

SIMULATION OF BUBBLES RISE IN CAVITY USING LATTICE BOLZTMANN
METHOD

WAN MOHAMAD DASUQI BIN WAN MA'SOR

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

Faculty of Mechanical Engineering
UNIVERSITI MALAYSIA PAHANG

DECEMBER 2010

SUPERVISOR'S DECLARATION

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

Signature:

Name of Supervisor: MUHAMAD ZUHAIRI BIN SULAIMAN

Position: LECTURER

Date: 6/12/2010

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: WAN MOHAMAD DASUQI B WAN MA'SOR

Id Number: MA07052

Date: 6/12/2010

**Dedicated, in thankful appreciation for support,
encouragement and understanding to
my beloved family and friends.**

ACKNOWLEDGEMENTS

First of all, I would like to express my heartily gratitude to my research supervisor, Associate Mr. Muhamad Zuhairi bin Sulaiman for his guidance, advices, efforts, supervision and enthusiasm given throughout for the progress of this research.

In preparing this thesis, I was in contact with many people, lecturers, and training engineers. They have contributed towards my understanding and thoughts. Without their continued support and interest, this thesis would not have been the same as presented here.

I would like to express my sincere appreciation to my parents for their support to me all this year. Without them, I would not be able to complete this research. Besides that, I would like to thank my course mates and my friends especially my section M06 for their help, assistance and support and encouragement.

ABSTRACT

This thesis explains the simulation of bubbles rise in cavity using Lattice Boltzmann Method. The two phase bubbles simulated using Lattice Boltzmann Equation. The simulation was done on two dimensional only and multiple bubbles motion under buoyancy was carried out. Three configurations of the bubble simulation was being simulated using LBM code development. Free energy model was reviewed and the base on isotropy approach (Yonetsu Approach) used and Galilean invariance are also considered. Three variations of configuration used which are distance between bubbles, configuration base on differential of Eotvos number, and configuration base on different Kappa number. The result obtained from simulation is proven by lattice Boltzmann method which to be used for multiphase. Simulation results are validated and has good agreement with previous studies. The experimental setup for bubble rise was-fabricated and due to some limitations, results obtained are not well satisfied.

ABSTRAK

Thesis ini menerangkan mengenai kajian terhadap simulasi buih yang naik dari kaviti melalui aliran dua fasa berdasarkan kaedah Lattice Boltzmann. Simulasi buih ini berasaskan dari persamaan kaedah Lattice Boltzmann. Simulasi ini hanya dilakukan dalam dua dimensi analisis sahaja dan pada masa yang sama apungan ke atas pergerakan buih berganda akan dapat dipelajari. Tiga jenis rupa bentuk simulasi buih yang menggunakan LBM akan di simulasikan melalui perisian Visual C++. Model tenaga bebas akan di tinjau kembali berdasarkan model yang terbaru yang berasas pendekatan isotropik(Pendekatan Yonetsu). Ketidaksamaan Galliean juga akan di pertimbangkan. Rupa bentuk yang disimulasikan di dalam thesis ini antaranya adalah rupa bentuk disebabkan pendekatan jarak diantara buih, rupa bentuk yang disebabkan perbezaan nombor Eotvos, dan terakhir sekali disebabkan perbezaan nombor Kappa. Keputusan yang diperolehi melalui simulasi ini telah dibuktikan melalui persamaan Lattice Boltzmann dan digunakan untuk dua fasa. Keputusan simulasi ini di perakui sah dan mendapat persetujuan yang baik dengan pembelajaran sebelum ini. Eksperimen penaikan buih secara manual juga telah dicipta dan bergantung ke atas tahap yang terhad, keputusan yang diperolehi masih belum sempurna.

TABLE OF CONTENT

		Page
SUPERVISOR'S DECLARATION		ii
STUDENT'S DECLARATION		iii
ACKNOWLEDGEMENTS		v
ABSTRACT		vi
ABSTRAK		vii
TABLE OF CONTENTS		viii
LIST OF TABLES		xi
LIST OF FIGURES		xii
LIST OF ABBREVIATIONS		xiii
LIST OF APPENDICES		xiv
CHAPTER 1	INTRODUCTION	
1.2	Background of Lattice Boltzmann Method	2
1.3	Problem Statement	4
1.4	Objectives	4
1.5	Project Scope	5
CHAPTER 2	LITERATURE REVIEW	
2.2	Lattice Boltzmann Method	7
2.3	Kinetic Theory	9
2.4	First Order Distribution Function	9
2.5	LBM Framework and Equations	12
2.6	Multiphase Lattice Boltzmann Method	15
2.7	Van De Waals Fluid	17
2.8	Maxwell equal area construction	20
2.9	Yonetsu's Approach	21

CHAPTER 3 METHODOLOGY

3.2	Flow Chart	24
3.3	Physical Properties	26
3.4	Code Development	27
3.5	Initial Condition	28
3.6	Advection	28
3.7	Collision	31
3.8	Boundary Condition	32
	3.8.1 Bounce back Boundaries	33
3.9	Output	34
3.10	Convergent	35
	3.10.1 Effect of Eotvos Number	35
	3.10.2 Effect of Kappa number	38
	3.10.3 Effect of Distance Increment	39
3.11	Experimental Setup	39

CHAPTER 4 RESULTS AND DISCUSSION

4.2	Effect by Distance Increment between Bubble (Case 1)	43
	4.2.1 Discussion (Case 1)	44
4.3	Effect by Eotvos number (Case 2)	45
	4.3.1 Discussion (Case 2)	46
4.4	Effect of kappa number, κ (Case 3)	47
	4.4.1 Discussion (Case 3)	48
4.5	Experimental Result	49

CHAPTER 5 CONCLUSION AND RECOMMENDATION

5.2	Conclusion	53
5.3	Recommendation	54

REFERENCES	55
APPENDIX A	56
APPENDIX B	57
APPENDIX C	58
APPENDIX D	59
APPENDIX E1	60
APPENDIX E2	61
APPENDIX E3	62
APPENDIX F1	63
APPENDIX F2	64
APPENDIX F3	65
APPENDIX G1	66
APPENDIX G2	67
APPENDIX G3	68
APPENDIX H	69
APPENDIX I1	70
APPENDIX I2	71
APPENDIX I3	72
APPENDIX J1	73
APPENDIX J2	74
APPENDIX J3	75
APPENDIX K1	76
APPENDIX K2	77
APPENDIX K3	78
APPENDIX L1	79
APPENDIX L2	80
APPENDIX L3	81

LIST OF TABLES

Table No	Title	Page
3.1	Differential location of each bubbles	41
3.2	Item used in the experiment	41
3.3	Criteria each item used in the experiment	41
4.1	Comparison the shape in differential position cases	44
4.2	Comparison shape deform base differential of Eo number	46
4.3	Comparison of shape coalesce with differential κ	48

LIST OF FIGURES

Figure No.	Title	Page
2.1	General concept of lattice Boltzmann Method	8
2.2	D2Q9 lattice and velocities.	13
2.3	D2Q9 velocity components	13
2.4	On-lattice and histogram views of the discrete single particle distribution	14
2.5	Isotherms plot of $\tilde{p} - \tilde{V}$	20
3.1	Project Flowchart	25
3.2	Example physical properties in simulation condition	26
3.3	Example Visual C++ software	27
3.4	Neighbor referencing	29
3.5	Illustration of mid-plane bounce back movement of direction specific densities f_a	33
3.6	Computed bubble shapes and rise velocities for the bubbles regimes	36
3.7	Bubble diagram for the shape and terminal rise velocity of gas bubbles in quiescent viscous liquids	37
3.8	Density gradient at the interface for various value of κ	38
3.9	Experimental setup for different location bubbles	40
3.10	Position of metal tube as a source of bubbles	40
4.2	Experimental result for differential position	51

LIST OF ABBREVIATIONS

CFD	-	Computational Fluid Dynamic
$f^{(i)}$	-	Distribution function
p	-	Pressure (pa)
F	-	Force (N)
Eo	-	Eotvos number Dimensionless
Re	-	Reynolds number Dimensionless
Mo	-	Morton number Dimensionless
σ	-	Surface tension Dimensionless
κ	-	Density gradient/kappa Dimensionless
D2Q9	-	Direction of bubbles Dimensionless
u	-	Velocity Dimensionless
f_{eq}	-	Equilibrium distribution function
ρ	-	Density of phase Dimensionless
ν	-	Kinematic shear viscosity Dimensionless
LBM	-	Lattice Boltzmann Method
ρ_L	-	Density of liquid phase Dimensionless
ρ_G	-	Density of gases phase Dimensionless
g	-	Gravity acceleration Dimensionless
d	-	Diameter of bubbles Dimensionless

LIST OF APPENDIX

Appendix	Title	Page
A	Sample of output in visual C++ software	56
B	AVS express software to show the result	57
C	Deformation shape when not coalesce	58
D	Deformation shape when can coalesce	59
E1	Deformation shape diagram at location ($^{36}/_{160}$, $^{80}/_{160}$, $^{124}/_{160}$)	60
E2	Deformation shape diagram at location ($^{40}/_{160}$, $^{80}/_{160}$, $^{120}/_{160}$)	61
E3	Deformation shape diagram at location ($^{44}/_{160}$, $^{80}/_{160}$, $^{116}/_{160}$)	62
F1	Deformation shape diagram at $Eo=10$	63
F2	Deformation shape diagram at $Eo=50$	64
F3	Deformation shape diagram at $Eo=100$	65
G1	Deformation shape diagram at $\kappa = 0.003$	66
G2	Deformation shape diagram at $\kappa = 0.005$	67
G3	Deformation shape diagram at $\kappa = 0.007$	68
H	Gantt Chart	69
I1	Code development for position ($^{36}/_{160}$, $^{80}/_{160}$, $^{124}/_{160}$)	70
I2	Code development for position ($^{40}/_{160}$, $^{80}/_{160}$, $^{120}/_{160}$)	71
I3	Code development for position ($^{44}/_{160}$, $^{80}/_{160}$, $^{116}/_{160}$).	72
J1	Code development for $Eo = 10$	73
J2	Code development for $Eo = 50$	74
J3	Code development for $Eo = 100$	75

K1	Code development for $\kappa = 0.003$	76
K2	Code development for $\kappa = 0.005$	77
K3	Code development for $\kappa = 0.007$	78
L1	Comparison of variation bubbles configuration (Case 1)	79
L2	Comparison of variation Eotvos number, Eo (Case 2)	80
L3	Comparison of variation Kappa number, κ (Case 3)	81

1.2 Background of Lattice Boltzmann Method

The lattice Boltzmann method (LBM) has been proposed as a mesoscopic approach to the numerical simulations of fluid motion on the statistical-thermodynamic assumption that a fluid consists of many virtual particles repeating collision and translation through which their velocities distributions converge to a state of local equilibrium. Lattice Boltzmann method, based on the lattice gas cellular automaton, possesses the advantages such as relatively easy implementation of boundary conditions on complicated geometry, high efficiency on parallel processing, and flexible reproduction of interface between multiple phases. The last point on which we focus in this study arises from the introduction of repulsive interaction between particles without any boundary condition for interface. As a result, LBM is more useful than other conventional methods for the numerical analysis of multiphase fluid system, where the flow pattern changes not only spatially but also temporally due to deformation, break up, and coalescence of droplet or bubbles (Orszag SA, 1995). Therefore its apply LBM to simulations of two-phase fluid motion. The low and high-density fluids in this study, referred to as bubble and liquid respectively, correspond to the pressurized fluids having a small density ratio such as steam and water in PWR nuclear power plants. Another advantage of LBM occurs on implementation of complicated boundary conditions to be seen in fuel assemblies, although the topic is not discussed in detail in this study. Lattice Boltzmann method therefore is a promising method suitable for simulating fluid flows in nuclear engineering.

In single-phase LBM, there have already been a lot of numerical results for incompressible viscous fluid flows. On the other hand, several kinds of multiphase fluid model have recently been proposed and applied to the simulations of phase separation and transition. The first immiscible multiphase model reproduced the phase separation by the repulsive interaction based on the color gradient and the color momentum between red-and-blue-colored particles representing two kinds of fluid. Shan and Chen proposed the gas-liquid model applicable to the phase transition with the potential between particles, while another gas-liquid model proposed by Swift. Simulates phase transitions consistent with the thermodynamics on the theory of Van de Waals-Chan-Hillard free energy.

(Kato, 2001) also presented a two phase model with a pseudo potential for van de Waals fluids. Lattice Boltzmann method was used to simulate the condensation of liquid droplets in supersaturated vapor, the two-phase fluid flow through sandstone in three dimensions and so on. However, it has not been applied to any quantitative numerical analysis of the motion of bubbles or droplet in two-phase flows under gravity in two and three dimensions, because the main purpose in previous works was to develop multiphase model and examine its property, or to reproduce fundamental phenomena in multiphase fluid where buoyancy effect is negligible.

Therefore, in LBM, we consider the buoyancy effect due to density difference in two phase fluid characterized with the non-dimensional numbers such as Eotvos and Morton numbers, and develop the three-dimensional version of the binary fluid model which is a newest one proposed by (Swift, 2001). This model using the free-energy approach has one important improvement that the equilibrium distribution of fluid particles can be consistently based on thermodynamics, compared with other models which are based on phenomenological models of interface dynamics. Consequently, the total energy, including the surface energy, kinetic energy, and internal energy can be conserved. Furthermore, it also reproduces Galilean invariance more properly than one-component fluid model.

Multiphase flow of fluids can be found everywhere either in natural environment phenomenon or in the technology evolution. Study of multiphase flow could contribute a better understanding on multiphase behavior. The knowledge of multiphase flow behavior is important in the development of equipment which directly related to multiphase problem. Lattice Boltzmann Method (LBM) is relatively new method and has a good potential to compete with traditional CFD methods. Recently it has been proved to be a promising tool to simulate the viscous flow (S. Chen and G. Doolen, 1998).

LBM base on derivation of kinetic theory which working in mesoscopic level instead of macroscopic discretization by traditional method. Instead of easy in incorporating with microscopic physics, it is also having shorter time compare to the current method. LBM have more advantages in multi-phase compare to traditional method.

In multi-phase, two main issues which are surface tension force modeling and interface recording have to be considered. In this study, we visualise the numerical results of bubble motions using LBM method by AVS Express software. The LBM coding will create by the software Microsoft Visual C++ SP6 and the result of the motion of bubbles will shown by AVS Express software.

1.3 Problem Statement

Bubbles rising simulation is complex via experiment. The buoyancy force is difficult to control in experiment. Derivation of basic Bhanatgar-Gross-Krook(BGK) Collision equation should be derive in macroscopic equation. Simulation by Lattice Boltzmann is not detail as like conventional CFD like VOF.

1.4 Objectives

To predict the bubbles motion under buoyancy force using the lattice Boltzmann method and investigate the bubble behavior of differential configuration of bubbles location. The bubbles also will be investigate by two more cases which is case by Eotvos number, Eo and the other one case by kappa number, κ .

1.5 Project Scopes

The study will perform numerical simulation and modelling base on Lattice Boltzmann Method. Two dimensional (2D) will be considered. Multiple bubbles motion under buoyancy force will be studied numerically. Three cases will be investigated which are simulated;

- (i) distance that effect of density interface by variation of Kappa number, κ ,
- (ii) effect of surface tension by variation of Eotvos number, Eo
- (iii) effect of bubbles configuration by variation of distance among bubble.

2.2 Lattice Boltzmann Method

The lattice Boltzmann method (LBM) is considerably as an alternatively approach to the well-known finite difference, finite element, and finite volume techniques for according the Navier- Stokes equations. Although as new comer in numerical scheme, the lattice Boltzmann (LB) approach has found recent successes in a host of fluid dynamical problems, including flows in porous media, magneto hydrodynamic, immiscible fluids, and turbulence. LB scheme is a scheme evolved from the improvement of lattice gas automata and inherits some features from its precursor, the LGA. The first LB model was floating- point version of its LGA counterpart each particle in the LGA model (represented by a single – bit Boolean integer) was replaced by a single – particle distribution function in the LB model (represented by floating point number). The lattice structure and the evolution rule remained the same. One important improvement to enhance the computational efficiency has been made for the LB method: the implementation of the BGK approximation (single relaxation time approximation). The uniform lattice structure was unchanged (Lallemand P, 1992).

The starting point in the lattice Boltzmann scheme is by tracking the evolution of the single- particle distribution, f_α . The concept of particle distribution has already well developed in the field of statistical mechanics while discussing the kinetic theory of gases and liquids. The definition implies that the probable number of molecules in a certain volume at certain time made from a huge number of particles in a system that travel freely, without collisions, for distances (mean free path) long compared to their sizes. Once the distribution functions are obtained, the hydrodynamic equations can be derived.

Although LBM approach treats gases and liquids as system consisting of individual particles, the primary goal of this approach is to build a bridge between the microscopic ad macroscopic dynamics, rather than to deal with macroscopic dynamic directly. In other words, the goal is to derive macroscopic equations from microscopic dynamics by means of statistic, rather than to solve macroscopic equation.

The LBM has a number of advantages over other conventional CFD methods. The algorithm is simple and can be implemented with a kernel of just a few hundred lines. The algorithm can also be easily modified to allow for the application for other, more complex simulation components. For example the LBM can be extended to describe the evolution of binary fluid mixtures, or extended to allow for more complex boundary conditions. Thus the LBM is an ideal tool in fluid simulation.

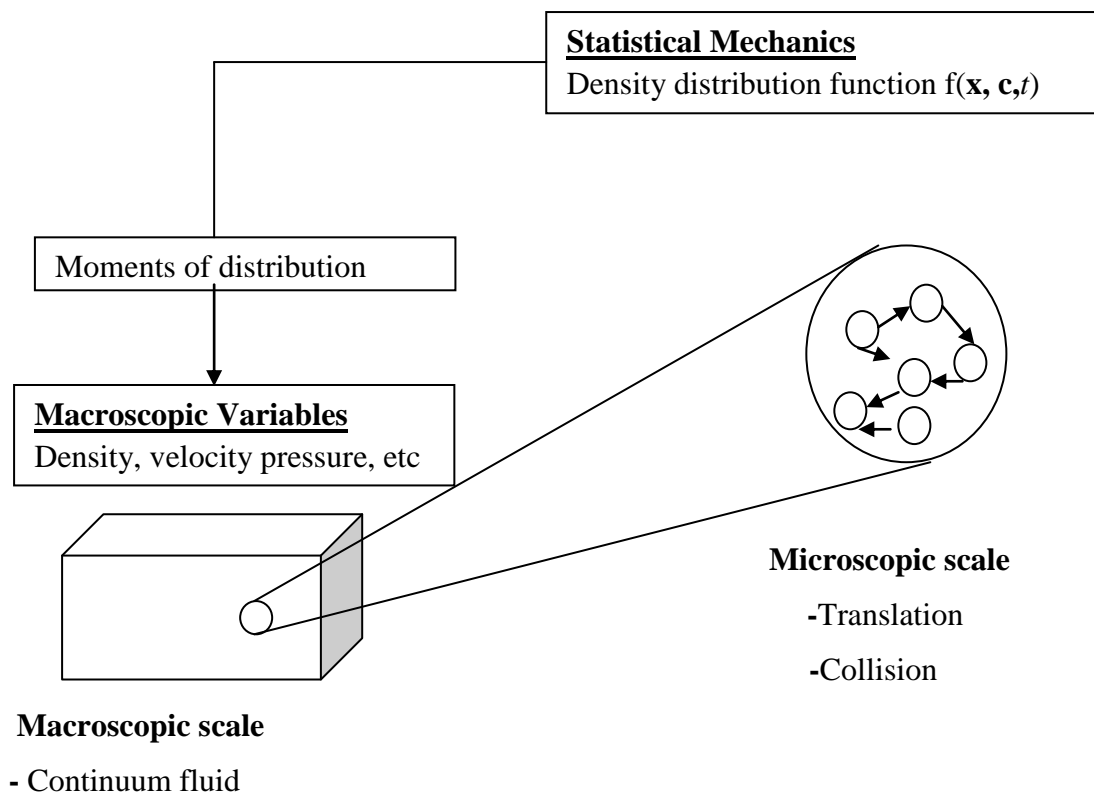


Figure 2.1: General concept of lattice Boltzmann Method

Source: Nor Azwadi Che Sidik, 2007

2.3 Kinetic Theory

Consider a dilute gas consisting of hard spherical particles moving at great velocity ($\sim 300 \text{ ms}^{-1}$). We limit their interaction to elastic collisions. Hypothetically, it would be possible to know the position vector (\mathbf{x}) and momentum (\mathbf{p}) of each individual particle at some instant in time. Such information would give the exact dynamical state of the system which, together with classical mechanics, would allow exact prediction of all future states. We could describe the system by a distribution function

$$f^{(N)}(\mathbf{x}^{(N)}, \mathbf{p}^{(N)}, t) \quad (2.1)$$

where N is the number of particles. Here the distribution is thought of as residing in a ‘phase space’, which is a space in which the coordinates are given by the position and momentum vectors and the time. Changes in Eq.(2.1) with time are given by the Liouville equation ($6N$ variables). However, this level of description is not possible for real gases, where $\sim 10^{23}$ (a mole of) particles are involved in just 20 liters of gas at atmospheric temperature and pressure. Fortunately we are usually interested only in low order distribution functions ($N = 1, 2$)(Orszag SA, 1995).

2.4 First Order Distribution Function

Statistical Mechanics offers a statistical approach in which we represent a system by an ensemble of many copies. The distribution

$$f^{(1)}(\mathbf{x}, \mathbf{p}, t) \quad (2.2)$$

gives the probability of finding a particular molecule with a given position and momentum; the positions and moments of the remaining $N-1$ molecules can remain unspecified because no experiment can distinguish between molecules, so the choice of which molecule does not matter.

This is the ‘Single particle’ distribution function. $f^{(1)}$ is adequate for describing all gas properties that do not depend on relative positions of molecules (dilute gas with long mean free path).

The probable number of molecules with position coordinates in the range $\mathbf{x} \pm d\mathbf{x}$ and momentum coordinates $\mathbf{p} \pm d\mathbf{p}$ is given by

$$f^{(1)}(\mathbf{x}, \mathbf{p}, t) d\mathbf{x} d\mathbf{p} \quad (2.3)$$

It introduce an external force \mathbf{F} that is small relative to intermolecular forces. If there are no collisions, then at time $t + dt$, the new positions of molecules starting at \mathbf{x} are

$$\mathbf{x} + (\mathbf{p}/m)dt = \mathbf{x} + (d\mathbf{x}/dt)dt = \mathbf{x} + d\mathbf{x} \quad (2.4)$$

and the new moments are

$$\mathbf{p} = \mathbf{p} + \mathbf{F}dt = \mathbf{p} + (d\mathbf{p}/dt)dt = \mathbf{p} + d\mathbf{p}. \quad (2.5)$$

Thus, when the positions and momenta are known at a particular time t , Incrementing them allows us to determine $f^{(1)}$ at a future time $t + dt$:

$$f^{(1)}(\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p}, t + dt) d\mathbf{x} d\mathbf{p} = f^{(1)}(\mathbf{x}, \mathbf{p}, t) d\mathbf{x} d\mathbf{p} \quad (2.6)$$

This is the streaming process.

There are however collisions that result in some phase points starting at (\mathbf{x}, \mathbf{p}) not arriving at

$$(\mathbf{x} + \mathbf{p}/m dt, \mathbf{p} + \mathbf{F} dt) = (\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p}) \quad (2.7)$$

and some not starting at (\mathbf{x}, \mathbf{p}) arriving there too. We set $\Gamma^{(-)} d\mathbf{x} d\mathbf{p} dt$ equal to the number of molecules that do not arrive in the expected portion of phase space due to

collisions during time dt . Similarly, we set $\Gamma^{(+)}d\mathbf{x}d\mathbf{p}dt$ equal to the number of molecules that start somewhere other than (\mathbf{x}, \mathbf{p}) and arrive in that portion of phase space due to collisions during time dt . If we start with Eq. (2.6) and add the changes in $f^{(1)}$ due to these collisions we obtain

$$f^{(1)}(\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p}, t + dt) d\mathbf{x}d\mathbf{p} = f^{(1)}(\mathbf{x}, \mathbf{p}, t) d\mathbf{x}d\mathbf{p} + [\Gamma^{(+)} - \Gamma^{(-)}] d\mathbf{x}d\mathbf{p}dt \quad (2.8)$$

The first order terms of a Taylor series expansion of the LHS of Eq. (2.8),

$$\begin{aligned} & f^{(1)}(\mathbf{x} + d\mathbf{x}, \mathbf{p} + d\mathbf{p}, t + dt) d\mathbf{x}d\mathbf{p} \\ &= f^{(1)}(\mathbf{x}, \mathbf{p}, t) d\mathbf{x}\nabla_{\mathbf{x}}f^{(1)} + d\mathbf{p}\cdot\nabla_{\mathbf{p}}f^{(1)} + \left(\frac{\partial f^{(1)}}{\partial t}\right) dt + \dots \end{aligned} \quad (2.9)$$

give the Boltzmann equation

$$\begin{aligned} & \left[f^{(1)}(\mathbf{x}, \mathbf{p}, t) + d\mathbf{x}\cdot\nabla_{\mathbf{x}}f^{(1)} + d\mathbf{p}\cdot\nabla_{\mathbf{p}}f^{(1)} + \left(\frac{\partial f^{(1)}}{\partial t}\right) dt + \dots \right] d\mathbf{x}d\mathbf{p} \\ &= f^{(1)}(\mathbf{x}, \mathbf{p}, t) d\mathbf{x}d\mathbf{p} + [\Gamma^{(+)} - \Gamma^{(-)}] d\mathbf{x}d\mathbf{p}dt \end{aligned} \quad (2.10)$$

Or

$$\mathbf{v}\cdot\nabla_{\mathbf{x}}f^{(1)} + \mathbf{F}\cdot\nabla_{\mathbf{p}}f^{(1)} + \frac{\partial f^{(1)}}{\partial t} = \Gamma^{(+)} - \Gamma^{(-)} \quad (2.11)$$

Note that this can be derived for an arbitrary number of different chemical components as well.

In its complete form with the collision operator written more explicitly, the Boltzmann equation is a nonlinear integral differential equation and is particularly complicated. According to Harris (1971), 50 years elapsed from the time that Boltzmann derived the equation before an approximate solution was found. With lattice Boltzmann methods, it approximately solve the equation from the particle perspective and focus on an equation strongly akin to Eq. (2.8). It explicitly contains the ‘collide and stream’ notion central to LBM.

2.5 LBM Framework and Equations

Lattice Boltzmann models vastly simplify Boltzmann’s original conceptual view by reducing the number of possible particle spatial positions and microscopic moment from a continuum to just a handful and similarly discretization time into distinct steps. Particle positions are confined to the nodes of the lattice. Variations in momenta that could have been due to a continuum of velocity directions and magnitudes and varying particle mass are reduced (in the simple 2-D model we focus on here) to 8 directions, 3 magnitudes, and a single particle mass. Figure 2.2 shows the Cartesian lattice and the velocities e_a where $a = 0, 1, \dots, 8$ is a direction index and $e_0 = 0$ denotes particles at rest.

This model is known as D2Q9 as it is 2 dimensional and contains 9 velocities. This LBM classification scheme was proposed by Qian et al. (1992) and is in widespread use. Because particle mass is uniform (1 mass unit or μ in the simplest approach), these microscopic velocities and moment are always effectively equivalent. The lattice unit (lu) is the fundamental measure of length in the LBM models and time steps (ts) are the time unit.

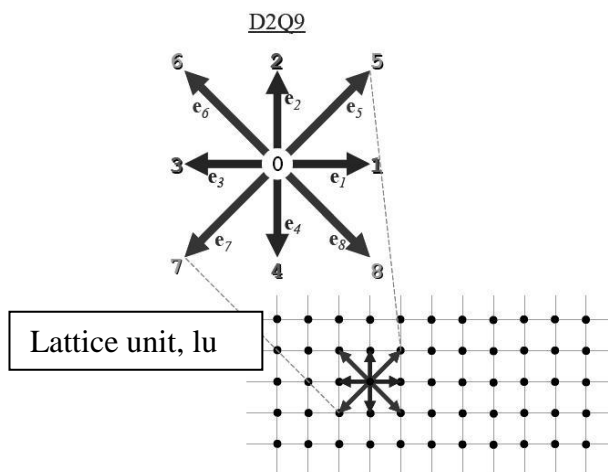


Figure 2.2: D2Q9 lattice and velocities.

Source: Qian et al. (1992)

The velocity magnitude of \mathbf{e}_1 through \mathbf{e}_4 is 1 lattice unit per time step or 1 lu ts^{-1} , and the velocity magnitude of \mathbf{e}_5 through \mathbf{e}_8 is $\sqrt{2}$ lu ts^{-1} . (While this is probably the most common velocity indexing scheme, be aware that others are in use.) These velocities are exceptionally convenient in that all of their x- and y-components are either 0 or ± 1 (Figure 2.3).

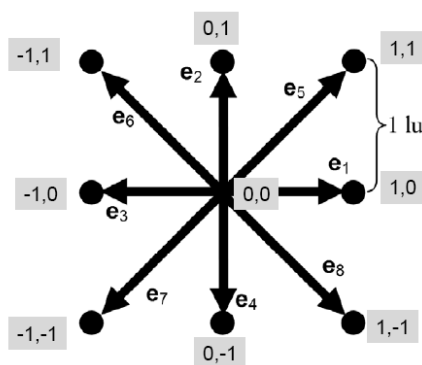


Figure 2.3: D2Q9 x, y velocity components

Source: Qian et al. (1992)

The next step is to incorporate the single-particle distribution function f , which is essentially the one that appears in Eq. (2.8), except that it has only nine discrete ‘bins’ instead of being a continuous function. The distribution function can conveniently be thought of as a typical histogram representing a frequency of occurrence (Figure 2.4). The frequencies can be considered to be direction-specific fluid densities. Accordingly, the macroscopic fluid density is

$$\rho = \sum_{a=0}^8 f_a \quad (2.12)$$

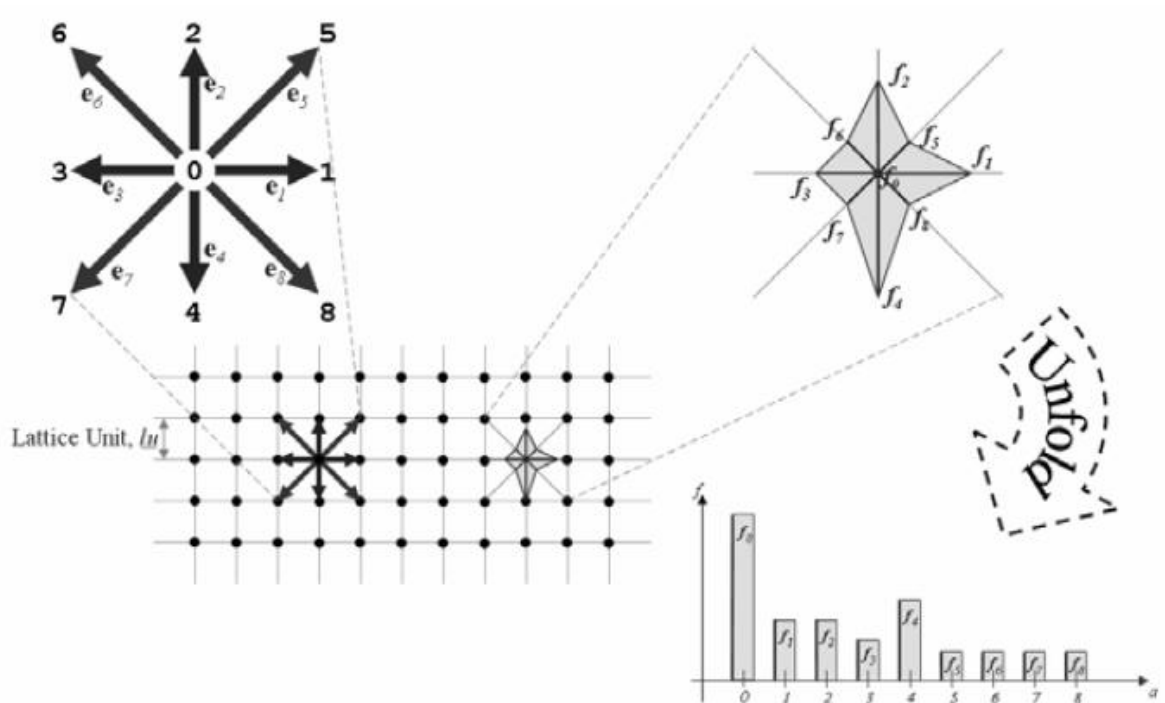


Figure 2.4: On-lattice and histogram views of the discrete single particle distribution function/direction-specific densities f_a .

Source: Qian et al. (1992)

The macroscopic velocity \mathbf{u} is an average of the microscopic velocities \mathbf{e}_a weighted by the directional densities f_a :

$$\mathbf{u} = \frac{1}{\rho} \sum_{a=0}^8 \mathbf{e}_a \quad (2.13)$$

This simple equation allows us to pass from the discrete microscopic velocities that comprise the LBM back to a continuum of macroscopic velocities representing the fluid's motion.

The next steps are streaming and collision of the particles via the distribution function. The simplest approach uses the Bhatnagar-Gross-Krook. Approximation for collision.

2.6 Multiphase Lattice Boltzmann Method

Evolution of the equation is important in the initial stage of simulation using Lattice Boltzmann Method. The equation is discretised in time and space, for a set of distribution function f_i . Model of two dimension nine-velocity model (D2Q9) is used evaluate the equation form as

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = \frac{1}{\tau} \left[f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t) \right] + F \quad (2.14)$$

where Δt is time step, \mathbf{e} is the particle's velocity, τ is the relaxation time for the collision, F is the external force and $i = 0, 1, \dots, 8$. The collision term at the right hand side of Equation 2.8 has been applied the BGK approximation.

The discrete velocity is expressed as

$$\begin{aligned} e_i &= (0, 0 \text{ for } i = 0, \\ e_i &= \cos(i-1)\pi/4, \sin(i-1)\pi/4 \text{ for } 1,3,5,7 \\ e_i &= 2^{1/2} \cos (i-1)\pi/4, \sin(i-1)\pi/4 \text{ for } i=2,4,6,8 \end{aligned}$$

f_{eq} is an equilibrium distribution function, the choice of which determines the physics inherent in the simulation.

The time evolution of the lattice consist of basically two steps which are: a streaming process, where the particle densities area shifted in discrete time steps through the lattice along the connection lines in direction i e to their next neighbouring nodes and a collisions step, where locally a new particle distribution is computed by evaluating the right hand side of Equation 2.14.

Free energy theory in two-phase lattice Boltzmann model states that the physical inherent in simulation described by the equilibrium. Then, a power series in local velocities is assumed.(Zaleski, G, 1991).

$$f_i^{eq} = A + B(e_{i,\alpha}u_\alpha) + C(e_{i,\alpha}e_{i,\beta}u_\alpha u_\beta) + Du^2 + G_{\alpha\beta}e_{i,\alpha}e_{i,\beta} \quad (2.15)$$

where the summation over repeated Cartesian indices is understood. The coefficient A, B, C, D and $G_{\alpha\beta}$ are determine by placing constraint on the moment of f_i^{eq} . In order that collision term conserves mass and momentum, the first moment of f_i^{eq} are constrained

$$\sum_i f_i^{eq} = \rho \quad (2.16)$$

$$\sum_i e_{i,\alpha} f_i^{eq} = \rho u_\alpha \quad (2.17)$$

The next moment of f_i^{eq} is chosen such that the continuum macroscopic equations approximated by evolution equation correctly describe the hydrodynamics of the component, non-ideal fluid. This gives

$$\sum e_{i,a} e_{i,\beta} f_i^{eq} = P_{a\beta} + \rho u_a u_\beta + \nu [u_a \partial_a (\rho) + u_y \partial_y (\rho) \delta_{a\beta}] \quad (2.18)$$

Where $\nu = c^2 \left(\tau - \frac{1}{2} \right) \Delta t / 3$ the kinematic shear viscosity and $P_{a\beta}$ is the pressure tensor. In order to fully constrain the coefficient A, B, C, D and $G_{a\beta}$ fourth condition is needed, which is

$$\sum e_{i,a} e_{i,\beta} f_i^{eq} = \frac{\rho c^2}{3} (u_a \delta_{\beta y} + u_\beta \delta_{ay} + u_\lambda \delta_{a\beta}) \quad (2.19)$$

2.7 Van De Waals Fluid

The Van De Waals equation of state is

$$\left(p + \frac{n^2 a}{V^2} \right) (V - nb) = nRT \quad (2.20)$$

where n is the mole number, a and b are constant characteristic of a particular gas and R is the gas constant. p , V and T are as usual the pressure, volume and temperature. If the isotherms of Van De Waals gas for p - V is plotted, one sees a point of reflection on the isotherm corresponding to the critical point of a gas. In other words we have

$$\left(\frac{\partial p}{\partial V} \right)_{T=T_c} \quad (2.21)$$

$$\left(\frac{\partial^2 p}{\partial V^2} \right)_{T=T_c} \quad (2.22)$$

n is set for convenience. The Van De Waals equation can also be written in the form

$$p = \frac{RT}{V - b} - \frac{a}{V^2} \quad (2.23)$$

the first and second derivatives of the above equations will be

$$\left(\frac{\partial p}{\partial V}\right)_{T=T_c} = -\frac{RT_c}{(V_b - b)^2} - \frac{2a}{V_c^3} \quad (2.24)$$

$$\left(\frac{\partial^2 p}{\partial v^2}\right)_{T=T_c} = \frac{2RT_c}{(V_c - b)^3} - \frac{6a}{V_c^3} \quad (2.25)$$

Solve both Eq. (2.24) and Eq. (2.25) for RT_c

$$RT_c = \frac{2a(V_c - b)^2}{V_c^3} \quad (2.26)$$

$$RT_c = \frac{3a(V_c - b)^3}{V_c^4} \quad (2.27)$$

Equating the right hand side of these last two equations

$$2a = 3a \frac{V_c - b}{V_c} \quad (2.28)$$

or finally

$$V_c = 3b \quad (2.29)$$

Substitute this result back into Eq. (2.26) will give

$$T_c = \frac{8a}{27bR} \quad (2.30)$$

by substituting these two results into Eq. (2.23) will give

$$p_c = \frac{8a}{27b^2} \quad (2.31)$$

Next we define the following ‘reduced’ quantities:

$$\tilde{p} = \frac{p}{p_c}; \tilde{V} = \frac{V}{V_c}; \tilde{T} = \frac{T}{T_c} \quad (2.32)$$

thus the Van De Waals equation becomes

$$\left(\tilde{p} + \frac{3}{\tilde{V}^2}\right)(3\tilde{V} - 1) = 8\tilde{T} \quad (2.33)$$

2.8 Maxwell equal area construction

Figure 2.5 shows the a plot of isotherms on a $\tilde{p}-\tilde{V}$ diagram for various \tilde{T} . For T greater than T_c , ($T > T_c$), the graph looks very much like the ideal gas isotherms. However, for T less than T_c , ($T < T_c$), a ‘loop’ (minimum and maximum) in \tilde{p} is occur. At this condition, the system separates into two phases, a gas of volume V_G and a liquid of volume V_L . The two coexisting phases both have the same pressure denoted by p_{LG} . The value of V_G , and V_L can be determined by recalling that at equilibrium condition, the chemical potentials of the two phases must be equal. As a result we come out with the situation that they can be found geometrically by the so called ‘Maxwell equal area construction’ as shown in the figure. For Example, for the value of $T=0.55$, the value of V_G , and V_L are 0.4523 (or density $\rho_G= 2.221$), 0.2043 (or density $\rho_L= 4.895$) respectively.

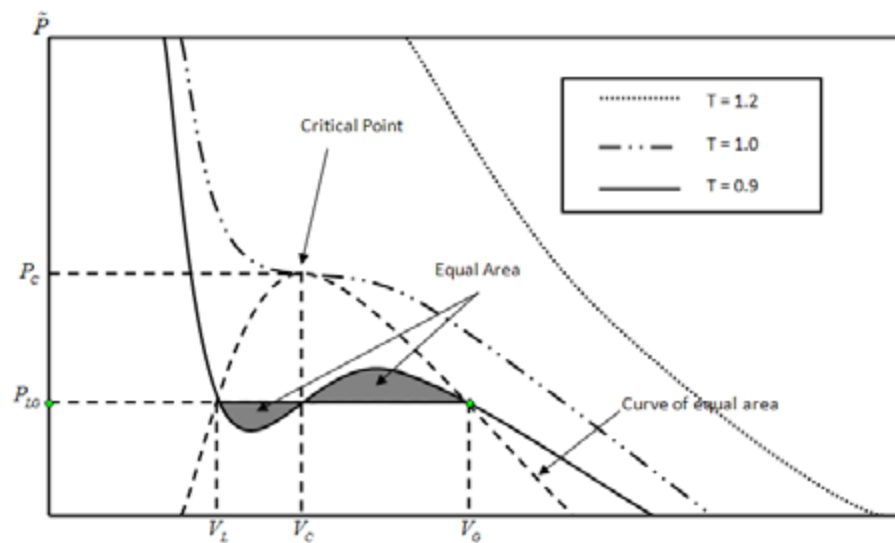


Figure 2.5: Isotherms plot of $\tilde{p} - \tilde{V}$

Source: C. S. N. Azwadi.2007

2.9 Yonetsu's Approach

The derivation of the coefficients A, B, C, D and $G_{\alpha\beta}$ proposed by Yonetsu was based on the isotropic tensor approach. In this thesis, the details derivation will not be shown and can be found in Yonetsu (Holdych, 2001).

$$A_2 = \frac{1}{12c^2} [p_o - \kappa(\partial_\gamma \rho)^2 - \kappa \rho \partial_{\gamma\gamma} \rho] \quad (2.34)$$

$$A_1 = 4A_2 \quad (2.35)$$

$$A_o = \rho - 4(A_1 + A_2) - \frac{3}{2c^2} [2vu_\gamma \partial_\gamma \rho + \kappa(\partial_\gamma \rho)^2] \quad (2.36)$$

$$B_2 = \frac{\rho}{12c^2} \quad (2.37)$$

$$B_1 = 4B_2 \quad (2.38)$$

$$C_2 = \frac{\rho}{8c^2} \quad (2.39)$$

$$C_1 = 4C_2 \quad (2.40)$$

$$D_2 = -\frac{\rho}{24c^2} \quad (2.41)$$

$$D_1 = 4D_2 \quad (2.42)$$

$$D_o = -\frac{2\rho}{3c^2} \quad (2.43)$$

$$G_{2xx} = \frac{1}{8c^4} [2vu_x \partial_x \rho + \kappa(\partial_x \rho)^2] \quad (2.44)$$

$$G_{2xy} = G_{2yx} = \frac{\nu}{8c^4} (u_x \partial_y \rho + u_y \partial_x \rho) + \frac{\kappa}{8c^4} (\partial_x \rho)(\partial_y \rho) \quad (2.45)$$

$$G_{2yy} = G_{2xx} \quad (2.46)$$

$$G_{1a\beta} = 4G_{2a\beta}, \text{ for all } a, \beta \quad (2.47)$$

Yonetsu et al.(2001) claimed that their model could predict well the phenomenon of bubble shear and give a very good agreement with analytical result for the Laplace' law pressure of the droplet-gas system.

3.2 Flow Chart

To achieve the objectives of this project, a methodology has been constructed like figure 3.2. The methodology flow chart is purposed to give guidelines and directions to successfully accomplish the main goal of this project.

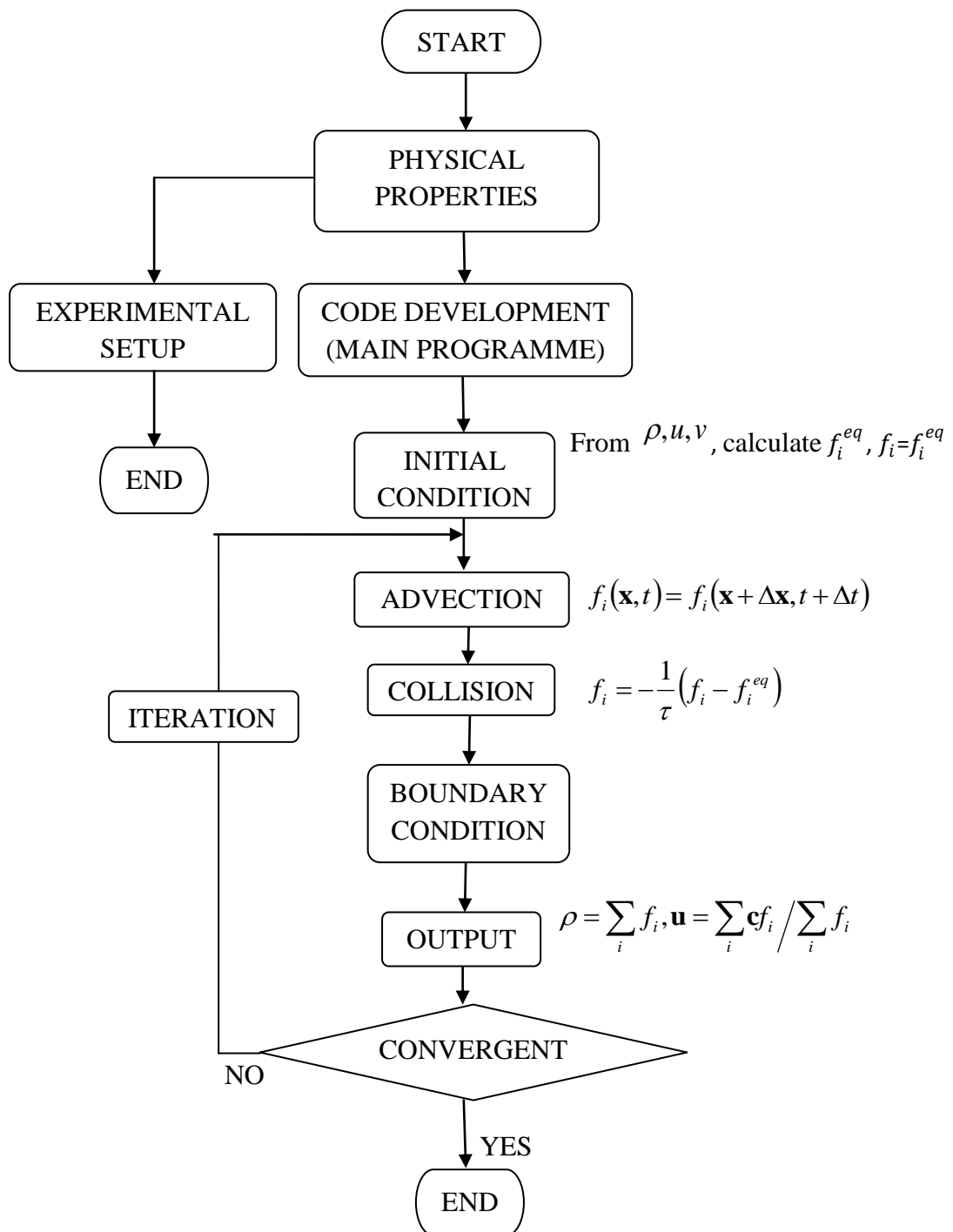


Figure 3.1: Project Flowchart

3.3 Physical Properties

The properties of air and water were used in the transport equations when the computational cell is in the liquid or the gas phase, respectively. At interface between the gas and liquid phases, the mixture properties of the gas and liquid phases based on the volume fraction weighted average were used. The density and viscosity in each cell at interface were computed by the application of following equations:

$$\rho = \alpha_G \rho_G + (1 - \alpha_G) \rho_L \quad (3.1)$$

Where ρ_G and ρ_L is density of gas and liquid phase respectively, while α_G is the volume fraction of gas. Schematic is shown as the figure 3.3.

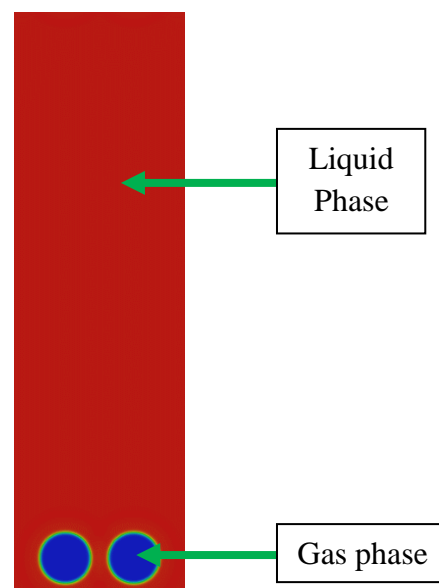


Figure 3.2: Example physical properties in simulation condition

3.5 Initial Condition

In the computational code, the value of density, velocity and viscosity will be dimensionless. The value of density liquid and gas phase according the unit S.I is $985.2\text{kg}/\text{m}^3$ and $1.092\text{kg}/\text{m}^3$ respectively. The value of liquid and gas phase is 4.895 and 2.211 respectively in dimensionless. The velocity is set starting from 0 which is from the initial condition. From the value of the density, velocity and viscosity it computational calculate to obtain the f_i^{eq} , and the $f_i=f_i^{eq}$.

```
_param.n_l = 4.895;// density of liquid
_param.n_g = 2.211;// density of gases

//set velocity
tmpNode.v.x = 0.0;
tmpNode.v.y = 0.0;
```


3.6 Advection

The BGK (Bhatnagar-Gross-Krook) is approximation that used in the simplest LBM. Succi(2001) provides excellent discussions of more complex models and the path to BGK. Streaming and collision (i.e., relaxation towards local equilibrium) look like this:


from
$$f_i(\mathbf{x}, t) = f_i(\mathbf{x} + \Delta\mathbf{x}, t + \Delta t) \quad (3.2)$$

to

$$f_a(\mathbf{x} + \mathbf{e}_a \Delta t, t + \Delta t) = f_a(\mathbf{x}, t) - \frac{[f_a(\mathbf{x}, t) - f_a^{eq}(\mathbf{x}, t)]}{\tau}$$



Streaming



collision

(3.3)

In streaming, the direction-specific densities f_a will move to the nearest neighbor lattice nodes. The scheme (originally due to Louis Colonna-Romano) shown in Figure 24 provides a convenient neighbor address relative to the point from which the f_s are being streamed. The ip , in , jp , and jn are computed at the beginning of their respective loops.

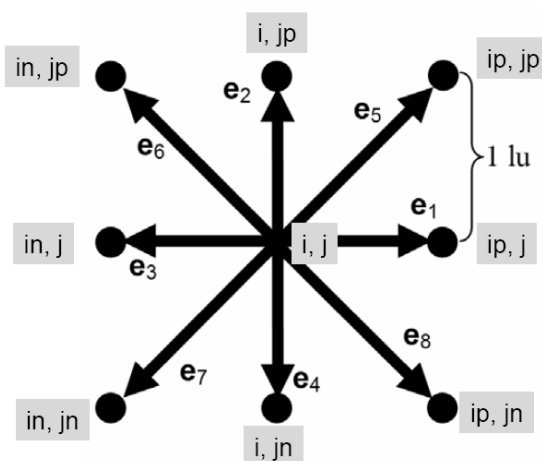


Figure 3.4: Neighbor referencing

Source: Louis Colonna-Romano.

```

// Streaming step.
for( j=0; j<LY; j++)
{
jn = (j>0 )?(j-1):(LY-1)
jp = (j<LY-1)?(j+1):(0 )
for( i=0; i<LX; i++)
{
if( !is_interior_solid_node[j][i])
{
in = (i>0 )?(i-1):(LX-1)
ip = (i<LX-1)?(i+1):(0 )
ftemp[j ][i ][0] = f[j][i][0];
ftemp[j ][ip][1] = f[j][i][1];
ftemp[jp][i ][2] = f[j][i][2];
ftemp[j ][in][3] = f[j][i][3];
ftemp[jn][i ][4] = f[j][i][4];
ftemp[jp][ip][5] = f[j][i][5];
ftemp[jp][in][6] = f[j][i][6];
ftemp[jn][in][7] = f[j][i][7];
ftemp[jn][ip][8] = f[j][i][8];
}
}
}

```

3.7 Collision

There are several possible types of collisions on the hexagonal lattice. Only two types are considered in the simplest FHP model; two-body collisions involve two particles while three-body collisions involve three. Two critical features of the lattice gas that allow it to simulate the Navier-Stokes equations are mass conservation and momentum conservation. Thus, it is essential that the microscopic-scale collisions honor mass and momentum conservation. In the lattice gas, all particles have the same mass and speed so that momentum conservation reduces to conservation of the vector sum of the velocities. Head-on collisions between two particles (or three particles approaching one another from $\pi/3 = 120^\circ$ separation) have no net momentum. Hence, the results of these collisions must also have zero net momentum. The code as shown below.

$$f_i = -\frac{1}{\tau} (f_i - f_i^{eq}) \quad (3.4)$$

```
// Collision step.
for( j=0; j<LY; j++)
for( i=0; i<LX; i++)
if( !is_liquid_node[j][i])
for( a=0; a<9; a++)
{
fij[a] = ftempij[a] - ( ftempij[a] - feqij[a])/tau;
}
}
```

3.8 Boundary Condition

In this chapter, it will give details for periodic, bounce back and following Zou and He (1997), constant pressure and constant velocity boundaries. Readers wishing to explore more advanced work on boundary conditions from Yu et al. (2003), Ginzburg and d’Humières (2003), Zhou et al. (2004), and the open literature. In this section, the boundary conditions for solute transport simulation will be presented. The computational code as below.

```
//set Boundary condition
//For Droplet in Shear flow

Cboundary tmpBound;

tmpBound.property = WALL;
tmpBound.v.x = _param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);
gamma_dot = 2.0 * _param.U_w / ( ( _param.ynum - 1 ) * _param.dy );
_param.Re = ( radius_droplet * radius_droplet ) * gamma_dot / _param.nu ;
_param.Ca = radius_droplet * _param.nu * gamma_dot / _param.sigma
```

3.8.1 Bounce back Boundaries

Bounce back boundaries come in several variants (Succi, 2001) and do not work perfectly (Gallivan et al, 1997 and Inamuro et al, 1995). Nevertheless, with proper consideration of the effective boundary location and for $\tau \approx 1$ (Chen et al, 1996), quite satisfactory results can be obtained as will be demonstrated below. Here we use the ‘mid-plane’ bounceback scheme in which the densities are temporarily stored inside the solids and re-emerge at the next time step. Figure 3.5 illustrates the process.

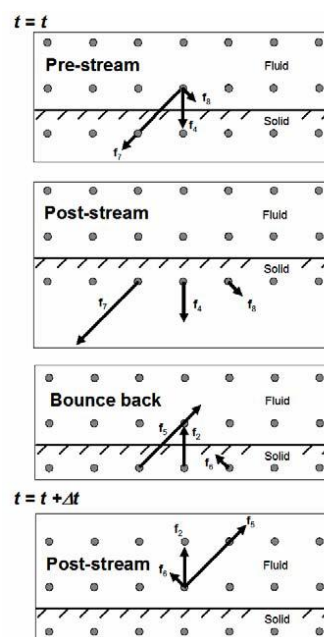


Figure 3.5: Illustration of mid-plane bounce back movement of direction specific densities f_a . The effective wall location is halfway between the fluid and solid nodes

Source: Sukop and Or, 2004

3.9 Output

Accordingly, the macroscopic fluid density is

$$\rho = \sum_i f_i \quad (3.5)$$

$$\mathbf{u} = \frac{\sum_i \mathbf{c} f_i}{\sum_i f_i} \quad (3.6)$$

The macroscopic velocity \mathbf{u} is an average of the microscopic velocities \mathbf{c} weighted by the directional densities.

Computational of the macroscopic fluid density and velocity via Equations. (3.5) and (3.6) simply involves looping through $i = 0, 1, \dots, 8$ and computing the appropriate sums as shown below.

```
// Computing macroscopic density, rho, and velocity, u= (ux,uy).
for( j=0; j<LY; j++)
{
for( i=0; i<LX; i++)
u_x[j][i] = 0.0;
u_y[j][i] = 0.0;
rho[j][i] = 0.0;
if( !is_liquid_node[j][i])
{
for( a=0; a<9; a++)
{
rho[j][i] += f[j][i][a];
u_x[j][i] += ex[a]*f[j][i][a];
u_y[j][i] += ey[a]*f[j][i][a];
}
u_x[j][i] /= rho[j][i];
```

```

u_y[j][i] /= rho[j][i];
}
}
}

```

3.10 Convergent

Convergent is the final output that its closest to the true result and logical. If the output obtain is away from the true result and the theory, the adjustable to the output will make again in code development step. In code development, the formula or value of equation will modify and will running back according the step. In convergent, we simulate of three type of effect which effect by Eotvos number, Kappa number, and Distance increment.

3.10.1 Effect of Eotvos Number

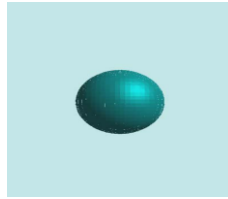
Grace (1973) has analysed a large body of experimental data on shapes and rise velocities of bubbles in quiescent viscous liquids and has shown that this data can be condensed into one diagram, provided that an appropriate set of dimensionless numbers is used. A copy of this diagram, taken from Clift et al. (1978) is reproduced in Fig. 3.9 where the dimensionless Morton (Mo), Eotvos (Eo) and Reynolds (Re) are given by

$$Eo = \frac{g\Delta\rho d^2}{\sigma} \quad (3.7)$$

$$Re = \frac{Ud}{\nu} \quad (3.8)$$

$$Mo = \frac{g\rho_L\Delta\rho\nu^4}{\sigma^3} \quad (3.9)$$

where the effective diameter d is defined as the diameter of a spherical bubble with the same volume as the bubbles.



Spherical: $Eo = 0.971$, $Mo = 1.26 \times 10^3$



Ellipsoidal: $Eo = 9.71$, $Mo = 0.1$



Skirted: $Eo = 97.1$, $Mo = 0.971$

Figure 3.6: Computed bubble shapes and rise velocities for the bubble regimes

Source: N.G. Deen, J.A.M. Kuipers, 2004

All of this shape is following the bubble diagram of Grace (1973) for the shape and terminal rise velocity of gas bubbles in quiescent viscous liquids. The bubbles diagram as shown in Figure 3.10

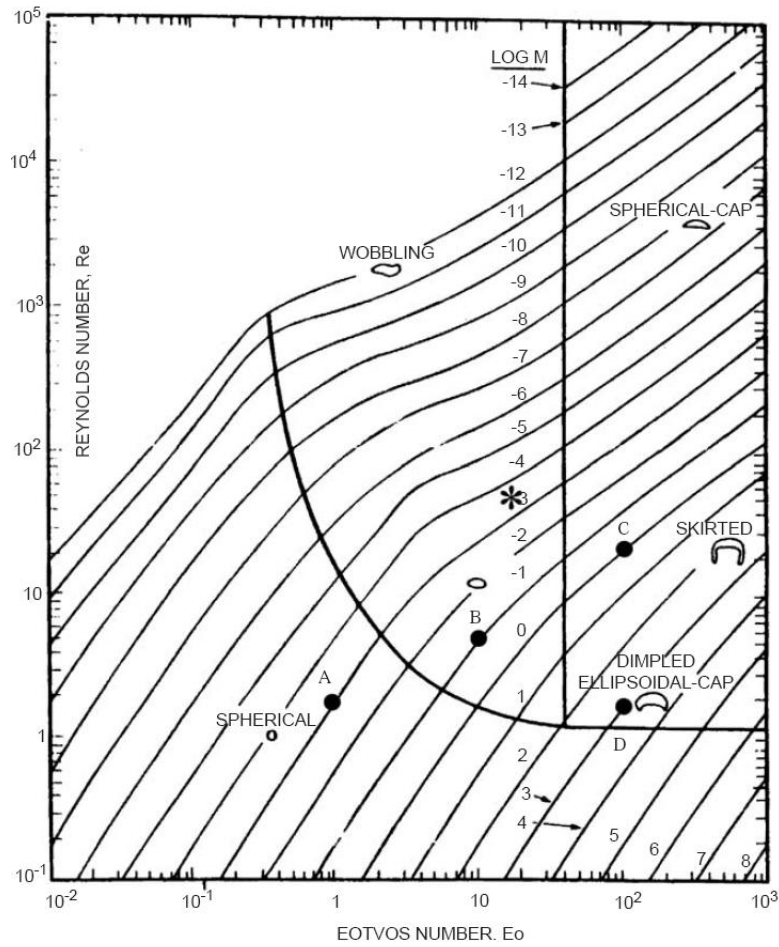


Figure 3.7: Bubble diagram for the shape and terminal rise velocity of gas bubbles in quiescent viscous liquids

Source: Grace (1973)

3.10.2 Effect of Kappa number

To determine the density profile at the interface for different values of kappa, κ can approve by equation (3.10)

$$W = \frac{\kappa}{2} (\partial_{\alpha} \rho)^2 \quad (3.10)$$

From the equation (3.10), know that gradient of density, ρ is inversely proportional to the kappa, κ . As can be seen from the graph from the Figure 3.11 , the value of κ is related to the density gradient at the interface

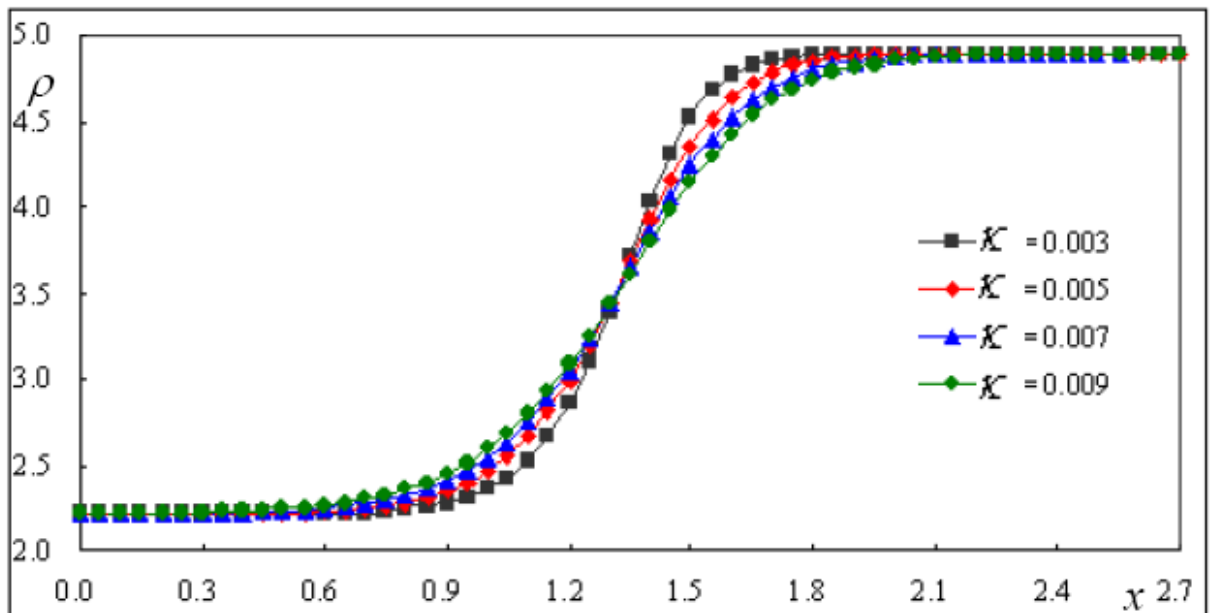


Figure 3.8: Density gradient at the interface for various value of κ

Source: Grace (1973)

3.10.3 Effect of Distance Increment

This section, it analyze effect the bubbles can collide when the position among each bubble will closer. This effect can calculate the position the bubble start to collide in the same of surface tension value. Experimental and simulation will compare together in this section. These for approve that the bubble at the most distant is absolutely not coalesce and at the certain distance that position bubble will be coalesce. It will shown in the chapter 4 for result and discussion.

3.11 Experimental Setup

In this section, it discuss about the preparing for equipment of experiment for bubble rise. To developing the experiment, the transparent box will use to easy for observation of bubble. The transparent box also guarantee in vacuum condition. The white sponge will use for closing the bottom of transparent box from water exit. This experiment used for analyze the effect of bubble interface with distance among of bubble decreasing. Actually, the result of this experiment will obtain more of one error causes by outside error such as force acting, pipe leakage and the position of metal tube. The apparatus shown as in Figure 3.9

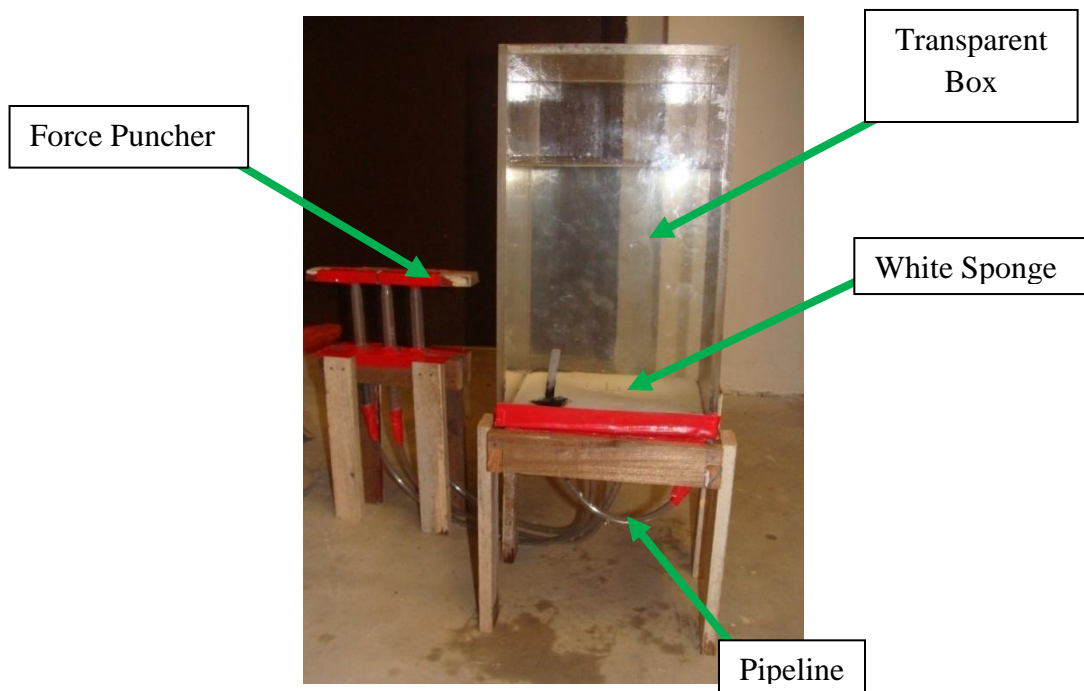


Figure 3.9: Experimental setup for different location bubbles

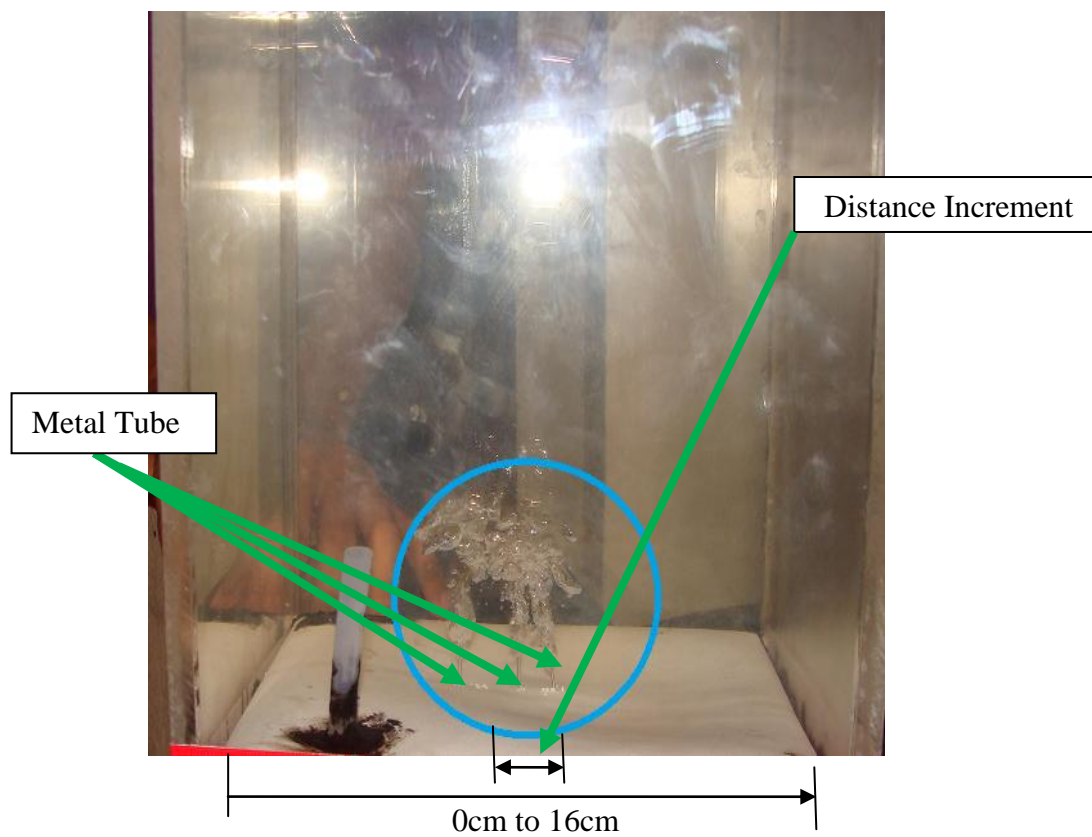


Figure 3.10: Position of metal tube as a source of bubbles

Figure 3.10 show the position among of metal tube. The distance every metal tube will more closer to each other according the value of distance given. The distance used for each experiment as shown in Table 3.1.

Table 3.1: Differential location of each bubbles

Position/No. Experiment	Left Side (cm)	Centre (cm)	Right Side (cm)
1	6.5	8.0	9.5
2	7.0	8.0	9.0
3	7.5	8.0	8.5

This experiment done according the specification of apparatus. The list specification of apparatus as shown in Table 3.2 that used to complete the experiment. This experiment also used the material that can avoid the water exit. The material and criteria used also shown in Table 3.3.

Table 3.2: Item used in the experiment

No.	Item	Length (cm)	Diameter (cm)	Wide (cm)
1.	Metal Tube	3.4	0.1	-
2.	Pipeline	48	0.5	-
3.	Transparent Box	33	-	16
4.	Shringe	6	1.8	-

Table 3.3: Criteria each item used in the experiment



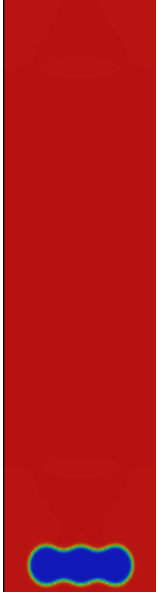
No.	Item	Part	Criteria
1.	White Sponge	Bottom	Can close back the hole causes by metal tube to change the position of metal tube
2.	Cork	Bottom	To avoid the water exit from white sponge if leakage
3.	Glass Transparent Box	Cover	Easy to detect any leakage and it impervious water
4.	Transparent Tube	Pipeline	To sure that pipeline have no bubble
5.	Shringe	Puncher	To sure the gas phase volume in same value

4.2 Effect by Distance Increment between Bubbles (Case 1)

This simulation to assess which position of bubbles will start to coalesce and deform together when rising. From the result of simulation and the position we setup, we saw which one bubbles will start to coalesce. The position will start from the most distant between bubbles where the position is $(\frac{36}{160}, \frac{80}{160}, \frac{124}{160})$ and follow up with the position $(\frac{40}{160}, \frac{80}{160}, \frac{120}{160})$ and finally $(\frac{44}{160}, \frac{80}{160}, \frac{116}{160})$. In simulation, we use the partial fraction value to coordinate the position. Cause of that in partial fraction, the smaller movement of position bubble can be adjusted.

In these simulations, gravitational is set to be 0.0001962, κ value is set to be 0.0075, density of fluid is set to 4.895 for liquid and 2.211 for gas. The periodic boundary condition is employed in all boundaries. In this study, the cavity shaped grid was used. Grid size used in these simulation is 201 x 681. In these simulation will show the better result compare from the experimental result. The result is shown as Table 4.1

Table 4.1: Comparison the shape in differential position cases

Position	$36/160, 80/160, 124/160$	$40/160, 80/160, 120/160$	$44/160, 80/160, 116/160$
Shape Coalesce			
Time Steps	130	8	7

4.2.1 Discussion (Case 1)

From the observation result, the result from the experimental is different with the simulation result. However, the direction of bubbles from the experimental is similar with the simulation whereby the direction is vertical. The force use in the experiment appear the bubbles is constant and similar to the simulation force. From the simulation, the bubbles start to coalesce when the position $(40/160, 80/160, 120/160)$. If looking at the time step, the position of $(36/160, 80/160, 124/160)$ still not coalesce although reach at 130 time steps. These cause by the interface of each bubbles. The distance between bubbles is mostly distant therefore coalesce between bubble will not occur. The reason is density gradient of interfaces is constant. Density gradient is related the kappa number. Therefore, if the position between bubbles a little close so the interface of




bubbles will pulled the bubbles to each other. Refer Appendix E1,E2, and E3 for full motion.

The bubbles coalesce from the experimental occurs when bubble reach at certain height. In experimental, the bubbles collides at certain height because the bubbles exit in smaller bubbles initially. When bubbles reach at certain height, the radius of bubbles increase. Therefore, the bubbles coalesce occur at certain height. These depend of velocity of bubbles in the water and it mostly differently with simulation which the velocity is constant. The result for experimental can refer on section 4.5.

4.3 Effect by Eotvos number (Case 2)

In this section, the two-dimensional triple bubbles rising under buoyancy is simulated. The density of each phase are taken as liquid is 4.895 and gas is 2.211. The periodic boundary condition is employed at all boundaries. Initially, it is located at the lower region (one sixth of the height) of computational domain of 201 x 681. In these simulation, the position for each bubbles is constant where the position bubbles is $(\frac{382}{1600}, \frac{800}{1600}, \frac{1218}{1600})$. The result obtain from this simulation is deform the shape of bubbles based of differentiation the Eotvos number at the same steps. The result as shown in Table 4.2.

Table 4.2: Comparison shape deform base differential of Eo number

Eotvos number	10	50	100
Shape Deform			
Time Step	135		

4.3.1 Discussion (Case 2)

From the observation of simulation result, the result is nearest to the theory result. The shape of bubbles increasing deform when the Eotvos number increase. Simulations have been done for Eo of 10, 50, and 20. Due to buoyancy force, the bubbles will move upward. In the meantime, the middle part of the bubbles will encounter a large deformation due to the hit of surrounding water. Eq. (4.1) indicates that the increase of Eo is equivalent to the decrease of the surface tension coefficient σ .

$$Eo = \frac{g\Delta\rho d^2}{\sigma} \quad (4.1)$$

It is well known that the surface tension force is to resist the deformation of the bubbles. In other words, the decrease of surface tension will enhance the deformation of




the bubbles. This phenomenon is clearly revealed in Figure 4.3.2 which display the bubbles shape.

For the case of $Eo = 10$, the shape of the bubbles does not change too much. This is because for this case, the surface tension force is strong, trying to keep its initial configuration. At $Eo = 50$, (Eo is increased and surface tension force decrease), the bubbles move up faster and the bubble's shape change. For all the cases, the bubbles is always kept symmetrically. Refer Appendix F1, F2, and F3 for full motion.

4.4 Effect of kappa number, κ (Case 3)

In simulation of kappa number, κ , gravitational is set to be 0.0001962, density of fluid is set to 4.895 for liquid and 2.211 for gas. The periodic boundary condition is employed in all boundaries. In this study, the grid size used in these simulation is 201 x 681. These simulation is constant in position between bubbles and also the Eotvos number. The Eotvos number use in this simulation is 10. The result obtain from this simulation is related the collision among of bubbles using the lower value kappa into the higher value. The result obtain shown in Table 4.3.

Table 4.3: Comparison of shape coalesce with different of κ .

Kappa number (κ)	0.003	0.005	0.0075
Shape Coalesce			
Time Step	121	6	6

4.4.1 Discussion (Case 3)

The result from simulation showing that kappa number is related the density of interface. When the kappa value used is 0.003, the density gradient of interface is decreases. Then the bubbles cannot coalesce together when density gradient decreases. When the kappa value increase to 0.005, the bubbles start to coalesce to each other, this cause the layer of interface is increases and attracted of each bubbles. Equation 4.2 indicates that the increase of kappa value will increase the gradient of density.

$$W = \frac{\kappa}{2} (\partial_{\alpha} \rho)^2$$

(4.2)

As can be seen from the graph, the value of κ is relate to the density gradient at the interface and also affect the width of interface. Refer Appendix G1, G2, and G3 for full motion.

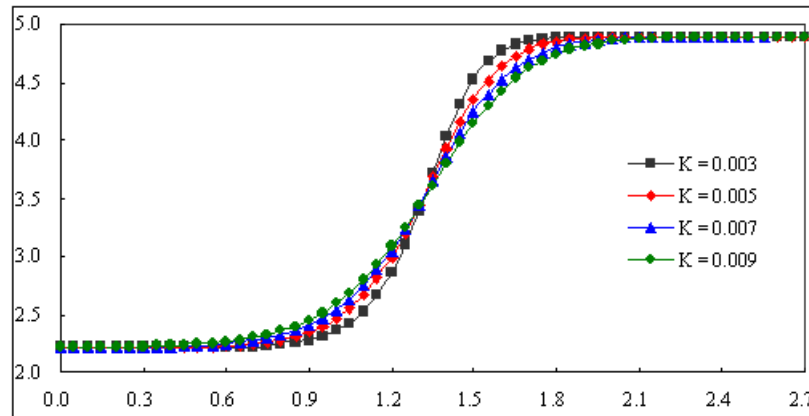


Figure 4.1: Density gradient at the interface for various value of κ

Source: Grace (1973)

4.5 Experimental Result for Case 1

From the experimental, the correctness of bubbles rise shape and also the collision is not hundred percent same with the simulation of Visual C++. The experimental result is only the expected result and it cannot to obtain the result whereby similar the simulation of software. Other words, the experimental result just for the observation result that justified the direction bubbles rise is similar with simulation result. The shape and collision of bubbles in experimental is absolutely not similar with the simulation result. The result obtain in experimental just comparison the simulation of bubbles in effect of bubbles by distance among each bubble. The result obtain shown in Figure 4.1.



Position: (6.5 cm, 8.0 cm, 9.5 cm)



Position: (7.0 cm, 8.0 cm, 9.0 cm)



Position: (7.5 cm, 8.0 cm, 8.5 cm)

Figure 4.2: Experimental result for differential position

The result in Figure 4.2 is show the effect of bubble when the position among of bubbles will closer. Refer to Table 3.1 in chapter 3 to know more details about the position among of each bubble. From the observation of the result in Figure 4.2, the bubbles will easy to collide when the position among bubble will more nearest. These cause by density gradient constant. When the thickness of density gradient thin it need move bubbles closer between other bubbles then it will collide when reach at the closer position. The position between bubbles can refer to Table 3.1.

5.2 Conclusion

A comprehensive computational fluid dynamic model has been developed and simulate for vertical multiphase flows. A detailed comparison of CFD simulation and the experimental data has been presented. From the simulation, it shows good quantitative agreement with lattice Boltzmann Method of the multiphase bubble rise. The simulated results indicate that the bubble coalesce is related to the Eotvos number, kappa number and also distance increment. Furthermore, from simulation, we know more detailed factor that could be the bubble appear and how the bubble can deform.

As that conclude, from the simulation result, the bubbles can coalesce when it closer together and the value of Kappa, κ is highest to obtain the bubbles coalesce. When the bubbles cannot coalesce each other, it will be effect configurations for each bubbles whether the bubbles size and shape will be smaller or bigger since bubbles raise upwards. Additional that, the increasing of Eotvos number will effecting the bubbles configuration. The bubbles deform from spherical until skirted and the bubbles will be separated when the bubbles reach at skirted configuration.

5.3 Recommendation

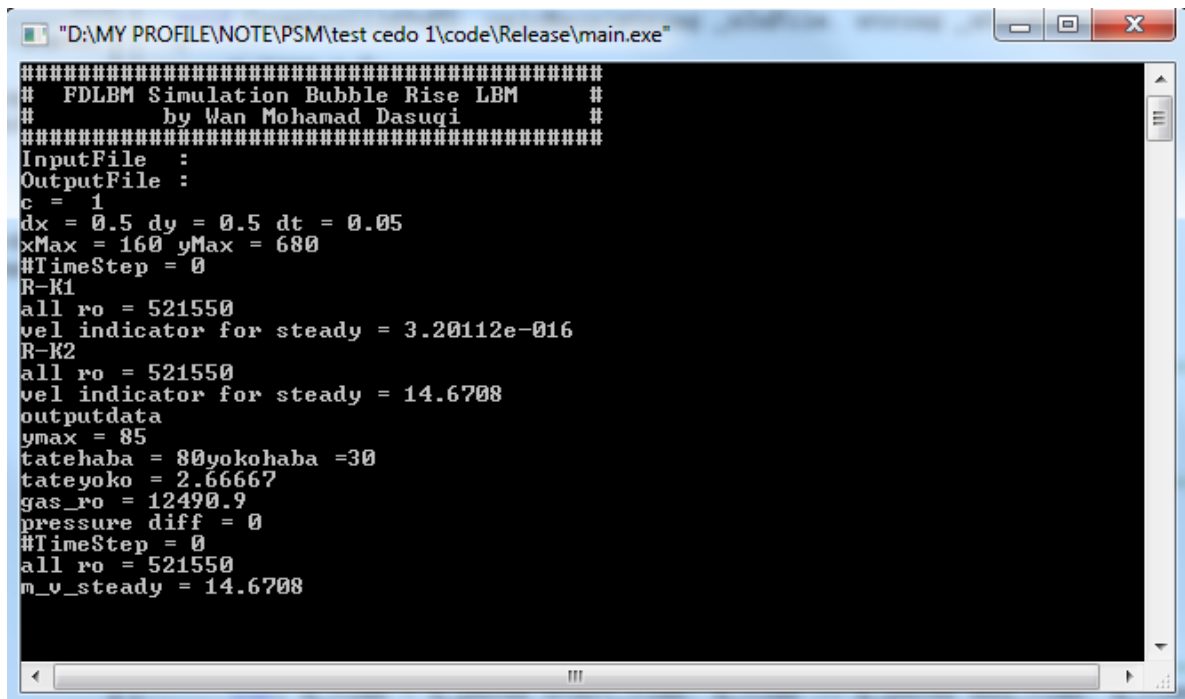
From the experimental result, the following recommendation are proposed:

- i. The experimental has develop is not enough to obtain the shape that similar with simulation result. Therefore, the equipment of experimental should be upgrade which can tracking bubbles accurately and can graphing by computational.
- ii. The simulation result has proven that bubbles deform effect by kappa number need more research to obtain more detail by upgrade the equation of kappa.
- iii. The simulation has proven from lattice Boltzmann method is the second best simulate in the multiphase of bubble. It will be the more advantage if can shows into 3 dimensional analysis whereby the result can shows in more detail.
- iv. During experimental, the bubbles exit is more bubbly. Therefore, the next experiment should be able to control the number of bubble and the velocity of bubble can constant.
- v. The simulation effect by distance increment between bubbles should to simulate many time to obtain the correct or closest position that the bubbles start to coalesce between other bubbles.

REFERENCES

- Clift, R., Grace, J.R., Weber, M., 1978. *Bubbles, Drops and Particles*. Academic Press, New York.
- Gunstensen, A.K., Rothman, D.H., Zaleski, G., 1991. *Lattice Boltzmann Model for Immiscible Fluids*, Physics Review A 43, 4320-4327.
- Holdych, Kato, Swift, D.J., Geogiadis, J.G., Buckius, R.O., 2001. Migration of a Van De Waals Bubbles: *Lattice Boltzmann Formulation*, Physics of Fluids 13, 817-825.
- H.W. Zheng, C. Shu , Y.T. Chew, *A lattice Boltzmann model for multiphase flows with Large density ratio*, Journal of Computational Physics 218 (2006) 353–371.
- Nor Azwadi Che Sidik, Takahiko Tanahashi, *Twophase Flow Simulation with Lattice Boltzmann Method*, Jurnal Mekanikal December 2007, No. 24,68 – 79
- Qian, YH, d'Humieres D, Lallemand P (1992) *Lattice BGK models for Navier-Stokes equation*. Europhys Lett 17:479-484
- Qian YH, Succi S, Orszag SA (1995) *Recent advances in lattice Boltzmann computing*. Ann Rev Comp Phys 30:195-242
- R. Zhang , X. He, S. Chen, *Interface and Surface Tension in Incompressible Lattice Boltzmann Multiphase Model*, Computer Physics communications 129 (2000) 121–130
- S. Chen, G. Doolen, *Lattice Boltzmann method for fluid flows*, Annu. Rev. Fluid Mech. 30 (1998)329–364.

APPENDIX A



```
"D:\MY PROFILE\NOTE\PSM\test cedo 1\code\Release\main.exe"
#####
# FDLBM Simulation Bubble Rise LBM #
# by Wan Mohamad Dasuqi #
#####
InputFile :
OutputFile :
c = 1
dx = 0.5 dy = 0.5 dt = 0.05
xMax = 160 yMax = 680
#TimeStep = 0
R-K1
all ro = 521550
vel indicator for steady = 3.20112e-016
R-K2
all ro = 521550
vel indicator for steady = 14.6708
outputdata
ymax = 85
tatehaba = 80 yokohaba = 30
tateyoko = 2.66667
gas_ro = 12490.9
pressure diff = 0
#TimeStep = 0
all ro = 521550
m_v_steady = 14.6708
```

Figure 6.1: Sample of output in visual C++ software

APPENDIX B

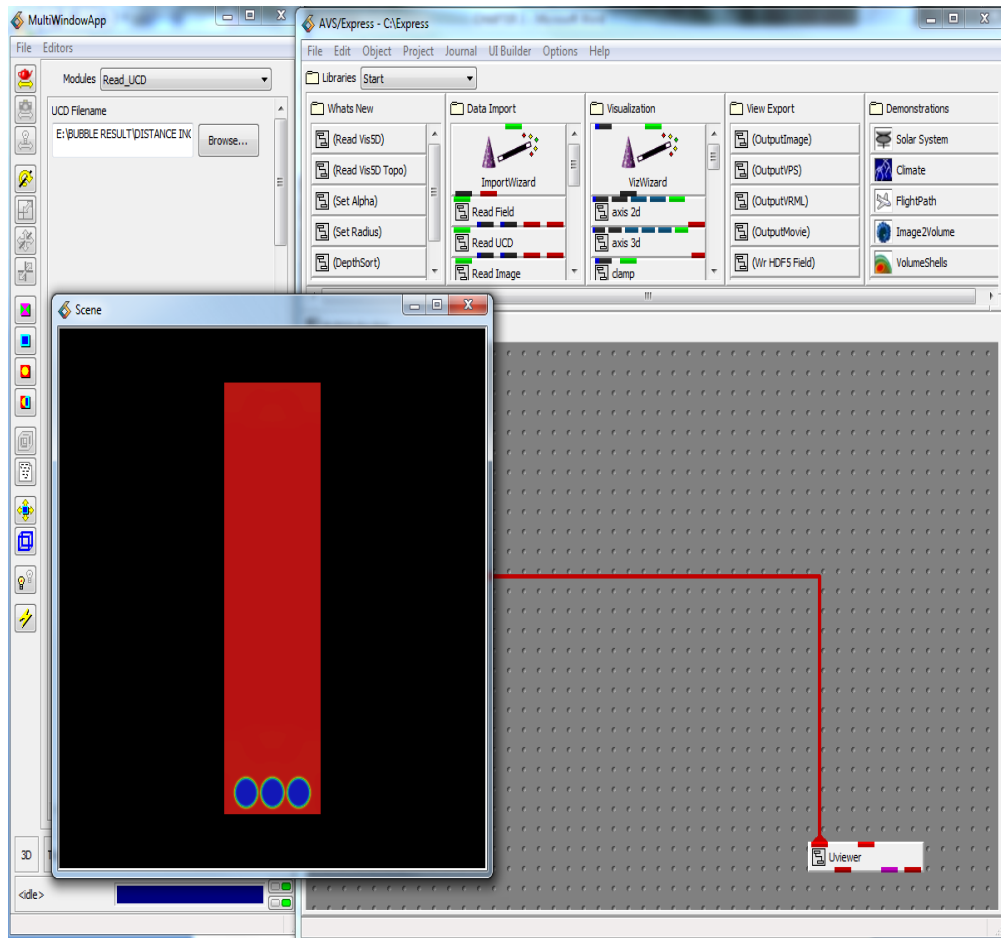


Figure 6.2: Avs express software to show the result

APPENDIX C

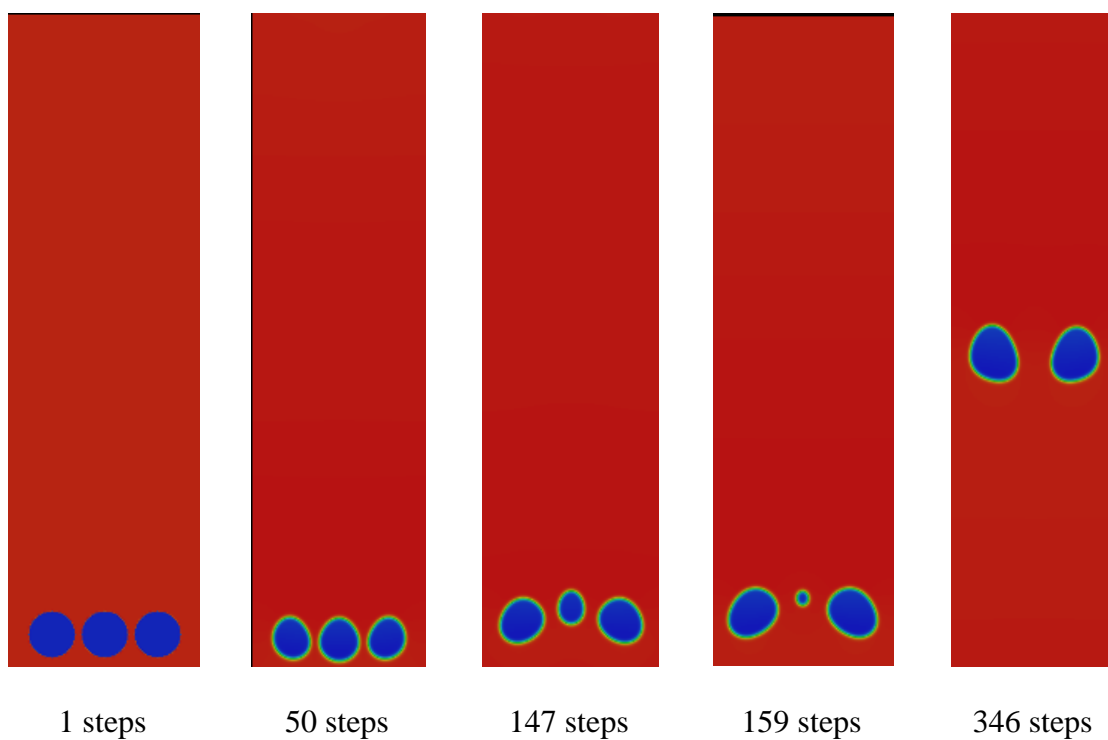


Figure 6.3: Deformation shape when not coalesce ($Eo = 10, \kappa = 0.0075,$)

APPENDIX D

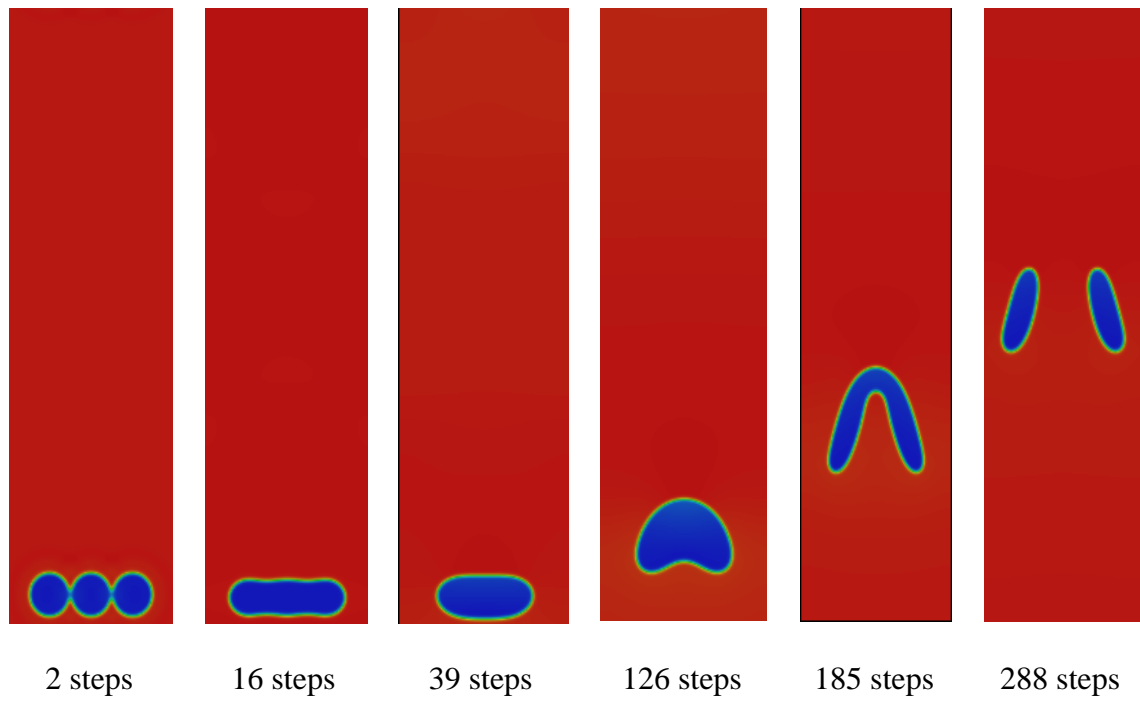


Figure 6.4: Deformation shape when can coalesce ($Eo = 10, \kappa = 0.0075$)

APPENDIX E1

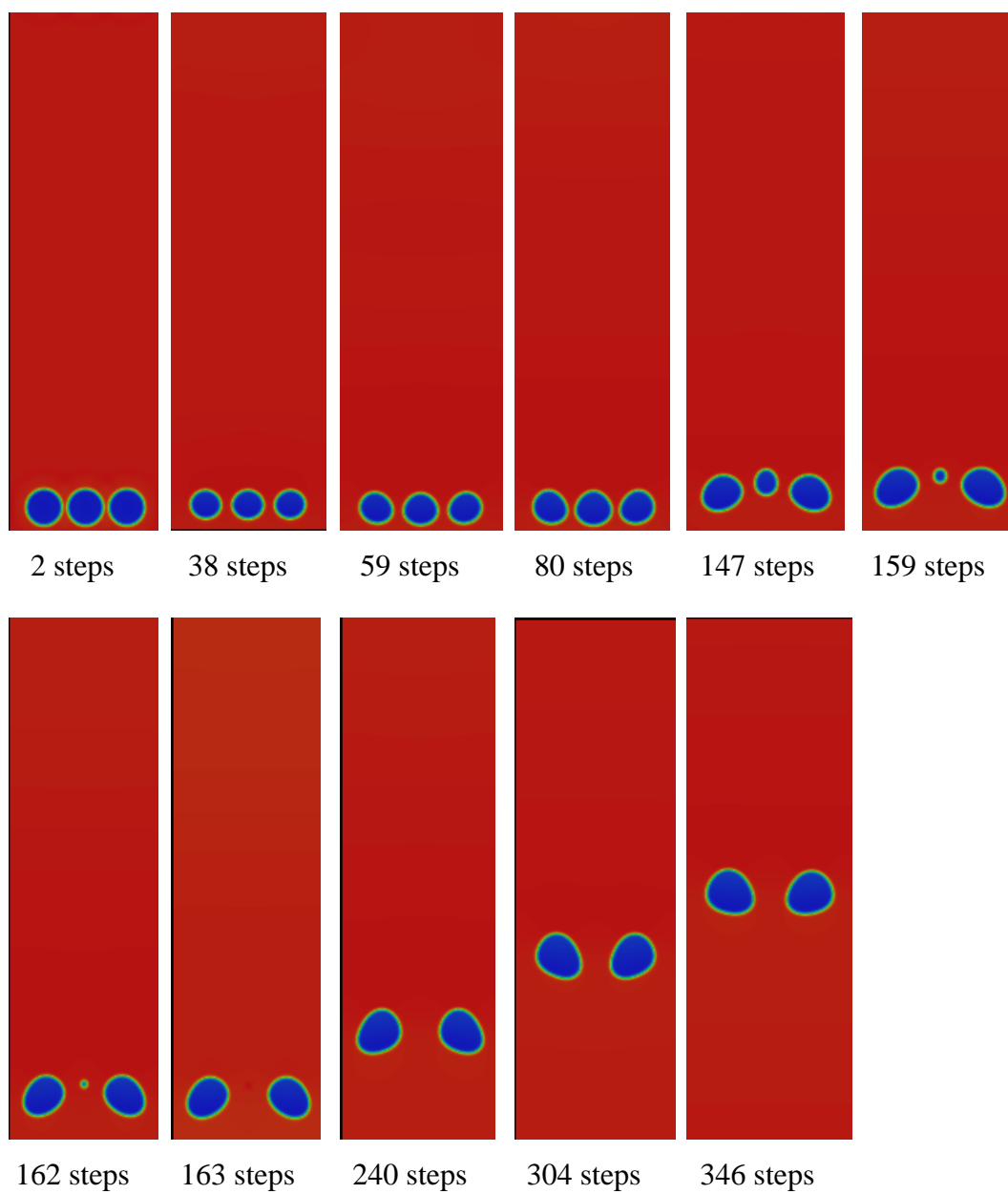


Figure 6.5: Deformation shape diagram at location $(\frac{36}{160}, \frac{80}{160}, \frac{124}{160})$

APPENDIX E2

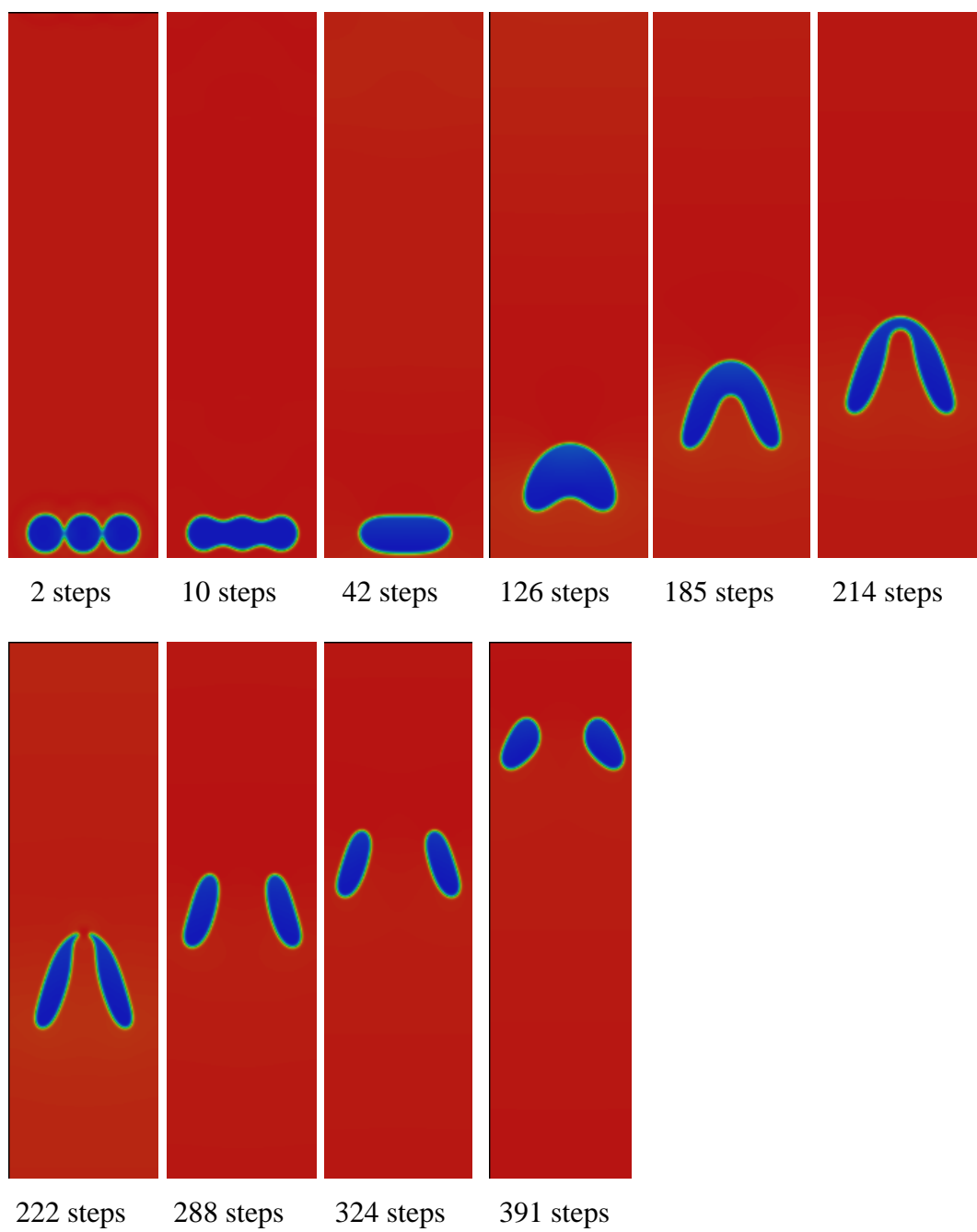


Figure 6.6: Deformation shape diagram at location $(\frac{40}{160}, \frac{80}{160}, \frac{120}{160})$

APPENDIX E3

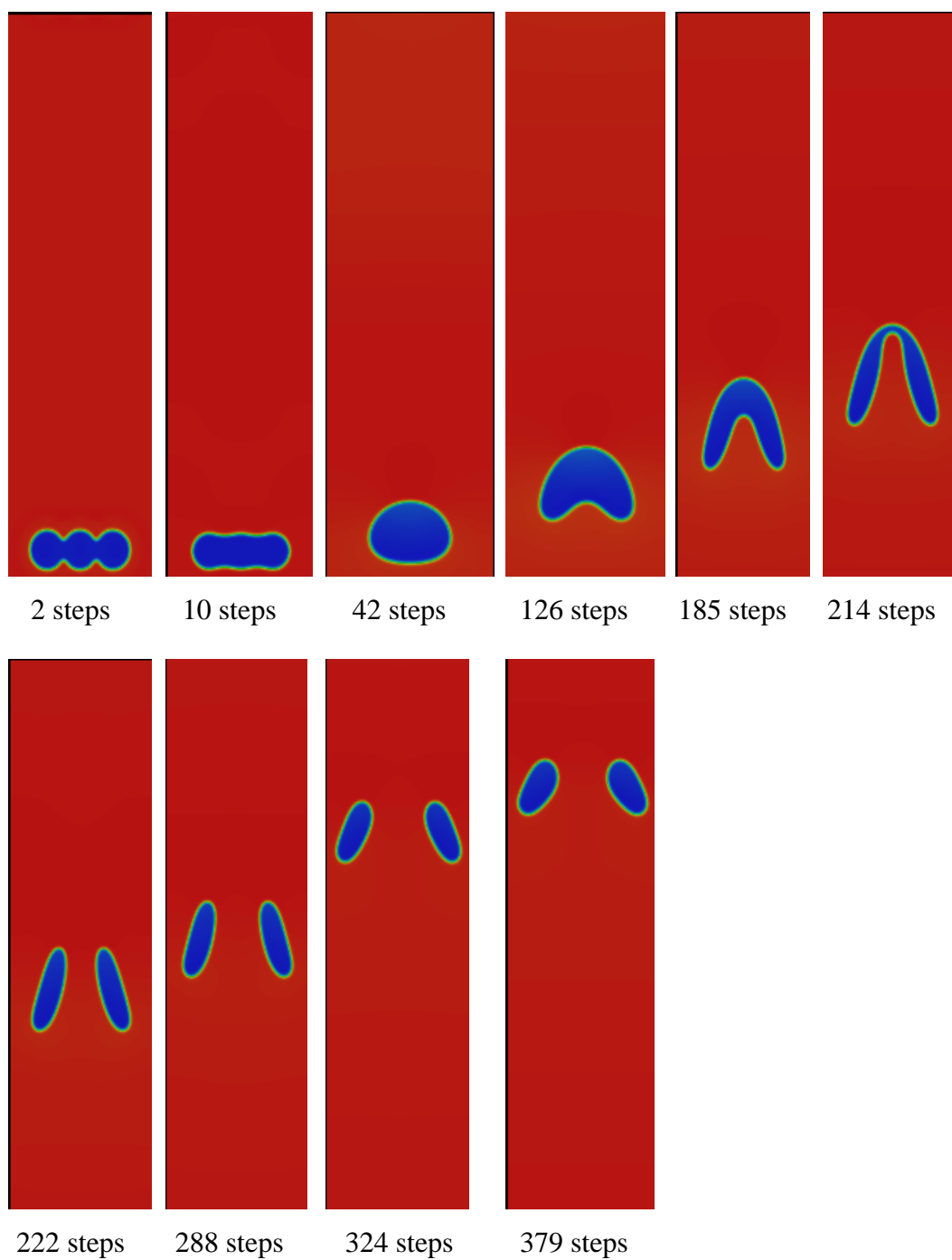


Figure 6.7: Deformation shape diagram at location $(\frac{44}{160}, \frac{80}{160}, \frac{116}{160})$.

APPENDIX F1

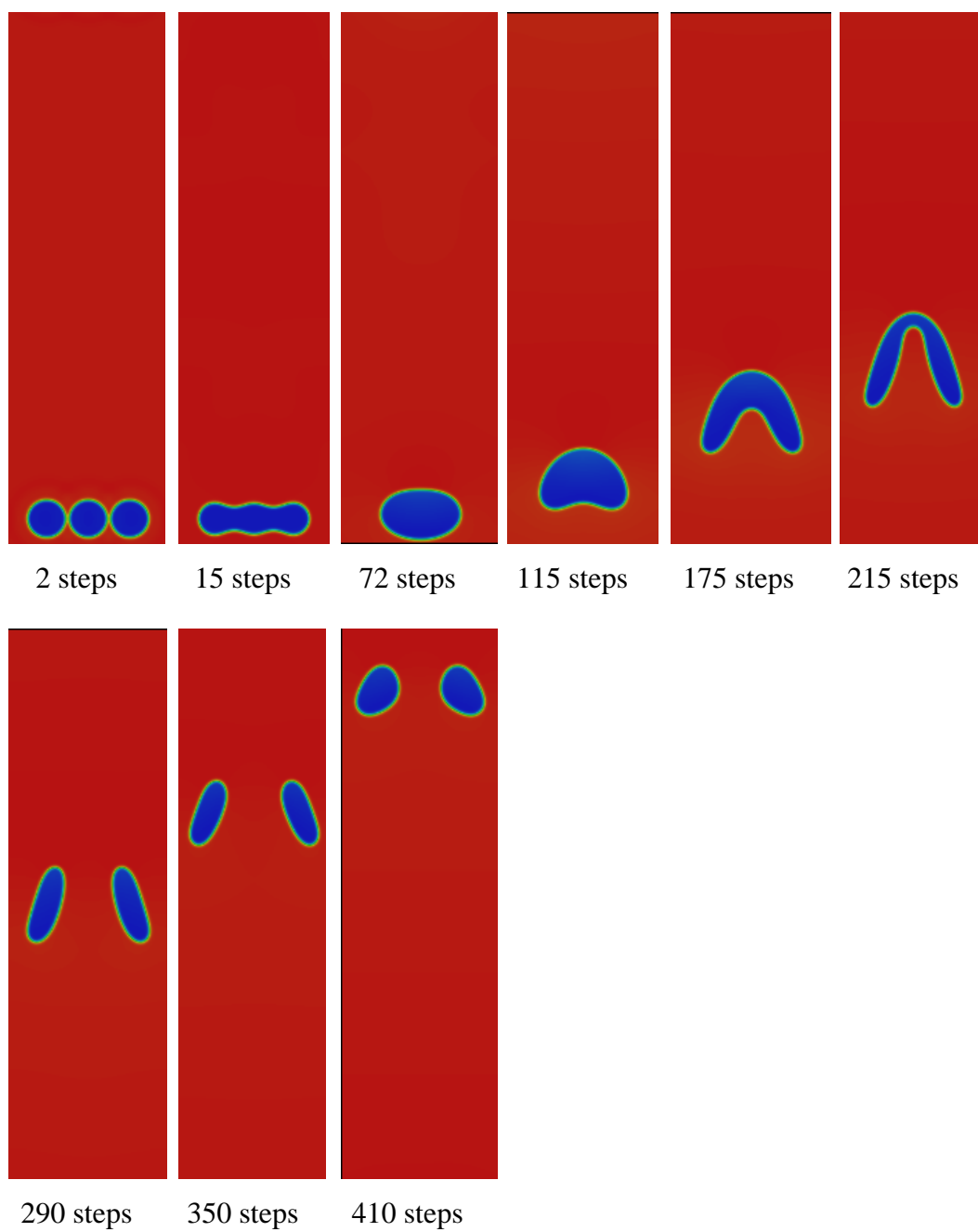


Figure 6.8: Deformation shape diagram at $Eo = 10$

APPENDIX F2

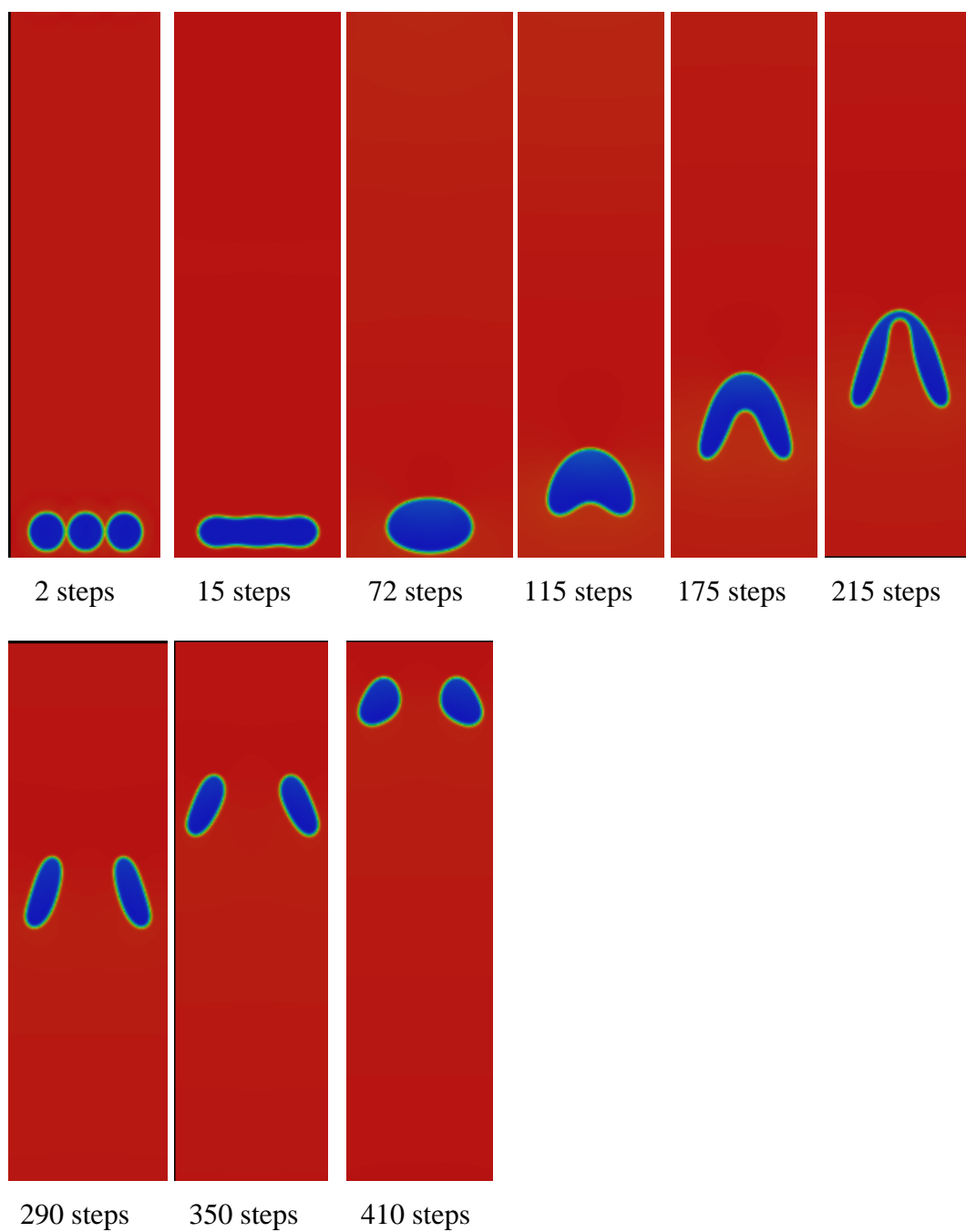


Figure 6.9: Deformation shape diagram at $Eo = 50$

APPENDIX F3

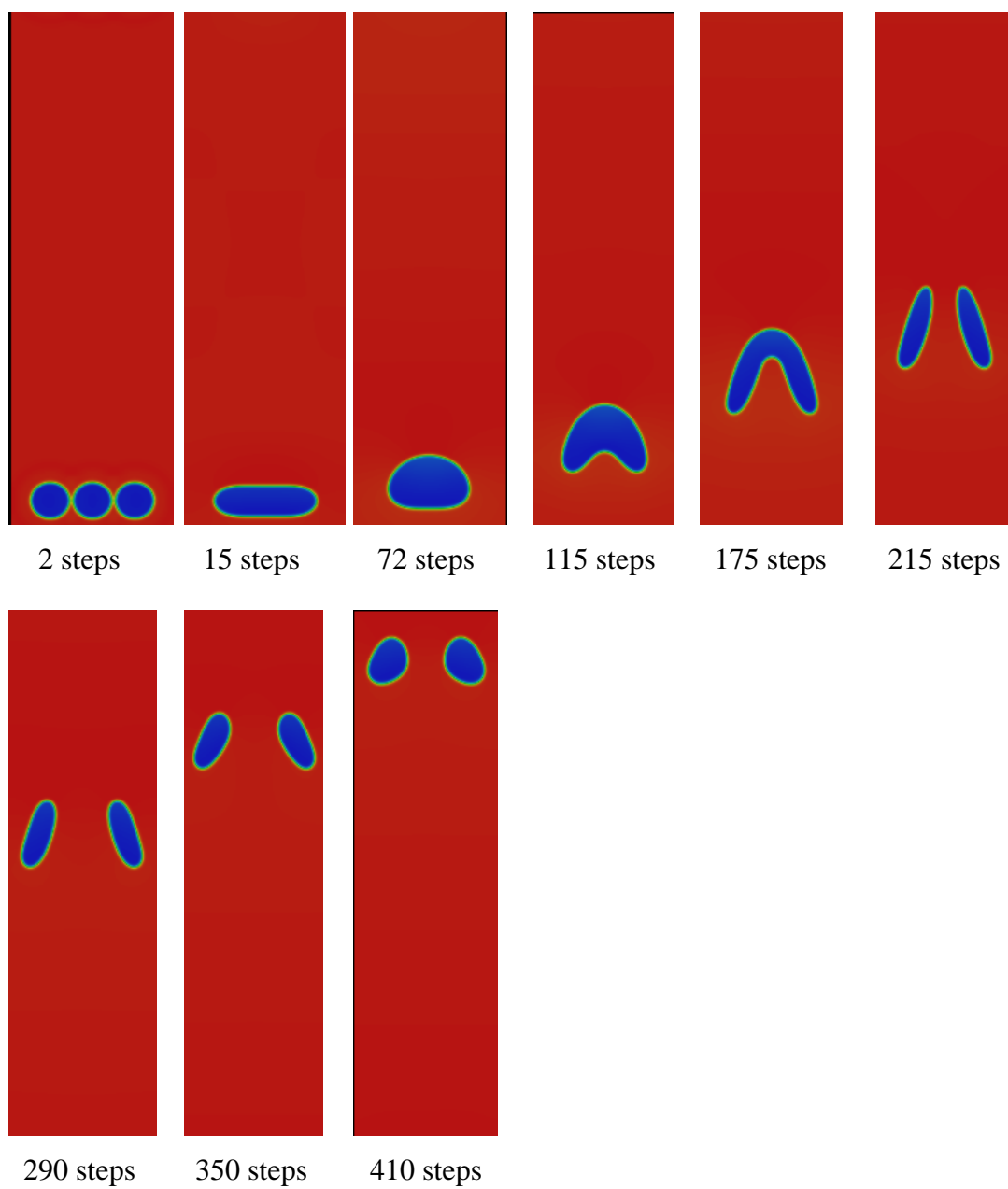


Figure 6.10: Deformation shape diagram at $Eo = 100$

APPENDIX G1

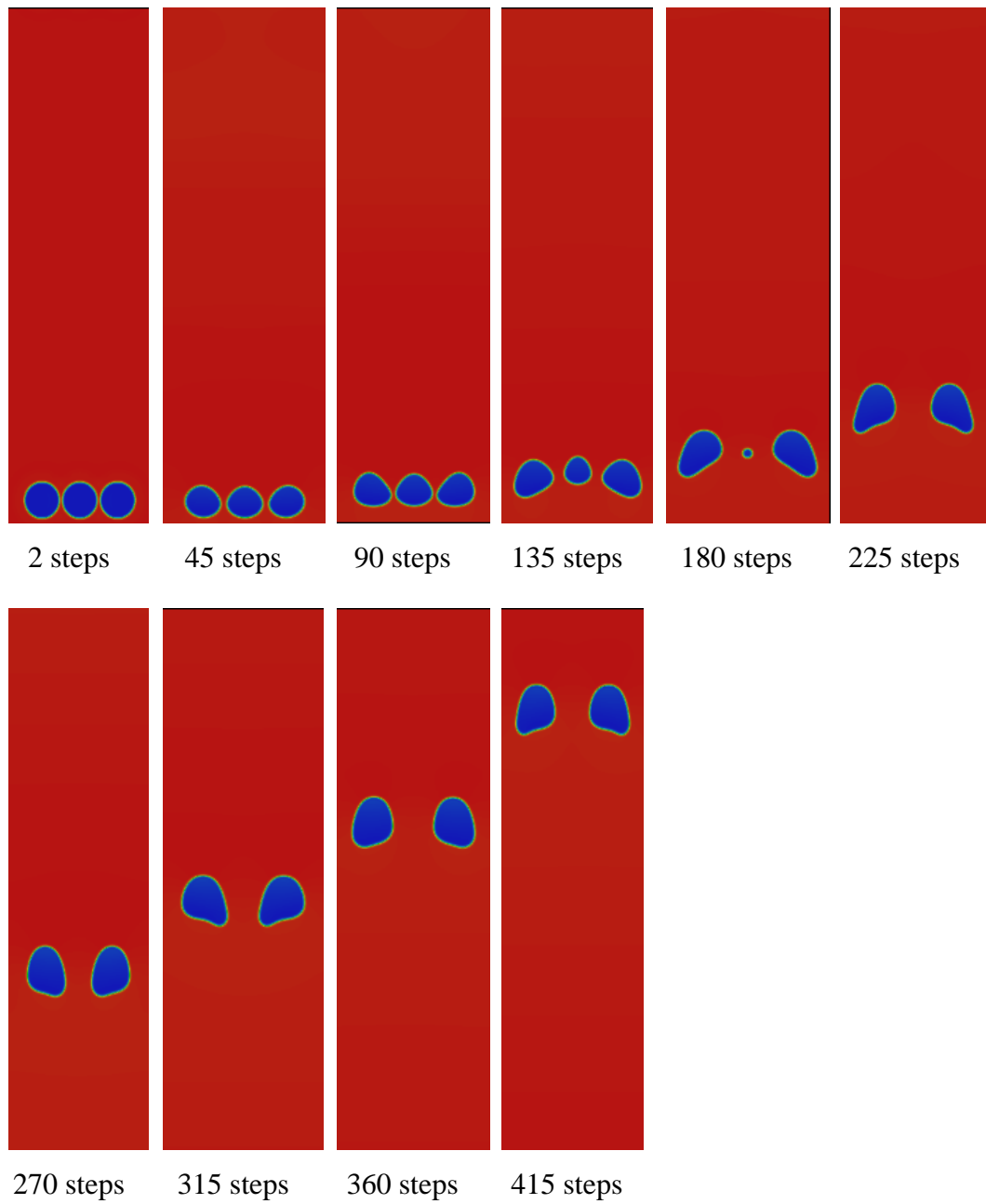


Figure 6.11: Deformation shape diagram at $\kappa = 0.003$

APPENDIX G2

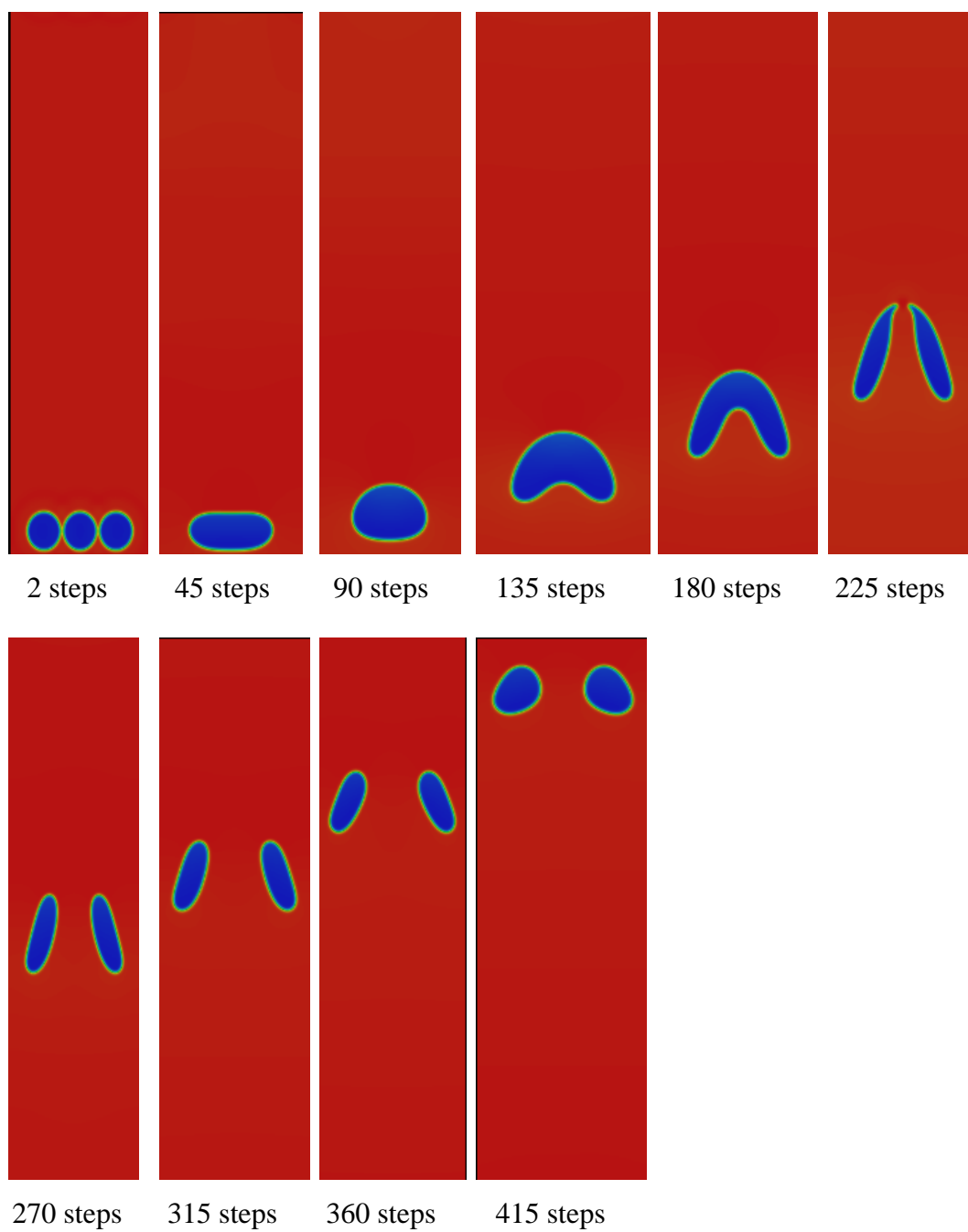


Figure 6.12: Deformation shape diagram at $\kappa = 0.005$

APPENDIX G3

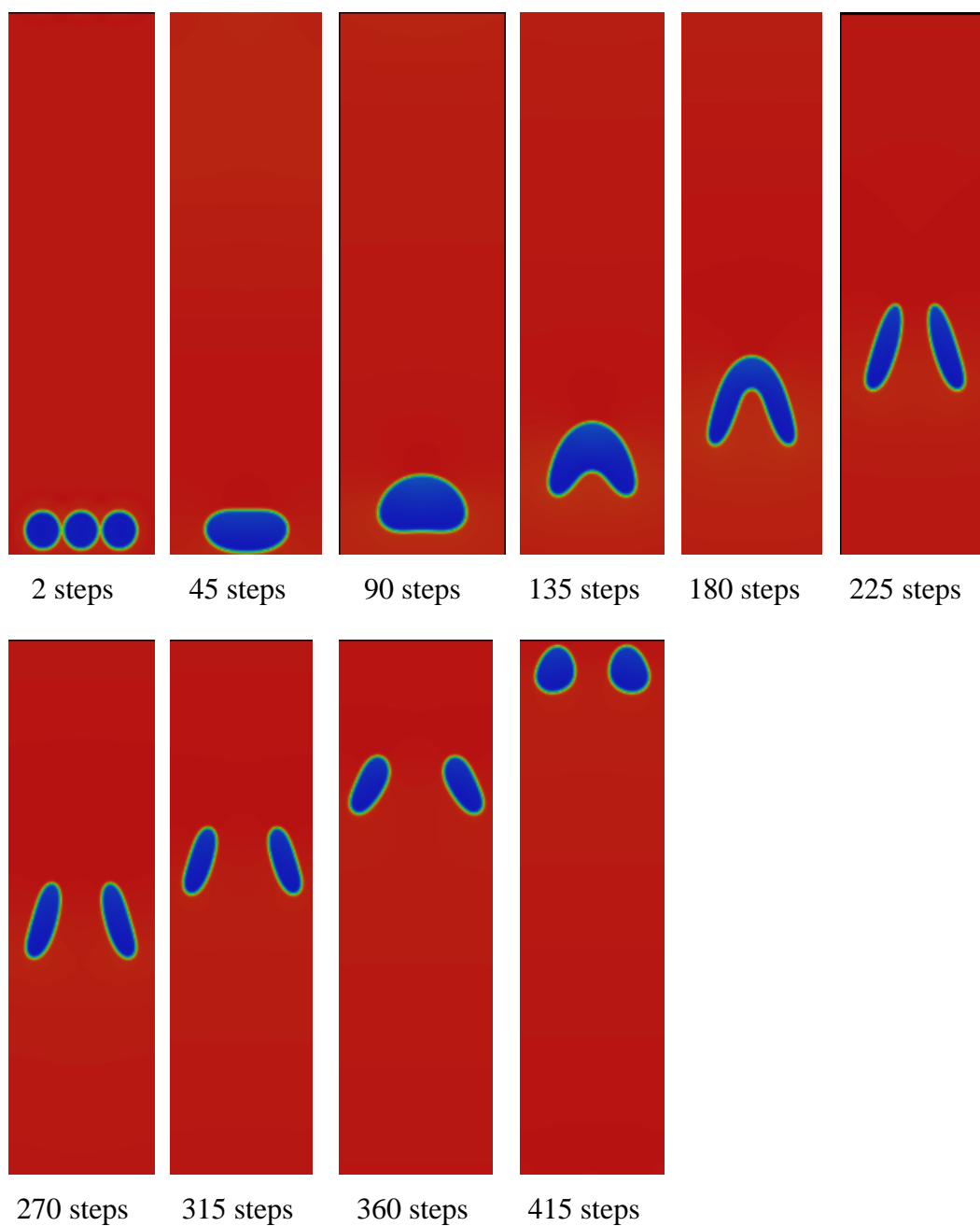


Figure 6.13: Deformation shape diagram at $\kappa = 0.007$

APPENDIX H

PROJECT ACTIVITES	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16
COLLECTING DATA																
ANALYSIS DATA																
EXPERIMENTAL DEVELOPMENT																
COLLECTING EXPERIMENT DATA																
CODE DEVELOPMENT																
CODE RUNNING																
COLLECTING CODE RUNNING																
THESIS WRITING(DRAFT1)																
PREPARING PRESENTATION																
THESIS WRITING(DRAFT2)																

Figure 6.14: Gantt Chart

APPENDIX I1

```

//Position(i,j)
tmpNode.pos.x = ((double) i) * _param.dx;
tmpNode.pos.y = ((double) j) * _param.dy;

//Init
tmpNode.init(_param);

//Sx^3 if (0 <= x <= 1)
if( ( tmpNode.pos.x - xMax *36*_param.dx /160.0 )*( tmpNode.pos.x - xMax *36*_param.dx /160.0 )
    +( tmpNode.pos.y - (radius_droplet + 5.0) )*( tmpNode.pos.y - (radius_droplet + 5.0) )
    <= radius_droplet * radius_droplet
    )
{
    tmpNode.ro = _param.n_g;
}
else if( ( tmpNode.pos.x - xMax *80*_param.dx /160.0 )*( tmpNode.pos.x - xMax *80*_param.dx /160.0 )
    +( tmpNode.pos.y - (radius_droplet + 5.0) )*( tmpNode.pos.y - (radius_droplet + 5.0) )
    <= radius_droplet * radius_droplet
    )
{
    tmpNode.ro = _param.n_g;
}
else if( ( tmpNode.pos.x - xMax *124*_param.dx /160.0 )*( tmpNode.pos.x - xMax *124*_param.dx /160.0 )
    +( tmpNode.pos.y - (radius_droplet + 5.0) )*( tmpNode.pos.y - (radius_droplet + 5.0) )
    <= radius_droplet * radius_droplet
    )
{
    tmpNode.ro = _param.n_g;
}
else
{
    tmpNode.ro = _param.n_l;
}

```

Figure 6.15: Code development for position $(\frac{36}{160}, \frac{80}{160}, \frac{124}{160})$

APPENDIX I2

```

//Position(i,j)
tmpNode.pos.x = ((double) i) * _param.dx;
tmpNode.pos.y = ((double) j) * _param.dy;

//Init
tmpNode.init(_param);

//S(x,y) if (radius_droplet <= radius_droplet)
if( ( tmpNode.pos.x - xMax *40*_param.dx /160.0 )*( tmpNode.pos.x - xMax *40*_param.dx /160.0 )
    +( tmpNode.pos.y - (radius_droplet + 5.0) )*( tmpNode.pos.y - (radius_droplet + 5.0) )
    <= radius_droplet * radius_droplet
    )
{
    tmpNode.ro = _param.n_g;
}

else if( ( tmpNode.pos.x - xMax *80*_param.dx /160.0 )*( tmpNode.pos.x - xMax *80*_param.dx /160.0 )
    +( tmpNode.pos.y - (radius_droplet + 5.0) )*( tmpNode.pos.y - (radius_droplet + 5.0) )
    <= radius_droplet * radius_droplet
    )
{
    tmpNode.ro = _param.n_g;
}

else if( ( tmpNode.pos.x - xMax *120*_param.dx /160.0 )*( tmpNode.pos.x - xMax *120*_param.dx /160.0 )
    +( tmpNode.pos.y - (radius_droplet + 5.0) )*( tmpNode.pos.y - (radius_droplet + 5.0) )
    <= radius_droplet * radius_droplet
    )
{
    tmpNode.ro = _param.n_g;
}

else
{
    tmpNode.ro = _param.n_l;
}

```

Figure 6.16: Code development for position $(40/160, 80/160, 120/160)$

APPENDIX I3

```

//Position(i,j)
tmpNode.pos.x = ((double) i) * _param.dx;
tmpNode.pos.y = ((double) j) * _param.dy;

//Init
tmpNode.init(_param);

//Size
if( ( tmpNode.pos.x - xMax *44*_param.dx /160.0 )*( tmpNode.pos.x - xMax *44*_param.dx /160.0 )
    +( tmpNode.pos.y - (radius_droplet + 5.0) )*( tmpNode.pos.y - (radius_droplet + 5.0) )
    <= radius_droplet * radius_droplet
    )
{
    tmpNode.ro = _param.n_g;
}
else if( ( tmpNode.pos.x - xMax *80*_param.dx /160.0 )*( tmpNode.pos.x - xMax *80*_param.dx /160.0 )
    +( tmpNode.pos.y - (radius_droplet + 5.0) )*( tmpNode.pos.y - (radius_droplet + 5.0) )
    <= radius_droplet * radius_droplet
    )
{
    tmpNode.ro = _param.n_g;
}
else if( ( tmpNode.pos.x - xMax *116*_param.dx /160.0 )*( tmpNode.pos.x - xMax *116*_param.dx /160.0 )
    +( tmpNode.pos.y - (radius_droplet + 5.0) )*( tmpNode.pos.y - (radius_droplet + 5.0) )
    <= radius_droplet * radius_droplet
    )
{
    tmpNode.ro = _param.n_g;
}
else
{
    tmpNode.ro = _param.n_l;
}

```

Figure 6.17: Code development for position $(44/160, 80/160, 116/160)$.

APPENDIX J1

```

_param.U_w = 0.0; // U_w = 0.0
_param.kappa = 0.0075 ; // kappa = 0.0075
_param.a = 9.0 / 49.0; // van der Waals 'a
_param.b = 2.0 / 21.0; // van der Waals 'b
_param.T = 0.55 ; // T = 0.55
_param.R = 1.0; // R = 1.0
_param.n_l = 4.895;
_param.n_g = 2.211;
_param.tau = 0.3708;
_param.sigma = 0.03033;

_param.c = 1.0;

_param.Cs = _param.c / ( sqrt(3.0) ); // Cs = c / sqrt(3)
_param.nu = _param.tau * _param.Cs * _param.Cs; // nu = tau * Cs^2

//check
cout << "c = " << _param.c << endl;
cout << "dx = " << _param.dx << " dy = " << _param.dy << " dt = " << _param.dt << endl;
cout << "xMax = " << xMax << " yMax = " << yMax << endl;

//set Boundary condition=====
//For Droplet in Shear flow

Cboundary tmpBound;

tmpBound.property = WALL;
tmpBound.v.x = _param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = _param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

gamma_dot = 2.0 * _param.U_w / ( ( _param.ynum - 1 ) * _param.dy ); // gamma_dot = 2 * U_w / (ynum - 1) * dy

_param.Re = ( radius_droplet * radius_droplet ) * gamma_dot / _param.nu ; // Re = R^2 * gamma_dot / nu
_param.Ca = radius_droplet * _param.nu * gamma_dot / _param.sigma; // Ca = R * nu * gamma_dot / sigma

```

Figure 6.18: Code development for $Eo = 10$

APPENDIX J2

```

_param.U_w = 0.0; // U_w = 0.0
_param.kappa = 0.0075 ; // \kappa = 0.0075
_param.a = 9.0 / 49.0; // van der Waals 'a
_param.b = 2.0 / 21.0; // van der Waals 'b
_param.T = 0.55 ; // T = 0.55
_param.R = 1.0; // R = 1.0
_param.n_l = 4.895;
_param.n_g = 2.211;
_param.tau = 0.3708;
_param.sigma = 0.0006066;

_param.c = 1.0;

_param.Cs = _param.c / ( sqrt(3.0) ); // Cs = 1.0 / sqrt(3)
_param.nu = _param.tau * _param.Cs * _param.Cs; // nu = tau * Cs^2

//check
cout << "c = " << _param.c << endl;
cout << "dx = " << _param.dx << " dy = " << _param.dy << " dt = " << _param.dt << endl;
cout << "xMax = " << xMax << " yMax = " << yMax << endl;

//set Boundary condition=====
//For Droplet in Shear flow

Cboundary tmpBound;

tmpBound.property = WALL;
tmpBound.v.x = _param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = _param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

gamma_dot = 2.0 * _param.U_w / ( ( _param.ynum - 1 ) * _param.dy ); // \dot{\gamma} = 2 * U_w / (y - 1) * dy

_param.Re = ( radius_droplet * radius_droplet ) * gamma_dot / _param.nu ; // Re = R^2 * \dot{\gamma} / \nu
_param.Ca = radius_droplet * _param.nu * gamma_dot / _param.sigma; // Ca = R * \dot{\gamma} / \sigma

```

Figure 6.19: Code development for $Eo = 50$

APPENDIX J3

```

_param.U_w = 0.0; // U_w = 0
_param.kappa = 0.0075 ; // kappa = 0.0075
_param.a = 9.0 / 49.0; // a = 9.0 / 49.0
_param.b = 2.0 / 21.0; // b = 2.0 / 21.0
_param.T = 0.55 ; // T = 0.55
_param.R = 1.0; // R = 1.0
_param.n_l = 4.895; // n_l = 4.895
_param.n_g = 2.211; // n_g = 2.211
_param.tau = 0.3708; // tau = 0.3708
_param.sigma = 0.0003033; // sigma = 0.0003033

_param.c = 1.0;

_param.Cs = _param.c / ( sqrt(3.0) ); // Cs = c / sqrt(3)
_param.nu = _param.tau * _param.Cs * _param.Cs; // nu = tau * Cs^2

//check
cout << "c = " << _param.c << endl;
cout << "dx = " << _param.dx << " dy = " << _param.dy << " dt = " << _param.dt << endl;
cout << "xMax = " << xMax << " yMax = " << yMax << endl;

//set Boundary condition=====
//For Droplet in Shear flow

Cboundary tmpBound;

tmpBound.property = WALL;
tmpBound.v.x = _param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

gamma_dot = 2.0 * _param.U_w / ( ( _param.ynum - 1 ) * _param.dy ); // gamma_dot = 2 * U_w / (ynum - 1) * dy

_param.Re = ( radius_droplet * radius_droplet ) * gamma_dot / _param.nu ; // Re = R^2 * gamma_dot / nu
_param.Ca = radius_droplet * _param.nu * gamma_dot / _param.sigma; // Ca = R * nu * gamma_dot / sigma
...

```

Figure 6.20: Code development for $Eo = 100$

APPENDIX K1

```

_param.U_w = 0.0; // U_w = |x|^\kappa
_param.kappa = 0.003 ; // \kappa = 0.003
_param.a = 9.0 / 49.0; // \eta = 9.0 / 49.0
_param.b = 2.0 / 21.0; // \eta = 2.0 / 21.0
_param.T = 0.55 ; // \tau = 0.55
_param.R = 1.0; // R = 1.0
_param.n_l = 4.895; // n_l = 4.895
_param.n_g = 2.211; // n_g = 2.211
_param.tau = 0.3708; // \tau = 0.3708
_param.sigma = 0.03033; // \sigma = 0.03033

_param.c = 1.0;

_param.Cs = _param.c / ( sqrt(3.0) ); // C_s = 1 / \sqrt{3}
_param.nu = _param.tau * _param.Cs * _param.Cs; // \nu = \tau * C_s^2

//check
cout << "c = " << _param.c << endl;
cout << "dx = " << _param.dx << " dy = " << _param.dy << " dt = " << _param.dt << endl;
cout << "xMax = " << xMax << " yMax = " << yMax << endl;

//set Boundary condition=====
//For Droplet in Shear flow

Cboundary tmpBound;

tmpBound.property = WALL;
tmpBound.v.x = _param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

gamma_dot = 2.0 * _param.U_w / ( ( _param.ynum - 1 ) * _param.dy ); // \dot{\gamma} = 2 * U_w / (y - 1) * dy

_param.Re = ( radius_droplet * radius_droplet ) * gamma_dot / _param.nu ; // Re = R^2 * \dot{\gamma} / \nu
_param.Ca = radius_droplet * _param.nu * gamma_dot / _param.sigma; // Ca = R * \dot{\gamma} / \sigma

```

Figure 6.21: Code development for $\kappa = 0.003$

APPENDIX K2

```

_param.U_w = 0.0; // U_w = 0.0
_param.kappa = 0.005 ; // kappa = 0.005
_param.a = 9.0 / 49.0; // a = 9.0 / 49.0
_param.b = 2.0 / 21.0; // van der Waals 'è|a
_param.T = 0.55 ; // van der Waals 'è|b
_param.R = 1.0; // R = 1.0
_param.n_l = 4.895;
_param.n_g = 2.211;
_param.tau = 0.3708;
_param.sigma = 0.03033;

_param.c = 1.0;

_param.Cs = _param.c / ( sqrt(3.0) ); // Cs = 1.0 / sqrt(3.0)
_param.nu = _param.tau * _param.Cs * _param.Cs; // nu = 0.3708 * 1.0 / sqrt(3.0) * 1.0 / sqrt(3.0)

//check
cout << "c = " << _param.c << endl;
cout << "dx = " << _param.dx << " dy = " << _param.dy << " dt = " << _param.dt << endl;
cout << "xMax = " << xMax << " yMax = " << yMax << endl;

//set Boundary condition=====
//For Droplet in Shear flow

Cboundary tmpBound;

tmpBound.property = WALL;
tmpBound.v.x = _param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

gamma_dot = 2.0 * _param.U_w / ( ( _param.ynum - 1 ) * _param.dy ); // gamma_dot = 2.0 * U_w / ( ynum - 1 ) * dy

_param.Re = ( radius_droplet * radius_droplet ) * gamma_dot / _param.nu ; // Re = R^2 * gamma_dot / nu
_param.Ca = radius_droplet * _param.nu * gamma_dot / _param.sigma; // Ca = R * nu * gamma_dot / sigma

```

Figure 6.22: Code development for $\kappa = 0.005$

APPENDIX K3

```

_param.U_w = 0.0; // U_w = 0.0
_param.kappa = 0.007 ; // kappa = 0.007
_param.a = 9.0 / 49.0; // a = 9.0 / 49.0
_param.b = 2.0 / 21.0; // b = 2.0 / 21.0
_param.T = 0.55 ; // T = 0.55
_param.R = 1.0; // R = 1.0
_param.n_l = 4.895; // n_l = 4.895
_param.n_g = 2.211; // n_g = 2.211
_param.tau = 0.3708; // tau = 0.3708
_param.sigma = 0.03033; // sigma = 0.03033

_param.c = 1.0;

_param.Cs = _param.c / ( sqrt(3.0) ); // Cs = c / sqrt(3)
_param.nu = _param.tau * _param.Cs * _param.Cs; // nu = tau * Cs * Cs

//check
cout << "c = " << _param.c << endl;
cout << "dx = " << _param.dx << " dy = " << _param.dy << " dt = " << _param.dt << endl;
cout << "xMax = " << xMax << " yMax = " << yMax << endl;

//set Boundary condition=====
//For Droplet in Shear flow

Cboundary tmpBound;

tmpBound.property = WALL;
tmpBound.v.x = _param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

tmpBound.property = WALL;
tmpBound.v.x = -_param.U_w;
tmpBound.v.y = 0.0;
_vBoundary.push_back(tmpBound);

gamma_dot = 2.0 * _param.U_w / ( ( _param.ynum - 1 ) * _param.dy ); // gamma_dot = 2 * U_w / (y - 1) * dy















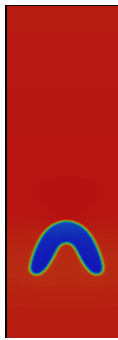
_param.Re = ( radius_droplet * radius_droplet ) * gamma_dot / _param.nu ; // Re = R^2 * gamma_dot / nu
_param.Ca = radius_droplet * _param.nu * gamma_dot / _param.sigma; // Ca = R * nu * gamma_dot / sigma

```

Figure 6.23: Code development for $\kappa = 0.007$










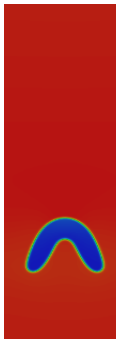
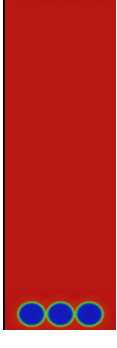



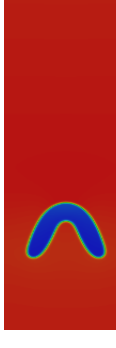
APPENDIX L1

Table 6.24: Comparison of variation bubbles configuration (Case 1)

CONFIGURATION	BUBBLES VARIATION				
(36 / 160) (80 / 160) (124 / 160)					
(40 / 160) (80 / 160) (120 / 160)					
(44 / 160) (80 / 160) (116 / 160)					
TIME STEPS	2	10	42	126	185

APPENDIX L2

Table 6.25: Comparison of variation Eotvos number, Eo (Case 2)

Eo	BUBBLES VARIATION				
10					
50					
100					
TIME STEPS	2	15	72	115	175

APPENDIX L3

Table 6.25: Comparison of variation Kappa number, κ (Case 3)

κ	BUBBLES VARIATION				
0.003					
0.005					
0.007					
TIME STEPS	2	45	90	135	180