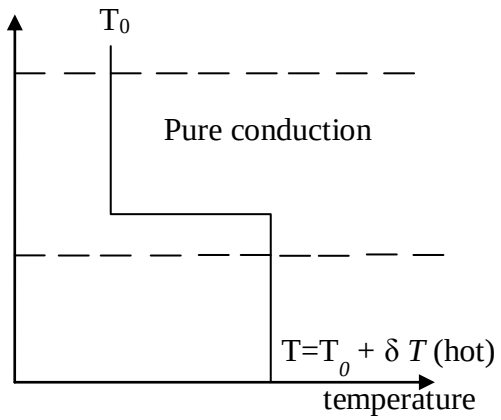


Figure 2.7: The relationship between top and bottom plates

The transport of hot fluid up and cold fluid down in the absence of convection and the temperature gradient is constant.

Two cases of interest:

- ∂T small: no convective motion, due to stabilizing effects of viscous friction.
- ∂T large: convective motion occurs.

Rayleigh in 1916 developed the theory which found the condition for the instability with two free surfaces. The instability would results if the temperature gradient $\beta = -\frac{dT}{dz}$ was large enough, so the Rayleigh number exceeds the critical value.

$$R_a = \frac{\alpha\beta g d^4}{k\nu} \tag{2.20}$$

Sources: Tadashi Watanabe 2004.

Where g is the acceleration due to gravity, α is the coefficient of thermal expansion, d is the depth of the chamber and k, ν are the thermal diffusivity and kinematics viscosity respectively. The parameter Ra represents the ratio of destabilizing buoyancy force to the stabilizing viscous force.

2.11.1 Rayleigh Bernard Convection Problem

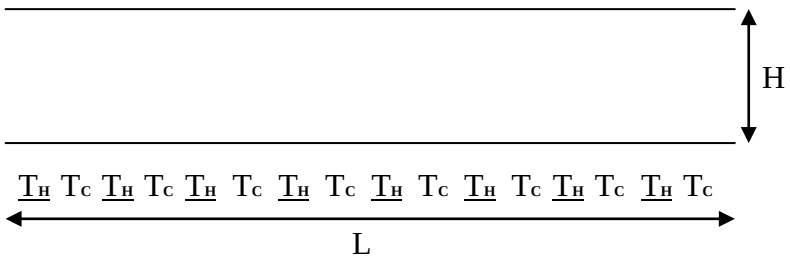


Figure 2.8: Geometry and boundary condition for Rayleigh Bernard problem.

where $\underline{T_H}$ is hot temperature

T_c is cool temperature

$$\frac{L}{H} = 10 \tag{2.21}$$

The ratio between the length, L and height, H is shown in equation 2.21 and the geometry and boundary condition for Rayleigh Bernard Convection show is Figure 2.8. The boundary condition for the Rayleigh Bernard Convection will be used in the simulation. The various values of length and height for Rayleigh Bernard Convection will be discussed in Chapter 4. The simulation results of the Rayleigh Bernard Convection also will be discussed in Chapter 4.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

In this chapter, will be show and explain briefly on step or procedure on how the simulation from start until get the result desired according to final year project which is Simulation of Rayleigh Bernard Convection using Lattice Boltzmann Method. Methodology is very important to conduct any simulation to make sure the process during the simulation run smoothly from the very beginning until we finish the simulation with the result desired. Figure 3.1 shows the steps by steps methodology in final year project. Furthermore, the boundary condition of Poiseulle flow, Couette flow and Porous Couette flow will be discussed in this chapter. After that, the simulation for every flow (Poiseulle, Couette and Porous Couette) also will be discussed. From the simulation result, the graph for each flow will be created.

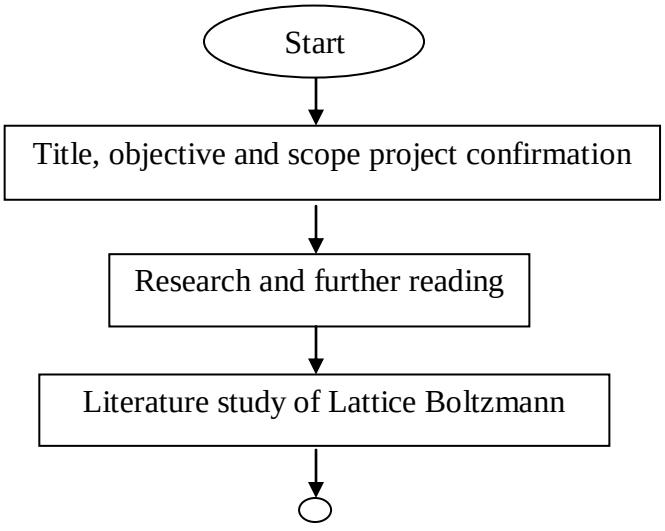


Figure 3.1: Flowchart of PSM

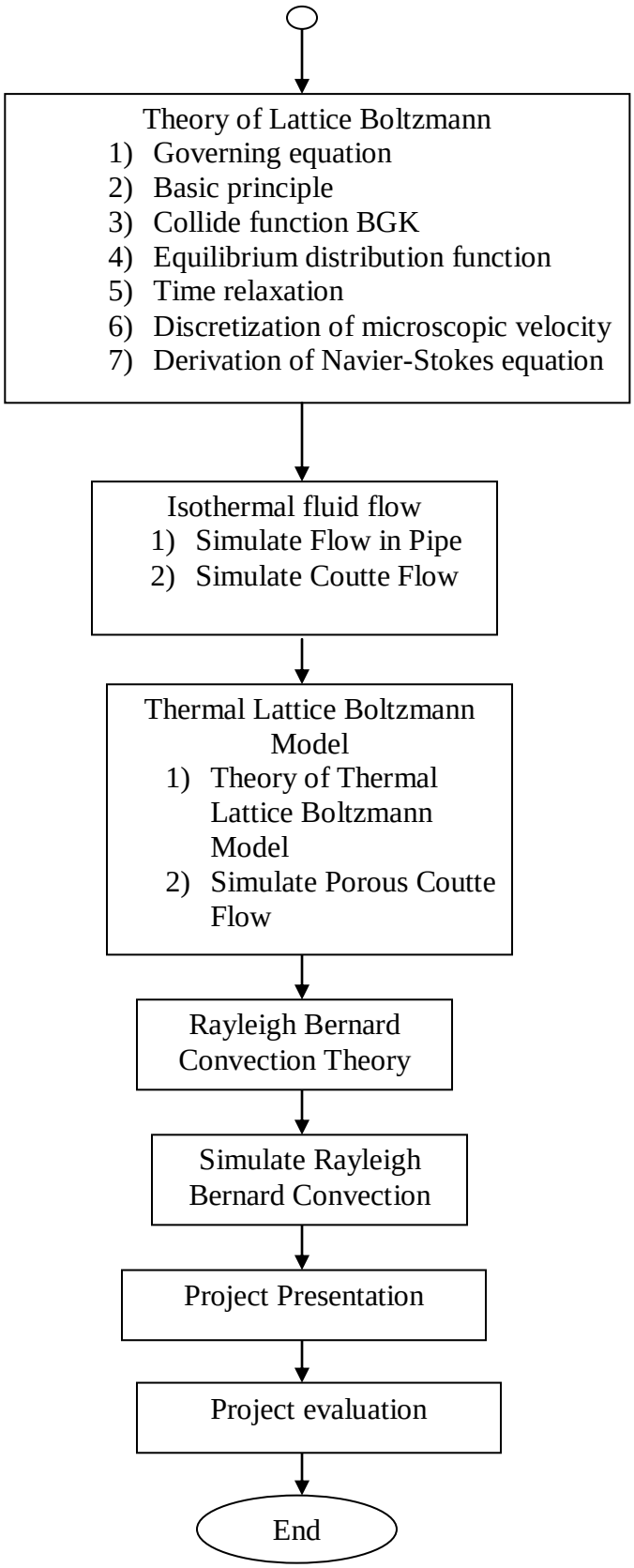


Figure 3.1: continue

3.2 COMPUTATIONAL FLUID DYNAMICS (CFD)

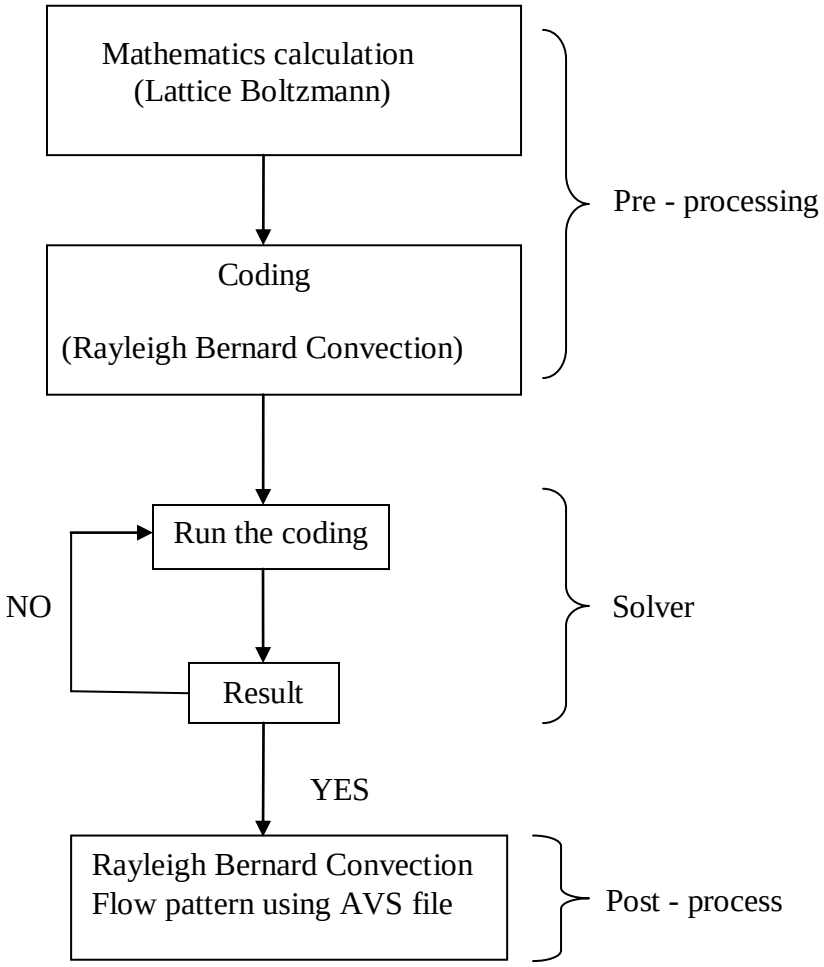


Figure 3.2: Computational Fluids Dynamics (CFD) flowchart

The simulation is conducted to study flow pattern of Rayleigh Bernard Convection using Compaq Visual Fortran software. By using CFD (Computational Fluid Dynamics), the steps or procedure can be explained from starting until get the results. CFD contains three main steps which are pre – processing, solver and post – process. CFD flowchart is shown in Figure 3.2. For the processing step, consisted two steps which are mathematical calculation and coding. Mathematical calculation is based to the Lattice Boltzmann equation. For this report, the coding is related to the problem whish is Rayleigh Bernard Convection. For solver consist two steps which are run the coding and get the result from the simulation. For the post-process steps, consist one step which

is the result from the Rayleigh Bernard Convection problem. AVS file is use to get the flow pattern of the Rayleigh Bernard Convection. AVS file is software that has been used to get the streamlines and the isotherms lines. Streamlines takes from the streamline absolute function while for the isotherms lines takes from the temperature function. The flow patterns for streamlines and isotherms will be discussed in Chapter 4.

3.3 ORIGINAL LBM ALGORITHM

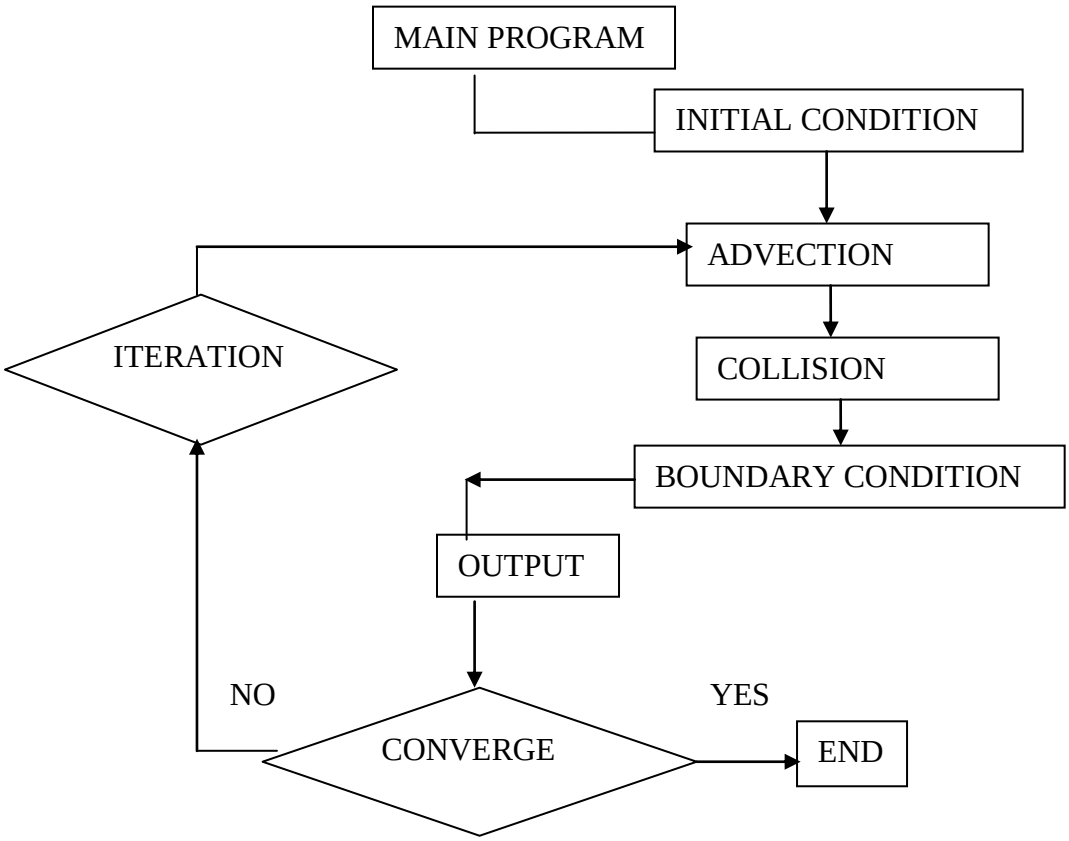


Figure 3.3: Original LBM algorithm flowchart

The algorithm flowchart for LBM is shown in Figure 3.3. From the algorithm flowchart, it consist advection and collision process. The initial values of density distribution f are specified at each grid point. Then, the system expands in the four steps. Four steps for algorithm flowcharts are advection process, collision process,

boundary conditions and lastly the simulation results. Firstly, the advection process is solved by applying the streaming process of the density distribution function. Advection process is initial position of the particles and then the particles stream (moves).Second steps is the collision process. Collision process is solved by Bhatnagar-Gross-Krook **(**BGK) collision model. The Bhatnagar-Gross-Krook collision model and equation have discussed earlier in Chapter 2. Third steps are defining the boundary conditions. The boundary conditions are defined based on the bounce back boundary conditions. Bounce back conditions is the possible direction that the particles will go after the collision process. There are two models of possible direction. First is 9-discrete velocity model (D2Q9) and second is 4-discrete velocity model (D2Q4). D2Q9 and D2Q4 have discussed earlier in Chapter 2. Last steps are the simulation process. The simulation results are based on steady state conditions. Poiseulle Flow, Couette Flow and Porous Couette Flow simulation results will discuss in subchapter 3.3. Rayleigh Bernard Convection simulation process will be discussed in Chapter 4.

3.4 SIMULATION RESULTS

3.4.1 Poiseulle Flow

In this numerical simulation, the velocity is not set because the Poiseulle flow is driven by a pressure gradient between inlet and outlet end of the channel. In this case the pressure gradient is set between the inlet and outlet end of the channel. The density is setting to get the different values between inlet and outlet end of the channel. The bounce back boundary conditions are applied at the top and bottom walls. Figure 3.3 show the boundary condition of the Poiseulle flow.

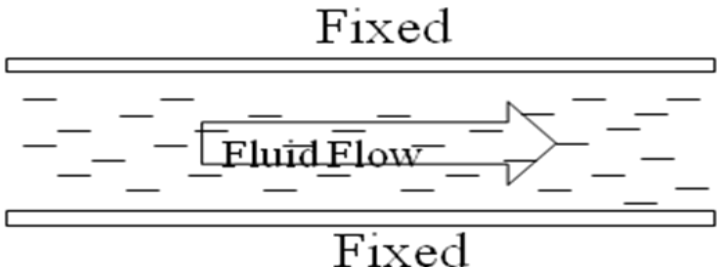


Figure 3.4: Poiseulle flow boundary condition

$$\sqrt{\sum_i \sum_x [f_i(x,t+1) - f_i(x,t)]^2 \Big/ \sum_i \times M \times N} \qquad (3.1)$$

The convergence criterion for Poiseulle flow is shown in equation 3.1 where M and N are the mesh number in x and y direction respectively. The measurement of velocity (u) and pressure along the channel (both after achieved the steady state) are measured in this simulation.

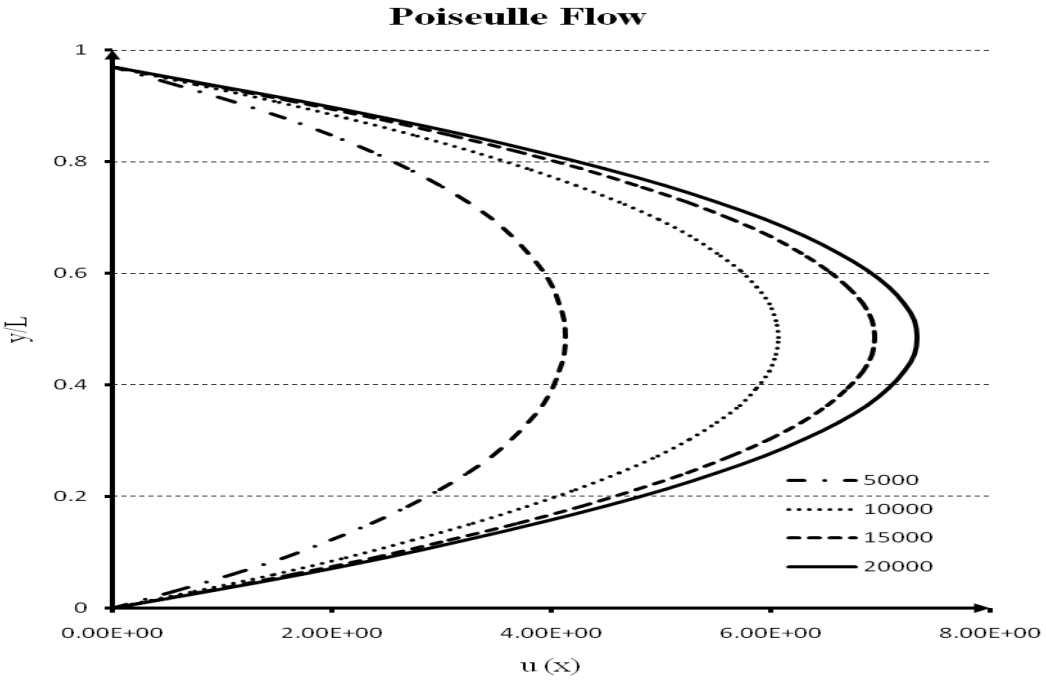


Figure 3.5: Poiseulle flow graph

Figure 3.5 show the Poiseulle flow simulation results. The density change between the two ends along the centerline of the channel is a straight line. In the inset the velocity profile across the channel is displayed for the stationary state. A lattice size 4 x 33 and time relaxation, τ_f = 0.55 is used in this test simulation. The result shows that the parabolic of the channel flow is obtained when the flow achieved the steady state.

3.4.2 Couette Flow

In this numerical simulation, the constant velocity is set at the top plate while the bottom plate is fixed. The initial conditions is related or similar to a null velocity everywhere expect on the top boundary, where the velocity is u = (U, 0). In this case, there is no pressure gradient between inlet and outlet end of the channel because the velocity is set to get the flow for this test simulation. The velocity on the plate is set at U = 0.05 (boundary condition in LBM units) and channel width is L = 32. Figure 3.6 show the boundary condition of the Couette flow.

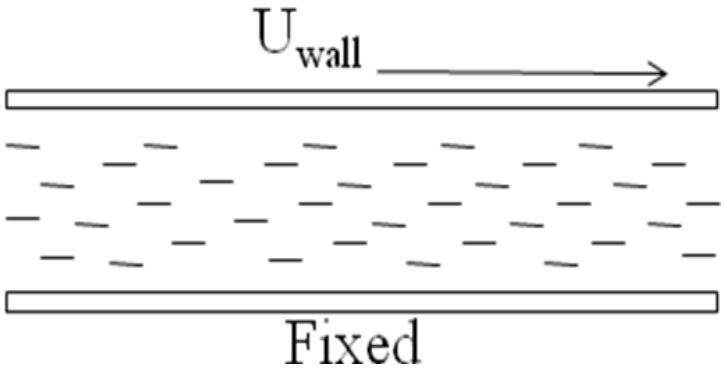


Figure 3.6: Couette flow boundary condition

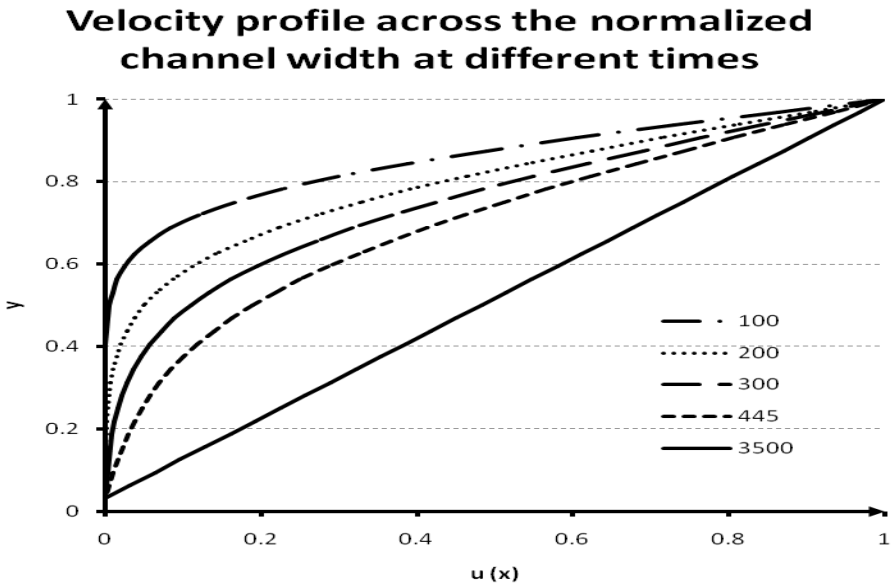


Figure 3.7: Couette flow graph

Figure 3.7 show a normalized velocity profiles for different times. The lattice size for this simulation is 4 x 32 and the relaxation time, τ_f = 1.0 is used. The velocity profiles are plotted at various values of times (t = 100, 200, 300, 445, 3500). All values of times are in LBM units. From the graph, x-axis is for periodic boundary conditions. The result shows that the velocity increases in linear from zero at bottom plate to U at top plate at the steady state.

3.4.3 Porous Couette Flow

This numerical flow has two finite parallel flat plates which the upper plate is cool, T_C and moves at speed U while bottom plate is hot, T_H and it is fixed. There is space by a distance of L between the two parallel plates. A constant normal flow of fluid is injected through the bottom hot plate and withdrawn at the same rate from the upper plate. Figure 3.8 show the boundary condition of Porous Couette flow boundary condition.

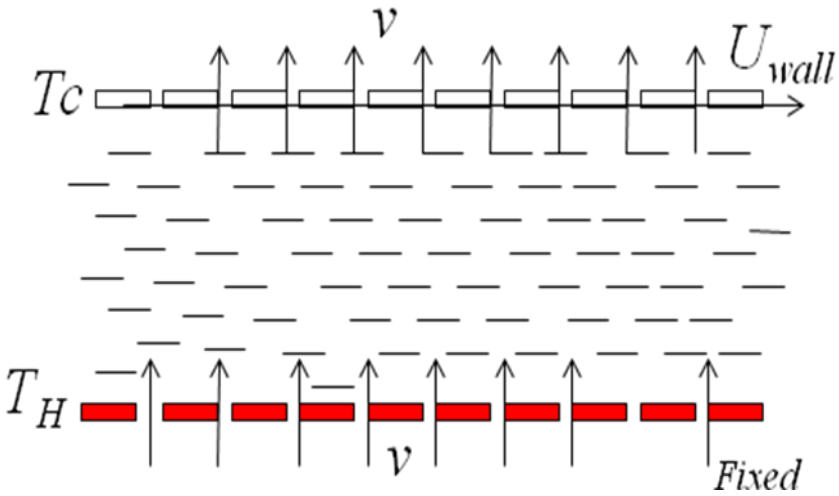


Figure 3.8: Porous Couette flow boundary condition

The analytical solution of the velocity field in steady state is given by Eq. (3.2) where Re is the Reynolds number (based on the inject velocity, v_o.)

$$u = U\left(\frac{\left(e^{\mathrm{Re}\,\bar{y}/l} - 1\right)}{e^{\mathrm{Re}} - 1}\right) \tag{3.2}$$

The temperature profile in steady state is given by Eq. (3.3) where ΔT = T_H – T_C is the temperature difference between the hot and cool walls and Pr = ν/X is the Prandtl number.

$$T = T_C + \Delta T\left(\frac{\left(e^{\mathrm{PrRe}\,\bar{y}/l} - 1\right)}{e^{\mathrm{PrRe}} - 1}\right) \tag{3.3}$$

Periodic boundary conditions are used at the inlet and outlet of the channel. For velocity, the non-equilibrium bounce back boundary condition is used. The non-equilibrium bounce back boundary condition is also used for the temperature.

The normalized temperature profile for various values of Reynolds number (Re = 5, 10, 20, 30), Ra = 100 and Pr = 0.71 and are shown in Figure 3.9. Prandtl number = 0.71 because in this simulation, air is used.

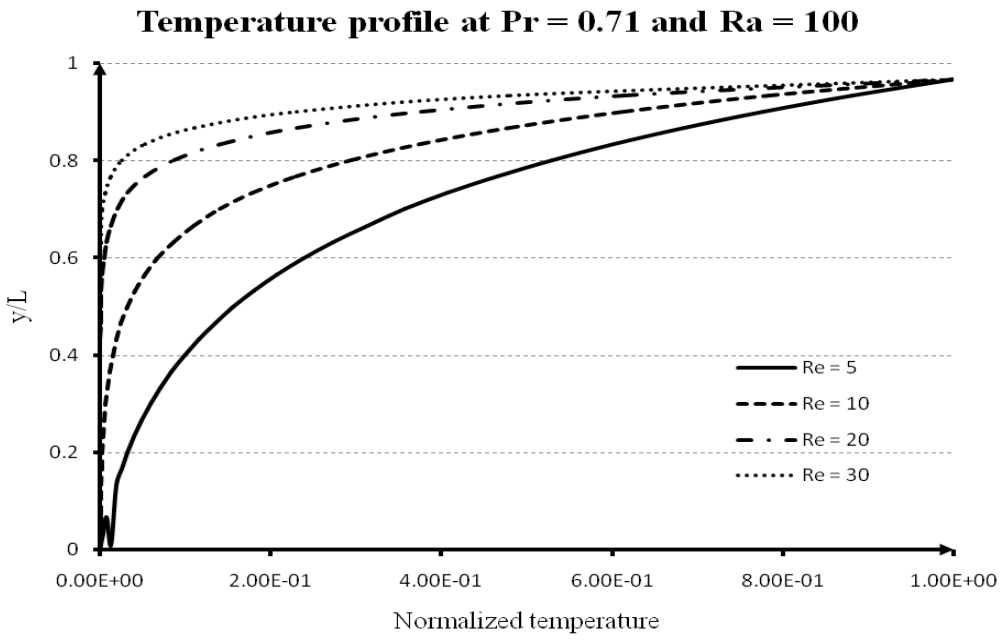


Figure 3.9: Porous Couette flow graph with different Re value

Figure 3.10 shown the result for Ra = 100, Re = 10 and Pr = 0.2, 0.8 and 1.5. The results in Figure 3.9 and Figure 3.10 show that each temperature and velocity profile have agreed the analytical solution.

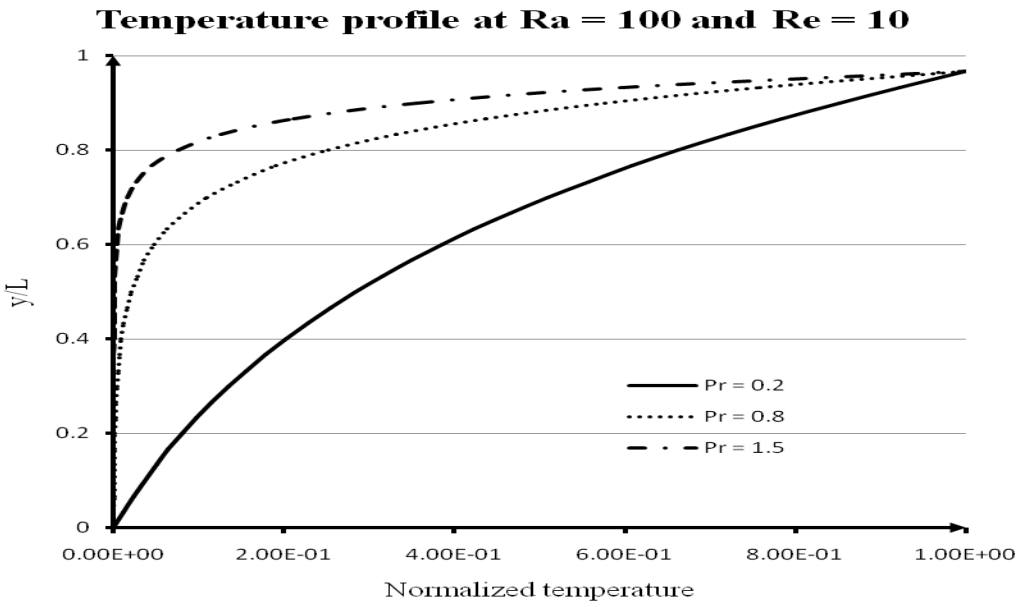


Figure 3.10: Porous Couette flow graph with different Pr value

To show that this model is suitable and numerically stable for a wide range of Rayleigh number, the computations for Ra = 10 till Ra = 60000 at Pr = 0.71 and Re = 10 have been done. The result is in Figure 3.11.

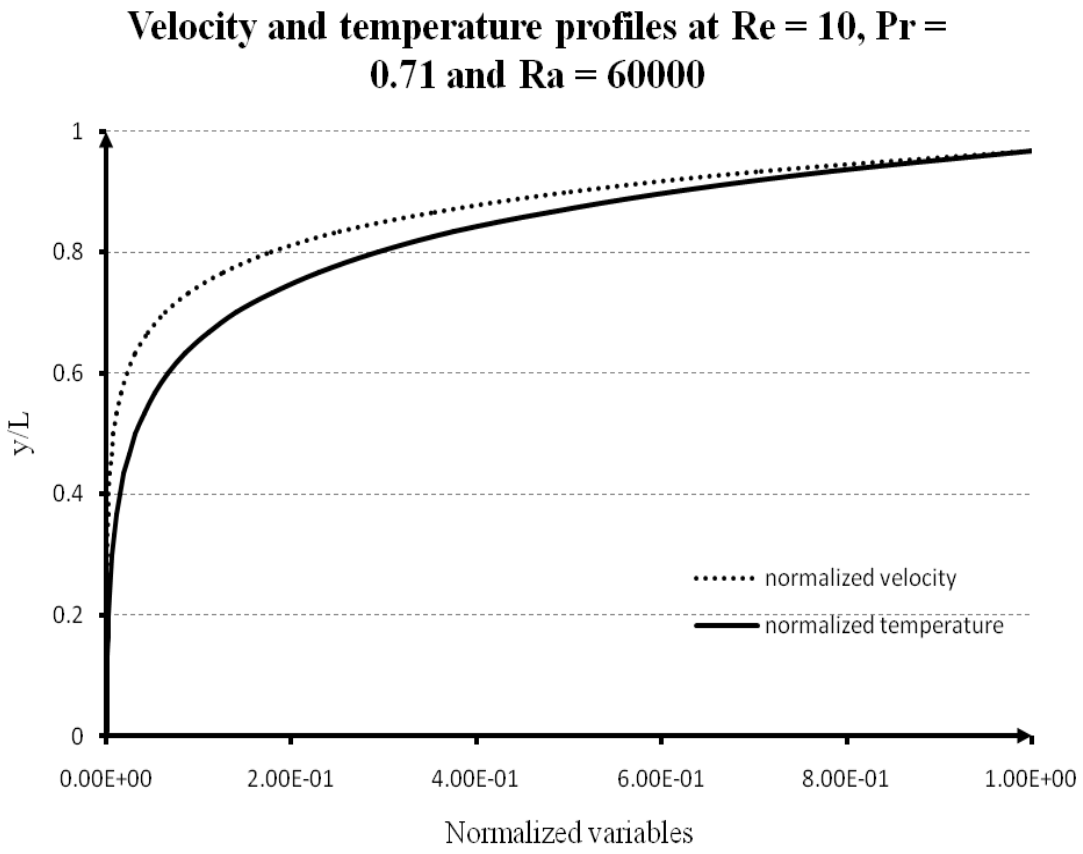


Figure 3.11: Porous Couette flow graph with Ra = 60000

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

In this chapter the results from the simulation will be analyze according to the project objectives. The data that had been gathered from the simulation will be used to compare the flow pattern for different Rayleigh number. From the data too, graph to show the relationship between the Nusselt number and Rayleigh number will be created. In this chapter too, the difference between the result and the graph for different Rayleigh number will be discussed. Furthermore, the flow pattern for different Rayleigh number also will be discussed in this chapter. From that the suitable flow pattern will be choose based on the Nusselt number.

4.2 RAYLEIGH BERNARD CONVECTION

Rayleigh Bernard system is an instability fluid that contain in two horizontal parallel plates. The bottom plate is heated randomly to produce a fixed temperature different while the upper plate is still in room temperature. When the temperature different between the upper and bottom plate is lower than critical value, the heat is transported by conduction only. Heat convection established when the temperature different between upper and lower plate is exceed the critical value.

D2Q9 and D2Q4 velocity model has been used. In this problem, the Prandtl number is fixed at 0.71. The value of 0.71 is for air. While for Rayleigh number is varied. In this problem, the value of Rayleigh is become the manipulated variables,

(Ra = 1000, 10000, 100000 and 1000000). The flow pattern for different Rayleigh number is studied.

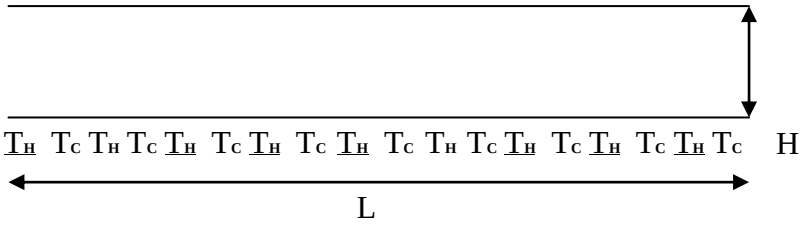


Figure 4.1: Geometry and boundary condition for Rayleigh Bernard problem.

where $\underline{T_H}$ is hot temperature

T_c is cool temperature

$$\frac{L}{H} = 10 \tag{4.1}$$

The ratio between the length, L and height, H is 10. The grid dependence test for the length, L is set to 100, 150, 200, and 250. The relationship between length and height is shown in Table 4.1.

Table 4.1: Relationship between length, L and height, H.

Length, L	Height, H
100	10
150	15
200	20
250	25

4.3 NUSSELT NUMBER AND RAYLEIGH BERNARD NUMBER

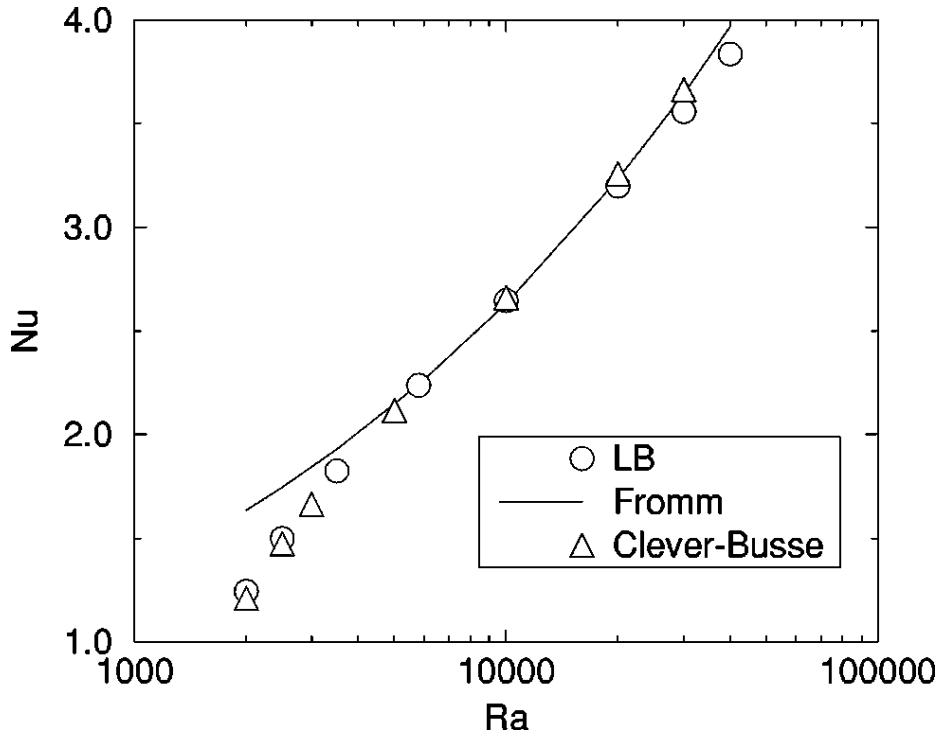


Figure 4.2: Relationship between Nusselt numbers and Rayleigh number

Sources: Tadashi Watanabe 2004.

Figure 4.2 shows that the steady-state Nusselt number as function of the Rayleigh number in 2D simulations. The LBE results agree with that of Clever and Busse for Rayleigh number less than 20 000 (X. Shan, 1997).

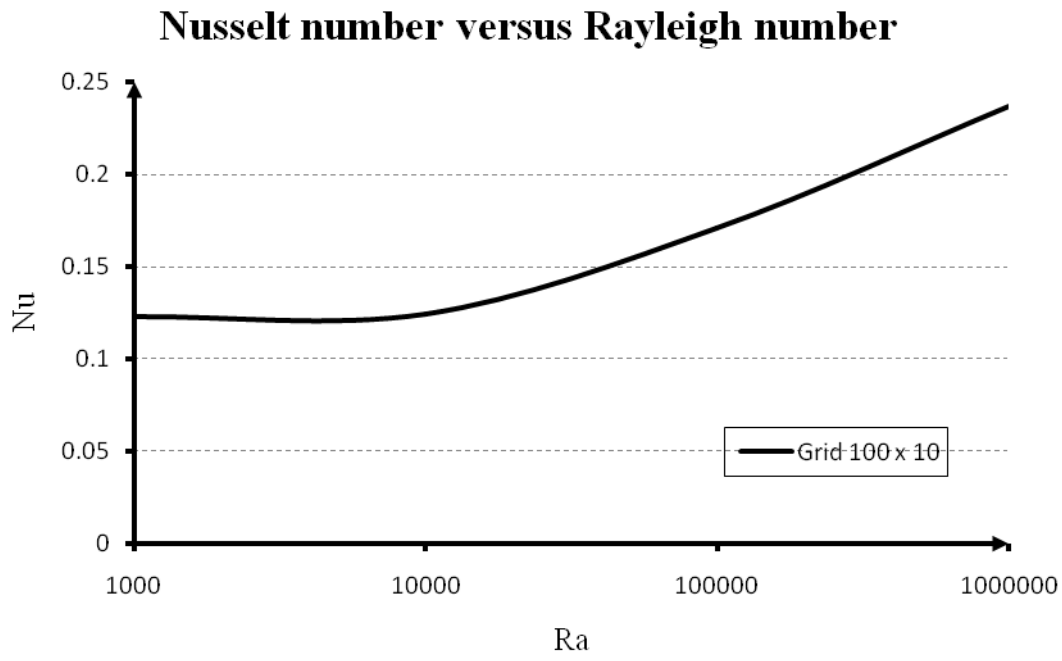


Figure 4.3: Relationship between Nusselt numbers with Rayleigh number for grid 100

From figure 4.3 shows that when Rayleigh number, Ra increase, the Nusselt number, Nu also increase. This means the Lattice Boltzmann equation for grid 100 x 10 results agree with Clever and Busse for Rayleigh number less than 20000 (X. Shan, 1997). Clever and Busse state that, Nusselt number increase according to the increasing value of Rayleigh number (X. Shan, 1997). That means that Nu number directly perpendicular with Ra number. Others grid for the boundary condition is show in Figure 4.4. From four grids, only grids 100 x 10 agree with Clever and Busse. While others grids disagree with Clever and Busse.

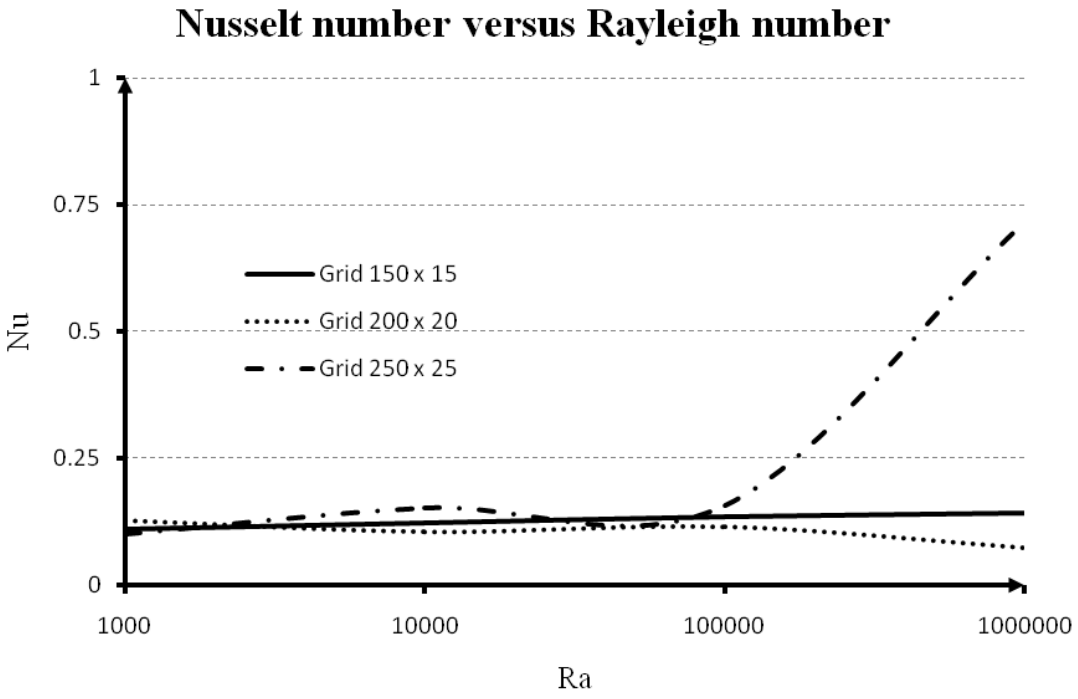


Figure 4.4: Relationship between Nusselt number and Rayleigh number

From figure 4.4 shows that when the Rayleigh number value increase, Nusselt number value will give the inconsistent value. Therefore, the Lattice Boltzmann equation for grid 150 x 15, 200 x 20 and 250 x 25 results disagree with Clever and Busse for Rayleigh number less than 20000. From graph 4.3 and 4.4, only for grid 100 x 10 the Lattice Boltzmann equation results agree with Clever and Busse while other grids (150 x 15, 200 x 20 and 250 x 25) results disagree with Clever and Busse. This is because there will be other possible causes or factors that make other grids did not agree with Clever and Busse Nusselt number versus Rayleigh number graph. The causes or factors will be discussed in Chapter 5 under the recommendation subchapter.

4.4 GRID DEPENDENCE TEST

Grid dependence test is produce by perform the graph of Nusselt number versus grid. From the graph, the optimum value will take as the flow pattern analysis. The purpose for grid dependence test is to find the optimum grid for each Rayleigh number. The relationship between grid number and Nusselt number is show in Table 4.2.

Table 4.2: Relationship between grid and Nusselt number for Ra = 1000.

Grid	Nusselt number
100 x 10	0.12279
150 x 15	0.12323
200 x 20	0.12702
250 x 25	0.10056

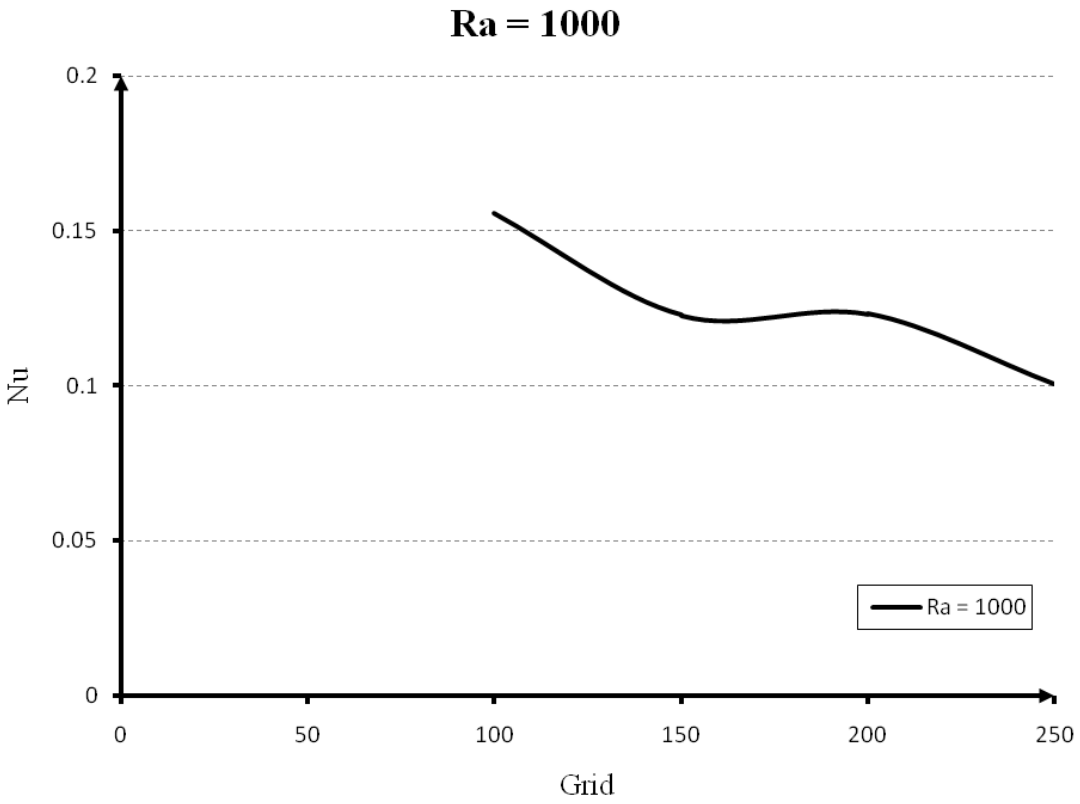


Figure 4.5: Graph Nusselt number versus grid for Ra = 1000

For Rayleigh number, Ra = 1000, four level grid (100, 150, 200, 250) have been tested and the shown in Figure 4.5. From the figure, grid 150 x 15 have been choose as the result for Ra = 1000. This is because this grid gives the optimum value for Nusselt number versus the grid as the result for the flow pattern. Optimum value here means the value that starting to flat or the lowest value for Nusselt number versus the grid.

Therefore, the effect of the flow pattern for Ra = 1000 (streamline and isothermal), just consider the grid for 150 x 15. The relationship between grid number and Nusselt number is show in Table 4.3.

Table 4.3: Relationship between grid and Nusselt number for Ra = 10000.

Grid	Nusselt number
100 x 10	0.12419
150 x 15	0.11057
200 x 20	0.10516
250 x 25	0.15233

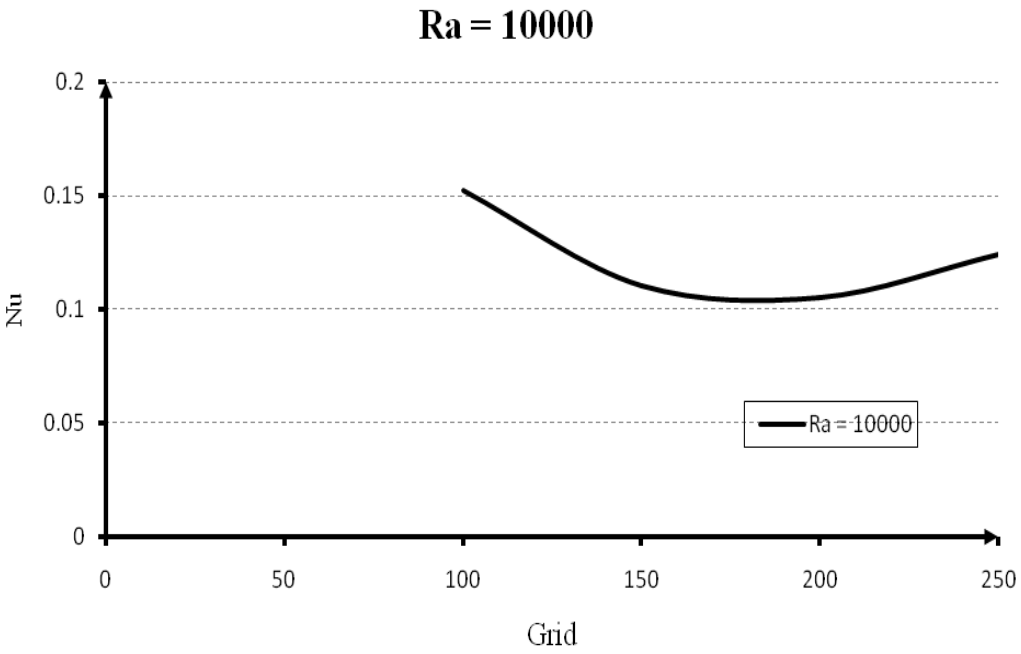


Figure 4.6: Graph Nusselt number versus Grid for Ra = 10000

For Rayleigh number, Ra = 10000, four level grid (100, 150, 200, 250) have been tested and the shown in Figure 4.6. From the figure, grid 150 x 15 have been choose as the result for Ra = 10000. This is because this grid gives the optimum value for Nusselt number versus the grid as the result for the flow pattern. Optimum value here means the value that starting to flat or the lowest value for Nusselt number versus

the grid. Therefore, the effect of the flow pattern for Ra = 10000 (streamline and isothermal), just consider the grid for 150 x 15. The relationship between grid number and Nusselt number is show in Table 4.4.

Table 4.4: Relationship between grid and Nusselt number for Ra = 100000.

Grid	Nusselt number
100 x 10	0.17055
150 x 15	0.13514
200 x 20	0.11522
250 x 25	0.15746

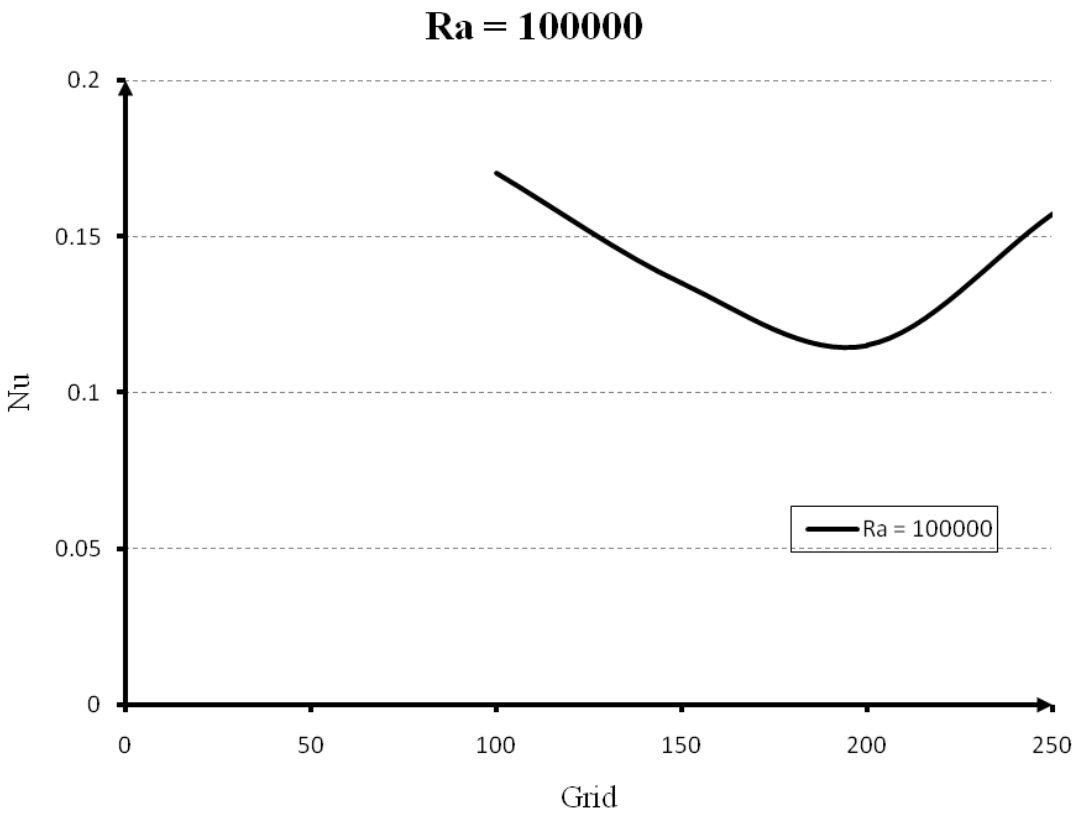


Figure 4.7: Graph Nusselt number versus Grid for Ra = 100000

For Rayleigh number, Ra = 100000, four level grid (100, 150, 200, 250) have been tested and the shown in figure 4.7. From the figure, grid 200 x 20 have been

choose as the result for Ra = 100000. This is because this grid gives the optimum value for Nusselt number versus the grid as the result for the flow pattern. Optimum value here means the value that starting to flat or the lowest value for Nusselt number versus the grid. Therefore, the effect of the flow pattern for Ra = 100000 (streamline and isothermal), just consider the grid for 200 x 20. The relationship between grid number and Nusselt number is show in Table 4.5.

Table 4.5: Relationship between grid and Nusselt number for Ra = 1000000.

Grid	Nusselt number
100 x 10	0.23680
150 x 15	0.14289
200 x 20	7.35887E-2
250 x 25	0.71564

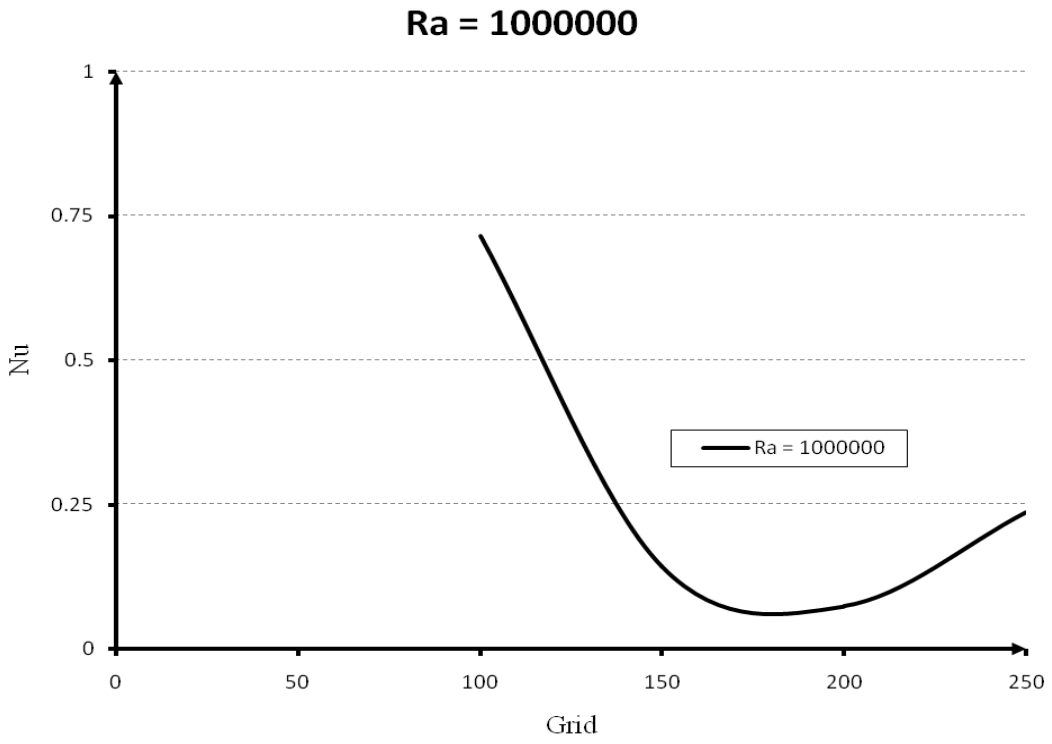


Figure 4.8: Graph Nusselt number versus Grid for Ra = 1000000

For Rayleigh number, Ra = 100000, four level grid (100, 150, 200, 250) have been tested and the shown in Figure 4.8. From the figure, grid 200 x 20 have been choose as the result for Ra = 100000. This is because this grid gives the optimum value for Nusselt number versus the grid as the result for the flow pattern. Optimum value here means the value that starting to flat or the lowest value for Nusselt number versus the grid. Therefore, the effect of the flow pattern for Ra = 100000 (streamline and isothermal), just consider the grid for 200 x 20.

4.5 VELOCITY PROFILE

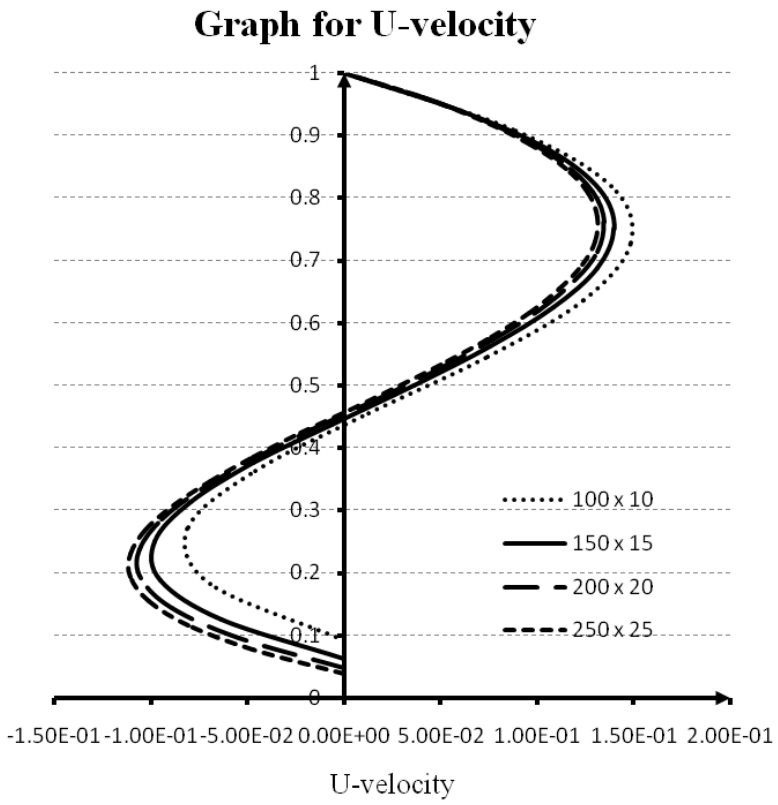


Figure 4.9: Horizontal velocity profile at Ra = 100000

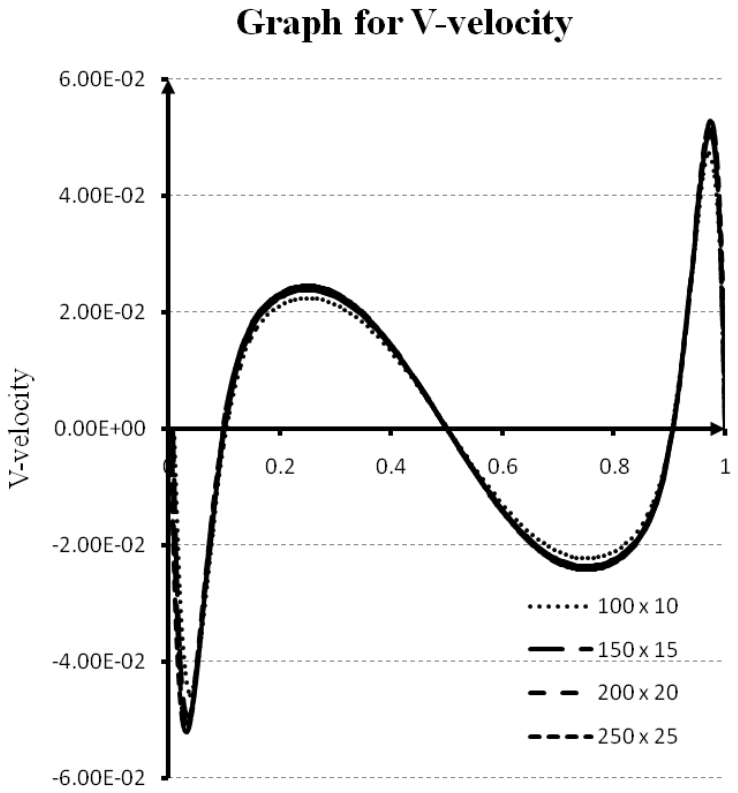
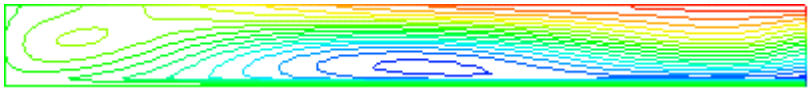


Figure 4.10: Vertical velocity profile at Ra = 100000

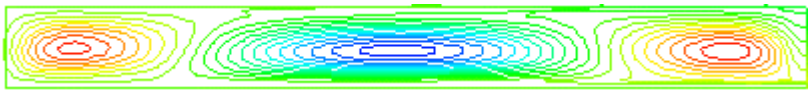
Figure 4.9 and Figure 4.10 shows the dimensionless horizontal and vertical velocity profile at Rayleigh number = 100000. Both figures are the velocity profile for each grid (100 x 10, 150 x 15, 200 x 20 and 250 x 25). The vertical and horizontal velocity profile is increasing from the bottom to the top walls and development of narrow boundary layers alongs the top wall. The peak values for both Figures (horizontal velocity and vertical velocity) are because of the intensified activities. Increasing the grid number is proportional to the increasing the Rayleigh number and increasing buoyancy force effect (C. S. Nor Azwadi, 2007). Both figures show the different velocity direction for top and bottom walls that correspond to the vortex development.

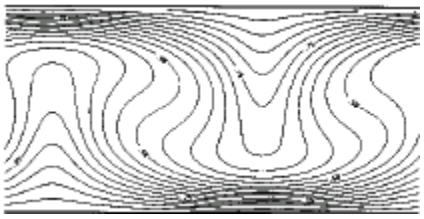
4.6 EFFECT OF THE RAYLEIGH NUMBER

4.6.1 Flow Pattern and Flow Intensity



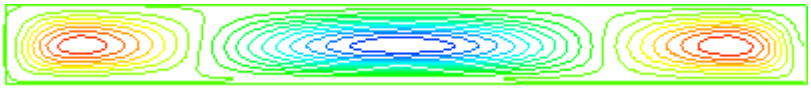
(a) Ra = 1000



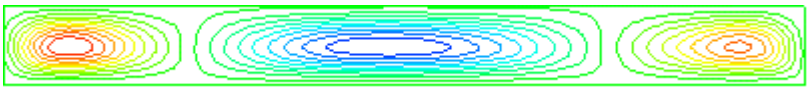


(b) Comparison with previous study for Ra = 10000

Sources: Xiaowen Shan 1997



(c) Ra = 100000



(d) Ra = 1000000

Figure 4.11: Streamline with different Rayleigh number

The streamlines shown in Figure 4.11 in which subfigures were arranged going down with increasing Rayleigh number. The temperature different between the upper

and bottom walls will cause the buoyant effects. These will make the recirculating vortices are formed which are clearly demonstrated by the closed streamlines. From the streamlines shown in Figure 4.11, the red lines shows the highest temperature followed by orange lines, yellow lines, green lines, bright blue lines and the lowest temperature is in dark blue lines.

The temperature difference between the upper and bottom walls will cause the temperature gradient in fluid; consequently density difference induced fluid motions. The hot fluid rises form the bottom until to the top of the plate and moves outwards along the top wall before turn downwards follow the sidewalls due to the effect of cooling.

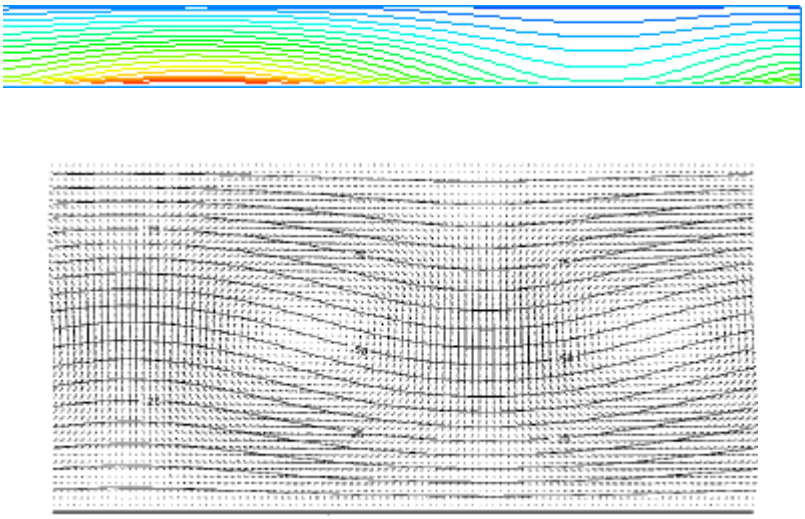
At lower Rayleigh number Figure 4.11 (a) and Figure 4.11 (b) the temperature different between the upper and bottom plate is lower than critical value, the heat is transported by conduction only. The vortices become larger and become more stretched to match the geometries of the enclosure as the Rayleigh number is increased. As increasing the Rayleigh number, the cores of the vortices also will slightly increased.

The streamlines become more packed to the upper and bottom walls as the Rayleigh number increased. The enhancement of the flow intensity resulted from the augmented flow area is compensated for the attenuation due to additional friction. Therefore, at both Ra = 1000 and 10000 the maximum stream functions are nearly identical. The vortices increased as the Rayleigh number increased because the higher fluid velocity which contributes to better the overall heat transfers.

At higher Rayleigh number Figure 4.11 (c) and Figure 4.11 (d) the temperature different between the upper and bottom plate is exceed the critical value, heat convection will established. The isotherms are distorted more because of the stronger convection effect that leads the stable stratification of the isotherms when increase the Rayleigh number. The magnitude of the velocity circulating in the enclosure increases as the Rayleigh number increase. This will make the thickness of the thermal boundary between the upper and bottom plates become larger.

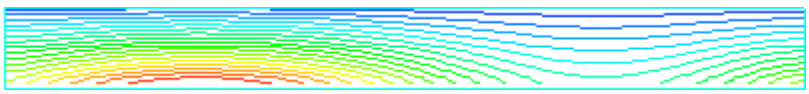
The vortices become larger and become more stretched to match the geometries of the enclosure as the Rayleigh number is increased. As increasing the Rayleigh number, the cores of the vortices also will slightly increased. The streamlines become more packed to the upper and bottom plates as the Rayleigh number increased. As the Rayleigh number is further increased (from Ra = 100000 to Ra = 1000000), the additional friction is negligible and the enhanced natural convection becomes dominant. Therefore the maximum stream functions at higher Rayleigh number is greater than at lower Rayleigh number.

4.6.2 Heat Transfer

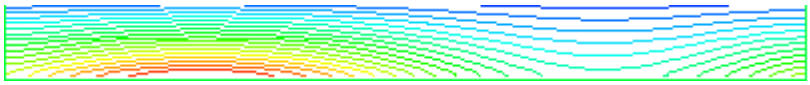


(a) Comparison with previous study for Ra = 1000

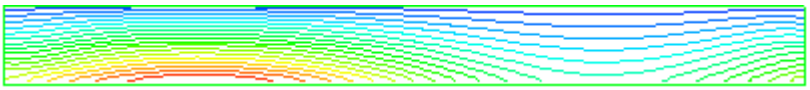
Sources: Xiaowen Shan 1997



(b) Ra = 10000



(c) Ra = 100000



(d) Ra = 1000000

Figure 4.12: Isotherms with different Rayleigh number

Results show above, is the isotherms for different Rayleigh number when the partially heated located at the bottom. The streamlines shown in Figure 4.12 in which subfigures were arranged going down with increasing Rayleigh number. The temperature difference between the upper and bottom walls will cause the temperature gradient in fluid; consequently density difference induced fluid motions and cause of the buoyant effects. The hot fluid rises form the bottom until to the top of the plate and moves outwards along the top wall before turn downwards follow the sidewalls due to the effect of cooling. From the isotherms shown in Figure 4.12, the red lines shows the highest temperature followed by orange lines, yellow lines, green lines, bright blue lines and the lowest temperature is in dark blue lines.

At lower Rayleigh number Figure 4.12 (a) and Figure 4.12 (b) the heat is transported by conduction only and the buoyancy driven by convection is very weak and unaffected. As the Rayleigh number increased the isotherms become more packed at the upper and lower plates. The isotherms become more concentrated to the upper region as increasing the Rayleigh number. Increased the Rayleigh number, the discrepancies for the pure conduction will become more considerable. The thermal boundary layer become thinner as the Rayleigh number increased because the higher fluid velocity which contributes to better the overall heat transfers.

At higher Rayleigh number Figure 4.12 (c) and Figure 4.12 (d) the heat is transported by convection and the separation of the thermal boundary layers is become closer. The isotherms along the heated bottom plate separate away from its upper zone and form a thermal plume impinging to the upper corner of the upper plate. This plume thermal plume formation become clear for Ra = 10000000. According to the asymmetric flow patterns, the isotherms for Ra = 100000 still follow the symmetric line but the symmetric line more closer compare to the lower Rayleigh number (Ra = 1000 and Ra = 10000). For Ra = 1000000, the symmetric line more closer compare to Ra = 100000. This shown that the isotherms for Ra = 100000 are more clear for the higher Rayleigh number.

4.7 DISCUSSIONS

In order to validate the current code, overall heat transfer rates of laminar natural convection in Rayleigh Bernard Convection enclosure and the computed velocity field were compared with experimental results in previous study. The streamlines and isotherms lines were set to be exactly identical to the experiment configurations. As shown in Figure 4.11 (b) and 4.12 (a), the streamline and isotherms were compared with the previous study. The numerical and the experimental data are connected in very good agreement for upper and bottom walls.

Cases with the top and bottom of the Rayleigh Bernard Convection boundary condition positioned were studied first. The Rayleigh Bernard Convection have two fixed walls where positioned at top and bottom. The effects of the Rayleigh number for Ra = 10^3, Ra = 10^4, Ra = 10^5 and Ra = 10^6 were investigated systematically. The aspect ratio for the boundary condition also was systematically investigated. The ratio, B is between the length, L and height, H of the boundary conditions as shows in equation 4.2.

$$\frac{L}{H} = 10 \tag{4.1}$$

The length and height that have been decided at 100, 150, 200 and 250 may put other values. This is because other values may produce better flow pattern of Rayleigh

Bernard Convection. The comparisons of the flow pattern and flow intensity and the heat transfer among all the Rayleigh numbers were presented.

The two dimensional Rayleigh Bernard Convection is simulated numerically using Lattice Boltzmann model. (2D) 9-discrete velocity model was used in this study while (3D) 27- discrete velocity model was used by (Tadashi Watanabe, 2004).

The Prandtl number is fixed at 0.71 because air is used in this study and the Rayleigh number $R_a = \frac{\alpha\beta g d^4}{k\nu}$ is varied where ΔT is the temperature difference between top and bottom walls while $\beta = -\frac{dT}{dz}$ is the temperature gradient. The top and bottom walls are applied no-slip boundary conditions using the bounce-back rule of the nonequilibrium density distribution. For the side boundaries, the periodic boundary condition is used.

The critical Rayleigh number at the static conductive state becomes unstable and given by the linear stability theory and this have been confirmed by the laboratory observations (X. Shan, 1997). The computation has to be carried out for a long time before the stable convection is developed since the development of the instability is very slow at near critical Rayleigh numbers. Computations were started from the static conductive state at several different Rayleigh numbers close to critical Rayleigh numbers to measure the critical Rayleigh number.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

Chapter one introduced about the relationship between macroscopic and microscopic approach have been discussed. For macroscopic approach, the solution is using Navier-Stokes equation (continuity and momentum equation) while for microscopic approach, the solution is using Lattice Gas Approach (LGA), Molecular Dynamic (MD) and Lattice Boltzmann method (LBM) by simulating fluid flows.

Chapter two the literature reviews for isothermal and thermal LBM has been discussed. The dimension less number (Prandtl number, Rayleigh number, Reynolds number, Nusselt number and Rayleigh Bernard Convection also has been discussed. In this chapter also discussed about the bounce back boundary condition. From the bounce-back boundary condition principle, 9-discrete velocity model (D2Q9) and 4-discrete velocity model (D2Q4) have been recovered.

The methodology (CFD flowchart) and the algorithm that used in the simulation to get the results have been explained in chapter three. Result for numerical solution isothermal LBM (Poiseulle flow and Couette flow) and thermal LBM (porous Couette flow) have been performed. Chapter four discussed about the flow pattern in the in Rayleigh Bernard Convection. These simulations used four difference grids and four different Rayleigh number. From four difference grids, the optimum value for each difference Rayleigh number have been figured by using the grid dependence test. The objective of the project was achieved when the simulation results agreed well with benchmark (previous study). For the streamlines, the vortices become larger and

become more stretched to match the geometries of the enclosure as the Rayleigh number is increased. For isotherms, when the Rayleigh number increased, isotherms become more packed at the upper plates and become more concentrated to the upper region.

Rayleigh Bernard Convection has become the favored example for the study of the spontaneous formation of structures in hydrodynamic systems. As using Rayleigh Bernard Convection code, it also can produce the flow patterns same as others codes for the convection. Rayleigh Bernard Convection is more accurate to the others ways because the fluid involves that placed between flat horizontal plates which are infinite in extent and are perfect heat conductors.

5.2 RECOMMENDATIONS

This subchapter will discussed about how to improve the simulation results and possible causes that might happen during run the simulation using Lattice Boltzmann Method (LBM). The recommendations must be take the action in next study to get more precisely results. The iteration for simulation result should increase to get more accurate results. This is because the data that obtained will be increased as the number of the iteration increase. So the data that gets from the simulation become more accurate. While running the project, there were many possible causes or factors that have been affected the simulation results accuracy. One of the factor is the simulation is converge early before finish the iteration value that have been decided. Decreased the condition of the convergence to make the simulation did not converge early. So for the next study, make the simulation coding that can accept all values of convergence. This can avoid the errors from occur and the coding can simulate until the iteration that have been decided. For this Rayleigh Bernard Convection coding, it is too long. Simplify the coding so the errors might be happen in the coding can be detect easier compare to long coding.

APPENDIX A

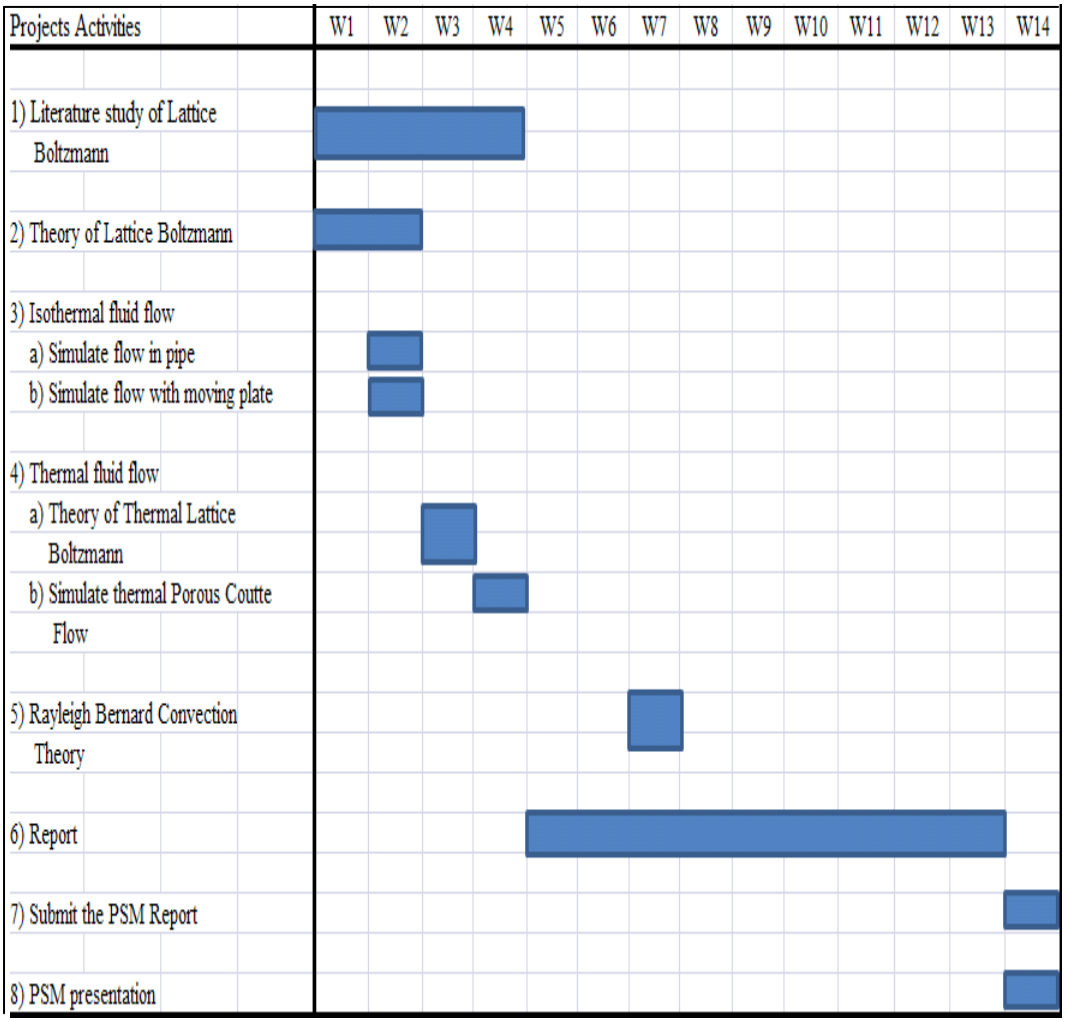


Figure 6.1: Gantt chart for PSM 1

APPENDIX B

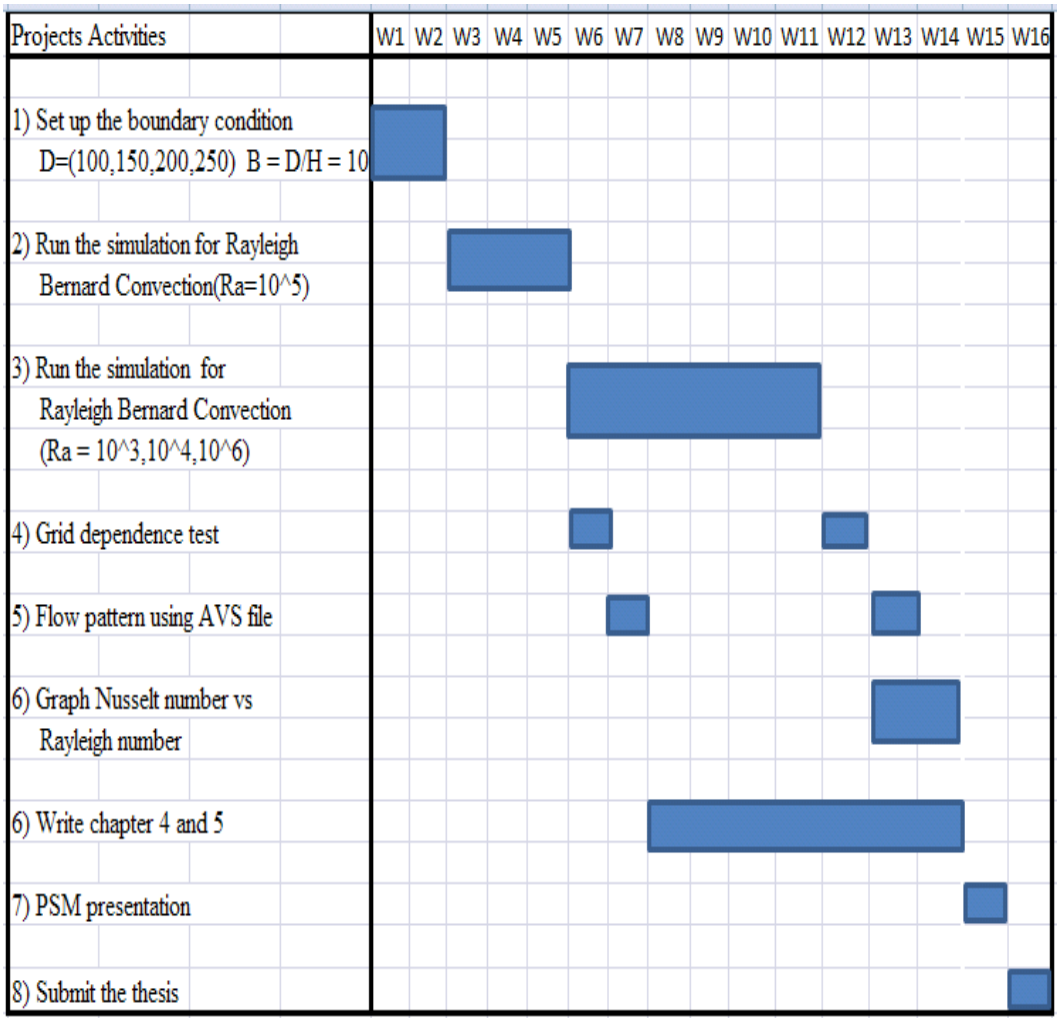


Figure 6.2: Gantt chart for PSM 2

APPENDIX C

Code Test for Rayleigh Bernard Convection

```
!  Lattice Boltzmann Method
!  based on Square Latteice
!
!  Natural convection from heated cubic
!
!  f5     f4     f3
!     .   |   .
!       . |f1
!  f6-----+----- f2
!       . | .
!     .   |   .
!  f7     f8     f9
!             Aspect ratio 3.0 (cubic diameter 80 elements)

          program natural_convection


        parameter(cd = 9, xd = 151, yd = 16, dd = 4)
           real*8  cx(1:cd), cy(1:cd)
           real*8  dx(1:dd), dy(1:dd)
           real*8  f(1:cd,1:xd,1:yd), f0(1:cd,1:xd,1:yd), ff(1:cd,1:xd,1:yd)
           real*8  g(1:dd,1:xd,1:yd), g0(1:dd,1:xd,1:yd), gg(1:dd,1:xd,1:yd)
           real*8  ux(1:xd,1:yd), uy(1:xd,1:yd), rho(1:xd,1:yd)
           real*8  temp(1:xd,1:yd),sumf,sumg
           real*8  vel(1:xd,1:yd)
           real*8  u0, rho0, pi,tauv,th,tc,pt,ra,nyu,ok,di,tauc,ri,re,lambda
!       real*8  velmaxn,velmaxn1

           u0 = 0.0                          ! wall velocity
           rho0 = 1.0                        ! constant density
           pt = 0.71                         ! Prandtl number
           ra = 100000                       ! rayleigh number
           th = 1.0                          ! bottom hot wall
           tc = 0.0                          ! up cool/initial wall temperature
!             uwell=0.1                              ! lid velocity
           lambda=0.5


           write(*,*) 'calculation start'
           write(*,*) 'nstep?'
     !     iteration step for display
           read(*,*) nstep
          write(*,*) 'nstep =',nstep
!             write(*,*) 'time relaxation?'
!             read(*,*) tauv
!             write(*,*) 'tauv =',tauv

           gra = (0.0577**2)/(yd)

           nyu = (gra*((xd)**3)*(th-tc)*pt/ra)**0.5
!             nyu = (gra*((xd)**3)*(th-tc)/(ri*re**2))**0.5 !mixed convection
!             nyu = (tauv - 0.5)/3.
           tauv = 3*nyu + 0.5
!             di  = (gra*((xd)**3)*(th-tc)/(ra*pt))**0.5 natural convection
           di = nyu/pt
           tauc = di + 0.5
!             gra = (nyu**2)*ra/(((xd)**3)*pt)
!             grad = (0.1731**2)/xd !incompressible limit
```

```
        write(*,*) 'nyu  =',nyu
        write(*,*) 'di   =',di
        write(*,*) 'Tauv =',tauv
        write(*,*) 'Tauc =',tauc
        write(*,*) 'gra =',gra
!       write(*,*) 'grad =',grad

        write(*,*) 'everything is ok?,press 0 if ok'
        read (*,*) ok

       cal   initial1
 (cx,cy,f,f0,g,g0,ux,uy,rho,temp,u0,rho0,th,tc,cd,xd,yd,dx,dy,dd,x1,x2,y1,y2,lambda)

       do iter = 1, 1000000

            call collide1 (f,f0,tauv,cd,xd,yd,g,g0,tauc,dd)

            call force (cx,cy,ux,uy,temp,f,f0,gra,cd,xd,yd)

            call stream1 (f,cd,xd,yd,g,dd)

            call boundary1 (f,f0,rho,u0,cd,xd,yd,g,g0,temp,dd,x1,x2,y1,y2)

            call         calc1
 (cx,cy,f,ux,uy,rho,u0,rho0,cd,xd,yd,temp,g,tc,th,dd,dx,dy,x1,x2,y1,y2,lambda)

            if (mod(iter,nstep) .eq.0) then
                        write(*,*) 'time step=', iter
                        write(*,*) 'velocity=',ux(xd/4,yd/4)
                        write(*,*) 'temperature= ',temp(xd/4,yd/4)
!                       write(*,*) 'velmax =', velmaxn1 - velmaxn
            end if

            call equilibrium1 (cx,cy,f0,ux,uy,rho,cd,xd,yd,dx,dy,g0,temp,dd)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
            !           check for convergence
          if (mod(iter,nstep) .eq.0) then

            write (*,*)'sum_f = ',sumf
                        do k = 1,cd
                                    do i = 1,xd
                                                do j = 1,yd
                                                            ff(k,i,j) = f(k,i,j)
                                      end do
                                    end do
                        end do

                        write (*,*)'sum_g = ',sumg
                        do k = 1,4
                                    do i = 1,xd
                                                do j = 1,yd
                                                            gg(k,i,j) = g(k,i,j)
                                      end do
                                    end do
                        end do

      endif

         if (mod(iter,nstep) .eq.1) then
                        sumf = 0.
                        do k = 1,cd
                                    do i = 1,xd
                                                do j = 1,yd
                                                            sumf = sumf +(f(k,i,j)-ff(k,i,j))**2
                                                end do
                                    end do
```

```
                end do

                sumf = (sumf/9.0*(yd)*(xd))**0.5

                !***** if converge*****::!
                if (sumf .le. 1.0d-3) then

                write(*,*)'solution_f converge'
!                               go to 100
                end if

                sumg = 0.
                do k = 1,4
                        do i = 1,xd
                                do j = 1,yd
                                        sumg = sumg +(g(k,i,j)-gg(k,i,j))**2
                                end do
                        end do
                end do

                sumg = (sumg/4.0*(yd)*(xd))**0.5

                !***** if converge*****::!
                if (sumg .le. 1.0d-3) then

                write(*,*)'solution_g converge'
!                               go to 100
                end if

        if (sumg .le. 1.0d-3 .and. sumf .le. 1.0d-3) then
        write(*,*)'both solution converge'

        go to 100
        end if

        end if

      end do

  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        !       check for convergence

!       if (mod(iter,nstep) .eq.0) then
!               do i = 1,xd
!                       do j = 1,yd
!                       vel(i,j) = ((ux(i,j)*(yd-1)/di)**2 + (uy(i,j)*(yd-1)/di)**2)**0.5
!                       end do
!               end do
!
!               velmaxn = abs(vel(1,1))
!               do i = 1,xd
!                       do j = 1,yd
!                               if(abs(vel(i,j)) .gt. velmaxn)  velmaxn = vel(i,j)
!                       end do
!               end do
!
!       endif

!       if (mod(iter,nstep) .eq.1) then
!               do i = 1,xd
!                       do j = 1,yd
!                       vel(i,j) = ((ux(i,j)*(yd-1)/di)**2 + (uy(i,j)*(yd-1)/di)**2)**0.5
!                       end do
!               end do

!               velmaxn1 = abs(vel(1,1))
```

```
!                       do i = 1,xd
!                               do j = 1,yd
!                                       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'
!                       go to 100
!                       end if

!             end do
! calculate nusselt number
      100               bnu = 0.0
!             do i = 1, xd
!                       do j = 1,yd
!                        if (i .eq. 1) then
! bnu = bnu + (((xd-1)/(di*(th-tc)*(xd)*(yd)))*((ux(i,j)*temp(i,j))-(di*(temp(i+1,j)-
 temp(i,j)))))
!                       else if (i .eq. xd) then
!  bnu = bnu + (((xd-1)/(di*(th-tc)*(xd)*(yd)))*((ux(i,j)*temp(i,j))-(di*(temp(i,j)-
 temp(i-1,j)))))
!                        else
!bnu= bnu + (((xd-1)/(di*(th-tc)*(xd)*(yd)))*((ux(i,j)*temp(i,j))- (di*(0.5*(temp(i+1,j)-
 temp(i-1,j))))))
!                        end if
!                        end do
!             end do

          open (unit=30,file='u vel1.dat',status='replace',action='write',iostat=ierror)
          write(30,*)'thermal diffusivity      ',di
         write(30,*)'Rayleigh number       ',ra
         write(30,*)'Prandtl number          ',pt
!     write(30,*)'Reynolds number          ',re
         write(30,*)'hydro relax. time       ',tauv
         write(30,*)'termo relax. time       ',tauc
         write(30,*)'solution converge at     ',iter
!     write(30,*)' nusselt number          ',bnu

         do j = 1,yd

         write(30,*) ux((xd+1)/2,j)*(yd-1)/di

         end do

         close(30)

         open (unit=31,file='v vel1.dat',status='replace',action='write',iostat=ierror)
         do i = 1,xd

         write(31,*) uy(i,(yd+1)/2)*(yd-1)/di

         end do

         close(31)

         open (unit=32,file='variables.dat',status='replace',action='write',iostat=ierror)
         write(32,*)'x-vel, y-vel, temp'
         do j = 1,yd
         do i = 1,xd
```

```
          write(32,*) ux(i,j)*(yd-1)/di,uy(i,j)*(yd-1)/di,temp(i,j)

          end do
          end do

          close(32)
          end

         subroutine                                                    initial1

 (cx,cy,f,f0,g,g0,ux,uy,rho,temp,u0,rho0,th,tc,cd,xd,yd,dx,dy,dd,x1,x2,y1,y2,lambda)
 !---------------------------------------------------------------------
         real*8  cx(1:cd), cy(1:cd), w(1:cd)
         real*8  dx(1:dd), dy(1:dd)
         real*8  f(1:cd,1:xd,1:yd), f0(1:cd,1:xd,1:yd)
         real*8  g(1:dd,1:xd,1:yd), g0(1:dd,1:xd,1:yd)
        real*8  ux(1:xd,1:yd), uy(1:xd,1:yd), rho(1:xd,1:yd)
        real*8  temp(1:xd,1:yd)
        real*8  u0,  rho0, pi,tc,th,lambda

        pi = 4.0*atan(1.0)
    cx(1) =  0.0
    cy(1) =  0.0
    do k = 2,9
       w(k) = 1.
       if(mod(k, 2) .eq. 1.) w(k) = sqrt(2.)
             cx(k) = w(k)*cos((k-2)*pi/4.0)
    cy(k) = w(k)*sin((k-2)*pi/4.0)
    enddo

        dx(1) = 1.0
        dy(1) = 1.0
        dx(2) = -1.0
        dy(2) = 1.0
        dx(3) = -1.0
        dy(3) = -1.0
        dx(4) = 1.0
        dy(4) = -1.0

   !    initial condition for velocity

       do i = 2,xd
             do j = 1,yd-1
                         ux(i,j) = u0; uy(i,j) = u0; rho(i,j) = rho0
                         temp(i,j)=tc

             end do
       end do
!!!!!INITIAL VELOCITY OF LID!!
       do i = 1,xd
                         ux(i,yd) = 0.0 ; uy(i,yd) = u0; rho(i,yd) = rho0
                         temp(i,yd) = tc
       end do

       do i = 1,xd
                         ux(i,1) = u0; uy(i,1) = u0; rho(i,1) = rho0
                         temp(i,1) = 1+lambda*(sin((2*pi*(i))/xd))
       end do


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
       call equilibrium1 (cx,cy,f0,ux,uy,rho,cd,xd,yd,dx,dy,g0,temp,dd)

       do k = 1,cd
```

```
            do i = 1,xd
                  do j = 1,yd
                        f(k,i,j) = f0(k,i,j)
                  end do
            end do
      end do

      do k = 1,dd
            do i = 1,xd
                  do j = 1,yd
                        g(k,i,j) = g0(k,i,j)
                  end do
            end do
      end do

      return
      end

   subroutine equilibrium1 (cx,cy,f0,ux,uy,rho,cd,xd,yd,dx,dy,g0,temp,dd)
!-----------------------------------------------------------------------
      real*8  cx(1:cd), cy(1:cd), f0(1:cd,1:xd,1:yd)
      real*8  dx(1:dd), dy(1:dd)
      real*8  g0(1:dd,1:xd,1:yd)
   real*8  ux(1:xd,1:yd), uy(1:xd,1:yd), rho(1:xd,1:yd)
      real*8  temp(1:xd,1:yd)
      real*8  u2, tmp
      integer i, j, k, m

   do i = 1,xd
            do j = 1,yd
                        u2 = ux(i,j)**2 + uy(i,j)**2
                        f0(1,i,j) = rho(i,j)*(1. - 3./2.*u2)*4./9.
                        do k = 1,4
                        m = k*2    ; tmp = cx(m)*ux(i,j) + cy(m)*uy(i,j)
                        f0(m,i,j) = rho(i,j)*(1. + 3.*tmp + 9./2.*tmp**2 -
                  3./2.*u2)/9.
                        m = k*2 + 1; tmp = cx(m)*ux(i,j) + cy(m)*uy(i,j)
                        f0(m,i,j) = rho(i,j)*(1. + 3.*tmp + 9./2.*tmp**2 -
            3./2.*u2)/36.
                        end do
            end do
       end do

      do i = 1,xd
            do j = 1,yd
                  do k = 1,dd
                        tmp = dx(k)*ux(i,j) + dy(k)*uy(i,j)
                        g0(k,i,j) = rho(i,j)*temp(i,j)*(1 + tmp )/4
                  end do
    end do
       end do

       return
       end


     subroutine collide1 (f,f0,tauv,cd,xd,yd,g,g0,tauc,dd)
!-----------------------------------------------------------------------
       real*8  f(1:cd,1:xd,1:yd), f0(1:cd,1:xd,1:yd), tauv
       real*8  g(1:dd,1:xd,1:yd), g0(1:dd,1:xd,1:yd), tauc
       integer i, j, k

     do k = 1,cd
            do i = 1,xd
                  do j = 1,yd
                        f(k,i,j) = f(k,i,j) - (f(k,i,j) - f0(k,i,j))/tauv
```

```
                        if (f(k,i,j) .le. -1) then
                            write(*,*)i,j,'error'
                        endif
                    end do
                end do
            end do

            do k = 1,dd
                do i = 1,xd
                    do j = 1,yd
                        g(k,i,j) = g(k,i,j) - (g(k,i,j) - g0(k,i,j))/tauc
                        if (g(k,i,j) .le. -1) then
                            write(*,*)i,j,'error'
                        endif
                    end do
                end do
            end do

            return
            end


            subroutine force (cx,cy,ux,uy,temp,f,f0,gra,cd,xd,yd)
!-------------------------------------------------------------------------
            real*8 f(1:cd,1:xd,1:yd), temp(1:xd,1:yd),cy(1:cd),cx(1:cd)
            real*8 f0(1:cd,1:xd,1:yd)
            real*8 ux(1:xd,1:yd), uy(1:xd,1:yd)
            integer i, j, k

            tempor = 0.0
            do i = 1,xd
        do j = 1,yd
            tempor = tempor + temp(i,j)
            tempave = tempor/(yd*yd)
            end do
            end do

            do k = 1,cd
        do i = 1,xd
        do j = 1,yd
            f(k,i,j) = f(k,i,j) + 3*gra*(cy(k)-uy(i,j))*f0(k,i,j)*(temp(i,j)-tempave)
            end do
            end do
            end do

            return
            end

            subroutine stream1 (f,cd,xd,yd,g,dd)
!-------------------------------------------------------------------
            real*8  f(1:cd,1:xd,1:yd),g(1:dd,1:xd,1:yd)
        real*8  tmp(1:cd,1:xd,1:yd),tmpg(1:dd,1:xd,1:yd)
            integer i, j, k
!
        do k = 1,cd
        do i = 1,xd
        do j = 1,yd
        tmp(k,i,j) = f(k,i,j)
            end do
            end do
            end do

            do k = 1,dd
        do i = 1,xd
        do j = 1,yd
        tmpg(k,i,j) = g(k,i,j)
```

```
      end do
      end do
      end do

      do k = 1,cd
 if(k .eq. 1) then
 do i = 1,xd; do j = 1,yd
    ii = i    ; jj = j
 f(k,ii,jj) = tmp(k,i,j)
    end do; end do

    else if(k .eq. 2) then
 do i = 1,xd-1; do j = 1,yd
    ii = i + 1; jj = j
 f(k,ii,jj) = tmp(k,i,j)
             end do; end do

   else if(k .eq. 3) then
 do i = 1,xd-1; do j = 1,yd - 1
   ii = i + 1; jj = j + 1
 f(k,ii,jj) = tmp(k,i,j)
   end do; end do

   else if(k .eq. 4) then
 do i = 1,xd; do j = 1,yd - 1
   ii = i    ; jj = j + 1
 f(k,ii,jj) = tmp(k,i,j)
   end do; end do

   else if(k .eq. 5) then
 do i = 2,xd; do j = 1,yd - 1
   ii = i - 1; jj = j + 1
 f(k,ii,jj) = tmp(k,i,j)
   end do; end do

   else if(k .eq. 6) then
 do i = 2,xd; do j = 1,yd
   ii = i - 1; jj = j
 f(k,ii,jj) = tmp(k,i,j)
   end do; end do

  else if(k .eq. 7) then
do i = 2,xd; do j = 2,yd
   ii = i - 1; jj = j - 1
f(k,ii,jj) = tmp(k,i,j)
   end do; end do

   else if(k .eq. 8) then
do i = 1,xd; do j = 2,yd
   ii = i    ; jj = j - 1
f(k,ii,jj) = tmp(k,i,j)
  end do; end do

  else if(k .eq. 9) then
do i = 1,xd-1; do j = 2,yd
  ii = i + 1; jj = j - 1
f(k,ii,jj) = tmp(k,i,j)
   end do; end do
   end if
   end do


  do k = 1,dd
if(k .eq. 1) then
do i = 1,xd-1; do j = 1,yd - 1
   ii = i + 1; jj = j + 1
```

```
 g(k,ii,jj) = tmpg(k,i,j)
    end do; end do

  else if(k .eq. 2) then
 do i = 2,xd; do j = 1,yd - 1
      ii = i - 1; jj = j + 1
 g(k,ii,jj) = tmpg(k,i,j)
   end do; end do

  else if(k .eq. 3) then
 do i = 2,xd; do j = 2,yd
     ii = i - 1; jj = j - 1
 g(k,ii,jj) = tmpg(k,i,j)
    end do; end do

   else if(k .eq. 4) then
 do i = 1,xd-1; do j = 2,yd
    ii = i + 1; jj = j - 1
 g(k,ii,jj) = tmpg(k,i,j)
  end do; end do

    end if
    end do


     return
    end


  subroutine boundary1 (f,f0,rho,u0,cd,xd,yd,g,g0,temp,dd,x1,x2,y1,y2)
!-----------------------------------------------------------------------
  real*8  f(1:cd,1:xd,1:yd),f0(1:cd,1:xd,1:yd)
  real*8  g(1:dd,1:xd,1:yd),g0(1:dd,1:xd,1:yd)
  real*8  temp(1:xd,1:yd)
  real*8  u0
  real*8  rho(1:xd,1:yd)

  ! nonequilibrium boundary conditon bottom
  do i = 2,xd-1                    !bottom
  f(4,i,1) = f(8,i,1)
  f(3,i,1) = f(7,i,1) !+ 0.5*(f(6,i,1)-f(2,i,1))
  f(5,i,1) = f(9,i,1) !+ 0.5*(f(2,i,1)-f(6,i,1))

  ! no slip bounce back
  g(1,i,1) = g(3,i,1)
  g(2,i,1) = g(4,i,1)
  end do

  do i = 2,xd-1                    !upper
  f(7,i,yd) = f(3,i,yd)
  f(9,i,yd) = f(5,i,yd) !+ 0.5*(f(6,i,1)-f(2,i,1))
  f(8,i,yd) = f(4,i,yd) !+ 0.5*(f(2,i,1)-f(6,i,1))

  ! no slip bounce back
  g(3,i,yd) = g(1,i,yd)
  g(4,i,yd) = g(2,i,yd)
  end do

  ! left and right boundary condition
  do j = 2,yd-1                    !left boundary
  f(2,1,j) = f(6,1,j)
  f(3,1,j) = f(7,1,j) !+ 0.5*(f(8,i,1)-f(4,i,1))
  f(9,1,j) = f(5,1,j) !+ 0.5*(f(4,i,1)-f(8,i,1))

  g(1,1,j) = g(3,1,j)!g0(1,1,j) + g(1,2,j) - g0(1,2,j)
  g(4,1,j) = g(2,1,j)! g0(4,1,j) + g(4,2,j) - g0(4,2,j)
```

```
    end do

    do j = 2,yd-1                   !right boundary
    f(5,xd,j) = f(9,xd,j) !+ 0.5*(f(8,i,1)-f(4,i,1))
    f(6,xd,j) = f(2,xd,j)
    f(7,xd,j) = f(3,xd,j) !+ 0.5*(f(4,i,1)-f(8,i,1))

    g(2,xd,j) =g(4,xd,j)! g0(2,xd,j) + g(2,xd-1,j) - g0(2,xd-1,j)
    g(3,xd,j) =g(1,xd,j)! g0(3,xd,j) + g(3,xd-1,j) - g0(3,xd-1,j)

    end do

    ! four corner boundary condition
    f(9,1,yd) = f(5,1,yd)
    f(8,1,yd) = f(4,1,yd)
    f(2,1,yd) = f(6,1,yd)
    g(4,1,yd) = g(2,1,yd)!g0(4,1,yd) + g(4,2,yd-1) - g0(4,2,yd-1)

    f(7,xd,yd) = f(3,xd,yd)
    f(6,xd,yd) = f(2,xd,yd)
    f(8,xd,yd) = f(4,xd,yd)
    g(3,xd,yd) = g(1,xd,yd)!g0(3,xd,yd) + g(3,xd-1,yd-1) - g0(3,xd-1,yd-1)

    f(3,1,1) = f(7,1,1)
    f(2,1,1) = f(6,1,1)
    f(4,1,1) = f(8,1,1)
    g(1,1,1) = g(3,1,1)!g0(1,1,1) + g(1,2,2)- g0(1,2,2)

    f(5,xd,1) = f(9,xd,1)
    f(4,xd,1) = f(8,xd,1)
    f(6,xd,1) = f(2,xd,1)
    g(2,xd,1) = g(4,xd,1)!g0(2,xd,1) + g(2,xd-1,2)- g0(2,xd-1,2)

    return
    end


subroutine    calc1       (cx,cy,f,ux,uy,rho,u0,rho0,cd,xd,yd,temp,g,tc,th,dd,dx,dy,x1,x2,y1,y2,lambda)
 !---------------------------------------------------------------------
    real*8  cx(1:cd), cy(1:cd), f(1:cd,1:xd,1:yd)
    real*8  dx(1:dd), dy(1:dd)
    real*8  temp(1:xd,1:yd),g(1:dd,1:xd,1:yd)
    real*8      ux(1:xd,1:yd), uy(1:xd,1:yd), rho(1:xd,1:yd)
    real*8  u0,rho0,th,tc,lambda
    integer i, j, k

    pi = 4.0*atan(1.0)

    do i = 1,xd
                do j = 1,yd
                            ux(i,j) = 0; uy(i,j) = 0; rho(i,j) = f(1,i,j);ux(i,yd) = 0.0
                            do k = 2,cd
                                        ux(i,j) =  ux(i,j) + f(k,i,j)*cx(k)
                                        uy(i,j) =  uy(i,j) + f(k,i,j)*cy(k)
                                        rho(i,j) = rho(i,j) + f(k,i,j)
                            end do
                end do
    end do

    do i = 1,xd
                do j = 1,yd
                            temp(i,j) = g(1,i,j);
                            do k = 2,dd
                                        temp(i,j) = temp(i,j) + g(k,i,j)
                            end do
                end do
```

```
    end do

 j = yd   ! top boundary
 do i = 1,xd
                ux(i,j) = 0.0
                uy(i,j) = u0
!               rho(i,j) = rho0
                temp(i,j) = tc
    end do

    j = 1  !bottom boundary
 do i = 1,xd
                ux(i,j) = u0
                uy(i,j) = u0
!               rho(i,j) = rho0
                temp(i,j) = 1+lambda*(sin((2*pi*(i))/xd))
    end do

    i = 1 ! left boundary
    do j = 2, yd-1
                ux(i,j) = u0
                uy(i,j) = u0
!               rho(i,j) = rho0
                temp(i,j) = tc
    end do

    i = xd ! left boundary
    do j = 2, yd-1
                ux(i,j) = u0
                uy(i,j) = u0
!               rho(i,j) = rho0
                temp(i,j) = tc
    end do

!   do i = x1+1,x2-1
!               do j = y1+1,y2-1
!                       ux(i,j) = 0.0; uy(i,j) = 0.0
!                       temp(i,j) = 0.0
!                       rho(i,j) = 0.0
!               end do
!   end do


    do i = 2,xd-1
                do j = 2,yd-1
                        if (i .ge. x1 .and. i .le. x2 .and. j .ge. y1 .and. j .le. y2) then !              define          cubic
    wall temperature
                                temp(i,j) = tc
                                ux(i,j) = u0
                                uy(i,j) = u0
!                               if (i .eq. x1 .or. i .eq. x2 .or. j .eq. y1 .or. j .eq. y2) then
!                               rho(i,j) = rho0
!                               end if
                        else
                        if (rho(i,j) .ne. 0.) then
                                ux(i,j) =  ux(i,j)/rho(i,j)
                                uy(i,j) =  uy(i,j)/rho(i,j)
                                temp(i,j) = temp(i,j)/rho(i,j)
                        else
                                ux(i,j)  = 0.
                                uy(i,j)  = 0.
                                temp(i,j) = 0.
                        endif
                        end if
                end do
    end do
```

```
!   temperature for inner cubic boundary

!   do j = y1,y2
!               temp(x1,j) = temp(x1-1,j)
!               temp(x2,j) = temp(x2+1,j)
!
!   end do
!   do i = x1,xd
!               temp(i,y1) = temp(i,y1-1)
!               temp(i,y2) = temp(i,j+1)
!   end do

    ! upper adiabatic wall
    do j = 1,yd
    temp(1,j) = temp(2,j)
    end do
    do j = 1,yd
    temp(xd,j) = temp(xd-1,j)
    end do
!   do j = 2, y1 - 1
!   temp(xd,j) = temp(xd-1,j)
!   end do

!   do j = y2 +1, yd-1
!   temp(xd,j) = temp(xd-1,j)
!   end do


return
    end
```

References

This guide is prepared based on the following references;

C. S. Nor Azwadi, "*The development of simplified thermal lattice Boltzmann models for the simulation of thermal fluid flow problems*", PhD Thesis, Universiti Tekonologi Malaysia (2007).

Y. Shi, T. S. Zhou and Z. L. Guo, "*Thermal lattice Bhatnagar-Groos-Krook model for flows with viscous heat dissipation in incompressible limit*", Phys. Rev. E, 70,pp66310-66319 (2004).

Bhatnagar, P.L., Gross, E.P and Krook, M., "*A model for collisional processes in gases I: small amplitude processes in charged and in nutral one-component systems*", Phys. Rev.94, pp511-525 (1954).

S. Chen and G.D. Doolen, Annu, "*A novel thermal model for the lattice Boltzmann method in incompressible limit*", J. Comp. Phys., 146, pp282-300 (1998).

Gary D. Doolen, "*Lattice Gas Methods for Partial Differential Equations*", Addison-Wesley, Redwood City, CA, (1990)

Y. H. Qian, D. Humieres and P. lallemand, "*Lattice BGK models for Navier-Stokes equation*", Europhys. Letters, 17, pp479-484 (1992).

C. Shu, Y. Peng and Y. T. Chew, "*Simulation of natural convection in a square cavity by Taylor series expansion and least square-based lattice Boltzmann method*", Int. J. Mod. Phys. C, 13, pp1399-1414 (2002).

S. Hou, Q. Zou, S. Chen, G. D. Doolen, and A. C. Cogley, "*Simulation of cavity flow by the lattice Boltzmann method*", J. Comput. Phys. pp118:329 (1995).

U. Frisch, B. Hasslacher, and Y. Pomeau, "*Lattice-gas automata for Navier-Stokes equations*", Phys. Rev. Lett. 56, pp1505-1508 (1986).

T. Watanabe, "*Flow pattern and heat transfer rate in Rayleigh-Bernard convection*", Phys. Fluids, Vol. 16, No. 4 (2004).

X. Shan, "*Simulation of Rayleigh-Bernard convection using a lattice Boltzmann method*", Phys. Review E, Vol. 55, No. 3 (1997).

X. Shan and X. He, "*Discretization of the velocity space in the solution of the Boltzmann equation*", Phys. Rev. Lett. 80, pp65-68 (1998).

Xiaoyi He and Li-Shi Luo , "*Theory of the lattice Boltzmann method: From the Boltzmann equationto the lattice Boltzmann equation*," Phys. Rev. E Vol.56, No. 6 (1997).

J. A. Given and E. Clementi, ''*Molecular dynamics and Rayleigh–Be´nard convection*,'' J. Chem. Phys. 90, 7376 (1989).

T. Watanabe and H. Kaburaki, "*Particle Simulation of Three-dimensional Convection Patterns in a Rayleigh-Benard System*", Phys. Rev. E, Vol.56, pp.1218-1221 (1997).

M. Rohde, D. Kandhai, J. J. Derksen, and H. E. A. Van den Akker, "*Improved bounce-back methods for no-slip walls in lattice-Boltzmann schemes: Theory and simulations*", Phys. Rev. E 67, (2003).