

Efficient Computations for Multiphase Flow Problems Using Coupled Lattice Boltzmann-Level Set Methods

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften

Der Fakultät für Mathematik der
Technischen Universität Dortmund
vorgelegt von

Seyed Mohammad Amin Safi

im Januar 2016

Dissertation

Efficient Computations for Multiphase Flow Problems Using Coupled
Lattice Boltzmann-Level Set Methods

Seyed Mohammad Amin Safi

Tag der mündlichen Prüfung: 14. 3. 2016

Erstgutachter : Prof. Dr. Stefan Turek

Zweitgutachter: Dr. habil. Fathollah Varnik

Drittgutachter: Prof. Dr. Dimitri Kuzmin

to Aida

Acknowledgements

Firstly, I would like to express my sincere gratitude to Prof. Stefan Turek for his valuable and persistent support and guidance throughout the past four years. He never stopped persuading and motivating me to move forward and that has always been the strongest force to keep me proceed on the right track.

I am incredibly grateful for the invaluable discussions and consultations provided by Dr. Abderrahim Ouazzi, Dr. Otto Mierka and Dr. Hogenrich Damanik who provided me with clear insights and understandings of my research from the very first stages of the work, as well as Prof. Dominik Göddeke to whom I owe a great deal of my high performance scientific computing knowledge. I also would like to acknowledge the efforts of those who virtually helped me with their remarkable research on numerical solution of multiphase flows and LB methods, especially Dr. Taehun Lee for his renowned contributions to the field of two-phase LBM, and for providing short but fruitful dialogues on this matter.

My utmost gratitude goes indeed to my dear friend, Shaghayegh Nazari, who kindly accepted to proofread the thesis and assisted improving the final text of the thesis.

I shall also thank my colleagues and friends in the chair of Applied Mathematics in TU Dortmund, Babak Sayyid Hosseini, Dr. Masoud Nickaeen, Dr. Evren Bayraktar, Sven Buijssen, Ramzan Ali and Absaar Ul Jabbar, with whom my PhD years were all about countless unforgettable cherishing moments.

And last, but not the least, I thoroughly thank my beloved wife, Aida, who has always been a source of patience, sacrifice and trust for me and has stood by my side during the difficult times of my research with an everlasting faith in my capabilities and a deep respect to my ambitions.

S. M. Amin Safi

Abstract

Multiphase flow simulations benefit a variety of applications in science and engineering as for example in the dynamics of bubble swarms in heat exchangers and chemical reactors or in the prediction of the effects of droplet or bubble impacts in the design of turbomachinery systems. Despite all the progress in the modern computational fluid dynamics (CFD), such simulations still present formidable challenges both from numerical and computational cost point of view.

Emerging as a powerful numerical technique in recent years, the lattice Boltzmann method (LBM) exhibits unique numerical and computational features in specific problems for its ability to detect small scale transport phenomena, including those of interparticle forces in multiphase and multicomponent flows, as well as its inherent advantage to deliver favourable computational efficiencies on parallel processors.

In this thesis two classes of LB methods for multiphase flow simulations are developed which are coupled with the level set (LS) interface capturing technique. Both techniques are demonstrated to provide high resolution realizations of the interface at large density and viscosity differences within relatively low computational demand and regularity restrictions compared to the conventional phase-field LB models. The first model represents a sharp interface one-fluid formulation, where the LB equation is assigned to solve for a single virtual fluid and the interface is captured through convection of an initially signed distance level set function governed by the level set equation (LSE). The second scheme proposes a diffuse pressure evolution description of the LBE, solving for velocity and dynamic pressure only. In contrast to the common kinetic-based solutions of the Cahn-Hilliard equations, the density is then solved via a mass conserving LS equation which benefits from a fast monolithic reinitialization.

Rigorous comparisons against established numerical solutions of multiphase NS equations for rising bubble problems are carried out in two and three dimensions, which further provide an unprecedented basis to evaluate the effect of different numerical and implementation aspects of the schemes on the overall performance and accuracy. The simulations are eventually applied to other physically interesting multiphase problems, featuring the flexibility and stability of the scheme under high Re numbers and very large deformations.

On the computational side, an efficient implementation of the proposed schemes is presented in particular for manycore general purpose graphical processing units (GPGPU). Various segments of the solution algorithm are then evaluated with respect to their corresponding computational workload and efficient implementation outlines are addressed.

Contents

1	Introduction	1
2	Governing equations in macro- and microscales	4
2.1	Macroscopic conservation laws	4
2.2	Extension to multiphase flows	6
2.2.1	Regularization of discontinuities	8
2.2.2	Inclusion of the surface tension forces	9
3	The Lattice Boltzmann Method	12
3.1	The Lattice Boltzmann Method	12
3.1.1	Treating force terms in LBM	16
3.2	Multiphase LB models	18
3.2.1	Early multiphase LB models	19
3.2.2	Pressure evolution multiphase LBM	20
4	Interface capturing	25
4.1	Choice of the interface capturing technique	25
4.2	Signed distance level set method	26
4.2.1	Numerical solution of the signed distance level set equation	28
4.3	Level set reinitialization	29
4.3.1	PDE-based reinitialization	30
4.3.2	Brute-force reinitialization	31
4.4	Conservative phase-field level set method	32
4.4.1	Numerical solution of the phase-field level set equation	33
4.5	Numerical experiments	34
4.5.1	Benchmarking problems	34
4.5.2	Benchmarking results	36
5	Coupled lattice Boltzmann-level set methods	44
5.1	The one-fluid coupled LB-LS model	45
5.1.1	The one-fluid LB formulation	45
5.1.2	Space discretization for one-fluid LBE	47
5.1.3	Time integration for one-fluid LBE	48
5.2	Pressure evolution coupled LB-LS method	50
5.2.1	Pressure evolution LBE with CSF forcing	51
5.2.2	Temporal and spatial discretization of pressure evolution LBE	52
5.3	Multiple relaxation time collision for multiphase LBM	53
5.3.1	Choice of the relaxation parameters	57

5.3.2	Summary of the coupled LB-LS algorithm	59
5.4	Numerical tests	60
6	Results and benchmarking	65
6.1	Rising bubble benchmarks	66
6.1.1	Problem definition	66
6.1.2	Classification	67
6.1.3	Parameters	68
6.1.4	Benchmark quantities	68
6.1.5	Error estimation	69
6.1.6	Review of the numerical tools	70
6.2	Relaxation time regulation	71
6.3	2D rising bubble benchmarking	74
6.3.1	Test case 1	74
6.3.2	Test case 2	80
6.3.3	Effect of interface thickness	86
6.4	Refined-grid level set	87
6.5	3D rising bubble	90
6.5.1	Test case 1	91
6.5.2	Test case 2	95
6.5.3	Isotropy and the choice of the lattice stencil	98
6.6	Effects of arithmetic precision	101
6.7	Application to non-benchmark flows	106
6.7.1	Rayleigh-Taylor instability	106
6.7.2	Droplet Splashing on thin liquid film	108
6.7.3	Collision of binary droplets	111
6.7.4	Bubbles bursting at free surface	115
7	Parallel implementation and computational performance	116
7.1	GPGPU programming model	116
7.1.1	Performance Optimization	118
7.2	Efficient implementations for coupled LB-LS schemes	119
7.2.1	Workload analysis	119
7.2.2	Performance optimization for two-phase flows	120
7.2.3	Implementation outlines	121
7.3	Performance analysis	123
8	Conclusion and outlook	128
8.1	Summary and conclusions	128
8.2	Outlook of the future works	129
8.2.1	Coupling with Cahn-Hilliard equation	129
8.2.2	Parallel AMR implementations	131
8.2.3	Multiple GPGPU implementations	131
A	Appendix	132
A.1	Weighted essentially non-oscillatory (WENO) scheme	132
B	Appendix	135
B.1	Transformation matrices for MRT collision	135

Bibliography

138

Introduction

Multiphase flow simulations have always been in the centre of interest for numerical scientists as well as chemical and mechanical engineers. A swarm of vapour bubbles generated inside power-plant pumps could cause severe damages due to the impact of high speed bubbles on the expensive metallic impellers, necessitating accurate numerical predictions before putting the machine into operation. Several other examples could be found in various disciplines where the existence of non-homogeneous mixtures of fluids are vital for instance in the production of polymer materials or in tracking the concentration of chemicals or medicines as they are carried over by the blood to the target tissues in the patient's body.

Situations like those mentioned above share a number of distinctions from single phase flows; they involve two or more fluids often having significantly different density or viscosity and more notably, a very thin interface is separating these fluids which is formed to maintain equilibrium between the surface tension forces and the imbalance of the thermodynamic pressure due to the molecular interaction between fluids of different types [32, 98]. In the eyes of a numerical scientist, such aspects are usually seen as locally sharp gradients of surface characteristics and fluid properties which are transported in space and time by flow or mass diffusion. An accurate numerical description of such unique features requires having very fine grid resolutions to realise the interface gradients and in some cases employing stabilization techniques to keep the interface smooth as it convects.

Despite offering remarkable numerical accuracy due to their solid mathematical foundations, direct numerical solutions of the Navier-Stokes (NS) equations often face operational complications when applied to multiphase flows of real scales. The notorious nonlinearity of the NS equations when combined with the discontinuities of the interface region effectively impedes the simulations.

Moreover, since the flow is governed by the capillary effects, excessive limitations on the temporal resolution, i. e., more time steps, are in fact inevitable if the surface tension forces are added explicitly [25]. Considering that the overall implicit scheme of most NS solvers has to solve one or more linear system of equations in each time step, this implies that the capillary effects could acts as a formidable computational barrier. Implicit inclusion of force terms would then be considered as a solution [56], but the subsequent computational complications are yet to be questioned. Consequently, one may think of computationally less sophisticated kinetic-based methods which basically march in time in an explicit fashion with naturally slow paces, enabling them to detect rapid temporal changes without imposing an extra cost. This is where the lattice Boltzmann model (LBM), as the most evolved variant of kinetic schemes, turns into an encouraging alternative [30, 70, 137].

The application of LB models has, for a long time, been established in the simulations of single phase flows in conventional as well as complex flow regimes and geometries. The method is recognised as a fledgling approach in specific disciplines where non-regular boundaries, small

length and time scales and long simulation times make the other choices unfavourable to implement or computationally unaffordable, e. g. in flows through porous media [7, 47], shallow water flows [33, 42, 144] or slip flows in micro channels [66, 114] and blood flow in human artery [14, 16]. In addition to such numerical features, the computational work-flow of the method puts it among the most desirable candidates for massive parallel implementations, delivering remarkable parallel efficiencies on compute clusters or even a single general purpose graphical processing unit (GPGPU) which fits into an office PC [80, 124]. This later aspect has even motivated direct numerical simulations (DNS) using LBM for turbulent flows [78, 130] with the application in transportation technology [4].

By the early 1990s, attempts to extend LBM to two- and multiphase flows, however, couldn't succeed to compete with the well-established methods based on the NS equations, particularly in terms of maximum tolerable density and viscosity ratios and stability at large Re numbers. From a historical point of view, one may refer to the popular and computationally light-weight interaction potential models of Shan and Chen [112] or the colour-fluid models of Gunstensen et al. [45] and Rothman and Keller [103]. The two most critical issues with such models preventing them to address real systems were, on one hand, the lack of concrete physical understandings to incorporate interface forces into the kinetic model of the LBM, and on the other hand the large truncation errors emerging from the inconsistent numerical discretization of the surface tension forces and the pressure gradients [75].

The evolution of more thorough knowledge of the kinetics of the LB equation by the late 1990s together with the introduction of more stable numerical techniques into the method led to the development of a new class of phase-field schemes which could realise a smooth, but yet diffuse, interface between the phases at large density and velocity differences. The model developed by He and Shan [52], which was later improved by Lee and Lin [75], as well as the free energy model of Swift et al. [121] are among the most popular of such models. In general, phase-field LB methods mostly incorporate the interface forces through high order derivatives of the chemical potential parameters. In addition, the thermodynamic and numerical inconsistencies arising from the variable density are carried over to a second LBE to recover the Cahn-Hilliard equation for minimizing the chemical potential and capture the interface, thus leaving the first LBE to solve for pressure and velocity only.

Nevertheless, the computational demand of these two-LBE schemes goes excessively beyond that of the early, weakly stable Shan-Chen or colour-fluid LB models, since a second $DnQm$ LBE has to be solved over the entire domain while both LBEs require spatial discretizations of force terms in all lattice directions with further increase in the memory requirements on the part of the computing hardware. In addition, calculation of the chemical potential terms requires third and fourth order derivatives of the concentration profile, which subsequently implies having a very fine lattice in the vicinity of the interface.

On the other hand, among numerous LB-based multiphase flow studies carried out to date [9, 19, 31, 86, 87, 96, 139], one could hardly come across comprehensive and quantitative comparisons against reference solutions of the multiphase NS equations. The comparisons are rather made mostly in the so-called *picture norm* against those from the experimental studies [21, 32] or against other similar LBM simulations, leaving the users oblivious of the effects of time and lattice resolution or the role of different force discretization schemes in building up the overall accuracy or computational performance. Consequently, most commercial users are still unsure regarding the reliability of the proposed schemes for practical purposes.

The aim of this thesis is thus twofold. Firstly, the thesis proposes two schemes which contribute to circumventing the unnecessary computational and regularity restrictions associated with the conventional phase-field LB models. Inspired by the ideas of the multiphase NS community, lies in the core of both techniques a coupling of an LBM flow solver with a level set interface

capturing engine, which replaces the second LBE in the phase-field LB models. While the first scheme suggests a sharp interface coupled model based on the continuum surface force approach (CSF), the second variant employs a modified diffuse interface level set equation accompanied by a diffuse interface LBE for the flow field and features a remarkable overall mass conservation and flexibility to capture extreme interface deformations. By avoiding a second LBE for interface capturing and its associated high order derivatives, both schemes bring promising savings in the required memory and simulation time.

The other aspiration of the thesis is to carry out benchmark comparisons against reference numerical data available for multiphase flows. To this end, the widely used data of the standard rising bubble problems based on finite element solutions of the NS equations are picked where the results produced by different academic and commercial codes [56] [6] could provide a clear measure, how close a kinetic-based approach could get to the solution of the macroscopic governing equations. This further creates a valuable basis to evaluate how critical the choices of force discretization, microscopic stencil, etc. could be in producing results which must not only converge, but shall also give rise to the anticipated values. It is therefore shown for both schemes that consistent description of the interface forces together with numerically plausible spatial discretization and time integration play key roles in constructing stable solutions. The lessons learnt would not only apply to the current methods, but can also be extended to other LB-based approaches.

Eventually, and as every new LB model has to stand the test of a preserved parallel efficiency, it will be shown towards the end of the thesis, how to efficiently implement the suggested algorithms on parallel machines and in particular on modern GPGPUs, and how far one may go in exploiting peak memory bandwidth and floating point operation power of such architectures.

The thesis is organised as follows. Chapter 2 presents an overview of the macroscopic equations governing incompressible multiphase flows along with proper boundary conditions. A survey on the foundations of the LB method and the conventional multiphase LB models is presented in Chapter 3. Chapter 4 elaborates on the choice of the interface capturing technique and eventually describes the solution of the level set equation, including reinitialization, mass conservation and the calculation of surface properties. Having established the methods used for resolving the interface, chapter 5 then discusses the proposed coupled LBM-level set schemes and focuses on the numerical aspects pertained to discretization and stabilization of the solution model. The numerical results for benchmarking of rising bubble simulations against academic and commercial solutions of the NS equations in two and three dimensions are provided in chapter 6, followed by subsequent discussions on the choice of the physical properties, relaxation parameters, lattice stencils and arithmetic precision. A preliminary study on the use of a refined grid for the level set and thus its implications on the overall accuracy and computational cost is also presented in this chapter. The chapter is eventually closed with the application of the proposed schemes to non-benchmark flows of physical importance and interest, which consists of Rayleigh-Taylor instability, droplet splashing on a thin liquid film, binary droplet collisions and bubbles bursting at the free surface. Chapter 7 is concerned with the computational features of the solvers and implementation on parallel processors including CPU and GPGPU architectures. A summary of the work and the outlook for future studies is eventually presented as the closure of the thesis in chapter 8.

2

Governing equations in macro- and microscales

Although the current thesis is mainly concerned with kinetic approaches, it is crucial to construct the mathematical bases which describe the incompressible multiphase systems since the proposed schemes try to closely mitigate macroscopic hydrodynamics of multiphase systems. A mathematical description of the different fluxes and forces which lead to the Navier-Stokes equations together with the definition of proper boundary conditions is first presented in this chapter. The chapter eventually focuses on the specific boundary conditions required to define a two-phase flow as well as a preliminary discussion on the inclusion of the surface tension forces in the governing equations.

2.1. Macroscopic conservation laws

Consider a fluid particle with density ρ moving freely with velocity \mathbf{u} in space and time which contributes to the overall pattern of the flow together with billions of other particles. A question posed to a mathematician or engineer is then how to accurately quantify the fluid motion without having to track each and every particle which is interacting with the neighbouring ones. This less preferred view of tracking individual particles in the flow is called *Lagrangian* approach in which the evolution of the fluid's mass and momentum per unit volume is governed by (energy is ignored as only isothermal flows are concerned here)

$$\frac{D\rho}{Dt} = 0 \quad (2.1.1)$$

$$\rho \mathbf{a} = \sum F \quad (2.1.2)$$

where the first equation implies the conservation of mass for the density ρ and the second one is Newton's second law for momentum $\rho \mathbf{a}$, in which F is the net force experienced by the particle as a result of interacting with its neighbours. Despite the very simple form of the above equations, they present a couple of practical issues; Firstly, the term "fluid particle" is a vague term to define, as one has to exactly specify how many fluid molecules contribute to forming a particle. Secondly, individually tracking billions of particles poses such a high computational cost that will make the simulation impractical for large problems.

However, in a real fluidic system, one may assume that the particles are fairly close to each other so that the bulk of the fluid could be considered as a continuum. This so-called *Eulerian* view implies that fluid properties are continuous across the domain of interest and their evolution could be divided into the local changes and the changes induced by the velocity through an arbitrary close volume in space. As such, one could integrate over the volume Ω to measure the bulk mass and the momentum. Equation (2.1.1) then requires the temporal changes of the bulk mass to be balanced with the net flux of mass ($\rho \mathbf{u}$) along the normal direction \mathbf{n} , integrated over the surface

S of the closed volume. Similarly, equation (2.1.2) could be translated into the rate of change of the momentum which is balanced with the volume force F_{vol} integrated over the volume together with the surface flux of momentum ($\rho\mathbf{u}\mathbf{u}$), and the surface forces F_{surf} integrated over the surface S . The integrations eventually lead to a new form of the conservation equations

$$\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega = - \int_S (\rho\mathbf{u}) \cdot \mathbf{n} dS \quad (2.1.3)$$

$$\frac{\partial}{\partial t} \int_{\Omega} \rho\mathbf{u} d\Omega = - \int_S \rho\mathbf{u}\mathbf{u} \cdot \mathbf{n} dS + \int_{\Omega} F_{vol} d\Omega - \int_S F_{surf} \cdot \mathbf{n} dS \quad (2.1.4)$$

where \mathbf{n} is the normal to the surface. By using the divergence theorem to convert surface integrals to volume integrals, one arrives at

$$\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega = - \int_{\Omega} \nabla \cdot (\rho\mathbf{u}) d\Omega \quad (2.1.5)$$

$$\frac{\partial}{\partial t} \int_{\Omega} \rho\mathbf{u} d\Omega = - \int_{\Omega} \nabla \cdot (\rho\mathbf{u}\mathbf{u}) d\Omega + \int_{\Omega} F_{vol} d\Omega - \int_{\Omega} \nabla \cdot F_{surf} d\Omega. \quad (2.1.6)$$

Writing these conservation laws in the form of PDEs, simply means to shrink the volume into an infinitesimal differential element to have

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{u}) = 0 \quad (2.1.7)$$

$$\frac{\partial}{\partial t} (\rho\mathbf{u}) = -\nabla \cdot (\rho\mathbf{u}\mathbf{u}) + F_{vol} - \nabla \cdot F_{surf}. \quad (2.1.8)$$

The above methodology could be followed to obtain the transport equations associated with other fluid properties, e. g., internal energy or chemical potential. The external surface force denoted by the stress tensor F_{surf} is composed of the shear stress T which acts against the main coordinate directions in space, and the normal stress exerted by the pressure p which encompasses three major effects of hydrodynamic pressure p_d , thermodynamic pressure p_t and hydrostatic pressure p_s . For the moment, the diagonal pressure tensor $p\mathbf{I}$ is considered as a whole and its components will be dealt with in the next chapter. Nevertheless, this general classification of the force tensor leads to a modified form of the momentum equation

$$\frac{\partial}{\partial t} (\rho\mathbf{u}) = -\nabla \cdot (\rho\mathbf{u}\mathbf{u}) + F_{vol} + \nabla \cdot (p\mathbf{I} + T). \quad (2.1.9)$$

Before proceeding with the classification of more specific forms of conservation PDEs of practical use, one needs to make two more assumptions; the first one is related to the fluid properties and picks a particular case out of several possibilities where the shear stress tensor is proportional to the strain tensor through a constant viscosity μ

$$T_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (2.1.10)$$

Such an approximation could be valid to a high degree of certainty for water, air and a number of other industrial or biochemical substances, and the fluids with such a property are called *Newtonian* fluids. The second assumption has to do with the flow itself, stating that the density does not change with time and space such that the conservation of mass in equation (2.1.7) leads to a divergence free velocity field describing the continuity equation

$$\nabla \cdot \mathbf{u} = 0. \quad (2.1.11)$$

As a result of the two critical expressions in equations 2.1.10 and 2.1.11, the momentum equation (2.1.8) in an incompressible regime under Newtonian flow assumption reduces to

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + F_{vol} + \nabla \cdot (\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)). \quad (2.1.12)$$

The combination of the continuity equation (2.1.11) and equation (2.1.12) for momentum constitutes the widely-known Navier-Stokes equations (NSE). Analogous to other types of PDEs, the closure of the NSE is also attained by defining a set of ample boundary conditions. Boundary conditions are mainly of Dirichlet and/or Neumann type. These generic boundary conditions depend on the type of the simulation and may vary from one problem to another. Dirichlet boundary conditions prescribe an absolute value for a variable q on the boundary $\partial\Omega_D$ as

$$\mathbf{q} = g \quad \text{on} \quad \partial\Omega_D \quad (2.1.13)$$

where for example setting $\mathbf{q} = \mathbf{u}$ and $g = 0$ along the solid walls results in the so-called no-slip boundary condition. Alternatively, one may consider the Dirichlet condition for a certain component of velocity as in the no-flux boundary condition

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on} \quad \partial\Omega_D \quad (2.1.14)$$

which prohibits any mass and momentum flux through the boundary, along the normal direction \mathbf{n} .

Neumann conditions, however, impose a fixed change or gradient on $\partial\Omega_N$. A common example is to rule zero velocity gradient at the outlet of channels or manifolds to approximate no change in flow variables at the far downstream of the flow field

$$\mathbf{n} \cdot (-p\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) = 0 \quad \text{on} \quad \partial\Omega_N. \quad (2.1.15)$$

A mixed boundary conditions could thus be realised as in the application of the free slip boundary condition

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad , \quad \mathbf{t} \cdot (\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) = 0 \quad \partial\Omega_S \quad (2.1.16)$$

where \mathbf{t} is the unit tangent vector. A specific realisation of such a boundary could be observed in a two dimensional flow of a Newtonian fluid of $\mu = const$ along a horizontal flat surface, where $u_y = 0$ and $\partial u_x / \partial y = 0$ which implies no flux through the boundary in addition to enforce vanishing tangential stresses along the surface. While letting u_x to take a non-zero value, this combination leads to a free slip flow at the boundary.

2.2. Extension to multiphase flows

While the NS equations are developed with the aim to describe the evolution of mass and momentum in incompressible flows which also include multiphase flows, then the question which arises is why shall one need to treat these equations in a different way when dealing with two or more phases? The answer could be summarized by explaining a couple of specific aspects of a multiphase flow system.

Firstly, the two immiscible phases, denoted by Ω_1 and Ω_2 in figure 2.1, are basically of different physical properties which means that the values of ρ and μ depend on the spatial location and could not be treated as constants any more. This implies that these variables must be taken into account in the numerical operations if any further integration or differentiation is to be applied to the governing equations during the solution process. Examples are the integration in finite elements method or the force term differentiation in LB scheme as will be discussed in the next chapter.

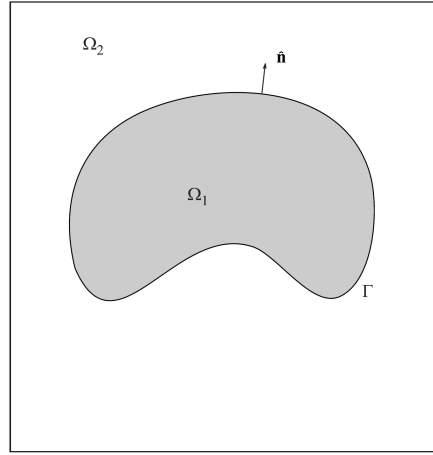


Figure 2.1: Configuration of a typical two-phase flow problem [56]

Moreover, the interface region separating the two fluids experiences a surface tension force F_s which balances the gradients of the pressure and velocity in the separation band and thus has to be considered as a new boundary condition imposed along the interface Γ

$$[-p\mathbf{I} + (\mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T))]_{|\Gamma} \cdot \mathbf{n} = F_s. \quad (2.2.1)$$

Lastly, the existence of the interface prohibits any mass flux across the phases which puts a restriction on the velocity field around the interface implying a second additional boundary condition applied in the normal direction to the interface.

$$\mathbf{u} \cdot \mathbf{n} = 0|_{\Gamma} \quad (2.2.2)$$

As a result of such distinctions, the incompressible NSE for multiphase flows, along with the new boundary conditions, could be summarized in the following form.

$$\left\{ \begin{array}{ll} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + F_{vol} + \nabla \cdot (\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)). & \text{in } \Omega \\ \nabla \cdot \mathbf{u} = 0. & \text{in } \Omega \\ \mathbf{u} = g & \text{on } \partial\Omega_D \\ \mathbf{n} \cdot (-p\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) = 0 & \text{on } \partial\Omega_N \\ [-p\mathbf{I} + (\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T))]_{|\Gamma} \cdot \mathbf{n} = F_s & \text{on } \Gamma \\ \mathbf{u} \cdot \mathbf{n}|_{\Gamma} = 0 & \text{on } \Gamma \end{array} \right. \quad (2.2.3)$$

Having the governing equations and their boundary conditions established for a general multiphase flow problem, let us now turn our focus to other implementation aspects such as treating the discontinuities in the properties and the mechanisms to introduce the surface tension forces into these equations.

2.2.1. Regularization of discontinuities

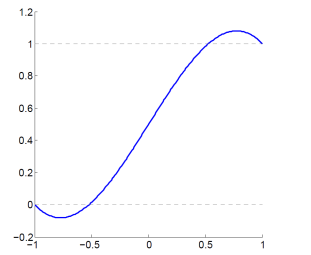
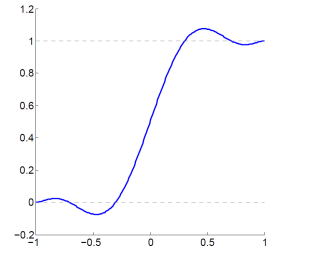
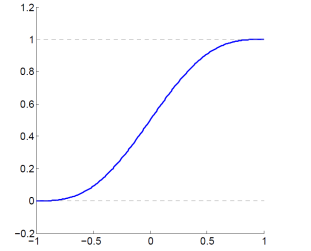
Type	Realisation	Transition functions
$H(v_1)$		$v_1(\xi) = \frac{1}{2} + \frac{1}{8}(9\xi - 5\xi^3)$
$H(v_2)$		$v_2(\xi) = \frac{1}{2} + \frac{1}{256}(525\xi - 1225\xi^3 + 1323\xi^5 - 495\xi^7)$
$H(v_3)$		$v_3(\xi) = \frac{1}{2}(1 + \xi + \frac{1}{\pi}\sin(\pi\xi))$

Table 2.1: Transition functions approximating a Heaviside function (from [56])

The existence of the interface causes the properties to change in a discontinuous way across the interface which is problematic for the numerical solutions where the space is discretized and discontinuities could not be conveniently modelled. Consequently, and as a widespread convention, the discontinuous dependence of the density and viscosity on the space could be regularized based on the signed distance $d(\Gamma, x)$ from any point in space to the interface through a Heaviside function initially proposed by Tornberg [126]

$$H(x) = \begin{cases} 1 & , \quad d \geq \varepsilon \\ v(d/\varepsilon) & , \quad |d| < \varepsilon \\ 0 & , \quad d \leq -\varepsilon \end{cases} \quad (2.2.4)$$

where v is the transition function, which together with the interface thickness ε , fully determines the regularization. Moreover, in order for the numerical method to converge to the correct hydrodynamics as $h \rightarrow 0$, the thickness has to be chosen as $\varepsilon = \mathcal{O}(h)$ and thus be proportional to the resolution. It has been discussed in [126] that since the analytical error of replacing the discontinuous Heaviside function with the regularized form increases with higher powers of ε , then the use of transition functions with high number of vanishing moments tends to preserve the original accuracy of the numerical scheme. Table 2.1, contains some examples of the transition functions [56]. When using the polynomial functions, one must be careful as the resulting values tend to produce slight over/undershoots at the interface edges, which in turn may result in negative values of den-

sity and viscosity that could potentially deteriorate the numerical simulation. In contrast, the *sine* function offers a smooth regularization and is thus widely preferred in practical implementations of numerical multiphase flow solutions.

Eventually, regardless of the choice of the transition function, any fluid property θ could be obtained continuously based on the Heaviside function as

$$\theta^s = \theta_1 + (\theta_2 - \theta_1)H \quad (2.2.5)$$

where subscripts 1 and 2 denote the properties pertained to the phase 1 and phase 2. By reconstructing smooth variations θ^s for originally discontinuous properties, numerical discretization and integration methods can be directly applied to the entire NSE.

2.2.2. Inclusion of the surface tension forces

The principal origin of the surface tension forces could be identified in two different ways. The first one is based on the fact that the surface tension forces tend to minimize the surface area of the interface, hence leading to the following geometrical description of the surface tension force

$$F_s = \sigma \kappa \mathbf{n} \quad (2.2.6)$$

where κ and \mathbf{n} are the local curvature and normal vector to the interface and σ is the surface tension coefficient which could be considered as a constant under the assumption of having a non-thermal flow with constant composition of the chemical impurities at the interface. The representation of equation (2.2.6) requires one to obtain the surface normal vector and curvature through differentiating either the concentration field or a smooth implicit function, depending on the choice of the interface capturing technique. This subject is left at the moment to be discussed extensively in chapter 4.

Whichever approach is pursued to track the interface and recover the surface properties, one must figure out a way to realise a boundary condition associated with such forces. An immediate choice would be to treat the interface as a sharp boundary and then apply the Dirichlet condition explicitly on the boundary as in equation (2.2.1). The approach is in fact practically difficult to implement as the interface does not necessarily coincide with the grid points and one also needs to have the exact location of the interface at every time stamp so as to deploy the boundary conditions there. Alternatively, the forces could be included implicitly via an immersed boundary acting as a volume force which is concentrated over a prescribed interface thickness ε and regularized by delta function $\delta(\Gamma, x)$ at the interface [113],

$$F_s = \sigma \kappa \mathbf{n} \delta_\varepsilon(\Gamma, x). \quad (2.2.7)$$

The technique is also known as the Continuum Surface Force (CSF) approach first used in the early works of Peskin et al. [95], [25], since it smears the force to some extent and eliminates the need to reconstruct the interface and to solve for two fluids separately. Inserting this new force term into equation (2.1.12) gives the CSF form of the multiphase NSE as

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \nabla \cdot (\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \sigma \kappa \mathbf{n} \delta_\varepsilon(\Gamma, x) + F_{vol}. \quad (2.2.8)$$

The CSF approach is widely preferred in both sharp and diffuse interface methods with various regularizations which must suit the flow solver and be compatible with the selected interface capturing scheme [57, 65, 88, 118]. A regularized Delta function then has to be defined such that its maximum is reached in the middle of the interface where the distance to the interface d is zero and vanishes at its edges, i. e., where $|d| = \varepsilon$, which quantitatively means

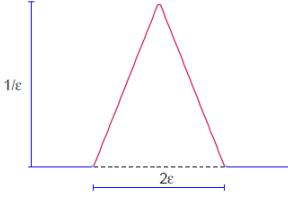
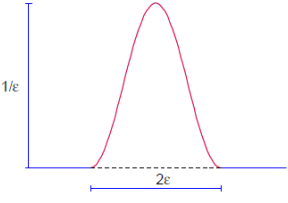
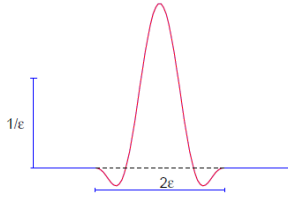
Type	Realisation	Kernel functions
$\delta_\varepsilon(\phi_1)$		$\phi_1(\xi) = 1 - \xi $
$\delta_\varepsilon(\phi_2)$		$\phi_2(\xi) = \frac{1}{2}(1 + \cos(\frac{\pi\xi}{2}))$
$\delta_\varepsilon(\phi_3)$		$\phi_3(\xi) = \frac{312}{512}(3 - 20\xi^2 + 42\xi^4 - 36\xi^6 + 11\xi^8)$

Table 2.2: Different realisations of the regularized Delta function over a thickness of 2ε (from [56])

$$\delta_\varepsilon(x) = \begin{cases} \frac{1}{\varepsilon}\phi(x/\varepsilon) & , \quad |x| \leq \varepsilon \\ 0 & , \quad |x| > \varepsilon \end{cases} \quad (2.2.9)$$

where ϕ is a characteristic function determining the kernel shape. Table 2.2 lists some examples of the regularized delta functions used in different approaches, including linear, cosine and high order polynomial forms. An extensive discussion on the use of various regularization functions and the associated errors could be found in the works of Tornberg et al. [35, 126].

In contrast to the above geometrical view, the dominated perspective in the chemical and material engineering literature is to relate the surface tension forces to the chemical potential μ_c

$$F_s = -C\nabla\mu_c \quad (2.2.10)$$

where C is the concentration of the phases defined as

$$C = \begin{cases} 1 & \text{if } \rho(x) = \rho_1 \\ 0 & \text{if } \rho(x) = \rho_2 \end{cases} \quad (2.2.11)$$

and ρ_1, ρ_2 are the densities of phase 1 and 2, respectively. At the interface, the concentration field is formed in a way to minimize the chemical potential based on equations, e. g., the Cahn-Hilliard (CH) equation for diffuse interfaces. This requires μ_c to depend on the thermodynamics of the species via a high order polynomial of C and $\nabla^2 C$ (see chapter 3 for more on this correlation). Equation (2.2.11) may seem to be autonomous from the surface properties and offer a minimal computational effort by differentiating a scalar field. Nevertheless, one must take into account

the exhaustive effort to numerically control the nonlinearity involved in the solution of the CH equation for capturing the concentration field, as well as the non-smooth variations of μ_c which degrades the subsequent numerical evaluation of $\nabla\mu_c$. Interested readers are referred to the works of Anderson et al. [11], Gurtin et al. [50] and Boyer [24] for useful discussions on physics and numerics of diffuse interface models including coupled NS-CH methods.

The Lattice Boltzmann Method

In this chapter the essentials of the LB method are first presented, including a brief overview starting from the basic lattice gas methods towards the evolution of discrete Boltzmann equation and eventually the LB equation itself. Then the relevance of the resulting kinetic equations with the macroscopic equations in chapter 2 are uncovered which pave the way for explaining the conventional multiphase LB schemes. Some key elements of these models are of historical and physical importance and will be used as foundations of the proposed multiphase schemes discussed in chapter 5.

3.1. The Lattice Boltzmann Method

The lattice Boltzmann method has its roots deep on one hand in the development of Lattice Gas (LG) methods during 1980s, and on the other hand in kinetic theory and statistical mechanics of gases [46, 116, 133]. The LG methods were first proposed in the milestone 1986 paper by Frisch, Hasslacher and Pomeau [39] to be used for the solution of the Navier-Stokes equations where they showed the evolution of the mass and momentum of dilute gases could be modelled by setting a group of particle distributions on discrete lattice nodes and then letting them collide locally and stream to the neighbouring nodes in discrete spatial directions. Figure 3.1 shows a hexagonal (hence six-velocity) stencil in LG Cellular Automata (LGCA) model of Frisch, Hasslacher and Pomeau (the so-called FHP model) with the corresponding collision rules.

In the FHP model, the mass and momentum are recovered by taking the moments of the distributions as:

$$\rho \mathbf{u} = \sum_{\alpha=1}^6 C_{\alpha} N_{\alpha} \quad , \quad \rho = \sum_{\alpha=1}^6 N_{\alpha} \quad (3.1.1)$$

where $c_{\alpha}, \alpha = 1, \dots, 6$ are the discrete particle velocities and N_{α} is the particle distribution. Despite the locality and linearity of LGCA, the model resembled several shortcomings with regards to isotropy, Galilean invariance, and most importantly, recovering the viscosity which would eventually lead to widespread noises as the collision of the particles is governed by ad-hock non-deterministic rules rather than physically authentic ones [133]. Consequently, the model could not be adopted to mitigate the real physics of the flow.

A profound physical support to the LGCA methods was then brought about by the application of the LG ideas to solve the historically renowned Boltzmann equation in thermodynamics and kinetic theory of gases as first proposed by McNamara and Zanetti in 1998 [84]. Before proceeding with the outcome of this combination, it is vital to shortly explain the structure of the Boltzmann equation itself.

In essence, the Boltzmann equation describes the microscopic evolution of particle distribu-

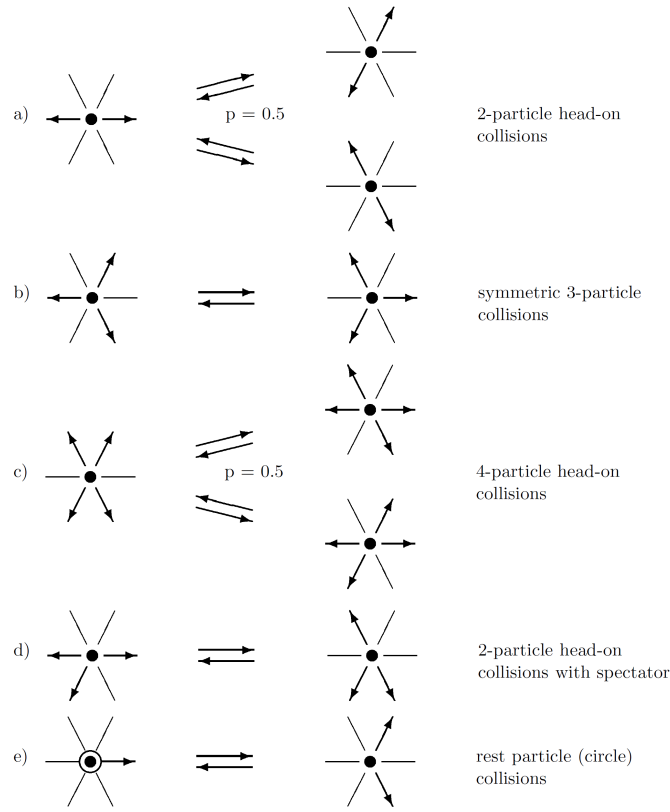


Figure 3.1: The collision rules in the hexagonal lattice gas model of FHP. Note that p denotes the probability (from [133]).

tions in space \mathbf{x} and time t and along a certain velocity ξ , thus defined as $f(\mathbf{x}, t, \xi)$. The inclusion of velocity in defining f_α creates the so-called velocity phase space rather than the usual time-coordinate space. The original Boltzmann equation in continuous form without the forcing term reads as

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \xi \cdot \nabla f = Q(f) \quad (3.1.2)$$

where ξ is the constant microscopic velocity. As seen in equation (3.1.2), the transport of the distribution functions is balanced by the collision of the particles through the integral Q which conserves mass, momentum and energy [46]. Moving on with Boltzmann's assumptions, mass and macroscopic momentum are then obtained by integrating the distribution functions over all possible velocity directions as

$$\rho(\mathbf{x}, t) = \int_{\xi} f(\mathbf{x}, t, \xi) d\xi \quad , \quad \rho \mathbf{u}(\mathbf{x}, t) = \int_{\xi} \xi f(\mathbf{x}, t, \xi) d\xi. \quad (3.1.3)$$

In order to avoid integrating the distribution functions over all possible velocities, a discrete Boltzmann equation (DBE) was proposed by confining the possible velocities to a group of discrete velocities c_α , analogous to the LG methods. This simplification eventually turns equation (3.1.2) into

$$\frac{\partial f_\alpha}{\partial t} + c_\alpha \nabla f_\alpha = Q(f_\alpha) \quad (3.1.4)$$

and replaces the integral terms in equation (3.1.3) with sums as

$$\rho(x, t) = \sum_{\alpha=0}^n f_{\alpha}(x, t) \quad , \quad \rho \mathbf{u}(x, t) = \sum_{\alpha=0}^n c_{\alpha} f_{\alpha}(x, t). \quad (3.1.5)$$

The calculation of the highly complex collision integral is hard to be sought analytically, and therefore an approximation to Ω_{α} was suggested by Koelman [67] and Qian et al. [99] through a relaxation of the distributions towards a local equilibrium state $f_{\alpha}^{eq}(\rho, \mathbf{u})$

$$\frac{\partial f_{\alpha}}{\partial t} + c_{\alpha} \nabla f_{\alpha} = -\frac{1}{\lambda} (f_{\alpha} - f_{\alpha}^{eq}(\rho, \mathbf{u})) \quad (3.1.6)$$

where λ denotes the relaxation time. This collision model is often regarded as the Bhatnagar-Gross-Krook (BGK) model as such a collision rule was first proposed in 1954 by Bhatnagar, Gross and Krook [23] for the continuous Boltzmann equation. The equilibrium state could be identified based on analysing the original collision integral so as to maximize the total entropy of the system and gives rise to the following exponential Maxwell-Boltzmann equilibria

$$f_{\alpha}^{eq}(\rho, \mathbf{u}) \equiv \frac{\rho}{(2\pi c_s^2)^{D/2}} \exp\left[-\frac{(\xi - \mathbf{u})^2}{2c_s^2}\right] \quad (3.1.7)$$

where $c_s = \sqrt{RT}$ is the speed of sound, D is the dimension of the space and T and R are the temperature and the gas constant, respectively. Note that $R = K_B/m$ where K_B is the Boltzmann constant and m is the molar mass of the fluid of interest. The discretization of the velocity space leads to the following polynomial form of the equilibrium

$$f_{\alpha}^{eq}(\rho, \mathbf{u}) = w_{\alpha} \rho \left[1 + \frac{c_{\alpha} \cdot \mathbf{u}}{c_s^2} + \frac{[c_{\alpha} \cdot \mathbf{u}]^2}{2c_s^4} + \frac{[\mathbf{u} \cdot \mathbf{u}]}{2c_s^2} \right]. \quad (3.1.8)$$

where the weight factors w_{α} are in fact quadrature constants obtained so as to satisfy the momentum equation as we travel from the continuous BE to its discrete counterpart [46]

$$\int \xi f_{\alpha}^{eq}(x, t, \xi) d\xi = \sum_{\alpha} w_{\alpha} c_{\alpha} f_{\alpha}^{eq}(x, t, c_{\alpha}). \quad (3.1.9)$$

It could be shown [67, 99, 133] that by applying the Chapman-Enskog expansion for f_{α} around f_{α}^{eq} up to the second second order, as well as moments of f_{α} up to tensors of rank 4 in c_{α} , assuming a nearly incompressible limit where $Ma \ll 1$ to cancel the $\mathcal{O}(\mathbf{u}^2)$ terms, and inserting the expansion into the DBE, one can recover the macroscopic mass and momentum¹ equations as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (3.1.10)$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho c_s^2 \mathbf{I} + \rho \mathbf{u} \mathbf{u}) - \frac{\lambda}{3} (\nabla \cdot [\nabla (\rho \mathbf{u}) + \nabla (\rho \mathbf{u})^T]) = 0. \quad (3.1.11)$$

Comparing equations (3.1.10) (3.1.11) to its original form in the NSE (2.1.12), reveals two exquisite conclusions to recover the pressure and viscosity as

$$p = \rho c_s^2 \quad , \quad \nu = \lambda/3. \quad (3.1.12)$$

where the equation for pressure resembles the ideal gas equation of state (EOS) in thermodynamics and is often called the single phase EOS of the LBM.

The dynamics of equation (3.1.6) exhibits a major distinction from the macroscopic Eulerian or Lagrangian views described in section 2.1. In fact, the non-local advection term in equation

¹Without the forcing term

(3.1.6) is linear since c_α is a constant and instead, the non-linear collision term is local. This offers a great advantage from the numerical point of view as it relinquishes the stabilization concerns while convecting ∇f_α , and allows the LG methodology to directly take hold on the DBE. This was in fact achieved by Sterling and Chen [115] through non-dimensionalizing equation (3.1.6) and applying implicit and trapezoidal time integration to the convective and collision terms of the DBE, respectively

$$\begin{aligned} & \frac{f_\alpha(x, t + \Delta t) - f_\alpha(x, t)}{\Delta t} + c_\alpha \frac{f_\alpha(x + \Delta x, t + \Delta t) - f_\alpha(x, t + \Delta t)}{\Delta x} \\ &= -\frac{1}{2\lambda} (f_\alpha(x, t + \Delta t) - f_\alpha^{eq}(x, t + \Delta t)) - \frac{1}{2\lambda} (f_\alpha(x, t) - f_\alpha^{eq}(x, t)). \end{aligned} \quad (3.1.13)$$

To render the scheme into an explicit form, a new distribution function is defined as

$$\bar{f}_\alpha = f_\alpha + \frac{1}{2\lambda} (f_\alpha - f_\alpha^{eq}). \quad (3.1.14)$$

Then, by choosing a lattice of $\Delta x = 1$ and a time resolution of $\Delta t = 1$, and setting the lattice velocity to be $c_i = \Delta x / \Delta t$, the above discretization reduces to a

$$\bar{f}_\alpha(x + \Delta x_\alpha, t + \Delta t) - \bar{f}_\alpha(x, t) = -\frac{1}{\tau} (\bar{f}_\alpha(x, t) - f_\alpha^{eq}(x, t)) \quad \text{with} \quad \tau = \lambda + 0.5. \quad (3.1.15)$$

Inspired by the LG framework, the obtained *perfect shift* equation could be readily split into two steps; a local collision to relax towards equilibrium and to arrive at a post-equilibrium state \bar{f}_α^*

$$\bar{f}_\alpha^*(x, t) = \bar{f}_\alpha(x, t) - \frac{1}{\tau} (\bar{f}_\alpha(x, t) - f_\alpha^{eq}(x, t)), \quad (3.1.16)$$

followed by a streaming step to propagate \bar{f}_α^* along c_α .

$$\bar{f}_\alpha(x + \Delta x_\alpha, t + \Delta t) = \bar{f}_\alpha^*(x, t). \quad (3.1.17)$$

Analogous to equation (3.1.5), the hydrodynamic properties now read as

$$\rho(x, t) = \sum_\alpha \bar{f}_\alpha(x, t) \quad , \quad \rho \mathbf{u}(x, t) = \sum_\alpha c_\alpha \bar{f}_\alpha(x, t). \quad (3.1.18)$$

Finally, as a result of the nondimensionalization process, the sound speed is simplified to $c_s = 1/\sqrt{3}$.

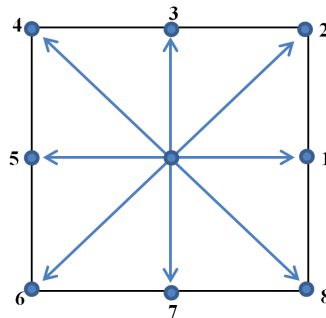


Figure 3.2: The 2 dimensional, 9 velocity D2Q9 lattice stencil

At this point, the set of discrete velocities (the lattice *stencil*) along with their corresponding weight factors have to be arranged in a way to preserve the isotropy of the underlying hydrodynamics. This feature is in fact reflected in the derivation of a fourth-order tensor which appears

during the Chapman-Enskog process in the convective term of the recovered NSE and needs to be isotropic so as to guarantee the isotropy of the resulting macroscopic equations [133]. In fact, any chosen lattice stencil must acquire a minimal symmetry and possess velocities and weight factors which guarantee the isotropy criteria. A number of 2D and 3D lattice stencils are then qualified and proposed for practical use in LBM.

An example of a prominent stencil which preserves the symmetry of hydrodynamics in 2D is the famous $D2Q9$ stencil depicted in figure 3.2. which comes with the following set of weight factors w_α and microscopic velocities c_α

$$w_\alpha = \begin{cases} 4/9, & \alpha = 0 \\ 1/9, & \alpha = 1, 3, 5, 7 \\ 1/36, & \alpha = 2, 4, 6, 8 \end{cases} \quad (3.1.19)$$

$$c_\alpha = \begin{cases} (0, 0), & \alpha = 0 \\ (\cos\theta_\alpha, \sin\theta_\alpha)c, & \alpha = 1, 3, 5, 7 \\ (\cos\theta_\alpha, \sin\theta_\alpha)\sqrt{2}c, & \alpha = 2, 4, 6, 8 \end{cases} \quad (3.1.20)$$

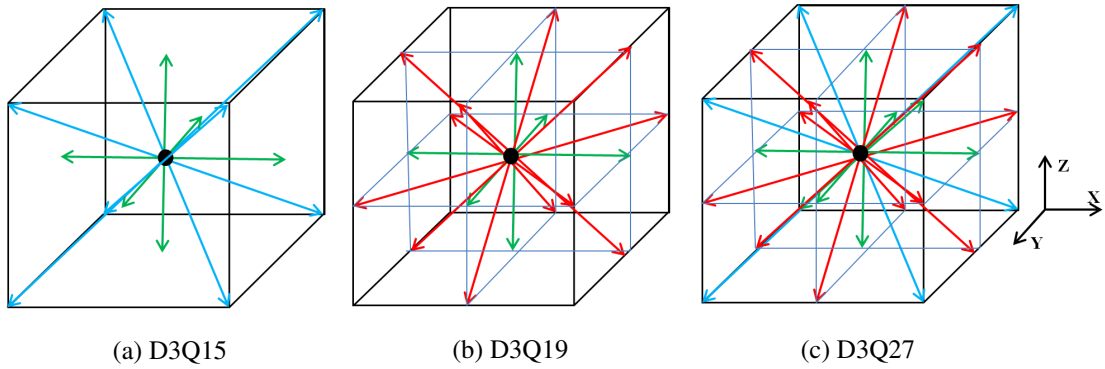


Figure 3.3: Different lattice stencils for three dimensional LBM

In 3D, however, different discrete velocity models have been proposed and used in the literature ranging from the minimal $D3Q13$ stencil, which imposes minimum memory and computation requirements [124], to the $D3Q15$, $D3Q19$ and the highly dense $D3Q27$ stencil. However, only the last three stencils are of practical use in multiphase flows as a high degree of symmetry is crucial in the evaluation of interface forces. The velocity models for these three stencils are illustrated in figure 3.3, while the corresponding velocities and weight factors are summarized in table 3.1.

3.1.1. Treating force terms in LBM

Early approaches to include forcing terms in LBM were those based on the LG schemes where, analogous to the distribution functions, the force vector \mathbf{F} is dispersed among the lattice directions and the contribution to each direction is obtained by applying the weight factors to the net force as

$$F_\alpha = \frac{w_\alpha}{c_s^2} c_\alpha \cdot \mathbf{F}. \quad (3.1.21)$$

Although the method tries to enforce a macroscopic force to the microscopic lattice in a non-physical way, yet as an advantage, it preserves the way the velocity and equilibrium distributions are obtained since it is directly added to equation (3.1.15) without the need to re-discretize the

Stencil	Weight factors	Velocities	Directions
D3Q15	2/9	(0, 0, 0)	$\alpha = 0$
	1/9	($\pm 1, 0, 0$), (0, $\pm 1, 0$), (0, 0, ± 1)	$\alpha = 1, \dots, 6$
	1/72	($\pm 1, \pm 1, \pm 1$)	$\alpha = 8, \dots, 14$
D3Q19	1/3	(0, 0, 0)	$\alpha = 0$
	1/18	($\pm 1, 0, 0$), (0, $\pm 1, 0$), (0, 0, ± 1)	$\alpha = 1, \dots, 6$
	1/36	($\pm 1, \pm 1, 0$), ($\pm 1, 0, \pm 1$), (0, $\pm 1, \pm 1$)	$\alpha = 7, \dots, 18$
D3Q27	8/27	(0, 0, 0)	$\alpha = 0$
	2/27	($\pm 1, 0, 0$), (0, $\pm 1, 0$), (0, 0, ± 1)	$\alpha = 1, \dots, 6$
	1/54	($\pm 1, \pm 1, 0$), ($\pm 1, 0, \pm 1$), (0, $\pm 1, \pm 1$)	$\alpha = 7, \dots, 18$
	1/216	($\pm 1, \pm 1, 0$)	$\alpha = 19, \dots, 26$

Table 3.1: Values of weight factors and microscopic velocities for three dimensional lattice stencils

continuous Boltzmann equation. As a result, the LBE with the forcing term reads as

$$\bar{f}_\alpha(x + \Delta x_\alpha, t + \Delta t) - f_\alpha(x, t) = -\frac{1}{\tau}(\bar{f}_\alpha(x, t) - \bar{f}_\alpha^{eq}(x, t)) + \frac{w_\alpha}{c_s^2} c_\alpha \cdot \mathbf{F} \quad (3.1.22)$$

where the mass and momentum are conserved since

$$\sum_\alpha F_i = 0 \quad , \quad \sum_\alpha c_\alpha F_\alpha = \mathbf{F}. \quad (3.1.23)$$

The method could be successfully applied to bulk forces, i. e., pressure or gravity driven flows. However, large spurious velocities tend to grow in flows involving large local gradients provoked by force terms, e. g., those of surface tension and pressure gradient in multiphase flows. Guo et al. [46] carried an analysis on the residual terms which appear in the eventual hydrodynamic equations as a result of using different forcing schemes, where the LG-based method was revealed to generate low order residuals in both macroscopic mass and momentum equations which explain the large errors in practice.

Alternatively, He et. al [52] started from the original continuous Boltzmann equation with the force term

$$\frac{\partial f}{\partial t} + \xi \cdot \nabla f + \mathbf{F} \cdot \nabla_\xi f(x, t, \xi) = Q(f), \quad (3.1.24)$$

and assumed f to inherit its principal contribution from f^{eq} .

$$F = \mathbf{F} \cdot \nabla_\xi f(x, t, \xi) \approx \mathbf{F} \cdot \nabla_\xi f^{eq}(x, t, \xi). \quad (3.1.25)$$

Considering equation (3.1.7) for f^{eq} , the above expression could be further expanded as

$$\mathbf{F} \cdot \nabla_\xi f^{eq}(x, t, \xi) = -\frac{(\xi - \mathbf{u}) \cdot \mathbf{F}}{c_s^2} f^{eq}(x, t, \xi) \quad (3.1.26)$$

which, under the discretized velocity space description, gives

$$F_\alpha = -\frac{(c_\alpha - \mathbf{u}) \cdot \mathbf{F}}{c_s^2} f_\alpha^{eq}(x, t) \quad (3.1.27)$$

and is added to the right-hand side of the LB 1 (3.1.2). Since the force term is of the same order as the distribution functions, the same time integration rule as for the collision in equation (3.1.13) must be applied to the forcing term, leading to an extended LBE for the modified distribution g_α

$$\bar{f}_\alpha(x + \Delta x_\alpha, t + \Delta t) - f_\alpha(x, t) = -\frac{1}{\tau}(\bar{f}_\alpha(x, t) - \bar{f}_\alpha^{eq}(x, t)) - (1 - \frac{1}{2}\tau) \frac{(c_\alpha - \mathbf{u}) \cdot \mathbf{F}}{c_s^2} f^{eq}(x, t) \quad (3.1.28)$$

where \bar{f}_α now encompasses the force term as well

$$\bar{f}_\alpha = f_\alpha + \frac{1}{2\lambda}(f_\alpha - f_\alpha^{eq}) + \frac{(c_\alpha - \mathbf{u}) \cdot \mathbf{F}}{2c_s^2} f^{eq}(x, t, c_\alpha). \quad (3.1.29)$$

In order to remove the dependence of the force term on the relaxation time one may also define a modified equilibrium \bar{f}_α^{eq} , which includes the forcing term

$$\bar{f}_\alpha^{eq} = f_\alpha^{eq} + \frac{(c_\alpha - \mathbf{u}) \cdot \mathbf{F}}{2c_s^2} f^{eq}(x, t, c_\alpha) \quad (3.1.30)$$

and thus turns equation (3.1.28) into

$$\bar{f}_\alpha(x + \Delta x_\alpha, t + \Delta t) - \bar{f}_\alpha(x, t) = -\frac{1}{\tau}(\bar{f}_\alpha(x, t) - \bar{f}_\alpha^{eq}(x, t)) - \frac{(c_\alpha - \mathbf{u}) \cdot \mathbf{F}}{c_s^2} f^{eq}(x, t). \quad (3.1.31)$$

It will be shown later in chapter 5 that this later form of LBE saves a considerable amount of computation time in multiple relaxation time (MRT) collision schemes since a matrix multiplication operation to apply upon the force term is hence avoided. In addition, the Chapman-Enskog expansion of the modified equation recovers the hydrodynamic equations with no low order residual.

Other methods to account for the force terms include the moment-expansion scheme by Luo [83] and the method of Guo et. al [49] which is based on a modified moment-expansion scheme. Such schemes are based on the expansion of the original force term in equation (3.1.27) via its moments up to the second order of ξ . Although the resulting force terms do recover the correct hydrodynamics, yet the expression for the force becomes more complex and leaves no significant computational merit over the method of He et. al. A nice survey and analysis of different schemes including those not discussed here is presented in the book by Guo and Shu [46].

3.2. Multiphase LB models

By the improvements in the hydrodynamics of the kinetic schemes towards the standard LBM, efforts has been made to incorporate phase separation and multiphase flow solutions into the LBM framework. As a result, several LB-based models have been proposed which sometimes realise the phase segregation phenomena from totally different points of view, leading to more than 10 distinct approaches. A survey over all these models goes beyond the scope of this thesis and the interested reader could refer to the reviews by Guo and Shu [46] and Prosperetti [98] as well as the book by Huang, Sukop and Lu [54]. Nevertheless, a few of these schemes have so far made it to commercial and academic codes including the early model of Gunstensen et al. [45], the model of Shan and Chen [112], the free energy model of Swift et al. [121], and the pressure evolution model of Lee and Lin [75]. Among these, the model of Shan and Chen is historically renowned as it has inspired the development of the later two schemes, and is thus shortly reviewed in the following pages. In addition, the fundamentals of the pressure evolution model will be discussed as the last topic of this chapter since it provides the ingredients for the proposed pressure evolution coupled scheme in chapter 5.

3.2.1. Early multiphase LB models

The early models include the colour fluid model of Gunstensen et al. [45] and the interaction potential model of Shan and Chen [112]. The colour fluid model is mostly based on modifying the collision term so as to include the collisions between fluid particles of different colours (namely blue and red). The phase separation is then modelled via performing collisions and perturbation steps, then recolouring the particles which includes solving a maximum value problem, and finally the streaming of the two sets of distribution functions pertained to the two colours. Early implementations reported to be unstable and computationally inefficient [8]. Although later efforts by Tölke et al [125] improved the model upon these issue, yet the approach did not capture a lasting interest as the dynamics of phase separation goes far from the macroscopic understandings of the phenomena.

On the contrary, the model of Shan and Chen leaves the overall collision step unchanged, and instead adds more of the thermodynamics since it sees the interface as a region where the non-monotonic thermodynamic pressure jumps to reach that of a different density. In this sense, a new internal force between the lattice points is introduced to the conventional LB equation, which is activated wherever there is a possibility that phases could be in equilibrium with each other at critical or sub-critical temperatures. This is also in accordance to the thermodynamics of fluids, where different phases start to form and separate as the temperature of a single or multicomponent mixture falls below a certain critical temperature. According to the model of Shan and Chen [112], the value of the inter-particle force in a single component fluid is obtained following the nearest neighbour interaction which is an approximation of the Lenard-Jones potential

$$\mathbf{F}(x, t) = -G\psi(x) \sum_{\alpha} w_{\alpha} \psi(x + c_{\alpha}, t) c_{\alpha} \quad (3.2.1)$$

where G is the interaction strength which acts like the effective temperature of the fluid and determines the surface tension, and ψ is the interaction potential which depends on the the equation of state of the fluid and mainly its density. In the original paper by Shan and Chen [112] this potential has the following form

$$\psi = 1 - \exp(-\rho). \quad (3.2.2)$$

Eventually, with a proper choice of G and initial values of the phase densities, the two phases start to segregate and reach equilibrium.

The appearance of the additional force term in the derivation of velocity is the main source to generate spurious velocities at the interface and therefore has attracted much interest among the researchers to derive the force in a way so as to minimize such side effects [77, 140, 141]. A more detailed understanding of the interaction model is gained via Taylor expansion of the equation (3.2.1)

$$\mathbf{F} = -\nabla \left(\frac{1}{2} c_s^2 G \psi^2 \right) - \frac{1}{2} c_s^2 G \psi \nabla \nabla^2 \psi + o(\Delta t^3) \quad (3.2.3)$$

and considering the fact that at the equilibrium, one must have

$$\nabla p = \nabla(\rho RT) - \mathbf{F}. \quad (3.2.4)$$

Inserting \mathbf{F} , integrating and further simplifying eventually gives the following non-ideal EOS

$$p = \rho RT + \frac{GRT}{2} \psi(\rho)^2 \quad (3.2.5)$$

Now, in order to be consistent with the thermodynamics, from the equation of state (3.2.5), we have

$$\psi = \sqrt{\frac{2(p - \rho RT)}{GRT}} \quad (3.2.6)$$

which can not simultaneously satisfy equation (3.2.2). This is the so-called thermodynamics inconsistency of the interaction potential model and is mainly due to the assumption that a molecule only interacts with its nearest neighbours as reflected in equation (3.2.1). This assumption comes from the Ising model which loses its validity in dense fluids, where unlike the present model, the interaction requires both short range repulsion and long range attraction to comply with the Lenard Jones potential [98]. As a result of such an inconsistency, the truncation errors which eventually appear in the hydrodynamics grow so large at the interface that the model suffers from non-physical diffusions and high spurious velocities as mentioned earlier. Moreover, as one increases the magnitude of the interaction strength G , the density ratio is increased which in turn magnifies the truncations error and the spurious velocities. Such large erroneous velocities cause the simulation to become unstable and thus the maximum permissible density ratio is often limited to 15-20. This is a serious disadvantage of the original Shan-Chen model, since in reality most of the liquid-gas systems involve density ratios larger than 100. For more elaborate discussions the reader may refer to the papers by Shan [110, 111].

In order to circumvent these unfavourable effects, several modifications have been proposed to control the spurious velocity and/or increase the maximum density ratio. This includes attempts to involve a second or more layers of neighbouring lattices in the evaluation of the interaction force, known as high order force calculation [106]. In addition, some researchers have proposed borrowing more realistic EOS from the thermodynamics which are based on both temperature and density of the fluid to reach higher density ratios [141].

3.2.2. Pressure evolution multiphase LBM

A major improvement upon the interaction potential model of Shan and Chen was brought about by a class of diffuse interface models first introduced by He et. al [53] and further improved by Lee and Lin in their outstanding paper of 2005 [75]. These methods are distinguished from the earlier LB models in two aspects: The first major distinction is the decoupling of the LBE from the variable density such that the variations in density do not enter the advective terms and the LBE is assigned to recover the macroscopic velocity and dynamic pressure directly from the distribution functions, such that no EOS is involved in recovering the pressure. The density is instead recovered by a second LBE which solves for the concentration profile and acts similar to the level set or volume of fluid methods for capturing the interface. This second LBE was later on replaced by an LBE form of the Cahn-Hilliard (CH) equation in [75] and [76].

Another feature of the scheme is the definition of the interface forces as a combination of the non-ideal part of the pressure and surface tension forces which are expected to balance each other at the interface. Unlike the colour-fluid and the interaction potential models, this form of the force means that the surface tension forces could now be adjusted explicitly. In essence, the scheme starts by defining the new pressure evolution distribution function g_α as

$$g_\alpha = f_\alpha c_s^2 + (p - \rho c_s^2) w_\alpha \quad (3.2.7)$$

which accordingly turns the equilibrium distribution into a new form as well

$$g_\alpha^{eq}(\rho, \mathbf{u}, p) = f_\alpha^{eq} c_s^2 + (p - \rho c_s^2) w_\alpha = w_\alpha \rho \left[p + \rho c_s^2 \left(\frac{c_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{[c_\alpha \cdot \mathbf{u}]^2}{2c_s^4} + \frac{[\mathbf{u} \cdot \mathbf{u}]}{2c_s^2} \right) \right]. \quad (3.2.8)$$

Such a formulation will be shown to bring a great advantage as the spurious velocities are controlled, provided a plausible space discretization is employed. In addition, the extended force term is introduced as

$$\mathbf{F} = \nabla(\rho c_s^2 - p) - \mathbf{F}_s + \mathbf{G} \quad (3.2.9)$$

where the term \mathbf{F}_s is a diffuse form of the surface tension forces and \mathbf{G} is a volume force, e. g., gravity. In order to rewrite the LBE for the evolution of g_α , one must take the total derivative of g_k as

$$\frac{Dg_\alpha}{Dt} = c_s^2 \frac{Df_\alpha}{Dt} + w_\alpha \frac{D\varphi(\rho)}{Dt} \quad (3.2.10)$$

where $\varphi = (p - \rho c_s^2)$ is the non-ideal part of the pressure. The transport of f is taken from the original LBE in equation (3.1.6). By defining $\tilde{\varphi}_\alpha = w_\alpha \varphi$, the transport of $\tilde{\varphi}_\alpha$ reads as

$$\frac{D\tilde{\varphi}_\alpha}{Dt} = \frac{\partial \tilde{\varphi}_\alpha}{\partial t} + c_\alpha \cdot \nabla \tilde{\varphi}_\alpha \quad (3.2.11)$$

The term $\partial \tilde{\varphi}_\alpha / \partial t$ is obtained by assuming that the material derivative of φ is zero since it is a function of density and the flow is assumed incompressible, i. e.

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = 0 \quad (3.2.12)$$

and hence multiplying by w_α gives

$$\frac{\partial \tilde{\varphi}_\alpha}{\partial t} + \mathbf{u} \cdot \nabla \tilde{\varphi}_\alpha = 0 \quad (3.2.13)$$

which implies $\partial \tilde{\varphi}_\alpha / \partial t = -\mathbf{u} \cdot \nabla \tilde{\varphi}_\alpha$ and eventually simplifies equation (3.2.10) to

$$\frac{Dg_\alpha}{Dt} = c_s^2 \frac{Df_\alpha}{Dt} + (c_\alpha - \mathbf{u}) \cdot \nabla \tilde{\varphi}_\alpha \quad (3.2.14)$$

or in terms of φ to

$$\frac{Dg_\alpha}{Dt} = c_s^2 \frac{Df_\alpha}{Dt} + w_\alpha (c_\alpha - \mathbf{u}) \cdot \nabla \varphi. \quad (3.2.15)$$

Now based on equation (3.1.6), one can expand Df_α / Dt including the force as instructed by equation (3.1.27) to arrive at the following pressure evolution LBE for g_α

$$\frac{\partial g_\alpha}{\partial t} + c_\alpha \cdot \nabla g_\alpha = \frac{1}{\lambda} (g_\alpha - g_\alpha^{eq}) + (c_\alpha - \mathbf{u}) \cdot [\nabla \varphi (\Gamma_\alpha(\mathbf{u}) - w_\alpha) + (\mathbf{F}_s + \mathbf{G}) \Gamma_\alpha(\mathbf{u})] \quad (3.2.16)$$

where $\Gamma_\alpha(\mathbf{u}) = f_\alpha^{eq}(\rho, \mathbf{u}) / \rho$. A noteworthy feature of equation (3.2.16) is that the sharp changes of density led by $\nabla \varphi = \nabla(c_s^2 \rho - p)$ are now multiplied by the relatively small magnitude of $(\Gamma_\alpha(\mathbf{u}) - w_\alpha)$ which diminishes the destructive effect of any truncation error arising from the discretized $\nabla \rho$ and hence improves the stability and maximum allowable density ratio. Nevertheless, the way in which the pressure and surface tension are defined could still trigger numerical errors and create spurious velocities at the interface.

In the original model of He and Chen [53], the pressure p is the total pressure which is the sum of the dynamic and thermodynamic pressures and the surface tension force is chosen to obey the following form

$$\mathbf{F}_s = \rho \gamma \nabla \nabla^2 \rho \quad (3.2.17)$$

where γ is related to the interface thickness ε and the surface tension coefficient σ . Lee and Fischer [74] later proposed a modified force term which transfers the chemical potential form of the thermodynamic pressure to the surface tension and leaves p to represent the dynamic pressure only. Then the diffuse form of \mathbf{F}_s already introduced in equation (2.2.10) is adopted such that

$$\mathbf{F} = \nabla(\rho c_s^2 - p) - C\nabla\mu_c + \mathbf{G} \quad (3.2.18)$$

where the chemical potential μ_c is a function of the concentration C as well as its Laplacian $\nabla^2 C$

$$\mu_c = 4\beta C(C-1)(C-0.5) - \mathcal{K}\nabla^2 C \quad (3.2.19)$$

and \mathcal{K} and β are chosen so as to control the interface thickness and surface tension coefficient. Moreover, replacing the total pressure with the dynamic part suggests to simplify equation (3.2.16) even more. In fact $\nabla p \sim \mathcal{O}(Ma^2)$ and hence the term $\nabla p(\Gamma_\alpha(\mathbf{u}) - w_\alpha)$ is reduced to $\mathcal{O}(Ma^3)$ and could be neglected in the incompressible limit [75] such that equation (3.2.16) could be rewritten as

$$\frac{\partial g_\alpha}{\partial t} + c_\alpha \cdot \nabla g_\alpha = \frac{1}{\lambda}(g_\alpha - g_\alpha^{eq}) + (c_\alpha - \mathbf{u}) \cdot [\nabla \rho c_s^2 (\Gamma_\alpha(\mathbf{u}) - w_\alpha) + (\mathbf{G} - C\nabla\mu_c)\Gamma_\alpha(\mathbf{u})]. \quad (3.2.20)$$

The concentration C varies between 0 and 1 and is obtained from the Cahn-Hilliard (CH) equation

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = \nabla \cdot (M\nabla\mu_c) \quad (3.2.21)$$

where the mobility M controls the mass diffusion rate and has a non-linear dependence on C in the original form of CH equation. According to the CH equation, the equilibrium concentration profile is obtained such that the free energy is minimized over the interface. In the specific case of a stationary flat interface of thickness ε , an analytical solution to this minimization problem is reached

$$C(d) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{2d}{\varepsilon}\right) \quad (3.2.22)$$

where d is the signed distance to the interface. This solution shows that the concentration asymptotically (diffusely) approaches 0 and 1 and differs from the commonly-used Heaviside function in sharp interface methods where the concentration of one phase is exactly nullified as the interface region is crossed towards the other phase.

In order to keep the solution of the entire coupled scheme on the lattice level equation (3.2.21) is solved via a second LBE for a new distribution function for C defined as $h_\alpha = (C/\rho)f_\alpha$. Then, similar to the procedure for g_α , one can write the transport of h_α by taking the total derivative D/Dt to get [76]

$$\frac{\partial h_\alpha}{\partial t} + c_\alpha \cdot \nabla h_\alpha = \frac{C}{\rho} \left[\frac{-1}{\lambda}(f_\alpha - f_\alpha^{eq}) + \frac{(c_\alpha - \mathbf{u}) \cdot \mathbf{F}}{c_s^2} \right] \quad (3.2.23)$$

which is simplified by substituting the definition of h_α for f_α to give

$$\begin{aligned} \frac{\partial h_\alpha}{\partial t} + c_\alpha \cdot \nabla h_\alpha &= \frac{-1}{\lambda}(f_\alpha - f_\alpha^{eq}) + f_\alpha \frac{D}{Dt} \left(\frac{C}{\rho} \right) + \frac{C}{\rho c - s^2} + \\ & (c_\alpha - \mathbf{u}) \cdot [\nabla \rho c_s^2 (\Gamma_\alpha(\mathbf{u}) - w_\alpha) + (\mathbf{G} - C\nabla\mu_c)\Gamma_\alpha(\mathbf{u})]. \end{aligned} \quad (3.2.24)$$

Similar to the approximation of the force term in equation (3.1.27), the remaining f_α on the right hand side could be approximated as $f_\alpha^{eq} = \rho\Gamma(\mathbf{u})$ such that the total derivative can be expanded as

$$f_\alpha \frac{D}{Dt} \left(\frac{C}{\rho} \right) \approx \left(\frac{\partial C}{\partial t} + c_\alpha \cdot \nabla C \right) \Gamma_\alpha - \frac{C}{\rho} \left(\frac{\partial \rho}{\partial t} + c_\alpha \cdot \nabla \rho \right) \Gamma_\alpha =$$

$$\left[(c_\alpha - \mathbf{u}) \cdot \nabla C + M \nabla^2 \mu_c \right] \Gamma_\alpha - \frac{C}{\rho} (c_\alpha - \mathbf{u}) \cdot \nabla \rho \Gamma_\alpha \quad (3.2.25)$$

where the continuity equation (2.1.11) and the CH equation (3.2.21) are used to account for $\partial \rho / \partial t$ and $\partial C / \partial t$, respectively and M is assumed to be a constant. Eventually, substituting the obtained expression for $f_\alpha D / Dt (C / \rho)$ in (3.2.24) yields the final LBE for h_α

$$\frac{\partial h_\alpha}{\partial t} + c_\alpha \cdot \nabla h_\alpha = \frac{-1}{\lambda} (h_\alpha - h_\alpha^{eq}) + M \nabla^2 \mu_c \Gamma_\alpha(\mathbf{u}) + (c_\alpha - \mathbf{u}) \cdot \left[\nabla C - \frac{C}{\rho c_s^2} (\nabla p + C \nabla \mu_c) \right] \Gamma_\alpha(\mathbf{u}). \quad (3.2.26)$$

The concentration is thus recovered through the zero order moment of the distribution function h_α as

$$C = \sum_\alpha h_\alpha. \quad (3.2.27)$$

Consequently, the fluid density and viscosity are obtained through the concentration as

$$\begin{cases} \rho(C) = \rho_g + (\rho_l - \rho_g) C \\ \mu(C) = \mu_g + (\mu_l - \mu_g) C \end{cases} \quad (3.2.28)$$

where the subscripts g and l denote the gas and liquid phases, respectively. In addition, the fluid pressure and velocity could be recovered using the moments in equation (3.1.5) modified in accordance with the definition of the new distribution function g_α in equation (3.2.7)

$$\rho(x, t) = \frac{1}{c_s^2} \sum_{\alpha=0}^n (g_\alpha - (p - \rho c_s^2) w_\alpha) \quad , \quad \rho \mathbf{u}(x, t) = \frac{1}{c_s^2} \sum_{\alpha=0}^n c_\alpha (g_\alpha - (p - \rho c_s^2) w_\alpha). \quad (3.2.29)$$

Remarks on pressure evolution LBM

The method has been shown to be stable for high density and viscosity ratios and employed for different case studies, e. g., droplet impact on liquid layers [75] and dry surfaces [76] as well as for rising bubble simulations [9], although the results were never compared against numerical benchmark data. However, one could obviously see that both equations (3.2.17) and (3.2.18) need three successive derivatives of the concentration profile. Furthermore, the second LBE in (3.2.26) will also require the fourth order derivative of C so as to recover the diffusion term of the CH equation. Such multilayer approximations, may lead to more computational complexity. Moreover, as the involved derivatives are basically obtained based on finite difference methods, using very high resolutions becomes inevitable to provide an adequately smooth density profile for numerical discretization. On the other hand, from the computational point of view, high degree of the derivatives requires multiple memory requests which often cause out-of-cache hits and severely impedes the solution on parallel compute machines. Moreover, the solution of the second LBE for the concentration means that the amount of the required memory is doubled as compared to single phase simulations which limits the maximum achievable resolution for 3D simulations on dense lattice stencils, e. g., D3Q19 or D3Q27 and on low memory GPGPUs. Consequently, coarse grid computations would be far from convergence and high resolution ones would be unaffordable if already not hitting the memory limits.

One remedy to alleviate the mentioned issues would be to utilize adaptive mesh refinement (AMR) techniques to concentrate the lattice around the interface only and thus save memory and

simulation times. Multiblock AMR-based LB simulations have their own complications as the change in lattice resolution implies different time steps for each lattice level which adds to the complexities of the scheme. Recently Fakhari and Lee [37, 38] proposed an AMR scheme based on a special finite difference discretization of the above two LBEs which allows for solving the LBEs on different lattice blocks with the same time step. However, similar to any other AMR scheme proposed to date, their scheme disturbs the structured layout of LBM's data arrays which could significantly degrade the parallel efficiency of the coupled solution.

Other alternatives, however, include using classical finite difference or element methods instead of the LBM for the solution of the CH equation or replace the CH equation with other interface capturing techniques, e. g., level set or volume of fluid methods. While the classical solutions of the CH equation are reported to have issues with controlling the non-linearity of the right hand side terms and large numerical mass loss [44, 132], the use of other interface capturing techniques could be a promising choice as the associated PDEs, particularly in the level set method, could be designed to conserve mass and have generally much lower degrees of non-linearity. Moreover, the resulting concentration profiles could be readily used to obtain the surface tension forces based on sharp or diffuse CSF techniques and hence avoid high order derivatives of equations 3.2.18 and 3.2.19. Such implementations are the subject of chapter 5, where the proposed sharp and diffuse coupled LBM-level set techniques will be discussed and examined for benchmark problems.

Before proceeding with the structure of the coupled schemes, it is necessary to first describe how interface capturing techniques work in general and what are the essential elements of the level set technique in particular for the problems of interest in this thesis.

Interface capturing

There has been a variety of interface tracking and capturing techniques proposed for multiphase flow problems. A major classification is done by whether the interface is tracked explicitly in a Lagrangian way or if an implicit Eulerian scheme is adopted to realise where the phases are separated in the computational domain. Lagrangian schemes are based on tracking the interface as an independent object (line or surface) which can be discretized to piecewise segments and hence the computational cells can be aligned with the interface. Although the method gives quite sharp representation of the interface, the computational algorithm is usually very complex as topology changes can not be handled automatically, and need periodic remeshing and additional techniques to realise unconventional deformations, e. g., merging and breakup.

Eulerian methods, on the other hand, are based on tracking or capturing the interface via an implicit function which moves on an arbitrarily defined computational mesh and thus does not have to deform or redistribute the grid points as the interface evolves. This further brings a number of desirable numerical and computational features and facilitates capturing of breakup or merging phenomena without additional algorithmic considerations. However, even those sharp interface variants of Eulerian schemes require to smear the interface over a number of grid cells to some extent so as to avoid the numerical discontinuity of the properties and hence introducing some artificial modifications will be inevitable. This means that in order to have the same resolution as in the Lagrangian schemes, the interface thickness has to be as small as possible which requires relatively high grid resolutions.

The coupled two-phase flow solvers proposed in this thesis are designed for large scale problems which involve complex interface changes and are desired to impose low computational costs and favourable parallel efficiency. Therefore, employing an Eulerian method which can enjoy regular meshing styles, is definitely preferred. As such, the most widely used variants of such methods are first reviewed in the following. The eventual choice, the level set method, is then briefly introduced for sharp and diffuse interfaces and proper numerical techniques are discussed for solving the resulting equations. The chapter is finally closed with studying the numerical and computational performance of the employed methods for some benchmark problems.

4.1. Choice of the interface capturing technique

The most widely used Eulerian interface *capturing* technique is the volume of fluid model first proposed by Hirt and Nichols [88]. It is based on recognizing the computational cells with their filling level or the so-called volume fraction; this means that cells are either empty and labelled by 0, completely filled and thus denoted by 1, or partially filled and hence get a value between 0 and 1 depending on the filling level. These partially filled levels are those intersected by the interface. An approximated description of the interface is then reconstructed based on the averaged filling level of such cells and their neighbours. The method was particularly proved to possess nice

mass preserving behaviour and is favourable for structured rectangular meshes as it makes the interface reconstruction straightforward. Nevertheless, the resulting interface shape will not be unique and its realisation depends very much on the choice of the reconstruction method which could vary between piecewise linear schemes (PLIC) [136] to higher order and complex parabolic reconstruction [100] or least square methods [101]. Moreover, since the interface is modelled as a sharp jump from 0 to 1, its convection with the flow velocity often needs special treatments to preserve the sharpness of the interface.

Another popular class of interface *tracking schemes* are those based on tracking a group of particle distributions which point to the front of the interface and are convected with the flow velocity. As the locations of the particles do not necessarily coincide with the computational grid, these often called *front tracking* methods, need to interpolate the velocity to the location of the particles. Conversely, in order to introduce the effect of the interface into the flow solver, a smeared delta function has to be constructed based on the location of the particles. Though the algorithm provides a precise description of the interface, it also suffers from the same difficulties as Lagrangian schemes in modelling breakup and merging of the interfaces and particles have to be periodically generated and removed to avoid kinks or unphysical interface shapes.

A more physically rich method already described in diffuse interface LB models is the phase-field model which is based on solving the Cahn-Hilliard (CH) equation for the concentrations field which minimizes the chemical potential energy over the interface and acts like a conservation equation for the concentration [82]. The implicit concentration parameter could then be considered as a diffuse representation of the interface. Beside the inherent mass conservation property of the CHE, a desirable feature of the method is that it can automatically realise breakup and merging of interfaces and does not need explicit manipulations by the user to treat such phenomena. Nevertheless, the original thermodynamically consistent CH equation with variable mobility is highly non-linear and numerical oscillations are inevitable if regular explicit time integration schemes are used [44]. Therefore, in most numerical simulations the mobility is taken as a constant to linearise the equation and thus to ease the numerical treatment [76]. Moreover, the mass preserving property of the equation is quite sensitive to the numerical solver and could be easily violated unless a very high resolution grid is used. Different ramifications are thus employed to preserve the mass, e. g., by replacing the equation with modified Allen-Chan equation [41], or through making up the truncation errors by adding them explicitly to the right hand side of the CH equation [132].

A combination of the desirable properties of the VOF and the phase-field method could be realised in the level set method (LSM), where an implicit function is defined on the entire domain, while its evolution is governed by a conservation equation. This strategy has led to several variants of the LSM, where sharp or diffuse representation of the interface could be realised depending on the choice of the implicit LS function. Two variants of the LS method are then discussed in the following and employed for the proposed coupled two-phase flow solvers. The first scheme, the classical variant of the LSM, is based on a signed distance function and allows for capturing a relatively sharp interface since the interface thickness does not control the quality of the resolved interface. The second scheme, a diffuse LSM, is based on the phase-field description of the interface but obeys the same conservation rule as in the sharp interface equation.

4.2. Signed distance level set method

The classical LSM was first introduced by Osher and Sethian in [92] where they proposed to capture the interface $\Gamma(t)$ implicitly via a higher dimensional function ϕ

$$\Gamma(t) = \{\mathbf{x} \in \mathbb{R}^d | \phi(\mathbf{x}, t) = v_{ls}\} \quad (4.2.1)$$

where v_{ls} is the iso-level or contour pointing to the location of the interface. By convention, the contour $\phi = 0$ is chosen to represent the interface curve in 2D or surface in 3D, while the negative and positive values simply refer to the inner and outer regions of the interface. In [92] it was suggested to initialize the function ϕ as the signed distance from the interface

$$\begin{cases} dist(\Gamma, \mathbf{x}) & , \mathbf{x} \in \Omega_1 \\ 0 & , \mathbf{x} \in \Gamma \\ -dist(\Gamma, \mathbf{x}) & , \mathbf{x} \in \Omega_2 \end{cases} \quad (4.2.2)$$

where Ω_1, Ω_2 denote the regions occupied by phase 1 and phase 2 respectively. An illustration of a 2D interface denoted by the iso-contour of $\phi = 0$ is given in figure 4.1 for a circular bubble.

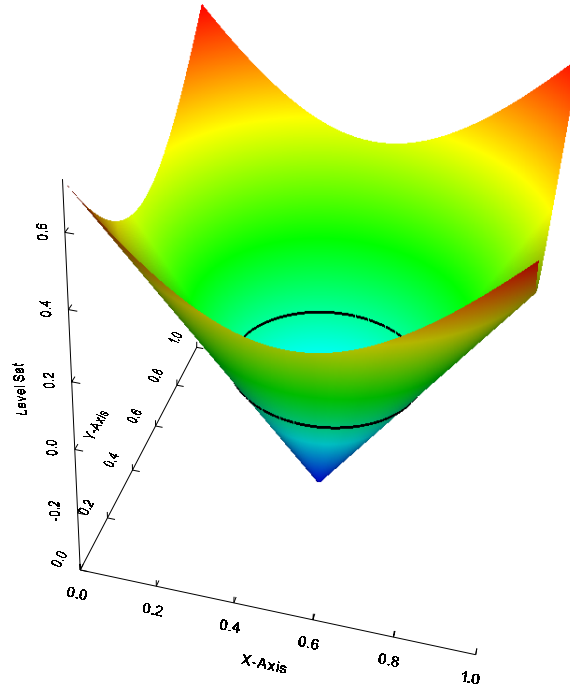


Figure 4.1: Implicit realisation of the interface illustrated by the solid iso-line of $\phi = 0$ for a 2D circular bubble.

Such a definition of the LS function has the advantage that the profile of ϕ is smooth since it varies steadily with the signed distance from the interface and does not jump over the interface. Consequently, advecting the LS field with velocity \mathbf{u} is numerically more convenient than moving a sharp jump, provided that the LS values remain to be, or close to, signed distance values. Another advantage of the signed distance property is that the bulk fluid properties could be readily related to the individual properties of each phase as

$$\begin{cases} \rho(\phi) = \rho_l H(\phi) + \rho_g (1 - H(\phi)) \\ \mu(\phi) = \mu_l H(\phi) + \mu_g (1 - H(\phi)) \end{cases} \quad (4.2.3)$$

where $H(\phi)$ is a regularized Heaviside function defined as

$$H(\phi) = \begin{cases} 0 & \phi < -\varepsilon \\ \frac{1}{2} \left[1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\varepsilon}\right) \right] & |\phi| \leq \varepsilon \\ 1 & \phi > \varepsilon \end{cases} \quad (4.2.4)$$

and ε is the interface thickness on each side. Consequently, a smoothed realization of the delta function can be derived from the Heaviside function [120] as

$$\delta_\varepsilon(\phi) = \frac{dH}{d\phi} = \begin{cases} 0 & \phi < -\varepsilon \\ \frac{1}{2} \left[\frac{1}{\varepsilon} + \cos\left(\frac{\pi\phi}{\varepsilon}\right) \right] & |\phi| \leq \varepsilon \\ 0 & \phi > \varepsilon. \end{cases} \quad (4.2.5)$$

Furthermore, the curvature as well as the normal vector to the interface are obtained based on the LS function ϕ

$$\mathbf{n}(\phi) = \frac{\nabla\phi}{|\nabla\phi|}, \quad \kappa(\phi) = \nabla \cdot \mathbf{n} = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right). \quad (4.2.6)$$

One may derive the evolution equation for the LS function starting with the fact that the $\phi(\mathbf{x}(t), t) = 0$ property of the interface must remain valid as the interface moves within the computational domain. In mathematical terms, the following property must hold true

$$\frac{\partial\phi}{\partial t} + \nabla\phi \cdot \frac{\partial\mathbf{x}(t)}{\partial t} = 0. \quad (4.2.7)$$

One way to reach the final level set equation (LSE) is to recognize $\partial\mathbf{x}(t)/\partial t$ as the Eulerian description of the velocity in fluid flow simulations. A more general way, however, is to consider the fact that the speed by which the front is propagated is $V = \mathbf{n} \cdot (\partial\mathbf{x}(t)/\partial t)$ and then use the definition of \mathbf{n} to have

$$\frac{\partial\phi}{\partial t} + V|\nabla\phi| = 0 \quad (4.2.8)$$

which is assumed valid for ϕ on the entire domain. Now, by choosing the speed function to be the velocity of the fluid normal to the interface, i. e., $V = \mathbf{u} \cdot \mathbf{n}$ and again using the definition of \mathbf{n} , one ends up with the global LSE for the transport of ϕ in a two-phase flow

$$\partial_t\phi + \mathbf{u} \cdot \nabla\phi = 0. \quad (4.2.9)$$

4.2.1. Numerical solution of the signed distance level set equation

The outer LBM flow solver requires the values of ϕ at the new time step $\phi(\mathbf{x}, t + \Delta t)$ if for example trapezoidal rule is to be applied to integrate the forcing terms in LBE. This implies that the values of velocity for the LSE come from the t time step and the integration in time must be explicit. Application of the second order Runge-Kutta scheme in time yields

$$\begin{aligned} \phi(x, t^*) &= \phi(x, t) + \Delta t L\phi(x, t) \\ \phi(x, t + \Delta t) &= \phi(x, t) + \frac{1}{2}\Delta t (L\phi(x, t) + L\phi(x, t^*)) \end{aligned} \quad (4.2.10)$$

where

$$L\phi = -\mathbf{u} \cdot \nabla\phi. \quad (4.2.11)$$

For the discretization of the convective term in space one may use different upwinding or other high order TVD schemes for stabilization. Here the 5th order weighted essentially non-oscillatory (WENO) scheme introduced in [81], and described in Appendix A, is employed which extends the first-order accurate upwind differencing to fifth-order spatial accuracy based on the smoothest possible interpolation for the function ϕ . It is worth noting that although the LS function

is generally known to have smooth variations in the computational field, the use of the WENO scheme ensures that the interface is accurately convected in the case that sharp kinks or edges of the interface might locally disturb the smoothness of the LS functions which eventually produces steep changes in the value of ϕ .

4.3. Level set reinitialization

As the solution of equation (4.2.9) advances in time, the LS function deviates from its initial favourable signed distance property. In fact, regardless of the type of initial prescription being signed distance or not, adjacent iso-contours pile up on each other and the interface starts to distort as the solution proceeds. This not only violates the initial assumption, but further degrades the quality of the reconstructed normals and curvatures. The only situation where the smooth LS retains its initial property is when the normal velocity along the characteristic normal to the interface remains constant, which is not generally fulfilled by the velocity fields in fluid flows. As a remedy, one may think of an extension velocity field in which a velocity field consistent with the mentioned condition is constructed around the interface while the velocity at the interface is preserved to be that of the original flow [5]. Unlike this approach which requires reconstructing a vector field, another popular solution is to reinitialize the scalar LS function so as to re-establish the signed distance property. Mathematically speaking, this is equivalent to solving the so-called Eikonal equation

$$|\phi(\mathbf{x})| = V(\mathbf{x}) \quad (4.3.1)$$

By setting $V(\mathbf{x}) = 1$, a signed distance field must be obtained as the solution. This general idea has led to a class of schemes with different numerical approaches to construct a signed distance field in the vicinity of the interface or on the entire computational domain. Apart from the accuracy of the obtained LS field and the reconstructed normals and curvatures, care must be taken that the original position of the interface could be modified during the reinitialization as this may cause significant mass loss. On the other hand, the computational overhead of the reinitialization must be reasonable and the parallel scaling has to be favourable when implemented on parallel compute machines.

A widely used approach proposed by Sethian in [108, 109] is the fast marching method. The method is based on the fact that the properties are propagated outward from the interface in the normal direction. One can then start from a band or a group of points adjacent to the interface being already close to distance function and then march in the normal direction through the new neighbouring nodes and update their values to signed distance by solving the Eikonal equation. The algorithm proceeds with admitting these updated points into the initial band until the band reaches a sufficient thickness to evaluate the surface tension forces [107, 108]. The scheme has been proved to be very robust in recovering the signed distance field and preserving the mass [58, 91]. Moreover, the computational cost in serial implementations scales with $\mathcal{O}(N \log N)$ which is quite favourable compared to other schemes which often scale with $\mathcal{O}(N^2)$. Nevertheless, a parallel implementation of the method on many-core architecture, e. g., GPGPUs, becomes problematic as updating ϕ at any certain point is not independent from the neighbouring points. The algorithm could potentially lead to write-after-write (WAW) situation in memory access which requires careful programming and brings additional overhead. Considering also the multiple branch diverging involved in the algorithm due to the search for new band members, the parallel performance can decay rapidly for large computations on GPGPUs and the method is thus not implemented in this work. For a recent study on the GPGPU implementation characteristics of the method one may refer to [29]. The other two popular schemes are the PDE-based technique and the brute-force method which are discussed in the following.

4.3.1. PDE-based reinitialization

One approach to solve the Eikonal equation (4.3.1) is to solve the following time dependent equation over an artificial time θ , as

$$\frac{\partial \phi}{\partial \theta} + |\nabla \phi| = 1 \quad (4.3.2)$$

for Ω_1 and Ω_2 simultaneously until the steady state is reached and equation (4.3.1) is thus satisfied. The problem with this approach is that there is no control over the position of the interface as it may be readily displaced during the solution of (4.3.2). To circumvent this issue, one could compute the signed distance function for all the grid points adjacent to the interface manually and then solve the following pair of equations

$$\begin{cases} \frac{\partial \phi}{\partial \theta} + |\nabla \phi| = 1 & , \mathbf{x} \in \Omega_1 \quad (\phi > 0) \\ \frac{\partial \phi}{\partial \theta} - |\nabla \phi| = -1 & , \mathbf{x} \in \Omega_2 \quad (\phi < 0) \end{cases} \quad (4.3.3)$$

with the boundary condition of $\phi = \phi_{old}, \mathbf{x} \in \Gamma$ [91]. Alternatively, Sussman et. al proposed in [118] to combine the above two equations in a single PDE using the sign function $W(\phi)$

$$\partial_{\theta} \phi + W(\phi)(|\nabla \phi| - 1) = 0 \quad (4.3.4)$$

which propagates with the normal velocity of magnitude 1 and hence quickly reaches steady state. For example by setting $\Delta x / \Delta \theta = 2$, and considering an interface thickness of m grid cells, the general rule of thumb would be to solve equation (4.3.4) for $2m$ time steps. The choice of the sign function has to be made with care since equation (4.3.4) is a hyperbolic equation. In particular, it has to be a smooth function to avoid discontinuity at the interface during the numerical solution. A few choices have been studied in the literature including the smoothed *sin* function in [98] or the rational function proposed by Peng et. al. [94],

$$W(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla \phi|^2 \Delta x^2}} \quad (4.3.5)$$

which has to be updated continuously as the iterative solution proceeds and is thus computationally demanding. Here, the original sign function suggested by Sussman et. al [118] is preferred

$$W(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \Delta x^2}} \quad (4.3.6)$$

which depends merely on ϕ_0 and has to be calculated only once. Finally, care must be taken that for the PDE-based scheme to be effective and accurate, ϕ_0 has to be already close to a signed distance function near the interface, meaning that equation (4.3.4) has to be solved $2m$ times at the end of each outer iteration.

For the solution of equation (4.3.4) one may use the second order Runge-Kutta method, although even a simple backward Euler would suffice since $\Delta \theta$ is often chosen to be very small

$$\phi(\mathbf{x}, \theta + \Delta \theta) = \phi(\mathbf{x}, \theta) - W(\phi)(|\nabla \phi| - 1)|_{\mathbf{x}, \theta}. \quad (4.3.7)$$

Discretizing $|\nabla \phi|$ also requires special care occurred to the hyperbolic nature of equation (4.3.4). Here the second order ENO scheme proposed in [127] is adopted. Assuming $D_x^+ \phi = \phi(x + \Delta x, y) - \phi(x, y)$ and $D_x^- \phi = \phi(x, y) - \phi(x - \Delta x, y)$, the derivative of ϕ in x direction denoted by $\nabla_x \phi$ is obtained as

$$\nabla_x \phi = \begin{cases} \tilde{D}_x^+ & \text{sgn}(\phi)D_x^+ \phi < 0 \quad \text{and} \quad \text{sgn}(\phi)D_x^- \phi < -\text{sgn}(\phi)D_x^+ \phi \\ \tilde{D}_x^- & \text{sgn}(\phi)D_x^- \phi > 0 \quad \text{and} \quad \text{sgn}(\phi)D_x^+ \phi > -\text{sgn}(\phi)D_x^- \phi \\ \frac{1}{2}(\tilde{D}_x^+ + \tilde{D}_x^-) & \text{otherwise} \end{cases} \quad (4.3.8)$$

with the modified differences defined as

$$\begin{aligned} \tilde{D}_x^+ &= D_x^+ \phi - \frac{1}{2}M(D_x^+ D_x^- \phi(x, y), D_x^+ D_x^- \phi(x + \Delta x, y)) \\ \tilde{D}_x^- &= D_x^- \phi + \frac{1}{2}M(D_x^+ D_x^- \phi(x, y), D_x^+ D_x^- \phi(x - \Delta x, y)) \end{aligned} \quad (4.3.9)$$

where the function $M(a, b)$ is given by

$$M(a, b) = \begin{cases} a & , \quad |a| < |b| \\ b & , \quad |b| \leq |a| \end{cases} \quad (4.3.10)$$

4.3.2. Brute-force reinitialization

The brute force reinitialization enforces the signed distance property by measuring the minimum distances of the grid points to the interface without solving the Eikonal equation. Although the implementation is straightforward, criticisms have been made on the mass conservation accuracy of the method since the interface might be displaced during the reinitialization. In addition, the computational cost scales with $\mathcal{O}(Nm)$ where m is the number of linear segments of the interface. In order to circumvent the mass loss issue, the weighted reinitialization method proposed in [128] is adopted here which applies a smoothed transition from the old LS values ϕ_0 to the new ones ϕ based on the obtained signed distance values ϕ_{dist}

$$\phi = \omega \phi_{dist} + (1 - \omega) \phi_0 \quad (4.3.11)$$

where ω is the weight factor. Since the brute-force method is highly effective in establishing a signed distance field, this weighted reinitialization is then performed periodically. Moreover, the algorithm tries to preserve the original values of ϕ for the points which contribute to the reconstruction of the interface by excluding them from the re-distancing computations. Eventually, the values of ω and the length of the period N_p depend on the velocity field and hence the deformation intensity and could be adjusted experimentally to control the mass conservation and minimizing oscillations in flow properties.

The periodic weighted method not only helps preserving the mass, but also lowers the computational cost as the global re-distancing is called only periodically. For the implementation on GPGPUs, however, one has to bear in mind that still the interface reconstruction involves $\mathcal{O}(N^2)$ searches to find the intersected cells which could drastically lower the parallel efficiency. This issue is addressed through utilizing a hybrid CPU-GPGPU framework, explained later in chapter 7. The numerical algorithm is nevertheless carried out in the following steps:

1. The interface is reconstructed by linear segments. This is realised by performing a search for intersected cells through checking if the two endpoints of any edge in the cell have different signs.
2. A piecewise linear interpolation is then used to find the intersection points. Each pair of points on a cell constitute an interface segment. A total of m segments will then represent the interface.

3. The distance from each of the grid points inside a band of 4ϵ in thickness is then calculated to all the reconstructed segments. For each segment, the minimum of the distances from the grid point to both endpoints as well as the middle point of the segment is saved to memory. The process gives m number of distances for each grid point.
4. The m distances are finally compared and their minimum, multiplied by the sign of ϕ_0 , is considered as ϕ_{dist} .

4.4. Conservative phase-field level set method

The signed distance LS function has the advantage of being a smooth function provided that it preserves its signed distance property. Nevertheless, even under a divergence-free velocity field, the LS function needs to be reinitialized as discussed above. On the other hand, even the most robust reinitialization schemes tend to displace the actual interface and additional measures have to be taken to control the mass loss as for example by adding constraints to the PDE-based method proposed by Sussman et. al [120], or through a geometrical approach, i. e., periodically calculating the peripheral as well as the total amount of mass loss in the brute-force method and then compensating the lost mass by dividing it by the interface length and adding the result to the global LS field [56]. While the former method adds more computational cost to the already expensive PDE-based method, the latter is not only costly but is also unphysical and its application becomes ambiguous in the case of interface breakup or merging.

On the other hand, the relatively sharp realization of the interface in the conventional LSE is of no merit if used in conjunction with a diffuse multiphase flow solver, e. g., the pressure evolution LBM, where one has to eventually replace the Heaviside function with diffuse alternatives, e. g., the tangent hyperbolic function as in equation (3.2.22).

In their 2005 paper, Olsson and Kreiss [90] proposed that if a LS function which directly points to the phases is initialized together with a conservative method to convect this function, then one could ensure that the mass remains conserved at the price of adding additional constraints to the LSE. Following the lines of [90], let us define the phase field function ψ whose value varies between 0 and 1 similar to the concentration C in the Cahn-Hilliard equation, where $\psi = 0.5$ level indicates the interface such that the fluid properties are directly recovered from ψ as

$$\begin{cases} \rho(\psi) = \rho_g + (\rho_l - \rho_g)\psi \\ \mu(\psi) = \mu_g + (\mu_l - \mu_g)\psi \end{cases} \quad (4.4.1)$$

while the normal to the interface along with the curvature could be obtained based on ψ

$$\mathbf{n}(\psi) = \frac{\nabla\psi}{|\nabla\psi|}, \quad \kappa(\psi) = \nabla \cdot \mathbf{n} = \nabla \cdot \left(\frac{\nabla\psi}{|\nabla\psi|} \right). \quad (4.4.2)$$

In addition, a smoothed delta function could be directly realised as $\delta(\psi) = |\nabla\psi|$. In order to capture the interface, the LSE to convect the phase field function ψ reads

$$\partial_t \psi + \mathbf{u} \cdot \nabla \psi = 0. \quad (4.4.3)$$

Assuming a divergence free velocity field $\nabla \cdot \mathbf{u} = 0$, one could directly deduce the conservation of ψ

$$\partial_t \psi + \nabla \cdot (\psi \mathbf{u}) = 0. \quad (4.4.4)$$

In order to preserve the thickness of the interface and keep the profile of ψ smooth as it convects, one may add a compression term $-\nabla \cdot (\psi(1 - \psi)\mathbf{n})$, acting exclusively on the interface

region, along with a balancing artificial diffusion $\eta \nabla^2 \psi$ to the right hand side of equation (4.4.3). Eventually, the following monolithic equation is solved in this work with the same length and time scales of LBM

$$\partial_t \psi + \mathbf{u} \cdot \nabla \psi = -\nabla \cdot (\psi(1-\psi)\mathbf{n}) + \eta \nabla^2 \psi \quad (4.4.5)$$

where the parameter η calibrates the numerical diffusion. This differs from the original algorithm suggested in [90], where the compression and diffusion are involved in a second reinitialization equation which is solved iteratively with small virtual time step analogous to the PDE-based reinitialization. On the other hand, considering the typically very small time steps of the LBE, a mere single iteration of equation (4.4.5) suffices to keep the smooth profile of ψ and preserve a constant interface thickness. The interface thickness parameter η could be chosen as a function of the physical resolution $h = \Delta x$ of the problem

$$\eta = \frac{h^\alpha}{2} \quad (4.4.6)$$

where the exponent α is chosen to be close to 1 so as to control the interface thickness and has a subsequent effect on the quality of the interface capturing as well as the overall mass conservation. Finally, solving equation (4.4.5) requires boundary and initial conditions. To obtain consistent initial conditions for ψ , one may set $\mathbf{u} = 0$ and numerically solve equation (4.4.5) to steady state. For the special case of a 2D circular bubble with $\psi = 0$ representing the inner region, an analytical solution is also available

$$\psi_0 = 1 - \frac{1}{1 + e^{d/\varepsilon}} \quad (4.4.7)$$

where d is the signed distance from the interface. Considering the derivatives in the source terms on the right hand side of equation (4.4.5), two boundary conditions are required for ψ and $\psi(1-\psi)\mathbf{n}$. In the particular case that a solid wall is encountered at the boundary, the homogeneous conditions analogous to those typically used for the Cahn-Hilliard equations could be applied [131]

$$\left. \frac{\partial \psi}{\partial n} \right|_{wall} = 0 \quad (4.4.8)$$

$$\nabla \cdot (\psi(1-\psi)\mathbf{n})|_{wall} = 0. \quad (4.4.9)$$

While the implementation of a periodic boundary is straightforward, the above conditions are also implemented in the case of slip boundaries where the phase-field is assumed to extend steadily across the boundary.

4.4.1. Numerical solution of the phase-field level set equation

Time integration of equation (4.4.5) is performed by the second order Runge-Kutta scheme

$$\begin{aligned} \psi(x, t^*) &= \psi(x, t) + \Delta t L\phi(x, t) \\ \psi(x, t + \Delta t) &= \psi(x, t) + \frac{1}{2} \Delta t (L\psi(x, t) + L\psi(x, t^*)) \end{aligned} \quad (4.4.10)$$

where

$$L\phi = [-\mathbf{u} \cdot \nabla \psi \mathbf{n}] - \nabla \cdot (\eta \psi(1-\psi) \frac{\nabla \psi}{|\nabla \psi|}) + \eta \nabla^2 \psi. \quad (4.4.11)$$

Apart from the CFL condition of $C = u\Delta x/\Delta t < 1$ for the convective term, the explicit time stepping also puts an upper bound on the diffusion number $\eta\Delta t/\Delta x^2$ and hence the maximum permissible time step

$$D = \frac{\eta\Delta t}{\Delta x^2} \leq D_m \quad (4.4.12)$$

Olsson and Kreiss suggested $D_m = 0.25$ in their simulations for the rigid rotation of a circular disk. On the other hand, numerical experiments with more stringent velocity fields, e. g., a shear velocity field suggest a safe upper bound of $D_m = 0.15$ for the monolithic implementations of equation (4.4.6) which puts a somewhat stricter limitation on the maximum allowable time step. The monolithic approach, however, is expected to bring a significant saving in computational cost which outweighs the decelerating effect of smaller time steps, although large time steps are not of interest when the time resolution has to match that of the LBM.

For the discretization in space, central differencing is used for the diffusion and compression terms, while the 5th order WENO scheme is employed for $\nabla\psi$ in the convective term, making the overall discretization total variation bounded (TVB) as speculated by Osher and Fedkiw [91]. It must be again emphasized that the high order accuracy and hence large spacial support of the WENO scheme although being beyond the second order accuracy and requires relatively higher overhead as compared to 3rd order ENO or other TVD schemes, yet guaranties preserving a smooth interface which is crucial to recover accurate pressure and velocity fields.

4.5. Numerical experiments

To have a more comprehensive understanding of the numerical performance of the employed LS schemes, the accuracy of sharp and diffuse interface tracking methods discussed above is studied through performing numerical experiments to solve a number of benchmark problems. Before defining these problems, it is helpful to rewrite the LSEs in suitable forms as required when eventually coupling them with solutions of the LBE. Specifically, in order to be consistent with the microscopic velocities from the LB flow solver, one may convert the LSEs into non-dimensional forms

$$\partial_t^*\phi + \mathbf{u} \frac{\Delta t}{\Delta x} \cdot \nabla^*\phi = 0. \quad (4.5.1)$$

$$\partial_t^*\psi + \mathbf{u} \frac{\Delta t}{\Delta x} \cdot \nabla^*\psi = -\frac{\Delta t}{\Delta x} \nabla^* \cdot (\psi(1-\psi)\mathbf{n}) + \eta \frac{\Delta t}{\Delta x^2} \nabla^{2,*}\psi \quad (4.5.2)$$

where the non-dimensional difference operators are $\partial_t^* = \Delta t \partial_t$, $\nabla^* = \Delta x \nabla$, $\nabla^{2,*} = \Delta x^2 \nabla^2$. Considering $\Delta t_{lb} = \Delta x_{lb} = 1$ in the LBM framework, one may replace $\mathbf{u}\Delta t/\Delta x = \mathbf{u}_{lb}$ to obtain

$$\partial_t^*\phi + \mathbf{u}_{lb} \cdot \nabla^*\phi = 0. \quad (4.5.3)$$

$$\partial_t^*\psi + \mathbf{u}_{lb} \cdot \nabla^*\psi = -\beta \nabla^* \cdot (\psi(1-\psi)\mathbf{n}) + D \nabla^{2,*}\psi. \quad (4.5.4)$$

where $\beta = \Delta t/\Delta x$.

4.5.1. Benchmarking problems

A group of widely used benchmark problems with known exact solutions are selected and described below, where the time evolution of each problem will be studied through quantitative and qualitative measurements.

Configuration and error measurement

It must be noted that all foregoing numerical experiments use $U_0 = 1$ and its lattice counterpart $U_0^* = 0.02$ which is a typical lattice velocity and is well below the maximum incompressible limit of $\mathbf{u}_{lb} < 0.1$. Moreover, for a characteristic length of $L_0 = 1$, the lattice length is set to $L_0^* = L_0/h$. By default, a lattice length of $L_0^* = 200$ is set for all the upcoming experiments, unless otherwise mentioned. From the stability point of view, this choice of velocity and length scale corresponds to a CFL number of $C = 0.02$ and diffusion number of $D = 0.022$ which satisfies both convection and diffusion stability criteria. Finally, the non-dimensional time is redefined based on the new characteristic values and the number of lattice iterations t_{lb}

$$T = \frac{L_0^*}{U_0^* t_{lb}} \quad (4.5.5)$$

In order to quantify the errors of the numerical solutions, the following approximations for the relative errors in L_1 , L_2 and L_∞ are used

$$\begin{aligned} L_1 \text{ error :} \quad & ||e_1|| = \left(\frac{\sum_{i=1}^N |\varphi_c - \varphi_0|}{\sum_{i=1}^N |\varphi_0|} \right) \\ L_2 \text{ error :} \quad & ||e_2|| = \left(\frac{\sum_{i=1}^N |\varphi_c - \varphi_0|^2}{\sum_{i=1}^N |\varphi_0|^2} \right)^{1/2} \\ L_\infty \text{ error :} \quad & ||e_\infty|| = \frac{\max_i |\varphi_c - \varphi_0|}{\max_i |\varphi_0|} \end{aligned}$$

where φ_0 and φ_c are the exact and computed values of the concentration field, respectively. The values of φ are thus obtained based on $H(\phi)$ and ψ for the sharp and phase field LS functions, respectively. In addition, the temporal area inside the interface is measured during the simulations and compared for different methods to evaluate their mass preserving behaviour.

Linear translation of a circular disk

The most elementary problem is the linear translation of a circle under the constant velocity field

$$\begin{cases} u_x(x, y) = U_0^* \\ u_y(x, y) = U_0^* \end{cases} \quad (4.5.6)$$

while the disk itself is initially located at $(x, y) = (0.5, 0.5)$ and returns to its origin at $T = 1$. The particular feature of this problem is that it satisfies the conditions to preserve a signed distance field as the magnitude of velocity normal to the interface characteristic directions remains constant.

Rigid rotation of a disk

The second test case loses the nice properties of the linear velocity field and instead applies the rigid rotation velocity

$$\begin{cases} u_x(x, y) = -U_0^* \pi \left(\frac{y^*}{L_0^*} - 0.5 \right) \\ u_y(x, y) = U_0^* \pi \left(\frac{x^*}{L_0^*} - 0.5 \right). \end{cases} \quad (4.5.7)$$

The disk with the radius of $R = 0.2$ is located at $(x, y) = (0.3, 0.5)$ at $T = 0$ and completes one rotation at $T = 2$. Since the disk does not undergo any shear and hence no potentially irreversible breakup, the problem is suitable to check the accuracy of the sharp LSM when employing reinitialization schemes.

Disk under reversed shear flow

The rotating shear disk problem could be modified in a way that the velocity field mitigates a vortex-like shear flow where the flow direction reverses at $T = 1$ so as to recover the the original state at $T = 2$. The non-dimensional velocity field reads

$$\begin{cases} u_x(x, y) = -U_0^* \pi \cos \left[\pi \left(\frac{x^*}{L_0^*} - 0.5 \right) \right] \sin \left[\pi \left(\frac{y^*}{L_0^*} - 0.5 \right) \right] \\ u_y(x, y) = U_0^* \pi \sin \left[\pi \left(\frac{x^*}{L_0^*} - 0.5 \right) \right] \cos \left[\pi \left(\frac{y^*}{L_0^*} - 0.5 \right) \right]. \end{cases} \quad (4.5.8)$$

Zalesak's disk

The Zalesak's disk problem also consists of a circular disk with a rectangular notch of width $w = 0.04$, made halfway through the disk.

$$\begin{cases} u_x(x, y) = -U_0 \pi (y - 0.5) \\ u_y(x, y) = U_0 \pi (x - 0.5). \end{cases} \quad (4.5.9)$$

The disk of radius $R = 0.4$ is located at $(x, y) = (0.5, 0.5)$ and completes one revolution around its centre at $T = 2$. Adding the rectangular notch to the disk creates sharp edges whose final shapes show how strong the artificial diffusion effects can act in smearing the sharp edges and corners, specially in diffuse interface LS method.

Deforming disk

The last and the most formidable benchmark is the circular disk of radius $R = 0.25$ located at $(x, y) = (0.5, 0.5)$, subject to the following intense velocity field [102]

$$\begin{cases} u_x(x, y) = -U_0^* \pi \sin \left[4\pi \left(\frac{x^*}{L_0^*} - 0.5 \right) \right] \sin \left[4\pi \left(\frac{y^*}{L_0^*} - 0.5 \right) \right] \\ u_y(x, y) = -U_0^* \pi \cos \left[4\pi \left(\frac{x^*}{L_0^*} - 0.5 \right) \right] \cos \left[4\pi \left(\frac{y^*}{L_0^*} - 0.5 \right) \right] \end{cases} \quad (4.5.10)$$

which experiences a rapid reversing at $T = 0.5$ and thus may induce irreversible deformations. A smoothed version of this velocity field is thus proposed in [79] which reverses gradually around $T = 0.5$

$$\begin{cases} u_x(x, y) = -\cos(\pi T) U_0^* \pi \sin \left[4\pi \left(\frac{x^*}{L_0^*} - 0.5 \right) \right] \sin \left[4\pi \left(\frac{y^*}{L_0^*} - 0.5 \right) \right] \\ u_y(x, y) = -\cos(\pi T) U_0^* \pi \cos \left[4\pi \left(\frac{x^*}{L_0^*} - 0.5 \right) \right] \cos \left[4\pi \left(\frac{y^*}{L_0^*} - 0.5 \right) \right]. \end{cases} \quad (4.5.11)$$

4.5.2. Benchmarking results

Figures 4.2(a) to (c) show the evolution of the reversed shear disk problem using the sharp LSM with and without reinitialization. For the brute force method the reinitialization period of $N_p = 20$ and the re-distancing weight of $\omega = 0.1$ are used. The iso-lines of $\phi = -3$ and $\phi = 3$ evidently show that the thickness is significantly different in various positions around the interface if no reinitialization is applied. Employing the reinitialization schemes results in uniform realization of the interface band, although reversing the velocity causes erroneous displacements to creep up and disturb the smooth shape of the interface.

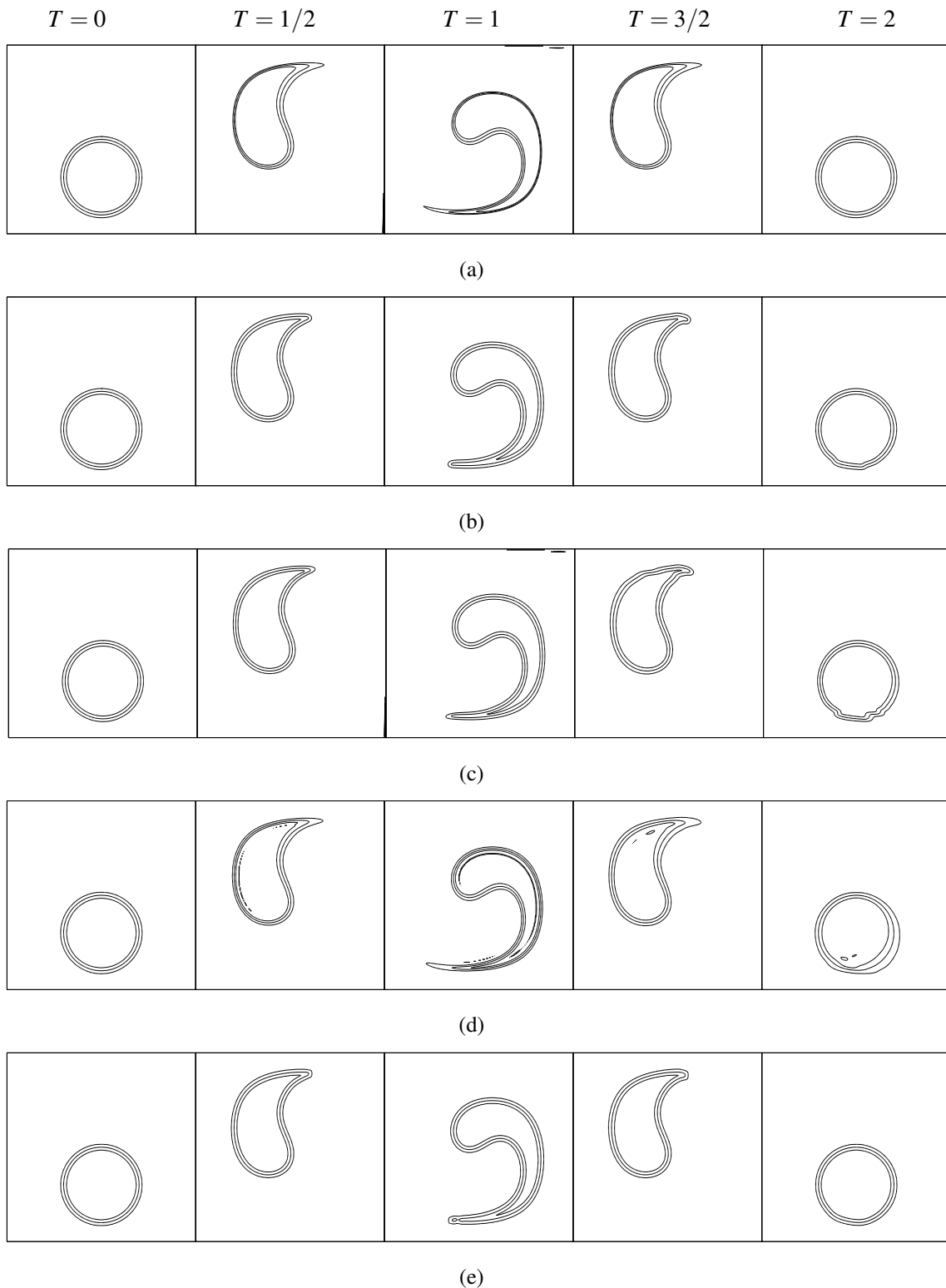


Figure 4.2: Temporal shapes of the interface for the reversed shear disk problem using (a) sharp LSM without reinitialization, (b) sharp LSM with brute force reinitialization, (c) sharp LSM with PDE-based reinitialization, (d) phase field LSM without reinitialization, (e) phase field LSM with $\eta = 0.004$ ($\varepsilon = 0.0175$).

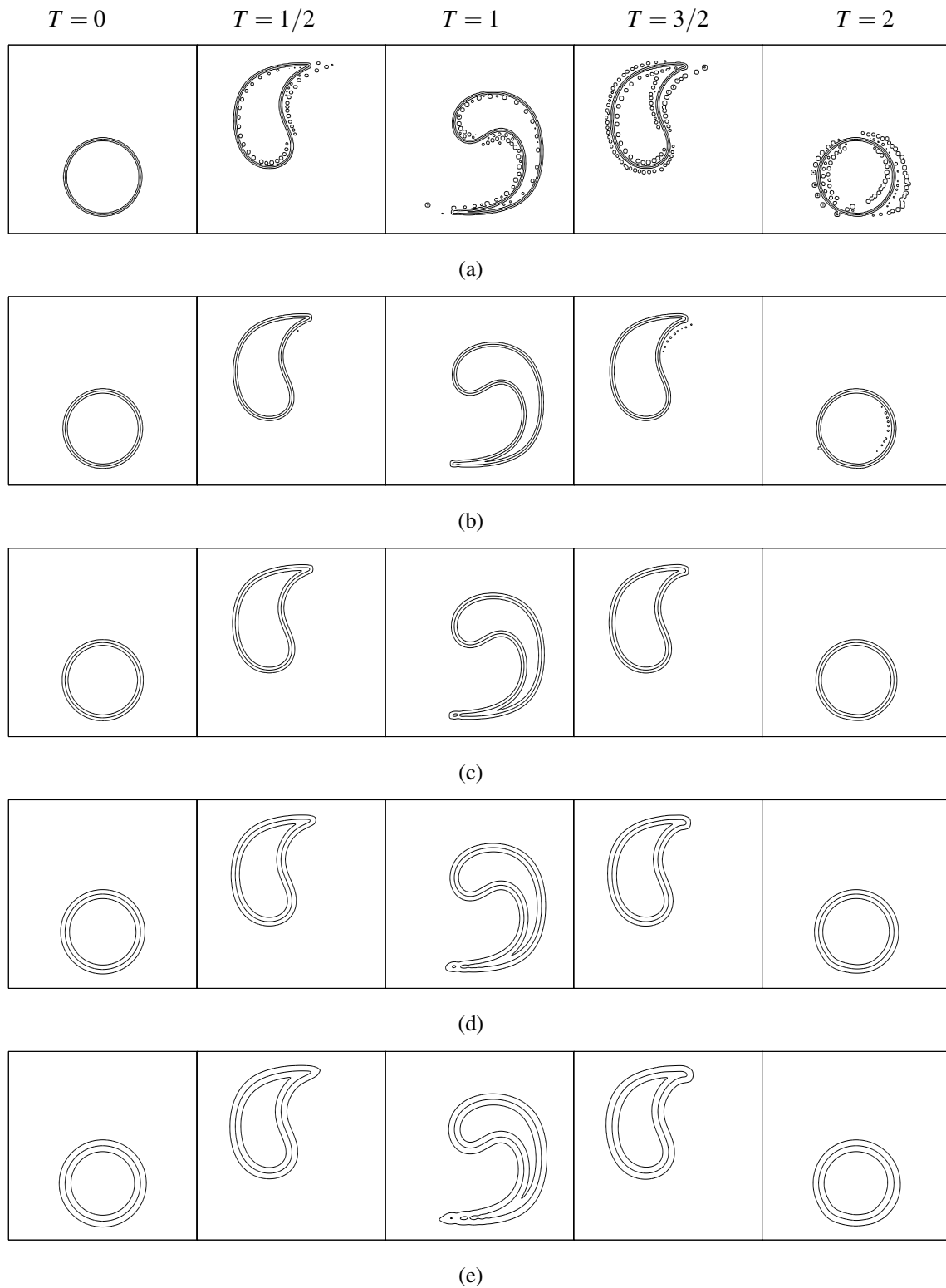


Figure 4.3: Temporal shape of the interface for the reversed shear disk problem using the phase field LSE with (a) $\eta = 0.002$, (b) $\eta = 0.003$, (c) $\eta = 0.004$, (d) $\eta = 0.006$, (e) $\eta = 0.008$.

Figure 4.2(d) and (e) present the same study for the phase-field LSM with and without the monolithic reinitialization. Here, the iso-lines of $\psi = 0.02$ and $\psi = 0.98$ are chosen to represent an interface band analogous to that of the sharp LSM. Despite using the 5th order WENO scheme, ignoring the monolithic reinitialization causes strong dispersion errors to degrade the solution and obviously prevents the phase-field LSM from recovering the initial interface band at $T = 2$. While the application of the monolithic reinitialization is very successful in preserving the interface thickness, the small imperfections in the final shape are due to the partial disintegration of the trailing edge at $T = 1$. Nevertheless, the disintegration could be delayed or vanished by using higher grid resolutions.

A noteworthy feature of the phase-field LSM is the dependence of the solution on the prescribed interface thickness ε , through the parameter η . In this sense, figure 4.3 shows that choosing a very small thickness induces such strong dispersion effect which the diffusion term could not rectify, while a very wide interface leads to excessive diffusion and early disintegration of the interface. This fact is also reflected quantitatively in table 4.1 for a grid size of $1/h = 200$ where the use of different values of thickness parameters η (accompanied by the corresponding thickness values ε) on the error norms is studied. It could be seen that taking $\varepsilon = 0.0175$, corresponding to 3.5 grid cells, gives the smallest relative error.

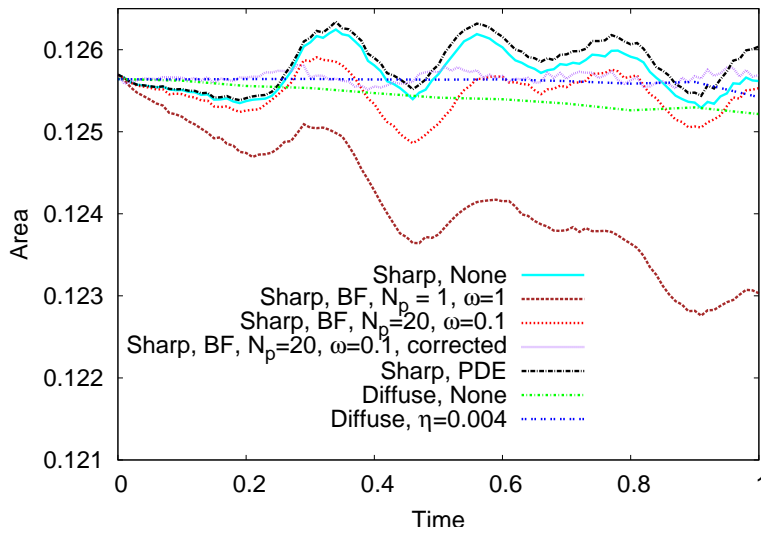


Figure 4.4: Evolution of the disk area under shear velocity field using different LSE implementations. BF=brute force reinitialization, PDE=PDE-base reinitialization, None= no reinitialization, Sharp= signed distance LSM, Diffuse= phase field LSM.

Table 4.1: Errors in the reversed shear disk problem using phase-field LSE with different values of η and $1/h = 200$.

η	ε	$\ e_1\ $	$\ e_2\ $	$\ e_\infty\ $
0.002	0.01	6.11e-3	3.04e-2	6.89e-1
0.003	0.0125	6.36e-4	5.02e-3	1.57e-1
0.004	0.0175	6.84e-4	5.68e-3	1.49e-1
0.006	0.025	1.11e-3	8.03e-3	1.82e-1
0.008	0.035	1.64e-3	9.82e-3	1.76e-1

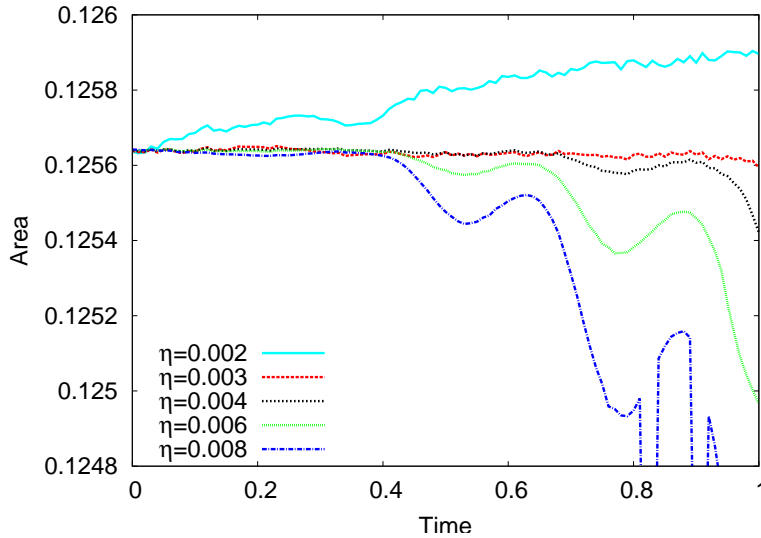


Figure 4.5: Evolution of the disk area under shear velocity field using phase field LSE with different values of η .

The effect of employing different schemes on preserving the initial mass is studied further in figure 4.4 for the reversed shear disk up to $T = 1.0$. The original sharp LSM shows large fluctuations in the temporal mass, while employing the reinitialization to establish the signed distance field seems to not have any positive influence on the mass conservation. It is only by adding the computationally expensive geometric mass corrections that the mass fluctuations are controlled. On the other hand, the phase-field LSM can remarkably preserve the mass and one may only discern decaying of the mass as the trailing edge of the distorted disk begins to disintegrate around $T = 0.8$.

Figure 4.5 also describes the effect of the interface thickness on the mass preserving behaviour of the phase field scheme with monolithic reinitialization. The discussions on the interface capturing quality could again explain the trend for the mass conservation. While the strong dispersion errors for $\eta = 0.002$ tend to overestimate the mass, increasing the thickness clearly adds too much diffusion, causing considerable mass losses.

Figures 4.6 and 4.7 indicate the qualitative results for the time evolution of the Zalesak's disk and the deforming disk problems, respectively. The sharp interface LSM with brute force reinitialization can best preserve the sharp corners of the notch in Zalesak's disk. The phase field LSM, however, tends to round the sharp corners to a very small extent which is an inevitable effect of the diffusion term of the phase field scheme. In the case of the deforming disk, the resolution is increased to $L_0^* = 1/h = 400$ so as to ensure that the very thin filaments generated around $T = 0.5$ are properly captured. The evolution of the disk shape exhibits the unique performance of the phase field LSM in recovering the original shape despite the intense deformation applied. While the remaining kinks produced by the reversed velocity field are evident in the final shape for the brute force reinitialization, the PDE-based reinitialization fails to build a uniform signed distance field around the interface due to the stringent deformation rates.

Tables 4.2 through 4.4 present the relative error norms on different grid levels using the sharp and phase field LS models. For the phase-field LSE, the thickness parameter of $\eta = 0.006$ is considered for all grid levels which implies solving the same physical problem regardless of the resolution. Note that using a smaller η is avoided so as to enforce at least 1 grid cell of thickness at the coarsest grid of $1/h = 50$.

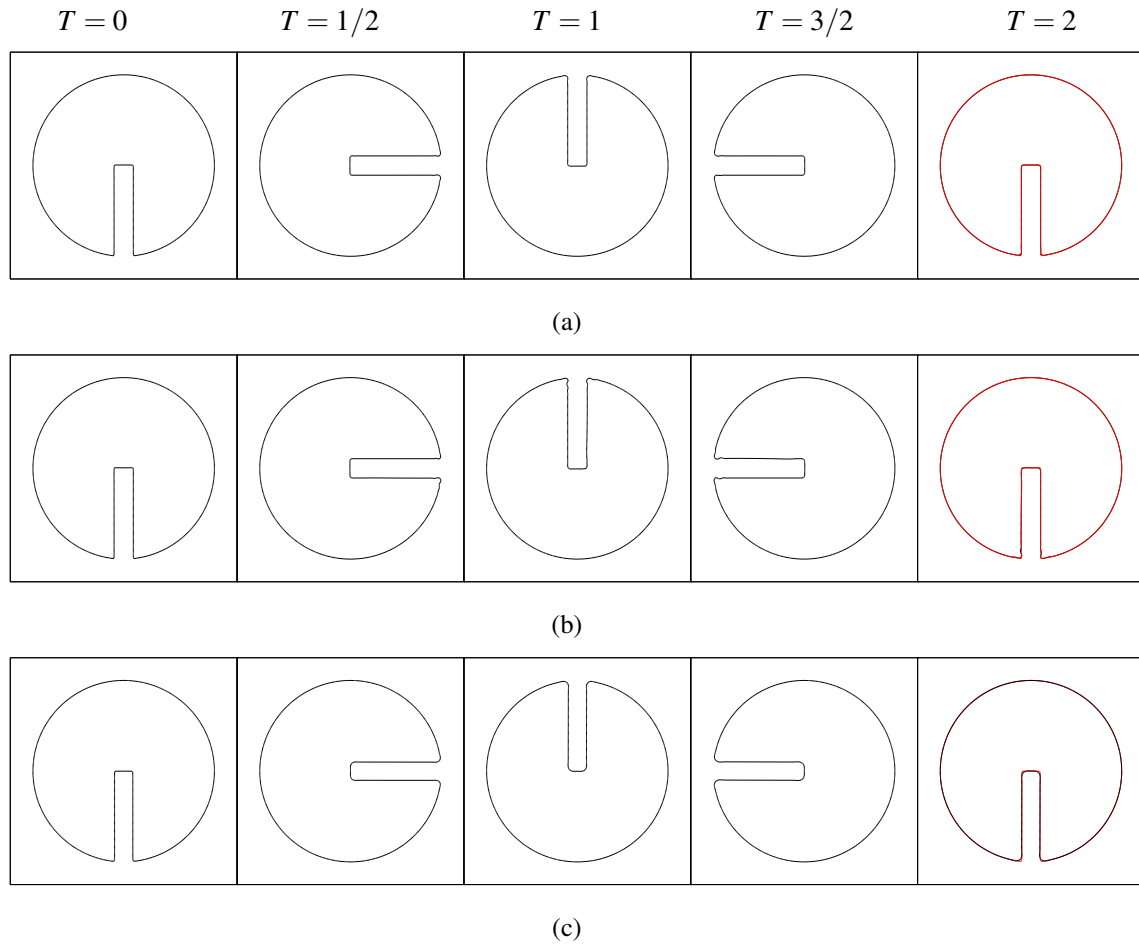


Figure 4.6: Temporal shape of the interface for the Zalesak's disk problem using (a) sharp LSE with brute force reinitialization, (b) sharp LSE with PDE-based reinitialization, (c) phase field LSE with $\eta = 0.004$. The solid red lines indicate the exact solution.

Table 4.2 contains the errors for the linear translation of a disk which requires no reinitialization when sharp LSE is used. The obtained rates of convergence (ROC) all confirm the second order accuracy dictated by the second order RK time integration. The large disproportional error values given by the coarse grid in the phase field LSM emphasizes the stronger dependence of the accuracy on the grid resolution in this method. Yet, the smoothness of ϕ in this particular problem causes the sharp LSM to give almost an order of magnitude smaller error values as compared to the phase field LSM even at high grid levels.

The errors pertained to the rigid rotating disk as well as the reversed shear disk problems are collected in tables 4.3 and 4.4, where the reinitialization is turned on for the sharp LSM so as to preserve the signed distance field. Although the second order convergence rate is recovered for the smooth problem of the rigid rotating disk, the convergence order drops below 2 for the case of the reversed shear disk. This is mainly due to the large displacement of the greatly deformed interface during reinitialization in the sharp LSM and the partial disintegration of the interface in the phase field LSM which is not fully reversible. Such large deformations in fact disturb the smooth LS field which in turn affects the accuracy of the WENO scheme. As also pointed out in [91], although the WENO scheme tends to remain highly stable in such situations, its high accuracy may decay to as low as first order in non-smooth regions of the flow. This situation causes the sharp LSM to lose its advantage of producing lower error magnitudes as seen by the reduced differences between the

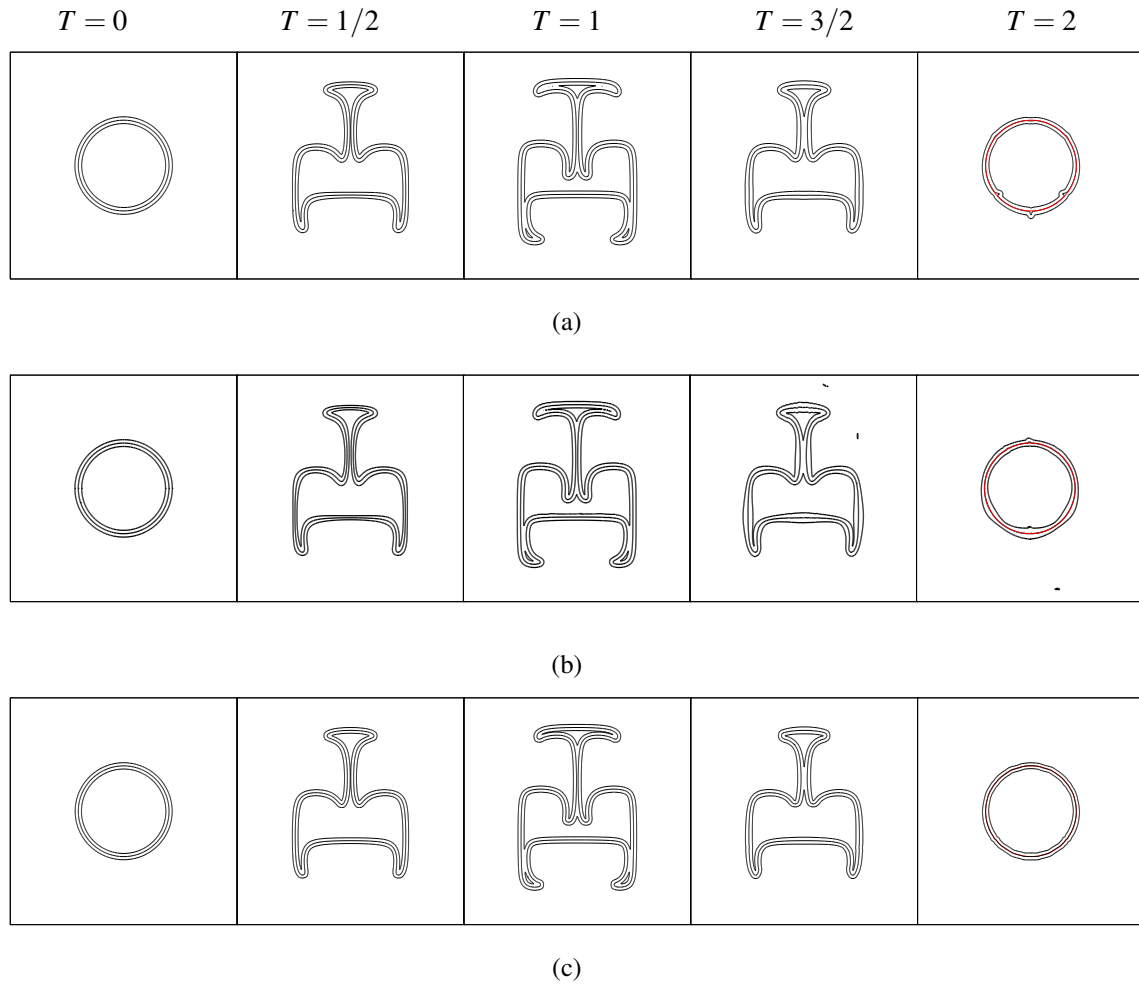


Figure 4.7: Temporal shape of the interface for the deforming disk problem using (a) Sharp LSE with brute force reinitialization, (b) Sharp LSE with PDE-based reinitialization, (c) phase field LSE with $\eta = 0.004$. The solid red line indicates the exact solution

errors of sharp and phase field LSM in table 4.4. The error levels also indicate that unlike the rigid rotating disk problem, the accuracy of the PDE-based reinitialization is strongly influenced by the poor quality of the pre-reinitialization values of ϕ .

Since the eventual coupled scheme runs on GPGPUs, the computation times on these machines for the two problems of rigid rotating disk and reversed shear disk are also collected in tables 4.3 and 4.4, respectively. Note that all the GPGPU implementations are carried out on a Tesla K20X GPGPU machine. The most noticeable fact is that the PDE-based reinitialization is associated with the highest GPU times since it involves an iterative process in each outer iteration. The cost of the method also scales up non-linearly since the $2m$ number of inner iterations scales with the resolution as $m = \varepsilon/h$. Additionally, the dependence of both reinitialization schemes on the length of the interface causes the GPU time to escalate in the reversed shear problem where the perimeter of the disk increases under deformation. In contrast, the cost of the phase field LSM scales linearly and is not affected by the shape of the interface, suggesting relatively lower run times for the reversed shear disk as compared to the sharp interface LSM implementations.

Table 4.2: Errors in the linear translation of a disk problem

$1/h$	$\ e_1\ $	ROC_1	$\ e_2\ $	ROC_2	$\ e_\infty\ $	ROC_∞
Sharp LSE, No reinitialization						
50	5.68e-4		1.88e-3		8.16e-4	
100	1.45e-4	1.96	4.79e-4	1.97	2.04e-4	2
200	3.74e-5	1.96	1.22e-4	1.97	5.19e-5	1.97
400	9.03e-6	2.05	2.95e-5	2.05	1.24e-5	2.06
Phase-field LSE, $\eta = 0.006, \varepsilon \approx 0.03$						
50	4.09e-2		1.13e-1		8.54e-1	
100	2.58e-3	3.98	1.05e-2	3.43	1.02e-1	3.07
200	2.45e-4	3.4	1.00e-3	3.39	1.11e-2	3.2
400	4.51e-5	2.44	1.61e-4	2.64	1.58e-3	2.81

Table 4.3: Errors and run times for the rigid rotating disk problem. GPU=GPGPU time (seconds)

$1/h$	GPU	$\ e_1\ $	ROC_1	$\ e_2\ $	ROC_2	$\ e_\infty\ $	ROC_∞
Sharp LSE, PDE reinitialization							
50	0.6	1.09e-3		9.13e-3		1.15e-2	
100	2	4.33e-5	4.66	2.26e-4	5.34	3.51e-4	5.03
200	10	9.12e-6	2.25	3.91e-5	2.53	1.96e-5	4.16
400	90	2.30e-6	1.99	9.56e-5	2.03	5.43e-6	1.85
Sharp LSE, Brute force reinitialization							
50	0.7	1.35e-3		5.47e-3		8.59e-3	
100	2	2.92e-4	2.20	1.22e-3	2.16	2.40e-3	1.84
200	7	8.30e-5	1.82	3.34e-4	1.87	2.47e-4	3.28
400	28	2.99e-5	1.47	1.21e-4	1.47	2.38e-5	3.38
Phase-field LSE, $\eta = 0.006, \varepsilon \approx 0.03$							
50	0.5	1.89e-2		5.42e-2		5.55e-1	
100	1.5	5.62e-4	5.07	2.16e-3	4.65	1.92e-1	4.86
200	6	1.29e-4	2.12	5.07e-4	2.09	4.81e-2	1.99
400	31	3.22e-5	2.00	1.26e-4	2.00	1.18e-3	2.03

Table 4.4: Errors and run times in the reversed shear disk problem. GPU=GPGPU time (seconds)

$1/h$	GPU	$\ e_1\ $	ROC_1	$\ e_2\ $	ROC_2	$\ e_\infty\ $	ROC_∞
Sharp LSE, PDE reinitialization							
50	0.7	3.89e-2		1.73e-1		5.67e-1	
100	3	6.60e-3	2.56	4.46e-2	1.96	3.24e-1	0.81
200	16	2.40e-3	1.46	1.48e-2	1.59	2.32e-1	0.48
400	117	1.62e-3	0.56	1.26e-2	0.24	2.84e-1	-0.29
Sharp LSE, Brute force reinitialization							
50	1	1.44e-2		5.68e-2		8.04e-2	
100	3	5.98e-3	1.27	2.50e-2	1.18	4.07e-2	0.98
200	9	2.42e-3	1.31	9.84e-3	1.35	1.71e-2	1.25
400	47	9.70e-4	1.32	3.79e-3	1.38	4.65e-4	5.20
Phase-field LSE, $\eta = 0.006, \varepsilon \approx 0.03$							
50	0.5	1.44e-2		3.70e-2		4.08e-1	
100	1.5	1.74e-3	3.05	1.09e-2	1.77	1.91e-1	1.10
200	6	1.11e-3	0.64	8.03e-3	0.44	1.81e-1	0.07
400	31	8.34e-4	0.42	6.04e-3	0.41	1.51e-1	0.27

Coupled lattice Boltzmann-level set methods

The problems already discussed with the two-LBE pressure evolution schemes have motivated developing a new generation of LB-based multiphase models which replace the LB or finite difference solution of the highly non-linear Cahn-Hilliard equation with conventional interface capturing schemes in a similar way as in the coupled multiphase solvers used in the Navier-Stokes community. Early attempts in this regard were focused on the use of volume of fluid (VOF) models in conjunction with a single-phase LB solver for the simulation of free surface flows [68]. In principle, free surface flows assume to have a considerably lighter fluid inside the gas phase such that the mass, momentum and shear stresses could be neglected outside of the liquid part. The interface thus needs to be captured in time and is treated as a moving boundary on which the surface tension forces are applied. This methodology has been pursued by Köner et al. [68] and Janssen et al. [60] with recent improvements on the realisation of the surface tension forces in [61] and the resulting coupled algorithms have been applied for the simulation of complex interface changes, e. g., braking dam, bursting of bubbles at the free surface and modelling of foam production and movement [10]. In a similar way, Yu et al. [138] used a particle LSM to track the interface in their free surface model with application to merging bubbles and breaking dams. However, a prominent deficit with free surface methods is the fact that the accuracy drops if the density ratio falls below 1000 since it does not take the dynamics of the gas phase into account. Hence, application of the model could not be extended to general multiphase flows.

Inspired by the free surface VOF-based schemes, Thommes et al. [123] and Becker et al. [20] suggested to solve the flow inside both phases separately using single-phase LBEs and then treat the interface as a curved boundary, on which a modified bounce-back scheme is applied to impose the surface tension and velocity boundary conditions. The interface itself is captured in a coupled fashion via finite difference solution of a LSE to realise a sharp interface. Finally each phase has to go through a refilling step so as to advance or retreat its mass, i. e., distribution functions, in accordance to their new position with respect to the advected interface. Despite the sophisticated boundary conditions and the advanced numerical tools employed, this two-fluid coupled scheme faces a fundamental issue when applied to high density and viscosity ratio multiphase problems, e. g., air-water system; in fact the solution of two fluids with different viscosities in the same computational domain implies having different lattices on each side since the lattice velocities c_α would be different depending on the fluid viscosity. This means that the data structures have to be redesigned in each iteration or two overlapping lattices have to be defined on the entire domain which in turn requires performing interpolations to transfer the interface data from the LS grid to at least one of the lattices. Consequently, one has to either accept restrictions of very low viscosity ratios or impose additional complexity and computational overhead due to using different lattices for each phase.

The disadvantages associated with the above two approaches motivate the need to design coupled LB schemes in which the limitations on density and viscosity ratios are in the same level as

in two-LBE models and both gas and liquid phases are solved, while the computational effort and complexity are reduced and parallel scalability of the solution is preserved. Consequently, in this chapter we propose two coupled models in which the flow field is solved by the LBM which is then coupled with LS interface capturing PDEs. As such, the first coupled model is a modified version of the one-fluid approach initially proposed by Mehravaran and Hannani [85, 86] in which one solves for a single virtual fluid and then moves the effect of variable density and viscosity to an extended force term including a sharp definition of the forces via the CSF forcing method. The formulation then allows coupling with a sharp interface capturing engine through a distance-based LS function. The second coupled model is based on the pressure evolution, diffuse interface LB scheme, already described in chapter 3, where the chemical potential formulation for the surface tension force is replaced with a diffuse CSF description of the forces and the interface is captured via solving for a phase-field level set function. For both of the coupled schemes consistent time integrations and force discretization are exploited to arrive at the final multiphase LBE, while the numerical solution of sharp and phase-field LSEs are carried out as described in chapter 4.

5.1. The one-fluid coupled LB-LS model

As already discussed in chapter 3, the main source of numerical errors in early multiphase LB models where the large truncation errors emerging from the sharp jumps of density and thus the thermodynamic pressure which leads to instability of the solutions for high density ratios and motivated the use of more thermodynamically consistent EOS and improved numerical approximation of the interaction forces, and above all, led to the invention of the pressure evolution LB models.

As an alternative pathway, one could skip these difficulties with modelling the interaction forces and start from the incompressible two-phase NS equations coupled with LS interface capturing. The constructed coupled multiphase NSE can then be recast into a suitable form, called one-fluid formulations here, to be mapped into LBM's kinetic framework as described below.

5.1.1. The one-fluid LB formulation

Following the discussions in chapter 2, a popular macroscopic approach for the solution of multiphase flow systems consists of solving the Navier-Stokes equations along with the surface tension boundary conditions realised using the CSF formulation as local smeared out forces acting on the interface Γ

$$\rho(\phi)\partial_t\mathbf{u} + \rho(\phi)\mathbf{u} \cdot \nabla\mathbf{u} + \nabla P - \nabla \cdot (\mu(\phi)(\nabla\mathbf{u} + \nabla\mathbf{u}^T)) = -\sigma\kappa(\phi)\delta_\epsilon(\phi)\mathbf{n}(\phi) \quad (5.1.1)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (5.1.2)$$

These equations need to be equipped with an interface capturing module to give the fluid properties and construct the surface tension forces. Sussman et al. [120] proposed to couple equation (5.1.1) with the LSE for the advection of the signed distance LS function ϕ . On the other hand, the approach to recover the single-phase NSE by the LBE could be repeated so as to also realise a two-phase LBE formulation based on the coupled two-phase NSE. However, a direct, yet naive mapping of the CSF formulation for the NSE to the LBM framework leads to severe numerical instabilities. Let us first rewrite the nearly incompressible NSE recovered by the Chapman-Enskog expansion of single-phase LBE

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla(\rho c_s^2 \mathbf{I} + \rho\mathbf{u}\mathbf{u}) - \frac{\lambda}{3} (\nabla \cdot [\nabla(\rho\mathbf{u}) + \nabla(\rho\mathbf{u})^T]) = -\sigma\kappa(\phi)\mathbf{n}(\phi)\delta_\epsilon(\Gamma, \phi) + F_v. \quad (5.1.3)$$

In fact, in a two-phase system, the simple linear EOS in the single-phase flow breaks down for a variable density since the term $\nabla\rho c_s^2$ represents only the dynamic pressure gradient and can not take into account the non-monotonic changes in the thermodynamic pressure over the interface. In order to rectify this issue with the variable density, the entire NS equation (5.1.1) can be divided by $\rho(\phi)$ to have [86]

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{\nabla P}{\rho(\phi)} - \frac{\nabla \cdot (\mu(\phi)(\nabla \mathbf{u} + \nabla \mathbf{u}^T))}{\rho(\phi)} = -\frac{\sigma\kappa(\phi)\mathbf{n}(\phi)\delta_\epsilon(\phi)}{\rho(\phi)}. \quad (5.1.4)$$

By introducing a virtual density $\bar{\rho} = 1$ to the convective terms, adding ∇p terms to the both sides and expanding the viscous term as

$$\frac{\nabla \cdot (\mu(\phi)(\nabla \mathbf{u} + \nabla \mathbf{u}^T))}{\rho(\phi)} = \frac{\mu(\phi)\nabla \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T)}{\rho(\phi)} + \frac{\nabla\mu(\phi) \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T)}{\rho(\phi)} \quad (5.1.5)$$

one ends up with a new form of the momentum equation

$$\bar{\rho}(\phi)\partial_t \mathbf{u} + \bar{\rho}(\phi)\mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \bar{\mu}(\phi)\nabla \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) = \bar{\mathbf{F}} \quad (5.1.6)$$

where $\bar{\mu} = \mu(\phi)/\rho(\phi)$ is the virtual viscosity. The modified force term $\bar{\mathbf{F}}$ is obtained as

$$\bar{\mathbf{F}} = -\frac{\sigma\kappa(\phi)\mathbf{n}(\phi)\delta_\epsilon(\phi)}{\rho(\phi)} + \frac{\nabla\mu(\phi) \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T)}{\rho(\phi)} + \nabla p \left(1 - \frac{1}{\rho(\phi)}\right). \quad (5.1.7)$$

Using the sharp interface definitions of $H(\phi)$ and $\delta_\epsilon(\phi)$ in equation (4.2.4) and (4.2.5) gives $\nabla\mu(\phi) = (\mu_l - \mu_g)\delta_\epsilon(\phi)\nabla\phi$ and hence

$$\bar{\mathbf{F}} = -\frac{\sigma\kappa(\phi)\mathbf{n}(\phi)\delta_\epsilon(\phi)}{\rho(\phi)} + \frac{(\mu_l - \mu_g)\delta_\epsilon(\phi)}{\rho(\phi)} [\nabla\phi \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] + \nabla p \left(1 - \frac{1}{\rho(\phi)}\right). \quad (5.1.8)$$

It is observed that $\bar{\mathbf{F}}$ comes with two new terms; the first one $\bar{\mathbf{F}}$ accounts for the jump in the viscous force term, while the second one adds the effect of the dynamic pressure gradient across the interface through the ∇p term. Eventually, the discrete Boltzmann equation along with the new extended force term based on the method of He et al. [53] reads as

$$\frac{\partial f_\alpha}{\partial t} + c_\alpha \nabla f_\alpha = -\Lambda(f_\alpha(x, t) - f_\alpha^{eq}(x, t)) - \frac{(c_\alpha - \mathbf{u}) \cdot \bar{\mathbf{F}}}{\bar{\rho}c_s^2} f_\alpha^{eq}(x, t) \quad (5.1.9)$$

where Λ is the generalized relaxation matrix and could be chosen to be of single relaxation time (SRT) or multiple relaxation time (MRT) type. The above reformulation means that the virtual density $\bar{\rho}$ is decoupled from the variable physical density $\rho(\phi)$. As this virtual density is assumed to be only nearly incompressible, the LBM's EOS is expected to be valid to obtain the pressure as

$$p = \bar{\rho}c_s^2 = \sum_\alpha f_\alpha. \quad (5.1.10)$$

Moreover, the fluid velocity could also be recovered based on the virtual density $\bar{\rho}$

$$\bar{\rho}\mathbf{u} = \sum_\alpha c_\alpha f_\alpha \quad (5.1.11)$$

5.1.2. Space discretization for one-fluid LBE

Thanks to the perfect shift principle of the LBE, the convective term of the LBE is already discretized through splitting the solution to collision and streaming steps and the only term which has to be discretized in space is the source term appearing on the right hand side of equation (5.1.9). Despite many discussions in LB community on time integration of the LBE and implementation of the forcing terms, little work has been done with regards to the spatial discretization of these terms. It was not until Lee and Lin [75] showed in their 2005 paper, that the aim to reach accurate solutions at high density ratios is not achieved without employing a numerically consistent space discretization for the force terms. It was revealed that even the stable pressure evolution method will produce non-physical results with high level of anisotropy if the force terms are discretized in a naive way. Although the present scheme differs from the pressure evolution LBE, it is discussed in the following how critical the force discretization could be at high density ratios.

Let us define the source term on the right hand side of equation (5.1.9) as

$$S_\alpha = \frac{(c_\alpha - \mathbf{u}) \cdot \bar{\mathbf{F}}}{\bar{\rho} c_s^2} f_\alpha^{eq}. \quad (5.1.12)$$

Inserting $\bar{\mathbf{F}}$ into S_α gives

$$S_\alpha = (c_\alpha - \mathbf{u}) \frac{f_\alpha^{eq}}{\bar{\rho} c_s^2} \cdot \left[-\frac{\sigma \kappa(\phi) \mathbf{n}(\phi) \delta_\epsilon(\phi)}{\rho(\phi)} + \frac{(\mu_l - \mu_g) \delta_\epsilon(\phi)}{\rho(\phi)} [\nabla \phi \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] + \nabla p \left(1 - \frac{1}{\rho(\phi)} \right) \right]. \quad (5.1.13)$$

which upon expanding the normal vector $\mathbf{n}(\phi)$ could be split into three parts

$$\begin{aligned} S_\alpha^\sigma &= \frac{f_\alpha^{eq}}{\bar{\rho} c_s^2} \frac{\sigma \kappa(\phi) \delta_\epsilon(\phi)}{\rho(\phi) |\nabla \phi|} (c_\alpha - \mathbf{u}) \cdot \nabla \phi \\ S_\alpha^\mu &= \frac{f_\alpha^{eq}}{\bar{\rho} c_s^2} \frac{(\mu_l - \mu_g) \delta_\epsilon(\phi)}{\rho(\phi)} (c_\alpha - \mathbf{u}) \cdot [\nabla \phi \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] \\ S_\alpha^p &= \frac{f_\alpha^{eq}}{\bar{\rho} c_s^2} \left(1 - \frac{1}{\rho(\phi)} \right) (c_\alpha - \mathbf{u}) \cdot \nabla p \end{aligned} \quad (5.1.14)$$

where S_α^σ , S_α^μ and S_α^p denote the surface tension, viscous and pressure source terms, respectively. In order to discretize the gradients applied to the velocity and the level set function in S_α^σ and S_α^μ , one can use the central differencing without any harm to the numerical stability since the level set function is smooth and the velocities do not jump across the interface. Unlike the pressure evolution LBE (3.2.20), in the one-fluid formulation there is no gradient of density in the force term. However, care must be taken that due to the EOS we have $\nabla p \approx \mathcal{O}(\nabla f)$ and hence the term $(c_\alpha \cdot \nabla p)$ in S_α^p contributes to the convective term on the left hand side of the LBE in the $-c_\alpha$ direction. This observation firstly suggests that $(c_\alpha \cdot \nabla p)$ has to be measured along the characteristic directions, i. e., the lattice velocities, and secondly reveals the need for some kind of stabilization of the convective contribution. A central differencing along characteristics is then defined as

$$c_\alpha \cdot \nabla^c p(x) = \frac{p(x + c_\alpha \Delta t) - p(x - c_\alpha \Delta t)}{2}. \quad (5.1.15)$$

While this second order directional differencing brings dispersion errors and thus numerical oscillations, a first order upwinding adds strong numerical diffusion and degrades the spacial discretization accuracy to first order. A third choice would be to use second order upwind differencing in the $-c_\alpha$ direction which is obtained by Taylor series expansion of $(c_\alpha \cdot \nabla p)^c$ up to second order in space around the point $x + c_\alpha \Delta t$ [76]:

$$c_\alpha \cdot \nabla^u p(x) \cong c_\alpha \cdot \nabla^c p(x + c_\alpha \Delta t) + c_\alpha \cdot \nabla^2 p(x + c_\alpha \Delta t) \quad (5.1.16)$$

where the second derivative reads as

$$c_\alpha \cdot \nabla^2 p(x + c_\alpha \Delta t) = c_\alpha \cdot \nabla \left[c_\alpha \cdot \nabla^c p(x + \frac{3}{2} c_\alpha \Delta t) - c_\alpha \cdot \nabla^c p(x + \frac{1}{2} c_\alpha \Delta t) \right] = p(x + 2c_\alpha \Delta t) - 2p(x + c_\alpha \Delta t) + p(x). \quad (5.1.17)$$

Substitution gives the second order upwind differencing along the characteristic

$$(c_\alpha \cdot \nabla p)^u = \frac{-p(x + 2c_\alpha \Delta t) + 4p(x + c_\alpha \Delta t) - 3p(x)}{2}. \quad (5.1.18)$$

A combination of central and upwind differencing would be to take an average of these two

$$(c_\alpha \cdot \nabla p)^{ave} = \frac{(c_\alpha \cdot \nabla p)^u + (c_\alpha \cdot \nabla p)^c}{2}. \quad (5.1.19)$$

Having these directional derivatives in hand, the rest of the non-directional derivatives of p as needed for $\mathbf{u} \cdot \nabla p$ can be obtained by integrating the central directional derivatives over all lattice directions as

$$\begin{aligned} \nabla_x p &= \sum_{\alpha=0}^n w_\alpha \frac{c_{\alpha,x} \cdot (c_\alpha \cdot \nabla p)^c}{c_s^2 \Delta t} \\ \nabla_y p &= \sum_{\alpha=0}^n w_\alpha \frac{c_{\alpha,y} \cdot (c_\alpha \cdot \nabla p)^c}{c_s^2 \Delta t} \\ \nabla_z p &= \sum_{\alpha=0}^n w_\alpha \frac{c_{\alpha,z} \cdot (c_\alpha \cdot \nabla p)^c}{c_s^2 \Delta t}. \end{aligned} \quad (5.1.20)$$

Nevertheless, one has to note that a naive central differencing along the main coordinate directions for $(c_\alpha \cdot \nabla p)$ as pursued in the work of Mehravaran [85] is believed to impose large numerical errors in high density ratio systems. The errors partly come from the fact that the 4-point spacial support of the main coordinate directions does not provide the required isotropy in LBM. In addition, ignoring the contribution of $c_\alpha \cdot \nabla p$ to the convective terms causes the instabilities to magnify for very low gas densities, i. e., high density ratios, since they are scaled by $1/\rho(\phi)$.

5.1.3. Time integration for one-fluid LBE

In a similar way to the time integration procedure of the DBE in chapter 3, a trapezoidal rule is applied to equation (5.1.9) to get

$$\begin{aligned} f_k(x + c_\alpha \Delta t, t + \Delta t) - f_\alpha(x, t) &= -\frac{\Delta t}{2} \Lambda(f_\alpha(x, t) - f_\alpha^{eq}(x, t)) + \\ &\quad -\frac{\Delta t}{2} \Lambda(f_\alpha(x + c_\alpha \Delta t, t + \Delta t) - f_\alpha^{eq}(x + c_\alpha \Delta t, t + \Delta t)) \\ &\quad + \frac{\Delta t}{2} S_\alpha(x, t) + \frac{\Delta t}{2} S_\alpha(x + c_\alpha \Delta t, t + \Delta t) \end{aligned} \quad (5.1.21)$$

To render the equation explicit, the conventional change of variables to \bar{f} and \bar{f}^{eq} is carried out:

$$\begin{aligned}\bar{f}_\alpha &= f_\alpha + \frac{1}{2}\Lambda(f_\alpha - f_\alpha^{eq}) - \frac{1}{2}S_\alpha \\ \bar{f}_\alpha^{eq} &= f_\alpha^{eq} - \frac{1}{2}S_\alpha\end{aligned}\quad (5.1.22)$$

Now, considering that S_α^p has to be discretized in t and $t + \Delta t$, one could eventually adopt three combinations for spatial discretization of the S_α^p term

- use central differencing along the main coordinate directions for both t and $t + \Delta t$.
- use directional central differencing ($c_\alpha \cdot \nabla^c p$) for $t + \Delta t$ and second order directional upwinding ($c_\alpha \cdot \nabla^u p$) for t .
- use directional averaged unwinding ($c_\alpha \cdot \nabla^{ave} p$) for both t and $t + \Delta t$.

The first choice followed by Mehravaran [85] will be only pursued to show the numerical implications later in the benchmarking studies. Picking up the second choice, nevertheless, yields:

$$\begin{aligned}\bar{f}_\alpha(x + c_\alpha \Delta t, t + \Delta t) &= f_\alpha(x, t) \frac{\Lambda}{2} (f_\alpha(x, t) - f_\alpha^{eq}(x, t)) + \frac{1}{2}S_\alpha^\sigma + \frac{1}{2}S_\alpha^\mu + \\ &\quad \frac{1}{2}(c_\alpha - \mathbf{u}) \frac{f_\alpha^{eq}}{\bar{\rho}c_s^2} \cdot \left[\nabla^u p \left(1 - \frac{1}{\rho(\phi)} \right) \right]\end{aligned}\quad (5.1.23)$$

where the remaining f_α can be expanded as

$$f_\alpha(\mathbf{I} + \frac{1}{2}\Lambda) = \bar{f}_\alpha + \frac{1}{2}\Lambda f_\alpha^{eq} + \frac{1}{2}(S_\alpha^\sigma + S_\alpha^\mu) + \frac{1}{2}(c_\alpha - \mathbf{u}) \frac{f_\alpha^{eq}}{\bar{\rho}c_s^2} \cdot \left[\nabla^c p \left(1 - \frac{1}{\rho(\phi)} \right) \right] \quad (5.1.24)$$

and f_α^{eq} can be replaced with the new equilibrium function from equation (5.1.22) where all the derivatives are obtained using central differencing. Substitution gives the following explicit LBE for the evolution of the virtual density

$$\begin{aligned}\bar{f}_\alpha(x + c_\alpha \Delta t, t + \Delta t) - \bar{f}_\alpha(x, t) &= (\Lambda + 2\mathbf{I})(\bar{f}_\alpha(x, t) - \bar{f}_\alpha^{eq}(x, t)) + S_\alpha^\sigma + S_\alpha^\mu + \\ &\quad (c_\alpha - \mathbf{u}) \frac{f_\alpha^{eq}}{\bar{\rho}c_s^2} \cdot \left[\nabla^{ave} p \left(1 - \frac{1}{\rho(\phi)} \right) \right]\end{aligned}\quad (5.1.25)$$

where the combination of the upwind and central terms for pressure gradients has led to the averaged directional derivative. The fluid pressure and velocity could now be recovered using \bar{f}_α based on equations (5.1.11) and (5.1.22). For the pressure

$$\begin{aligned}p = \bar{\rho}c_s^2 = c_s^2 \sum_\alpha f_\alpha &= c_s^2 \sum_\alpha \left[\bar{f}_\alpha + (c_\alpha - \mathbf{u}) \frac{f_\alpha^{eq}}{\bar{\rho}c_s^2} \cdot \bar{\mathbf{F}} \right] = \\ &= c_s^2 \sum_\alpha \bar{f}_\alpha + \sum_\alpha \left[(c_\alpha - \mathbf{u}) \frac{f_\alpha^{eq}}{\bar{\rho}} \cdot \bar{\mathbf{F}} \right]\end{aligned}\quad (5.1.26)$$

where the last sum could be expanded as

$$\sum_\alpha \left[(c_\alpha - \mathbf{u}) \frac{f_\alpha^{eq}}{\bar{\rho}c_s^2} \cdot \bar{\mathbf{F}} \right] = \bar{F}_i \left[\sum_\alpha c_{\alpha i} f_\alpha^{eq} + \sum_\alpha f_\alpha^{eq} u_i \right]. \quad (5.1.27)$$

By definition, both sums are equal to $\bar{\rho}u_i$ and thus cancel each other. Consequently, the pressure is recovered as

$$p = c_s^2 \sum_{\alpha} \bar{f}_{\alpha}. \quad (5.1.28)$$

As for the velocity, one starts with equation (5.1.11)

$$\begin{aligned} \bar{\rho} \mathbf{u} = \sum_{\alpha} c_{\alpha} f_{\alpha} = \sum_{\alpha} \left[\bar{f}_{\alpha} c_{\alpha} + c_{\alpha} (c_{\alpha} - \mathbf{u}) \frac{f_{\alpha}^{eq}}{\bar{\rho} c_s^2} \cdot \bar{\mathbf{F}} \right] \\ \sum_{\alpha} \bar{f}_{\alpha} c_{\alpha} + \sum_{\alpha} \left[c_{\alpha} (c_{\alpha} - \mathbf{u}) \frac{f_{\alpha}^{eq}}{\bar{\rho} c_s^2} \cdot \bar{\mathbf{F}} \right] \end{aligned} \quad (5.1.29)$$

and the second sum is expanded as

$$\sum_{\alpha} \left[c_{\alpha} (c_{\alpha} - \mathbf{u}) \frac{f_{\alpha}^{eq}}{\bar{\rho} c_s^2} \cdot \bar{\mathbf{F}} \right] = \bar{F}_i \left[\sum_{\alpha} c_{\alpha i} c_{\alpha i} f_{\alpha}^{eq} + u_i \sum_{\alpha} f_{\alpha}^{eq} c_{\alpha i} \right] \quad (5.1.30)$$

where $\sum_{\alpha} c_{\alpha i} c_{\alpha i} f_{\alpha}^{eq} = \bar{\rho} c_s^2$. The second sum, however, gives $\bar{\rho} \mathbf{u}^2$ which is eventually multiplied by $1/c_s^2$ in equation (5.1.30) to give $Ma^2 \approx 0$, by the assumption of very low Mach numbers in the LB hydrodynamics. Consequently, one is left with the first sum to have

$$\bar{\rho} \mathbf{u} = \sum_{\alpha} c_{\alpha} \bar{f}_{\alpha} + \frac{1}{2} \bar{\mathbf{F}} \quad (5.1.31)$$

Now, following the third discretization choice means replacing both the directional upwind differencing in (5.1.23) and the directional central differencing in equation (5.1.24) with the averaged differencing. This will end up in an identical form of the discretized LBE as in equation (5.1.25) except that all the derivatives for calculation of pressure and velocity in equations (5.1.28) and (5.1.31) are obtained using the averaged differencing.

5.2. Pressure evolution coupled LB-LS method

The pressure evolution coupled LB-LS model closely follows the diffuse interface methodology of the conventional pressure evolution scheme described in chapter 3, except for two major distinctions:

1. The high order definition of the surface tension forces based on the chemical potential μ_c is replaced with the diffuse CSF definition of these forces.
2. The interface is captured via solving phase-field LSE for the mass conserving phase-field LS function ψ , instead of the kinetic solution of the highly non-linear Cahn-Hilliard equation.

In fact the new strategy has some major advantages; firstly, the high order derivatives in the calculation of the surface tension and in the solution of the Cahn-Hilliard equation are avoided such that the truncation errors emerging from the finite difference discretization of these derivatives are reduced. Secondly, the lowered number of derivatives implies fewer arithmetic computations and memory accesses. Moreover, by avoiding the solution of a second LBE, significant reduction in memory space is achieved which allows for the simulation of larger problem sizes, in particular for three dimensional problems. As an example, while a D3Q19 LB solution of the Cahn-Hilliard equation requires at least 38 arrays for post-collision and post-streaming distribution functions, the data structure for the phase-field LSE requires only 10 arrays for the LS function and other auxiliary parameters for the convection, compression and diffusion terms.

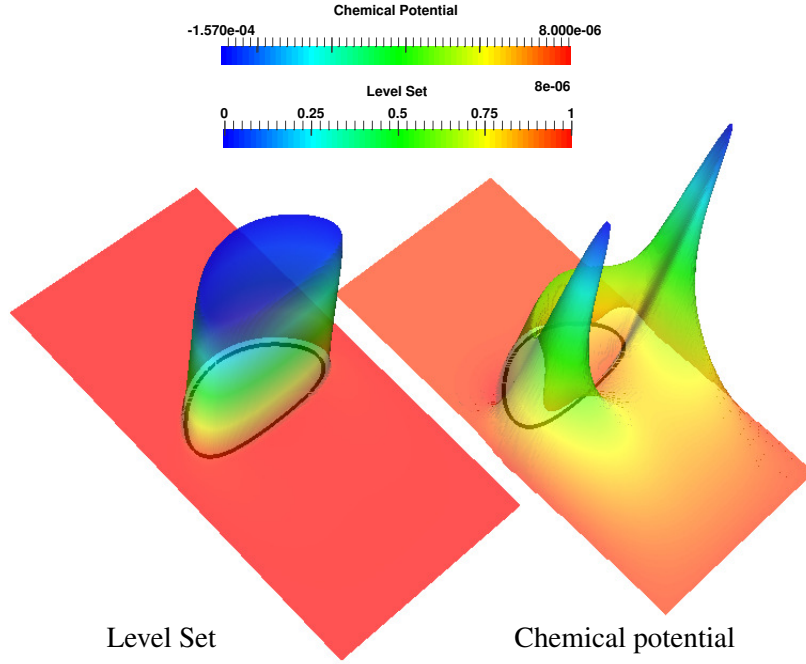


Figure 5.1: Variations in the scalar fields of the chemical potential μ_c , and the phase-field LS function ψ , for a typical rising bubble problem. The solid line shows the location of the interface

5.2.1. Pressure evolution LBE with CSF forcing

As the phase-field LS function gives the normal and curvature to the interface, then one can use the same CSF definition of the surface tension forces as in equation (2.2.7) so as to obtain the diffuse force in the pressure evolution scheme

$$\mathbf{F} = \nabla(\rho c_s^2 - p) + \sigma \kappa(\psi) \mathbf{n}(\psi) \delta_\epsilon(\psi) + \mathbf{G} \quad (5.2.1)$$

where a diffuse Delta function $\delta_\epsilon(\psi)$ can also be defined based on the variations of ψ

$$\delta_\epsilon(\psi) = |\nabla\psi| \quad (5.2.2)$$

which, together with the dependence of ρ on ψ via equation (4.4.1), gives the final form of the force as a function of ψ

$$\mathbf{F} = \Delta\rho c_s^2 \nabla\psi - \nabla p + \sigma \nabla \cdot \left(\frac{\nabla\psi}{|\nabla\psi|} \right) \nabla\psi + \mathbf{G} \quad (5.2.3)$$

with $\Delta\rho = (\rho_l - \rho_g)$. It must be emphasized that using the sharp interface version of the surface tension forces and ρ in equation (5.2.1) may end up in large numerical error in the simulation of high density ratio systems as the balancing ∇p term does not immediately vanish outside of the interface. In addition, comparison of the new forcing term to those of He and Chen [53] in equation (3.2.17) and Li and Liu [76] in equation (3.2.18), reveals that the the present scheme needs only two successive derivatives of ψ , while the variation of ψ is monotonic in the vicinity of the interface as compared to non-monotonic and more irregular changes of the chemical potential μ_c . To see this more clearly, figure 5.1 shows the profiles of ψ along with that of the chemical potential μ_c for a typical rising bubble. From the numerical point of view, it is evident that applying differencing tools to evaluate $\nabla\mu_c$ cannot properly resolve the sharp cusps and crests of μ_c , unless a very fine grid is used.

5.2.2. Temporal and spatial discretization of pressure evolution LBE

In order to carry out the time integration, the pressure evolution LBE is borrowed from chapter 3 with the CSF form of the surface tension forces and the generalized relaxation matrix Λ :

$$\frac{\partial g_\alpha}{\partial t} + c_\alpha \cdot \nabla g_\alpha = \Lambda_\alpha (g_\alpha - g_\alpha^{eq}) + (c_\alpha - \mathbf{u}) \cdot \left[\Delta \rho c_s^2 (\Gamma_\alpha(\mathbf{u}) - w_\alpha) \nabla \Psi + (\mathbf{G} + \sigma \nabla \cdot \left(\frac{\nabla \Psi}{|\nabla \Psi|} \right) \nabla \Psi) \Gamma_\alpha(\mathbf{u}) \right]. \quad (5.2.4)$$

Application of the trapezoidal rule for both terms on the right hand side gives

$$g_k(x + c_\alpha \Delta t, t + \Delta t) = g_k(x, t) - \frac{\Lambda}{2} (g_\alpha(x, t) - g_\alpha^{eq}(x, t)) - \frac{\Lambda}{2} (g_\alpha(x + c_\alpha \Delta t, t) - g_\alpha^{eq}(x + c_\alpha \Delta t, t)) \\ + \frac{\Delta t}{2} S_\alpha^\sigma(x, t) + \frac{\Delta t}{2} S_\alpha^p(x, t) + \frac{\Delta t}{2} S_\alpha^\sigma(x + c_\alpha \Delta t, t + \Delta t) + \frac{\Delta t}{2} S_\alpha^p(x + c_\alpha \Delta t, t + \Delta t) \quad (5.2.5)$$

where S_α^σ is the source term pertained to the surface tension and gravity and S_α^p denotes the force contribution from the density gradient

$$S_\alpha^\sigma = (c_\alpha - \mathbf{u}) \cdot \left[(\sigma \nabla \cdot \left(\frac{\nabla \Psi}{|\nabla \Psi|} \right) \nabla \Psi + \mathbf{G}) \Gamma_\alpha(\mathbf{u}) \right] \quad (5.2.6) \\ S_\alpha^p = (c_\alpha - \mathbf{u}) \cdot [\Delta \rho c_s^2 (\Gamma_\alpha(\mathbf{u}) - w_\alpha) \nabla \Psi].$$

As for the spacial discretization, the central differencing scheme will be used for $\nabla \Psi$ and $\nabla \cdot (\nabla \Psi / |\nabla \Psi|)$ in the surface tension terms. However, as the $\nabla \rho$ has a significant contribution to the convective term and will grow large for large density differences, directional upwind differencing is used for time t while directional central differencing is employed for $t + \Delta t$, such that

$$g_k(x + c_\alpha \Delta t, t + \Delta t) = g_k(x, t) - \frac{\Lambda}{2} (g_\alpha(x, t) - g_\alpha^{eq}(x, t)) - \frac{\Lambda}{2} (g_\alpha(x + c_\alpha \Delta t, t) - g_\alpha^{eq}(x + c_\alpha \Delta t, t)) \\ + \frac{\Delta t}{2} S_\alpha^\sigma(x, t) + \frac{\Delta t}{2} S_\alpha^\sigma(x + c_\alpha \Delta t, t + \Delta t) + \\ \frac{\Delta t}{2} (c_\alpha - \mathbf{u}) \cdot [\Delta \rho c_s^2 (\Gamma_\alpha(\mathbf{u}) - w_\alpha) \nabla^u \Psi] \Big|_{(x,t)} + \\ \frac{\Delta t}{2} (c_\alpha - \mathbf{u}) \cdot [\Delta \rho c_s^2 (\Gamma_\alpha(\mathbf{u}) - w_\alpha) \nabla^c \Psi] \Big|_{(x+c_\alpha \Delta t, t+\Delta t)} \quad (5.2.7)$$

and $c_\alpha \cdot \nabla^c$, $c_\alpha \cdot \nabla^u$ are central and upwind differencing in the lattice direction c_α , as defined for directional derivatives of pressure in equations (5.1.15) and (5.1.18). Equation (5.2.8) could then be rendered to an explicit form

$$\bar{g}_k(x + c_\alpha \Delta t, t + \Delta t) = \bar{g}_k(x, t) - (\Lambda + 2\mathbf{I})(\bar{g}_\alpha(x, t) - \bar{g}_\alpha^{eq}(x, t)) + \\ (c_\alpha - \mathbf{u}) \cdot \left[(\sigma \nabla^c \cdot \left(\frac{\nabla^c \Psi}{|\nabla^c \Psi|} \right) \nabla^c \Psi + \mathbf{G}) \Gamma_\alpha(\mathbf{u}) \right] \Big|_{(x,t)} + \\ (c_\alpha - \mathbf{u}) \cdot [\Delta \rho c_s^2 (\Gamma_\alpha(\mathbf{u}) - w_\alpha) \nabla^{ave} \Psi] \Big|_{(x,t)} \quad (5.2.8)$$

where $c_\alpha \cdot \nabla^{ave}$ is adopted from equation (5.1.19) and the following transformations are used

$$\bar{g}_\alpha = g_\alpha + \frac{\Lambda}{2} (\bar{g}_\alpha - \bar{g}_\alpha^{eq}) - \frac{1}{2} (S_\alpha^p + S_\alpha^\sigma) \quad (5.2.9) \\ \bar{g}_\alpha^{eq} = g_\alpha^{eq} - \frac{1}{2} (S_\alpha^p + S_\alpha^\sigma).$$

Similar to the discretization of the one fluid approach, derivatives other than directional ones as in $\mathbf{u} \cdot \nabla \rho$ are obtained by a weighted sum over the calculated directional central derivatives $c_\alpha \cdot \nabla$, as in equation (5.1.20).

In order to recover the pressure from the new distribution functions, one can use the new distribution functions \bar{g}_α

$$p = \sum_\alpha g_\alpha = \sum_\alpha \left[\bar{g}_\alpha - \frac{\Lambda}{2} (\bar{g}_\alpha - \bar{g}_\alpha^{eq}) + \frac{1}{2} (S_\alpha^p + S_\alpha^\sigma) \right] = \sum_\alpha \bar{g}_\alpha + \frac{1}{2} \sum_\alpha (S_\alpha^p + S_\alpha^\sigma). \quad (5.2.10)$$

The sum over the surface tension and gravity forces eventually reduces to zero since

$$\sum_\alpha (c_\alpha - \mathbf{u}) \Gamma_\alpha(\mathbf{u}) = \sum_\alpha c_\alpha \Gamma_\alpha(\mathbf{u}) - \mathbf{u} \sum_\alpha \Gamma_\alpha(\mathbf{u}) = 0 \quad (5.2.11)$$

while the other sum gives a non zero contribution

$$\sum_\alpha (c_\alpha - \mathbf{u}) (\Gamma_\alpha(\mathbf{u}) - w_\alpha) \nabla^c \psi = \mathbf{u} \cdot \nabla^c \psi \quad (5.2.12)$$

which eventually gives the pressure as

$$p = \sum_\alpha \bar{g}_\alpha + \frac{c_s^2 \Delta \rho}{2} \mathbf{u} \cdot \nabla^c \psi. \quad (5.2.13)$$

Recovering of the momentum from the first order moment of \bar{g}_α reads

$$\rho \mathbf{u} c_s^2 = \sum_\alpha c_\alpha g_\alpha = \sum_\alpha c_\alpha \bar{g}_\alpha + \frac{1}{2} \sum_\alpha c_\alpha (S_\alpha^p + S_\alpha^\sigma). \quad (5.2.14)$$

Unlike the previous case of pressure, the sum over $c_\alpha S_\alpha^p$ vanishes since

$$\sum_\alpha c_\alpha (c_\alpha - \mathbf{u}) (\Gamma_\alpha(\mathbf{u}) - w_\alpha) \nabla^c \psi \approx \mathbf{u}^2 = \mathcal{O}(Ma^2) \approx 0. \quad (5.2.15)$$

While the directional contribution from S_α^σ remains as

$$\sum_\alpha c_\alpha (c_\alpha - \mathbf{u}) \Gamma_\alpha = \sum_\alpha c_\alpha c_\alpha \Gamma_\alpha + \mathcal{O}(Ma^2). \quad (5.2.16)$$

Substituting the expression for Γ_α and neglecting high order terms, gives

$$\sum_\alpha c_\alpha (c_\alpha - \mathbf{u}) \Gamma_\alpha = c_s^2 \quad (5.2.17)$$

which thus simplifies the expression for momentum

$$\rho \mathbf{u} = \frac{1}{c_s^2} \sum_\alpha \bar{g}_\alpha + \frac{1}{2} (\sigma \nabla^c \cdot \left(\frac{\nabla^c \psi}{|\nabla^c \psi|} \right) \nabla^c \psi + \mathbf{G}). \quad (5.2.18)$$

5.3. Multiple relaxation time collision for multiphase LBM

It is now widely-known that the use of the single relaxation time (SRT) scheme introduced in equation (3.1.15), though being computationally cheap, is the source of numerical oscillations in the pressure and velocity field. Such oscillations may specially end up in severe instability in the case of very low viscosities, as may occur for example in the gas phase in multiphase flows. The destructive influence of the SRT scheme on the magnitude of the spurious velocities is also

well studied in the works of Lallemand and Luo [71] for single-phase flows and more recently by Yu and Fan [139] and Fakhari and Lee [36] for multiphase flows. Such unfavourable effects emerge from the fact that in the single relaxation time collision, all the hydrodynamic moments of the distribution function are forced to relax towards equilibrium through the same relaxation time τ , which is a function of viscosity ν . This implies that such moments are transported under the same physical time step which may violate the physical restrictions on the temporal integration, thus causing instability [96]. The situation aggravates when dealing with 3D simulations as the computational cost restricts the lattice size and hence increasing the lattice resolution can not be practically considered as a solution.

As a remedy to alleviate this issue, a number of modifications to the SRT model have been suggested. One approach is to use an entropic LB model (ELBM) proposed by Karlin et. al [64] and Ansumali and Karlin [12], [13] in which the equilibrium distribution functions are constructed in a way to minimize the convex Liapanov functional H , so as to guarantee the positive production of entropy in the system and hence prevent the distribution function to become negative. As a result, the relaxation time does not have to be fixed by the viscosity only and is governed by the new stability constraints leading to spacial variations of the relaxation time. The method has been shown to be remarkably stable for high Re numbers and even turbulent flows and was extended to multicomponent flows. Nevertheless, it has been criticized to impose excessively higher computational demand as compared to the SRT BGK model and lack of accuracy proofs to recover the NSE [134, 135].

A well-established method, however, is to perform a multiple relaxation time (MRT) collision in which different hydrodynamic moments are allowed to relax towards their equilibrium state under distinct relaxation times. The approach has not been proved to improve the stability for previously unstable SRT solutions, but suggests a more physically meaningful collision framework in which the different hydrodynamic moments could be treated to add desired complexities and specific features to the flow which are hard to be realised through the SRT scheme. Examples are simulating flows of viscoelastic materials [73] as well as acoustic and thermal flows with adjustable Prandtl numbers for simulating thermo-hydrodynamics [72]. Another remarkable feature of MRT methods is to treat different diffusion rates in anisotropic systems and more importantly to rectify the anisotropies and thus spurious currents emerging in isotropic systems such as multiphase flows [36, 96]. Although the computational cost of the MRT increases as compared to the SRT model, the additional workload consists mainly of arithmetic instructions which could be optimized and hidden by the memory access latencies in the overall intensive memory bound LB algorithm [96].

The implementation of the MRT for 2 and 3D flows has been discussed in [71] and [34]. Mukherji and Abaraham [87] regulated the method for 2D multiphase flows which was extended later on by Premnath and Abraham [96] for 3D multiphase flows up to D3Q15 and D3Q19 lattice stencils. In this section, the structure of the MRT model is discussed and the implementation for 2D and 3D stencils up to D3Q27 is presented with focus on the multiphase flow systems.

For the derived LB equations (5.1.25) and (5.2.8) for both one-fluid and pressure evolution schemes the collision term involves the generalized relaxation matrix $\Lambda + 2\mathbf{I}$. In the MRT framework, Lallemand and Luo [71] proposed in their renowned 2000 paper to rewrite this term as

$$\Lambda + 2\mathbf{I} = \mathbf{M}^{-1}\hat{\Lambda}\mathbf{M} \quad (5.3.1)$$

where $\hat{\Lambda}$ is a $n \times n$ the diagonal relaxation matrix and \mathbf{M} is the transformation matrix which converts the n distribution functions of the $DmQn$ lattice stencil to n hydrodynamic moments. In essence, this reformulation of the collision means that first the moments and their corresponding equilibrium states are constructed as

$$\begin{aligned}\hat{f} &= \mathbf{M}\bar{f} \\ \hat{f}^{eq} &= \mathbf{M}\bar{f}^{eq}.\end{aligned}\quad (5.3.2)$$

Then the new moments are relaxed individually through different relaxation times induced by the relaxation matrix $\hat{\Lambda}$. Eventually, the post-collision moments are returned back to the discrete kinetic space through the inverse transformation performed by the application of \mathbf{M}^{-1} .

Similarly, the $\hat{\Lambda} = \mathbf{M}\mathbf{A}\mathbf{M}^{-1}$ for D2Q9, D2Q15, D3Q19 and D3Q27 will be accordingly of the form

$$\begin{aligned}\hat{\Lambda} &= \text{Diag}\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9\} \\ \hat{\Lambda} &= \text{Diag}\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\} \\ \hat{\Lambda} &= \text{Diag}\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}, s_{17}, s_{18}, s_{19}\} \\ \hat{\Lambda} &= \text{Diag}\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}, s_{17}, s_{18}, s_{19}, \\ &\quad s_{20}, s_{21}, s_{22}, s_{23}, s_{24}, s_{25}, s_{26}, s_{27}\}.\end{aligned}\quad (5.3.3)$$

The procedure to derive the transformation matrices is closely connected to the definition of the hydrodynamic quantities via constructing different moments m_α from the distribution functions and the macroscopic velocities $c_{\alpha,i}$ in the Cartesian directions. For D2Q9 up to D3Q19 stencils, the moments are

$$\begin{aligned}\hat{f} &= [\rho, e, e^2, j_x, q_x, j_y, q_y, p_{xx}, p_{xy}] \\ \hat{f} &= [\rho, e, e^2, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, p_{ww}, p_{xy}, p_{yz}, p_{zx}, m_{xyz}] \\ \hat{f} &= [\rho, e, e^2, j_x, q_x, j_y, q_y, j_z, q_z, 3p_{xx}, 3\pi_{xx}, p_{ww}, \pi_{ww}, p_{xy}, p_{yz}, p_{zx}, m_x, m_y, m_z]\end{aligned}\quad (5.3.4)$$

where e, e^2 are the kinetic energy and square of kinetic energy, q_x, q_y, q_z are the components of energy flux, $p_{xx}, p_{xy}, p_{yz}, p_{zx}$ are the components of the traceless viscous stress tensor and p_{ww} comes from the combination $p_{ww} = p_{yy} - p_{zz}$, and m_{xyz} is the antisymmetric third order moment. Finally, π_{xx} and π_{ww} have the same symmetry as the diagonal part of p_{ij} and are made by the production of p_{ij} and the energy. It can be shown that each moment is originally constructed via one row of a primary transformation matrix $\tilde{\mathbf{M}}$. The final transformation matrix \mathbf{M} , however, is preferred to be in such a form to give a strictly diagonal relaxation matrix. This requires rewriting $\tilde{\mathbf{M}}$ in an orthonormal basis via the Gram-Schmidt procedure. The orthogonal set is built in increasing order of moments and then arranged in increasing order of complexity of the tensorial representation of the moments to obtain the matrix \mathbf{M} [96]. For the D2Q9 stencil the rows of \mathbf{M} could be already defined so as to give an orthonormal basis [34]

$$\begin{aligned}|\rho\rangle_\alpha &= |c_\alpha|^0, \\ |e\rangle_\alpha &= -4|c_\alpha|^0 + 3(c_{\alpha,x}^2 + c_{\alpha,y}^2), \\ |j_x\rangle_\alpha &= c_{\alpha,x}, \\ |j_y\rangle_\alpha &= c_{\alpha,y}, \\ |p_{xx}\rangle_\alpha &= c_{\alpha,x}^2 - c_{\alpha,y}^2, \\ |e^2\rangle_\alpha &= 4|c_\alpha|^0 - \frac{21}{2}(c_{\alpha,x}^2 + c_{\alpha,y}^2) + \frac{9}{2}(c_{\alpha,x}^2 + c_{\alpha,y}^2)^2, \\ |q_x\rangle_\alpha &= [-5|c_\alpha|^0 + 3(c_{\alpha,x}^2 + c_{\alpha,y}^2)]c_{\alpha,x}, \\ |q_y\rangle_\alpha &= [-5|c_\alpha|^0 + 3(c_{\alpha,x}^2 + c_{\alpha,y}^2)]c_{\alpha,y}, \\ |p_{xy}\rangle_\alpha &= c_{\alpha,x}c_{\alpha,y}\end{aligned}\quad (5.3.5)$$

which gives the transformation \mathbf{M} as

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & 2 & -1 & 2 & -1 & 2 & -1 & 2 \\ 4 & -2 & 1 & -2 & 1 & -2 & 1 & -2 & 1 \\ 0 & 1 & 1 & 0 & -1 & -1 & -1 & 0 & 1 \\ 0 & -2 & 1 & 0 & -1 & 2 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & -2 & 1 & 0 & -1 & 2 & -1 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \end{bmatrix}. \quad (5.3.6)$$

As a special case in 3D, where the obtained moments also encompass those in D3Q15 and D3Q19, the derivation of the transformation for the D3Q27 stencil is reviewed here. In their work for the 3D LBM-based large eddy simulation of turbulent flows, Suga et. al [117] performed a similar procedure as in [71] and derived the set of hydrodynamic moments for the D3Q27 stencil

$$\hat{f} = [\rho, j_x, j_y, j_z, e, p_{xx}, p_{yy}, p_{zz}, p_{xy}, p_{yz}, p_{zx}, q_x, q_y, q_z, q_x^2, q_y^2, q_z^2, e^2, e^3, \pi_{xx}, \pi_{yy}, \pi_{zz}, \pi_{xy}, \pi_{yz}, \pi_{zx}, m_x, m_y, m_z, m_{xyz}] \quad (5.3.7)$$

where in addition to those moments in D3Q19 stencil, the additional moments are the square of energy flux q_x^2, q_y^2, q_z^2 , the cube of energy e^3 , and other products of the second order tensors and energy $\pi_{xy}, \pi_{yz}, \pi_{zx}, \pi_{xy}$. Beside the density and the momentum in three directions, the rest of the moments are constructed in such a way to give the following form of the rows of $\tilde{\mathbf{M}}$

$$\begin{aligned} |e\rangle_\alpha &= (c_{\alpha,x}^2 + c_{\alpha,y}^2 + c_{\alpha,z}^2) & , & & |p_{xx}\rangle_\alpha &= (2c_{\alpha,x}^2 - c_{\alpha,y}^2 - c_{\alpha,z}^2), \\ |p_{yy}\rangle_\alpha &= (c_{\alpha,y}^2 - c_{\alpha,x}^2 - c_{\alpha,z}^2) & , & & |p_{xy}\rangle_\alpha &= (c_{\alpha,x}c_{\alpha,y}), \\ |p_{zz}\rangle_\alpha &= (c_{\alpha,z}^2 - c_{\alpha,x}^2 - c_{\alpha,y}^2) & , & & |p_{zx}\rangle_\alpha &= (c_{\alpha,z}c_{\alpha,x}), \\ |q_x\rangle_\alpha &= 3(c_{\alpha,x}^2 + c_{\alpha,y}^2 - c_{\alpha,z}^2)c_{\alpha,x} & , & & |q_y\rangle_\alpha &= 3(c_{\alpha,x}^2 + c_{\alpha,y}^2 - c_{\alpha,z}^2)c_{\alpha,y}, \\ |q_z\rangle_\alpha &= 3(c_{\alpha,x}^2 + c_{\alpha,y}^2 - c_{\alpha,z}^2)c_{\alpha,z} & , & & |q_x^2\rangle_\alpha &= \frac{9}{2}(c_{\alpha,x}^2 + c_{\alpha,y}^2 - c_{\alpha,z}^2)^2 c_{\alpha,x}, \\ |q_y^2\rangle_\alpha &= \frac{9}{2}(c_{\alpha,x}^2 + c_{\alpha,y}^2 - c_{\alpha,z}^2)^2 c_{\alpha,y} & , & & |q_z^2\rangle_\alpha &= \frac{9}{2}(c_{\alpha,x}^2 + c_{\alpha,y}^2 - c_{\alpha,z}^2)^2 c_{\alpha,z}, \\ |e^2\rangle_\alpha &= \frac{3}{2}(c_{\alpha,x}^2 + c_{\alpha,y}^2 - c_{\alpha,z}^2)^2 & , & & |e^3\rangle_\alpha &= \frac{9}{2}(c_{\alpha,x}^2 + c_{\alpha,y}^2 - c_{\alpha,z}^2)^3, \\ |\pi_{xx}\rangle_\alpha &= (2c_{\alpha,x}^2 - c_{\alpha,y}^2 - c_{\alpha,z}^2)(c_{\alpha,x}^2 + c_{\alpha,y}^2 + c_{\alpha,z}^2) & , & & |\pi_{yy}\rangle_\alpha &= (c_{\alpha,y}^2 - c_{\alpha,x}^2 - c_{\alpha,z}^2)(c_{\alpha,x}^2 + c_{\alpha,y}^2 + c_{\alpha,z}^2), \\ |\pi_{xy}\rangle_\alpha &= (c_{\alpha,x}c_{\alpha,y})(c_{\alpha,x}^2 + c_{\alpha,y}^2 + c_{\alpha,z}^2) & , & & |\pi_{yz}\rangle_\alpha &= (c_{\alpha,y}c_{\alpha,z})(c_{\alpha,x}^2 + c_{\alpha,y}^2 + c_{\alpha,z}^2), \\ |\pi_{zx}\rangle_\alpha &= (c_{\alpha,z}c_{\alpha,x})(c_{\alpha,x}^2 + c_{\alpha,y}^2 + c_{\alpha,z}^2) & , & & |m_x\rangle_\alpha &= c_{\alpha,x}(c_{\alpha,y}^2 - c_{\alpha,z}^2), \\ |m_y\rangle_\alpha &= c_{\alpha,y}(c_{\alpha,x}^2 - c_{\alpha,z}^2) & , & & |m_z\rangle_\alpha &= c_{\alpha,z}(c_{\alpha,x}^2 - c_{\alpha,y}^2), \\ |m_{xyz}\rangle_\alpha &= (c_{\alpha,x}c_{\alpha,y}c_{\alpha,z}), & & & & \end{aligned} \quad (5.3.8)$$

By collecting the resulting coefficients into the matrix $\tilde{\mathbf{M}}$ and using the Gram-Schmidt method [117]

$$\mathbf{M} = \tilde{\mathbf{M}} - \sum_{i=0}^k \frac{\tilde{\mathbf{M}}_k \cdot \mathbf{M}_i}{|\mathbf{M}_i|^2} \mathbf{M}_i \quad , \quad (k \geq 4), \quad (5.3.9)$$

the transformation \mathbf{M} will be obtained. Note that the first 4 rows are conserved and mutually independent and hence do not enter the orthonormalization procedure. The derivation of the matrix

\mathbf{M} for the D3Q15 and D3Q19 stencils follows similar rules and the final representation of matrices up to D3Q27 are given in Appendix B.

As to the two coupled multiphase flow models presented in this thesis, it is noteworthy that in comparison to the single-phase MRT model, here the transformation \mathbf{M} will be applied to \bar{f}_α and \bar{g}_α in the one-fluid and pressure evolution models, respectively. This means that the interpretation of \hat{f} and \hat{g} will be different in accordance to the definition of the original distribution functions. For example the zero-order moment will be interpreted as the virtual density $\bar{\rho}$ in the one-fluid model.

5.3.1. Choice of the relaxation parameters

A special extreme in selecting the relaxation parameters would be to set all s_α to $1/\tau$ which reduces the MRT scheme to the SRT collision. Beside this particular case, there have been very few works concerning the choice of the relaxation parameters and their impact on numerical accuracy. Lallemand and Luo [71] provided an extensive analysis for single-phase flows using the D2Q9 MRT model and analysed the behaviour of the hydrodynamic moments and their sensitivity to the relaxation times. First of all, the relaxation parameters pertaining to the conserved moments, i. e., mass and momentum could be chosen arbitrarily and are often set to zero since the moments are identical to their equilibrium values. They also concluded that the relaxations $s_v = 1/\tau$ for different stress tensors must be identical to preserve the isotropy of the system and are related to shear viscosity ν as

$$\tau = \frac{1}{s_v} = 3\nu + \frac{1}{2} \quad (5.3.10)$$

where the kinematic viscosity of the two-phase mixture $\nu = \mu/\rho$ is obtained based on equation (4.2.3) for the one fluid scheme, or equation (4.4.1) for the pressure evolution scheme. A more accurate derivation of τ for the pressure evolution formulation was proposed in [76] through an inverse linear combination of the primary relaxation times $\lambda_g = \tau_g - 0.5$ and $\lambda_l = \tau_l - 0.5$

$$\tau = \lambda + 0.5 = \left(\frac{\Psi}{\lambda_l} + \frac{1-\Psi}{\lambda_g} \right)^{-1} + 0.5. \quad (5.3.11)$$

where Ψ is the phase-field LS function. The relaxation for the kinetic energy was suggested in [71] to be related to the bulk viscosity ζ through a similar rule in the long wave-length limit

$$\zeta = \frac{1}{3} \left(\frac{1}{s_\zeta} - \frac{1}{2} \right). \quad (5.3.12)$$

Nevertheless, the definition of the bulk viscosity is highly problem-dependent and thus often requires the users to find the optimum values of s_ζ by experiment. Eventually, the relaxation times pertained to the square of energy and fluxes of energy were shown in [71] to have no significant effect on the transport coefficients and their values were chosen to be slightly greater than 1. Yu and Fan [139] and Premnath and Abraham [96] carried out simulations for 3D MRT multiphase flow problems following the same rule of thumbs for the choice of the relaxation parameters. Beside the positive consequences reported regarding the diminished level of spurious velocities at the interface using the MRT scheme, one could not find extensive quantitative measurements specially for the dynamic velocity field to show the effect of choosing different relaxation times other than those in [71]. Fakhari and Lee [36] performed studies on static bubbles and concluded that faster convergence for spurious kinetic energy level is obtained by under-relaxing the energy mode which means setting small values for s_ζ and thus scaling up the bulk viscosity.

In this work a more detailed study on the effect of the relaxation parameters on the accuracy of the benchmark simulations will be presented in chapter 6. Nevertheless, a summary of the general setting of the relaxation parameters is suggested as follows. For the D2Q9 lattice

$$\hat{\Lambda} = \text{Diag}\{0, s_{\zeta}, s_3, 0, s_5, 0, s_7, s_v, s_v\} \quad (5.3.13)$$

where s_3, s_5, s_7 are free parameters. For D3Q15 lattice

$$\hat{\Lambda} = \text{Diag}\{0, s_{\zeta}, s_3, 0, s_5, 0, s_7, 0, s_9, s_v, s_v, s_v, s_v, s_v, s_{15}\} \quad (5.3.14)$$

where s_3, s_5, s_7, s_9 and s_{15} are free parameters. For D3Q19 lattice

$$\hat{\Lambda} = \text{Diag}\{0, s_{\zeta}, s_3, 0, s_5, 0, s_7, 0, s_9, s_v, s_{11}, s_v, s_{13}, s_v, s_v, s_v, s_{17}, s_{18}, s_{19}\} \quad (5.3.15)$$

where s_3, s_5, s_7, s_9 plus s_{11}, s_{13} and s_{17} to s_{19} are free parameters. Finally, for D3Q27 lattice one has

$$\hat{\Lambda} = \text{Diag}\{0, 0, 0, s_{\zeta}, s_v, s_v, s_v, s_v, s_v, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}, s_{17}, s_{18}, s_{19}, s_{20}, s_{21}, s_{22}, s_{23}, s_{24}, s_{25}, s_{26}, s_{27}\} \quad (5.3.16)$$

with s_{10} to s_{27} as free parameters which are often chosen to be close to 1.0.

5.3.2. Summary of the coupled LB-LS algorithm

The solutions for the for one-fluid and pressure evolution coupled LBM-LS schemes described so far, are summarized in algorithms 1 and 2 for the two coupled methods. Note that the solution of the sharp LSE in the one-fluid approach may employ either the PDE-based or the periodic brute force reinitialization.

Algorithm 1 Computational algorithm for the one-fluid coupled LB-LS model

- 1: Initialize distribution functions as $\bar{f}_\alpha = \bar{f}_\alpha^{eq}$
 - 2: Initialize flow properties and force terms
 - 3: Initialize the LS function ϕ as well as \mathbf{n} and κ
 - 4: **for** $t_{lb} \leq t_{max}$ **do**
 - 5: Calculate $\nabla\phi$ sing WENO scheme.
 - 6: Solve LS equation (4.2.10) for $\phi(\mathbf{x}, t + \Delta t)$.
 - 7: **if** Brute force reinitialization **then**
 - 8: **if** $t_{lb} \% N_p = 0$ **then**
 - 9: Reconstruct the interface
 - 10: Calculate ϕ_{dist}
 - 11: Update ϕ using equation (4.3.11).
 - 12: **end if**
 - 13: **end if**
 - 14: **if** PDE-based reinitialization **then**
 - 15: Calculate $|\nabla\phi|$ using equation (4.3.9).
 - 16: Solve equation (4.3.4) for $2\epsilon/h$ iterations.
 - 17: **end if**
 - 18: Update \mathbf{n} and κ using equation (4.2.6).
 - 19: Calculate $c_\alpha \cdot \nabla\phi$ for all directions α .
 - 20: Update the force terms in equation (5.1.14).
 - 21: Do collision and streaming to get $\bar{f}_\alpha(x + c_\alpha\Delta t, t + \Delta t)$ using equation (5.1.25).
 - 22: Update p and \mathbf{u} using equations (5.1.28) and (5.1.31).
 - 23: **end for**
-

Algorithm 2 Computational algorithm for the pressure evolution coupled LB-LS model

- 1: Initialize distribution functions as $\bar{g}_\alpha = \bar{g}_\alpha^{eq}$
 - 2: Initialize flow properties and force terms
 - 3: Initialize the LS function ψ as well as \mathbf{n} and κ
 - 4: **for** $t_{lb} \leq t_{max}$ **do**
 - 5: Calculate $\nabla\psi$ sing WENO scheme.
 - 6: Calculate the compression and diffusion terms in equation (4.4.5)
 - 7: Solve LS equation (4.4.10) for $\psi(\mathbf{x}, t + \Delta t)$.
 - 8: Update \mathbf{n} and κ using equation (4.4.2).
 - 9: Calculate $c_\alpha \cdot \nabla\psi$ for all directions α .
 - 10: Update the force terms in equation (5.2.6).
 - 11: Do collision and streaming to get $\bar{g}_\alpha(x + c_\alpha\Delta t, t + \Delta t)$ using equation (5.2.8).
 - 12: Update p and \mathbf{u} using equations (5.2.13) and (5.2.18).
 - 13: **end for**
-

5.4. Numerical tests

In order to examine the performance of the two multiphase LB formulations with minimum influence from the interface tracking approaches, the 2D static bubble problem is studied, by which the accuracy of the solutions could be verified through measuring the spurious velocity level as well as checking the pressure variations against the Laplace law which relates pressure difference across the interface to the surface tension as

$$P_{in} - P_{out} = \frac{\sigma}{r_0}$$

where r_0 is the radius of the bubble. The bubble is located at $(x, y) = (0.5, 0.5)$ in a square domain with periodic boundary conditions on all sides. Two test cases are then defined with input values as in table 5.1 in macroscopic units. The lattice values for density and viscosity are assumed to be identical to the macroscopic ones. The surface tension coefficient in LB units σ_{lb} could be obtained based on dimensional analysis

$$\frac{\sigma \rho L_0}{\mu^2} = \frac{\sigma_{lb} \rho_{lb} L_0^*}{\mu_{lb}^2}.$$

Since $\rho_{lb} = \rho$, $\mu_{lb} = \mu$ and, $L_0^*/L_0 = 1/h$, then

$$\sigma_{lb} = \sigma h$$

The first test case denoted by STC1 has a low density ratio and high surface tension so as to check the accuracy of the surface tension implementation. The second test case denoted by STC2 has a low surface tension coefficient and instead a high density ratio.

Table 5.1: Physical parameters for the static bubble test cases

Test case	ρ_l	ρ_g	μ_l	μ_g	σ
STC1	1000	100	10	1	24.5
STC2	1000	1	10	0.1	1.96

The spurious velocity level is measured using the following norms

$$|u|_1 = \frac{\sum_{i=1}^N |\mathbf{u}_i|}{N}$$

$$|u|_2 = \left(\frac{\sum_{i=1}^N |\mathbf{u}_i|^2}{N} \right)^{1/2}$$

$$|u|_{max} = \max_i |\mathbf{u}_i|$$

where the macroscopic velocities are obtained by conversion from lattice units. Dimensional analysis again gives the macroscopic velocities

$$\frac{\rho_{lb} u_{lb} L_0^*}{\mu_{lb}} = \frac{\rho u L_0}{\mu} \quad \longrightarrow \quad u = u_{lb} / h$$

which magnifies the LB velocities by $1/h$ and therefore the obtained magnitudes must be distinguished from those reported in lattice units of other studies in the literature [36, 140]. In addition, the error in the Laplace law is calculated as

$$E_p = \frac{|P_{in} - P_{out}|}{\sigma / r_0} - 1$$

Table 5.2: Description of various differencing schemes used for force discretization.

Pressure force discretization	differencing description
C	central for $t + \Delta t$ and t
DCC	directional central for $t + \Delta t$ and t
DCU	directional central for $t + \Delta t$, and directional upwind for t
DAA	directional averaged for $t + \Delta t$ and t

At this stage of the work, all the free relaxation parameters of the MRT collision are set to 1.0, since only the steady state solution of a stationary problem is desired and the intermediate convergence rate is of minimal interest.

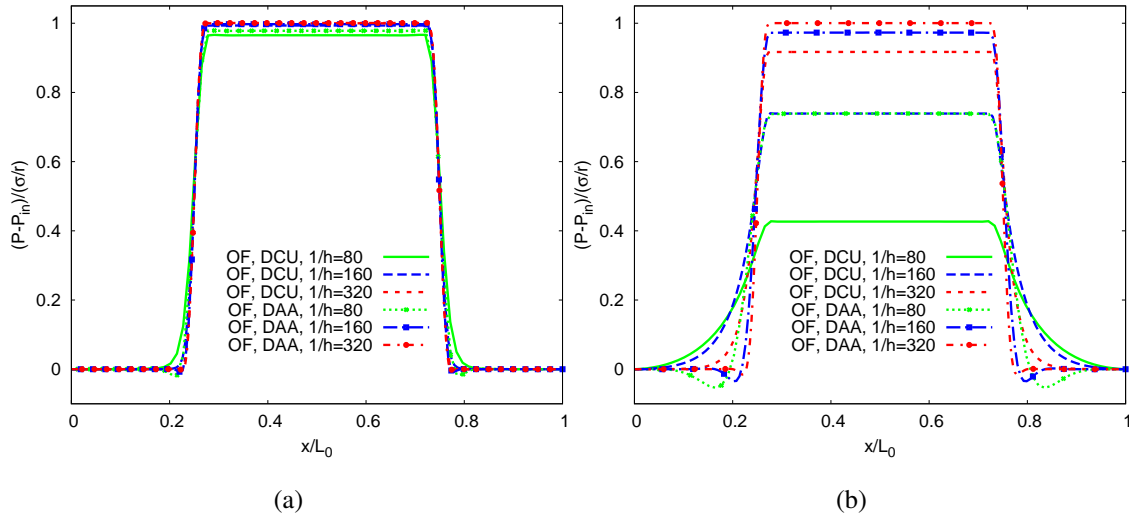
As the one-fluid scheme has never been analysed with regards to the accuracy of the force discretization in the literature, the performance of different discretization methods, as summarised in table 5.2, are investigated in tables 5.3 and 5.4 for the one-fluid scheme (OF), using $\varepsilon = 0.025$ for all lattice levels. For STC1 with low density ratio, all discretization schemes perform well as the dominating surface tension effect is implemented in the same way. Nevertheless, the simple central differencing produces larger spurious velocities due to the weaker isotropy in discretization of the pressure forces. Investigation of the second test case STC2, of higher density ratio, shows that both simple and directional central differencing fail to recover the correct pressure level, while confining the averaged differencing to the time t only, does not improve the accuracy of the pressure unless a high lattice resolution is employed. In contrast, applying the averaged differencing for both $t + \Delta t$ and t presents a favourable reduction in both velocity and pressure errors by refining the lattice. The difference between the performance of DCU and DAA schemes in recovering the pressure is studied more in figure 5.2. While both schemes agree well with the Laplace law for STC1, the DCU differencing tends to underestimate the pressure difference in STC2 due to the steep variations of \tilde{f}_α across the interface in the one-fluid approach.

Table 5.3: Pressure and velocity errors for static bubble STC1 using one-fluid (OF) scheme, $\varepsilon = 0.025$.

$1/h$	$ \mathbf{u} _1$	$ \mathbf{u} _2$	$ \mathbf{u} _{max}$	E_p
OF-C				
80	2.0e-3	2.3e-3	6.3e-3	2.9e-2
160	7.6e-4	9.4e-4	3.6e-3	6.2e-3
320	2.3e-4	2.9e-4	1.0e-3	8.6e-5
OF-DCC				
80	5.5e-5	1.2e-4	9.1e-4	4.4e-2
160	3.2e-5	5.3e-5	6.0e-4	1.1e-2
320	1.2e-5	3.5e-5	6.4e-4	1.1e-3
OF-DCU				
80	8.7e-5	1.5e-4	1.0e-3	3.6e-2
160	2.0e-5	5.1e-4	5.8e-4	6.7e-3
320	9.5e-6	3.1e-5	4.7e-4	2.3e-4
OF-DAA				
80	2.3e-4	3.3e-4	1.3e-3	2.2e-2
160	4.5e-5	7.3e-5	6.1e-4	2.6e-3
320	1.2e-5	3.1e-5	4.8e-4	6.4e-4

Table 5.4: Pressure and velocity errors for the static bubble STC2 using the one-fluid (OF) scheme, $\varepsilon = 0.025$.

$1/h$	$ \mathbf{u} _1$	$ \mathbf{u} _2$	$ \mathbf{u} _{max}$	E_p
OF-C				
80	6.5e-4	7.9e-4	2.9e-3	6.3e-1
160	6.5e-4	7.3e-4	2.1e-3	3.2e-1
320	4.2e-4	4.7e-4	1.0e-3	1.2e-1
OF-DCC				
80	4.9e-5	9.0e-5	7.7e-4	6.9e-1
160	9.6e-6	2.6e-5	2.9e-4	3.9e-1
320	2.4e-6	8.5e-6	1.5e-4	1.6e-1
OF-DCU				
80	1.5e-4	2.1e-4	1.2e-3	6.8e-1
160	3.7e-5	5.3e-5	3.0e-4	2.7e-1
320	5.6e-6	1.0e-5	1.2e-4	8.4e-2
OF-DAA				
80	5.5e-4	5.5e-4	3.3e-3	2.7e-1
160	1.0e-4	1.0e-4	1.1e-3	2.8e-2
320	1.2e-5	2.4e-5	2.0e-4	2.0e-4

**Figure 5.2:** Variations of pressure along the x direction at $y = 0.5$ for the static bubble (a) STC1, (b) STC2, using different discretizations of the one-fluid (OF) scheme, $\varepsilon = 0.025$.

Lee and Lin [75] presented a similar investigation on the use of different discretization schemes for the pressure evolution (PE) approach. They concluded that using DCU technique for $\nabla\psi$, as explained in section 5.2.2, provides higher stability and accuracy as compared to C and DCC types. Unlike the one-fluid approach, numerical experiment shows that the use of the DAA differencing for the pressure evolution scheme must be avoided as it rapidly triggers instabilities in the solution at high density ratios.

By choosing the DAA differencing for the one-fluid approach and DCU differencing for the pressure evolution approach, figure 5.3 shows the performance of the two methods in recovering the pressure on different lattice levels and interface thickness values ε/h . Contrary to the pressure

evolution method, the one-fluid method does not reduce the pressure error for a constant $m = \varepsilon/h$ on finer grids, i. e., the accuracy of the pressure does not improve if a thinner interface is desired on higher lattice resolutions. This is speculated to be a result of using the ideal gas EOS for recovering the pressure in the one-fluid approach which is valid for moderate pressure gradients, while its accuracy degrades for more steep pressure variations of a thin interface. In fact, the one-fluid method needs at least $m = 8$ to start reducing the errors for thinning interfaces. On the other hand, the PE approach gives generally lower pressure errors even on coarse grids since its underlying LBE solves specifically for the pressure. In addition, the scheme consistently diminishes the pressure and velocity errors by refining the lattice for thinning interfaces as also showed in tables 5.5 and 5.6 for $m = 4$.

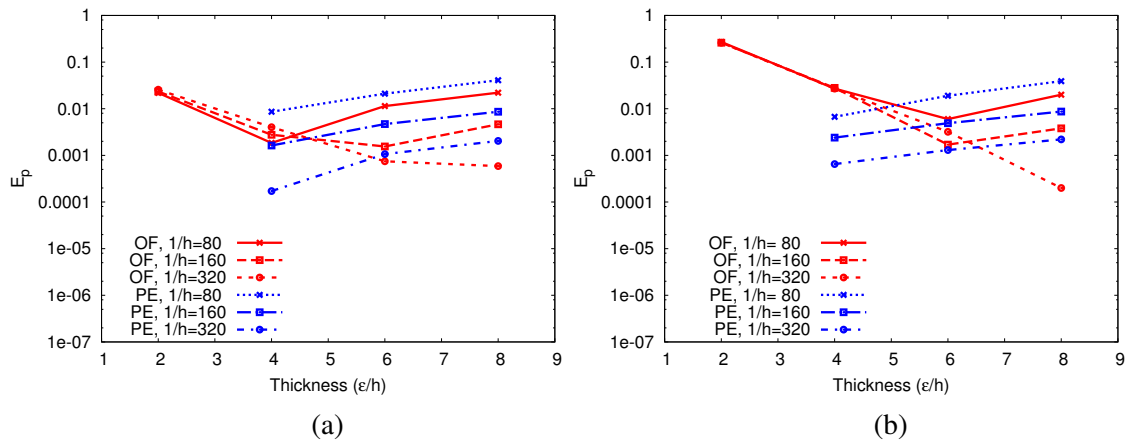


Figure 5.3: Pressure error for the static bubble(a) STC1, (b) STC2, using different values of interface thickness ε/h .

Table 5.5: Pressure and velocity errors for the static bubble STC1 using pressure evolution (PE) scheme, $\varepsilon/h = 4$.

$1/h$	$ \mathbf{u} _1$	$ \mathbf{u} _2$	$ \mathbf{u} _{max}$	E_p
80	3.0e-4	4.6e-3	1.8e-3	8.6e-3
160	1.1e-4	2.1e-4	1.1e-3	1.6e-3
320	7.1e-5	1.1e-4	8.8e-4	1.7e-4

Table 5.6: Pressure and velocity errors for the static bubble STC2 using pressure evolution (PE) scheme, $\varepsilon/h = 4$.

$1/h$	$ \mathbf{u} _1$	$ \mathbf{u} _2$	$ \mathbf{u} _{max}$	E_p
80	7.9e-5	1.1e-4	4.5e-4	6.7e-3
160	2.8e-5	3.9e-5	1.6e-4	2.4e-3
320	7.7e-6	1.1e-5	5.1e-5	6.5e-4

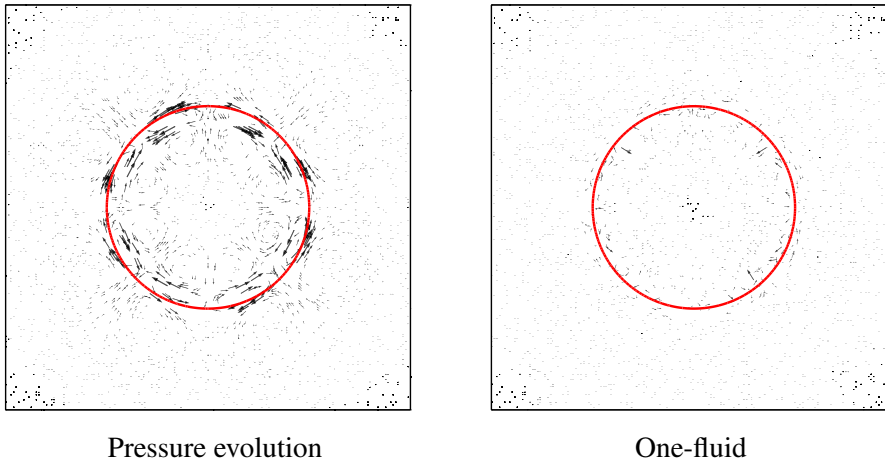


Figure 5.4: Spurious velocities in static bubble STC1.

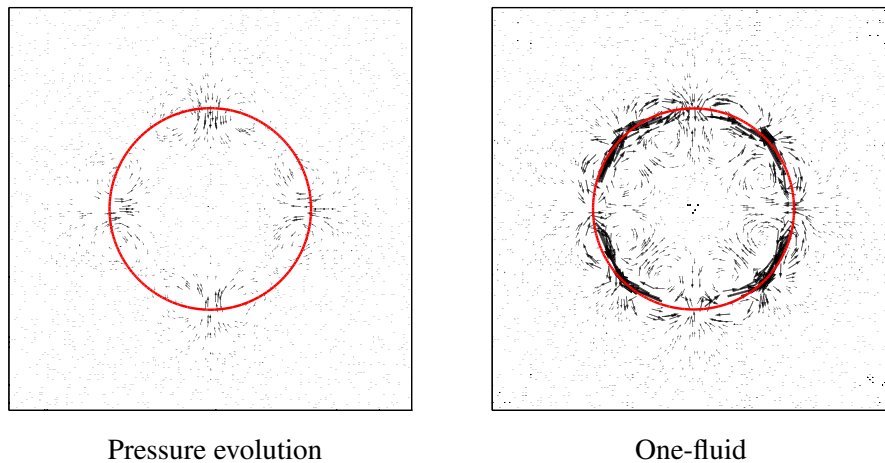


Figure 5.5: Spurious velocities in static bubble STC2.

A similar comparison between the two methods with regards to the magnitude and distribution of the spurious velocities is presented in figures 5.4 and 5.5 for both test cases. The lattice resolution is $1/h = 160$ and the interface thickness is set to $\varepsilon/h = 4$. The velocities reach their maximum values over the interface where the truncation error is large. It could be seen that, as compared to the pressure evolution scheme, the one-fluid scheme can eradicate the spurious velocities more effectively in the surface tension dominated problem of STC1. The reason could be associated with the sharp nature of the one-fluid scheme which limits the surface tension forces to the interface band, while the diffuse profile of the phase-field pressure evolution scheme allows the surface tension and pressure forces to extend their interaction far from the interface. In contrast, the spurious velocities in STC2 reveal the relative weakness of the one-fluid approach in dealing with high density ratios. Even with the DAA differencing, the large density difference tends to magnify the truncation errors in the one-fluid forcing term. On the other hand, in the absence of large surface tension effects, the pressure evolution model offers greatly diminished spurious currents.

Finally, it must be noted that despite having higher pressure errors and spurious velocities at large density ratios, the sharp interface one-fluid approach remains nevertheless stable for interfaces as thin as only 1 lattice cell in thickness, whereas the PE approach needs at least 3 lattice cells to resolve the interface (although still inaccurate) as also discussed in [9] and [74].

Results and benchmarking

The numerical test of the static bubble discussed in the last chapter has the advantage of having specific exact solutions, i. e., the bubble has to remain circular, spurious velocities must be minimized and the pressure difference has to obey the Laplace law. In case of non-stationary problems in two-phase flows, however, there is no known analytical solution as for most of other unsteady flow which is governed by incompressible Navier-Stokes equations. As such, most of the numerical methods introduced for multiphase flows stick to qualitative comparisons against experimental results to see whether the final steady shape of a bubble or droplet resembles that of the experiments, especially those given by Clift et al. [32], Bhaga and Weber [22] and Hartunian [51]. The drawback of being limited to these so-called *picture norm* studies is that in most cases, the exact initial and boundary conditions of the experiments are often not given or known, such that one can not directly consider them in the numerical implementations. Moreover, as for example for a rising bubble problem, experimental data do not provide any information regarding the temporal change in the bubble shape or velocity over the unsteady course of the experiment. Consequently, efforts have been made in the recent years to define two-phase flow benchmark problems which consist of simple configurations and specific initial and boundary conditions, where the performance of any numerical scheme could be clearly quantified based on measuring certain physical properties [55–58, 128]. This approach helps to remove confusion in comparing different numerical tools since, in addition to providing picture norm results, certain benchmark quantities and their associated error norms and convergence rates can be studied to evaluate the underlying numerics.

Unfortunately, benchmark computations have rarely been addressed for LB-based multiphase simulations, and the majority of the studies tend to present qualitative comparisons against experimental data for terminal shape of rising bubbles [9, 31, 85], or against other LB solutions, e. g., the works of Banari et. al [18] and Wang et. al [131, 132] for Rayleigh-Taylor instability and droplet splashing on a liquid surface. While the flow configurations of such problems are not uniformly defined among different works, the later two problems in particular involve complex interface deformations where the flow quantities tend to greatly vary with only small changes in the input values and thus make it hard to judge the performance of certain LB approaches.

In this chapter, the widely-known benchmark problems of rising bubbles proposed in [57], [128] and [6] are used to evaluate the accuracy of the coupled LB-LS schemes in 2D and 3D cases against the direct numerical solution of the two-phase incompressible Navier-Stokes equations. To this end, the problem configurations in two and three dimensions are first described and the different parameters used to quantify the flow are introduced. Before presenting the benchmarking results, regulating the relaxation parameters preliminarily discussed in chapter 5, is analysed in more detail to pick the best relaxation values for a stable solution in the case of unsteady flows. Then, the performance of the coupled schemes for low and high density ratio rising bubbles is studied through both qualitative and rigorous quantitative comparisons against the available macroscopic solutions. Having the benchmark solutions already in hand, also facilitates studying the role of

arithmetic precision, lattice stencil and the use of refined level set grids on the overall accuracy of a coupled LB-LS solution. Numerical experiments for some non-benchmark flows involving high Re numbers and/or large deformation rates are finally presented towards the end of this chapter.

6.1. Rising bubble benchmarks

The problem introduction and configuration, as well as the definition of benchmark quantities and flow characteristics are described in this section.

6.1.1. Problem definition

The 2D rising bubble benchmark problem, first introduced in the work of Hysing et al. [58], consists of a rectangular box of size 1×2 filled with a fluid of density ρ_1 . A circular bubble of density ρ_2 and radius $r_0 = 0.25$ is initially located at $(x, y) = (0.5, 0.5)$ inside the box. The bubble is allowed to rise due to the gravity-driven buoyancy force of $\mathbf{G} = (\rho_2 - \rho_1)\mathbf{g}$, with $\mathbf{g} = (0, -g)$, where the force acts exclusively upon the bubble. An illustration of the problem configuration is given in figure 6.1(a). The top and bottom boundaries are set to solid walls where the no-slip boundary conditions is imposed via the second order bounce-back scheme as described in [137]. The free slip boundary condition ($\mathbf{u} \cdot \mathbf{n} = 0, \mathbf{t} \cdot (\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) = 0$) is applied to the side walls.

Figure 6.1(b) also illustrates the configuration of the 3D problem, as given by Adelsberger et al. [6], where Ω_1 and Ω_2 refer to fluids 1 and 2, respectively. The problem is similar to its 2D counterpart except that all the boundaries are considered as no-slip walls. Moreover, due to the symmetry of the problem in the horizontal plane, only 1/4 of the problem is solved to save computational cost and memory, where the center of the bubble will be located at $(x, y, z) = (0, 0, 0.5)$ and the symmetry boundary condition is imposed on $x = 0$ and $y = 0$ planes.

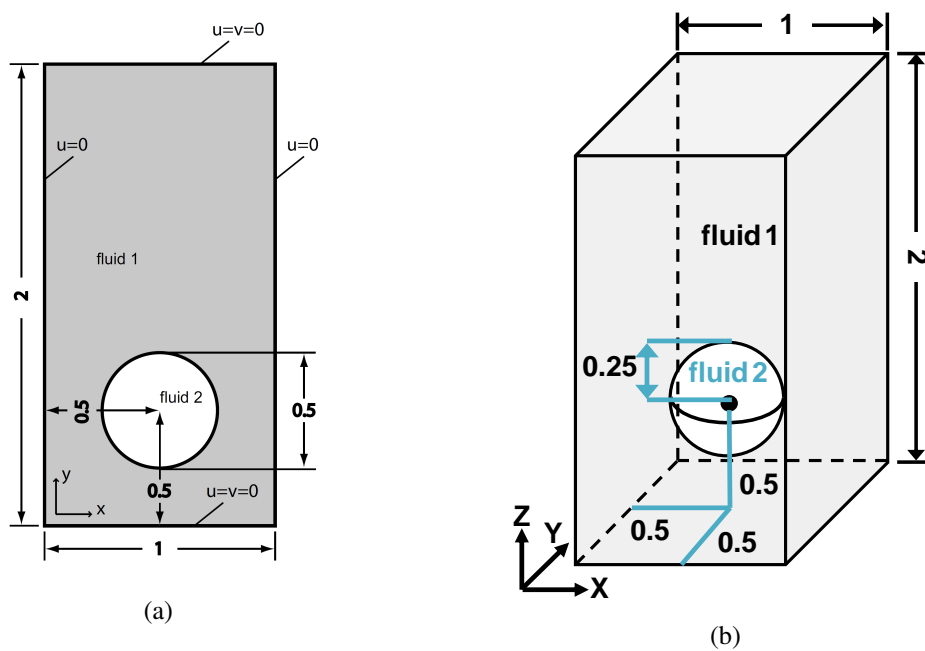


Figure 6.1: Illustration of the rising bubble problems in (a) 2D from [58], and (b) 3D.

6.1.2. Classification

Clift et al. [32] have discussed the different situations where the interaction of the viscous, inertia and surface tension forces leads to distinguishable shapes of rising bubbles which could be classified as distinct flow regimes based on three non-dimensional numbers, namely Eötvös number Eo , Morton number Mo , and bubble Reynolds number Re . The Eo number is the ratio of the gravity-driven inertia forces to the surface tension forces

$$Eo = \frac{4\rho_1 g(r_0)^2}{\sigma}.$$

Non-dimensionalization for the ratio of viscous to surface tension forces gives the Morton number

$$Mo = \frac{g\mu_1^4(\rho_1 - \rho_2)}{\sigma^3\rho_1^2}.$$

The Re number eventually shows how strong the inertia forces are compared to the viscous forces

$$Re = \frac{\rho_1\sqrt{g}(2r_0)^{3/2}}{\mu_1}.$$

To specify the flow regime, one could only rely upon the Eo and Re numbers and avoid the Mo number since the exponents of μ and σ imply large variations in the Mo number as shown in figure 6.2. One must note that r_0 must be replaced with $r_0^* = r_0/h$ if the non-dimensional numbers are defined in LB units. By setting Eo and Re numbers as input and choosing the values of ρ_1, ρ_2 and μ_1, μ_2 in lattice units, the corresponding values of g_{lb} and σ_{lb} could be simply derived based on the above definitions and are thus dependent on the lattice resolution

$$g_{lb} = \left(\frac{\mu_1 Re}{\rho_1 (2r_0^*)^{3/2}} \right)^2, \quad \sigma_{lb} = \frac{4\rho_1 g_{lb} (r_0^*)^2}{Eo}.$$

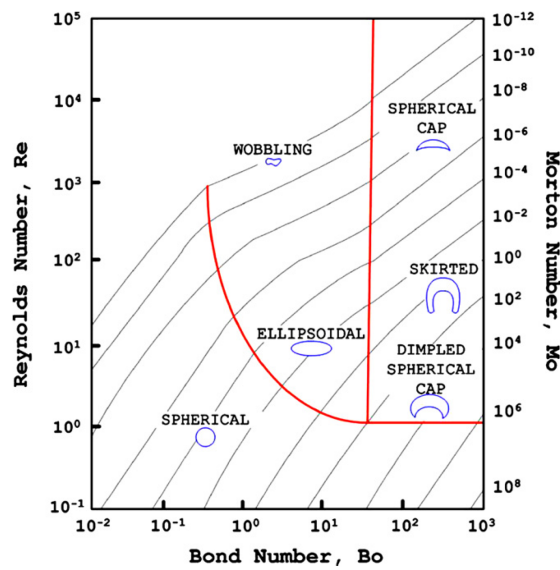


Figure 6.2: Bubble shape regime map from [32].

6.1.3. Parameters

Based on the geometry and boundary conditions described for the 2D and 3D flows, two test case problems are defined which from now on are referred to as TC1, TC2 in 2D and 3D, giving a total of four problems. The macroscopic input values pertained to TC1 and TC2 are summarized in table 6.1 where the density and viscosities are directly applied in the LB computations, while the values of σ and g have to be converted with respect to lattice resolution as described above. Additionally, to preserve consistency with the macroscopic time measurement in [56] and [6], the following equation is used to calculate the macroscopic time based on LB iteration number t_{lb}

$$T = t_{lb} \sqrt{\frac{g l b}{g L_0^*}}$$

where $L_0^* = 1/h$ is the characteristic length in lattice units. For both test cases, simulations are carried until $T = 3$.

Table 6.1: Physical parameters and dimensionless numbers for TC1 and TC2, applied to rising bubble problems.

Test case	ρ_1	ρ_2	μ_1	μ_2	g	σ	Re	EO	ρ_1/ρ_2	μ_1/μ_2
TC1	1000	100	10	1	0.98	24.5	35	10	10	10
TC2	1000	1	10	0.1	0.98	1.96	35	125	1000	100

Similar to the test cases defined for the static bubbles in chapter 4, the two rising bubble test cases are designed to examine the performance of the numerical solutions under low and high density and viscosity differences as well as for different ratios of inertia to surface tension forces. As such, the first case presents a low density ratio two-phase system, e. g., in water and oil mixtures where setting $EO = 10$ and $Re = 35$ implies strong surface tension effects which tend to retain the circularity of the bubble as also seen in the bubble shape map of figure 6.2. In contrast, the second test case imposes high density and viscosity ratios of 1000 and 100, respectively, which are pretty close to the characteristic values of an air-water system. According to the bubble map, the increased Eötvös number of $EO = 125$ allows for larger deformation of the bubble interface since the inertia effects outweigh the surface tension forces. The Re number, however, is kept constant such that the velocity and thus the bubble position will be similar in both cases, making it easier to concentrate on the two-phase flow parameters only.

6.1.4. Benchmark quantities

In order to quantify the dynamics of the bubble through its course of rise, a number of benchmark quantities are defined and used as in the following.

Bubble centroid: To track the bubble position during the rise process, the bubble centroid coordinate (x_c, y_c) is computed as

$$X_c = (x_c, y_c) = \frac{\int_{\Omega_2} \mathbf{x} dx}{\int_{\Omega_2} 1 dx}$$

where Ω_2 encompasses all the lattice points inside the bubble.

Bubble volume/area: During the simulation, it is desired that the bubble volume/area be perfectly conserved, which is, however, not the case as the interface would be displaced during the interface capturing process. Nevertheless, a piecewise linear interpolation is used to calculate

the area in 2D (accordingly, volume in 3D). As a by-product of the area/volume calculation, the length/area of the linear edges/faces, which construct the interface, can be summed up to obtain the temporal peripheral/area of the bubble.

Circularity: Having the bubble area A_b and its peripheral P_b at time t , one can calculate the bubble deformation parameter for the 2D problems, often called the circularity ϱ [129]

$$\varrho = \frac{P_a}{P_b} = \frac{\text{perimeter of area-equivalent circle}}{\text{perimeter of bubble}} = \frac{\pi d_a}{P_b},$$

where P_a denotes the perimeter or circumference of a circle with diameter d_a which has an area equal to that of a bubble with perimeter P_b . For a perfectly circular bubble or droplet, the circularity has to be equal to unity and decay as the bubble is deformed.

Sphericity: Similar to the 2D case, the roundness of a 3D bubble can be measured based on the sphericity parameter defined in [129] as

$$\Psi = \frac{A_a}{A_b} = \frac{\text{area of volume-equivalent sphere}}{\text{area of bubble}} = \frac{\pi^{1/3}(6V_b)^{2/3}}{A_b}.$$

Rise velocity: The mean velocity with which the bubble is rising is defined as

$$\mathbf{U}_c = \frac{\int_{\Omega_2} \mathbf{u} \, dx}{\int_{\Omega_2} 1 \, dx}$$

Bubble size: This last quantity describes the maximum extension (diameter) of the bubble in the main coordinate directions which together with the bubble circularity and sphericity further specifies the deformation state of the bubble. It is calculated as

$$d_i = \max_{p,q \in \Omega_2} |p_i - q_i|, \quad i = x, y, z.$$

where p and q are any two arbitrary points which belong to bubble.

6.1.5. Error estimation

In the absence of a known exact solution for the rising bubble problems, a set of relative error norms are defined so as to obtain a measure of convergence of the coupled LB-LS schemes in 2D

$$\begin{aligned} L_1 \text{ error :} & \quad ||e_1|| = \frac{\sum_{t=1}^{NTS} |q_t - q_{t,ref}|}{\sum_{t=1}^{NTS} |q_{t,ref}|} \\ L_2 \text{ error :} & \quad ||e_2|| = \left(\frac{\sum_{t=1}^{NTS} |q_t - q_{t,ref}|^2}{\sum_{t=1}^{NTS} |q_{t,ref}|^2} \right)^{1/2} \\ L_{\max} \text{ error :} & \quad ||e_{\infty}|| = \frac{\max_t |q_t - q_{t,ref}|}{\max_t |q_{t,ref}|} \end{aligned}$$

where q_t s are the temporal values of the bubble quantities and the reference quantities $q_{t,ref}$ are obtained at the finest lattice level of $1/h = 640$. For all the sums and all lattice levels, the number of time steps of $NTS = 480$ is used, such that the quantities are evaluated at uniformly identical time stamps.

6.1.6. Review of the numerical tools

The overall architecture of the different numerical tools including the present solvers and those used for benchmarking studies are shortly reviewed in the following.

Present models: The one-fluid (OF) and pressure evolution (PE) coupled LB-LS schemes, described in chapter 5, represent the models used for the present LB-based simulations. The forces in the LBE of the one-fluid scheme are discretized using directional averaged method for both t and $t + \Delta t$ (DAA), while the pressure evolution scheme uses directional central differencing for $t + \Delta t$ and directional upwind differencing for t (DCU). The one-fluid model employs a brute-force reinitialization which is called every $N_p = 25$ time steps for the base lattice of $1/h = 80$. By looking at equations (6.1.2) and (6.1.3) one infers that $\Delta t \propto h^2$, and thus consistency implies scaling N_p by a factor of 4 as the lattice is refined by a factor of 2. The pressure evolution coupled scheme relies upon its monolithic reinitialization and no explicit reinitialization is thus required. Unless otherwise mentioned, a constant relative interface thickness of $\varepsilon = 0.025$ is applied for the one-fluid model to avoid sharper pressure jumps on finer lattices, while the interface becomes thinner via prescribing $m = \varepsilon/h = 4$ for all lattice levels in the pressure evolution scheme.

FeatFlow: The FeatFlow software is a finite element solution package for solving incompressible Navier-Stokes equations. The software modules of TP2D and TP3D are designed for general transport phenomena and in particular for simulating incompressible 2D and 3D two-phase flows. The two-phase solver is based on coupling the Navier-Stokes equations with the signed distance-based LS equation. Space discretization in the 2D code consists of using rotated bilinear polynomial shape functions \tilde{Q}_1 for velocity, and Q_0 elements for pressure, while the 3D code makes use of Q_2 elements for velocity and P_1 elements for pressure. Besides, Q_1 and Q_2 elements are used for solving the LS equation in 2D and 3D codes, respectively. Time integration is out carried via a semi-implicit time stepping in the 2D code, while a Crank-Nicolson scheme is used in the 3D implementations. Moreover, brute-force reinitialization is used to keep the LS function close to signed distance values in both 2D and 3D versions. A full description of the solvers could be found in [55, 56] for 2D and [128] for 3D, while the information on the general framework of the FeatFlow software is provided in [3].

FreeLIFE: The FreeLIFE (Free-Surface Library of Finite Element) software is also designed for finite element solution of the Navier-Stokes equations for incompressible two-phase flows. It also employs a LS-based technique for capturing the interface. The Navier-Stokes problem is addressed using $P1$ -iso $P2$ elements for the velocity and $P1$ for the pressure. The sub-grid topology associated with the $P1$ -iso $P2$ element is also exploited for the solution of the LS transport equation [26]. Reinitialization is carried out via a local reconstruction (based on L2-projection) of the distance function near the interface and a fast marching technique for the far field [27, 93]. It must be noted that the results of FreeLIFE are only limited to the 2D test cases.

MoonMD: Another set of results for the 2D cases are provided by the MoonMD package [63] which is based on mapped finite element methods for discretizing partial differential equations. It solves the incompressible Navier-Stokes equations by inf-sup stable isoparametric finite elements [43]. For the two-phase flow simulations it employs an arbitrary Lagrangian-Eulerian (ALE) approach. The velocity components are discretized on simplex grids by quadratic basis functions enriched with cubic bubble functions, and the pressure by discontinuous piecewise linear elements. The movement of the interface was done in a Lagrangian manner after which the inner mesh points were fitted to the new interface by an elastic mesh update. More details on the two-phase flow framework in MoonMD could be found in [40].

NaSt3D: The NaSt3D software solves the Navier-Stokes equation on an equidistant grid. Chorin's projection method is used to decouple velocity and pressure fields, where a 2nd order explicit Adams-Bashforth scheme is employed to solve the velocity while the pressure is recovered via solving the Poisson equation. The interface is captured through a signed distance LS function

and the bubble mass is corrected using the local correction scheme of Sussman and Fatemi [119]. Additional information on the solution algorithm of two-phase flows using NaSt3D is provided in the work of Adelsberger et al. [6].

A summary of the most noticeable features of the different numerical packages described above is presented in table 6.2.

Table 6.2: Summary of the different numerical tools used for rising bubble benchmarking

Solver	Developer	Problem Dimensions	Flow solver	Time Integration	Interface Capturing
Present, OF	LS3, TU Dortmund	2D	LBM	explicit	phase-field LS
Present, PE	LS3, TU Dortmund	2D, 3D	LBM	explicit	sharp LS
FeatFlow	LS3, TU Dortmund	2D, 3D	FEM	semi-implicit	sharp LS
FreeLIFE	IAN, EPFL	2D	FEM	implicit	Lagrangian
MoonMD	IACS, Uni. Magdeburg	2D	FEM	implicit θ	sharp LS
NaSt3D	INS, Uni. Bonn	3D	FDM	explicit	sharp LS

6.2. Relaxation time regulation

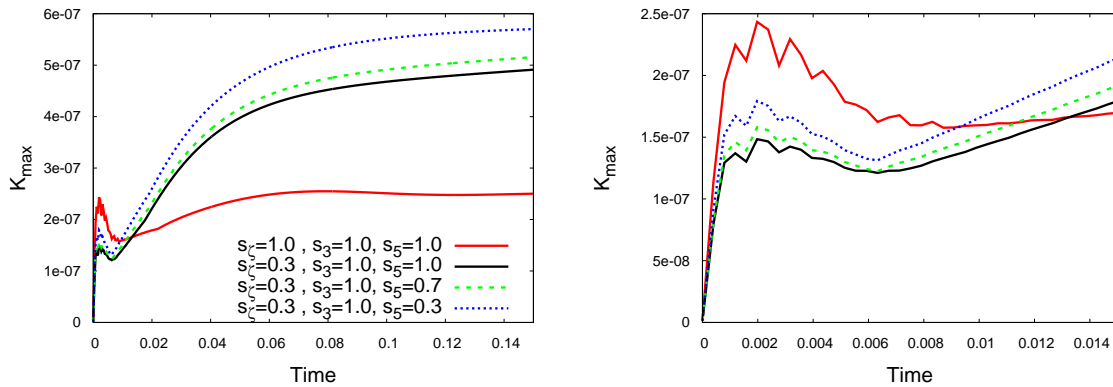


Figure 6.3: Evolution of the maximum kinetic energy K_{max} for the for the 2D static bubble STC2 using the one-fluid scheme (left) and the enlarged view (right).

To find optimum values for the relaxation times s_α , the experiments here will be limited to 2D static and dynamic problems with D2Q9 LB stencil, where the effect of the free parameters s_ζ, s_3, s_5, s_7 are investigated on the stability and accuracy. The associated moments are the energy e , the square of energy e^2 , and the fluxes of energy q_x and q_y , respectively. Preserving the symmetry then requires to have $s_5 = s_7$. The rest of the free relaxation parameters for the 3D discrete models are either pertained to the moments of similar origin as the above four key moments, which thus take identical values, or their relaxation rate does not significantly affect the solution and hence are set to 1. Note that the relaxation s_v for the viscous stress mode is derived based on the linear formulation of τ for the one-fluid method and the inverse linear formulation for the pressure evolution scheme as discussed in chapter 5. For all the experiments in this section a lattice resolution of $1/h = 160$ is used.

Figures 6.3 and 6.4 show the evolution of the maximum kinetic energy $K_{max} = |\mathbf{u}|^2/2$ for the high density ratio static bubble problem STC2 defined in chapter 5. It is clear that by under-relaxing the energy mode through choosing $s_\zeta < 1.0$, faster convergence to steady state could be

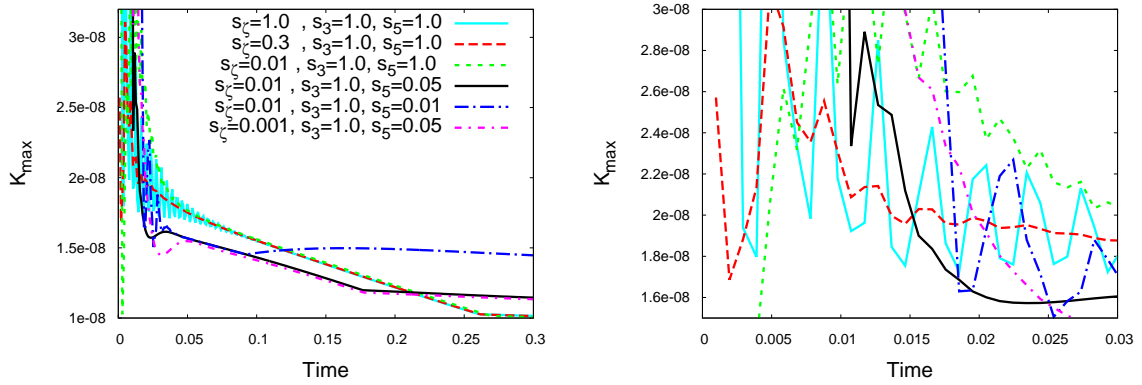


Figure 6.4: Evolution of the maximum kinetic energy K_{max} for the 2D static bubble STC2 using the pressure evolution scheme (left) and the enlarged view (right).

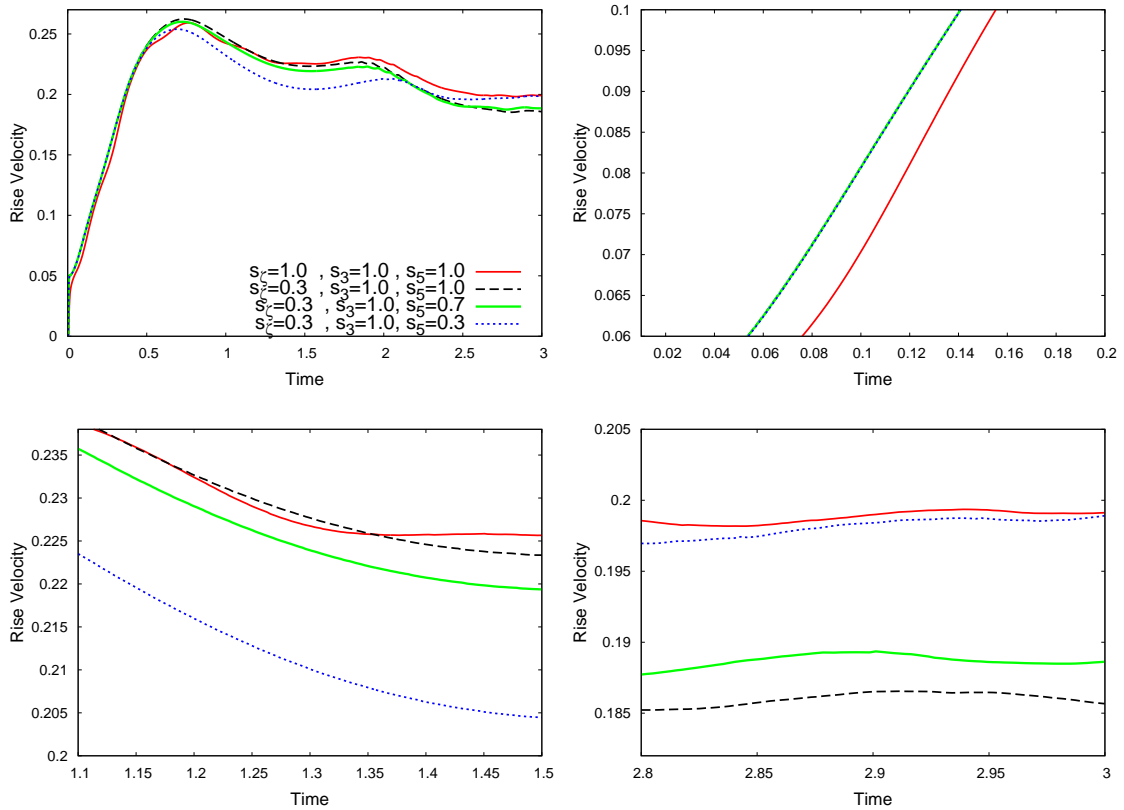


Figure 6.5: Rise velocity for TC2 using the one-fluid method and different choices of the free relaxation parameters.

achieved. Stability analysis for the minimum values gives $s_\zeta \geq 0.3$ in the one-fluid method and a less stringent limit of $s_\zeta \geq 0.01$ in the pressure evolution scheme. Applying the same principle to the e^2 , q_x and q_y moments, however, shows no positive effect on the convergence in the static bubble since the velocities are very small and there exists no strong bulk current.

The influence of different relaxation settings for the rising bubble problem of TC2 is studied in figures 6.5 and 6.6. The favourable effect of choosing a small s_ζ , with the same minimum limits

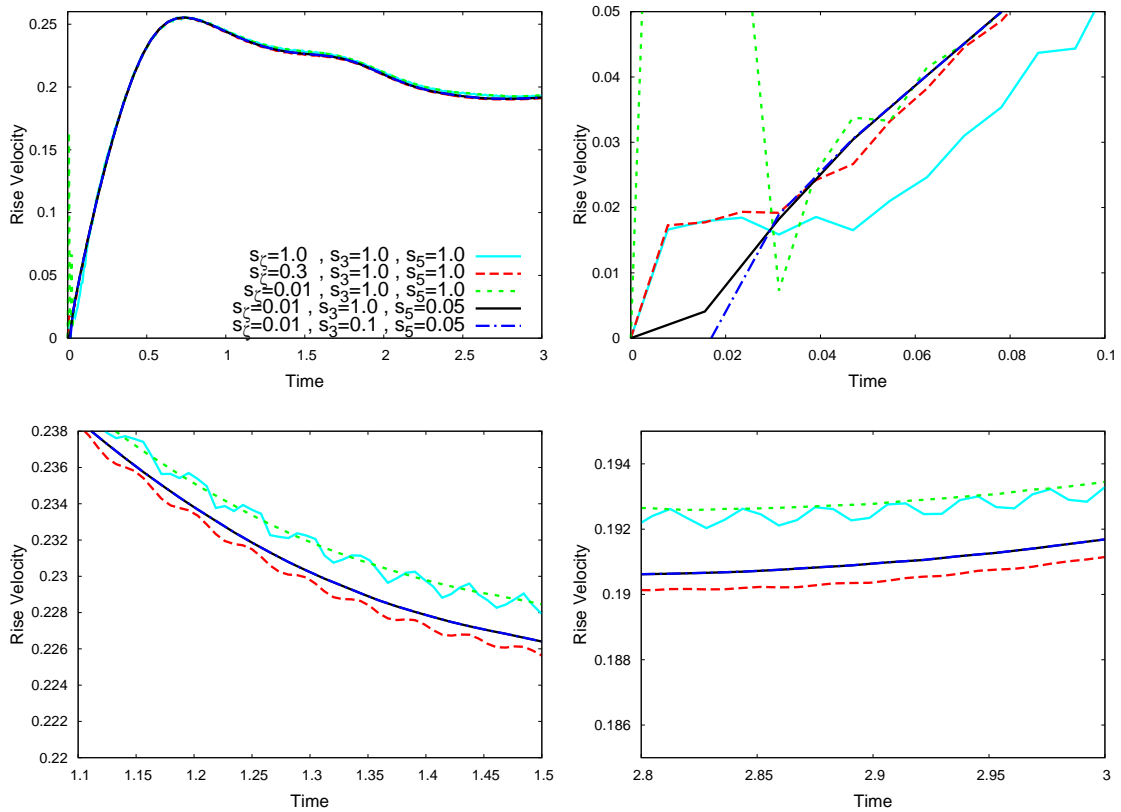


Figure 6.6: Rise velocity for TC2 using the pressure evolution method and different choices of the free relaxation parameters.

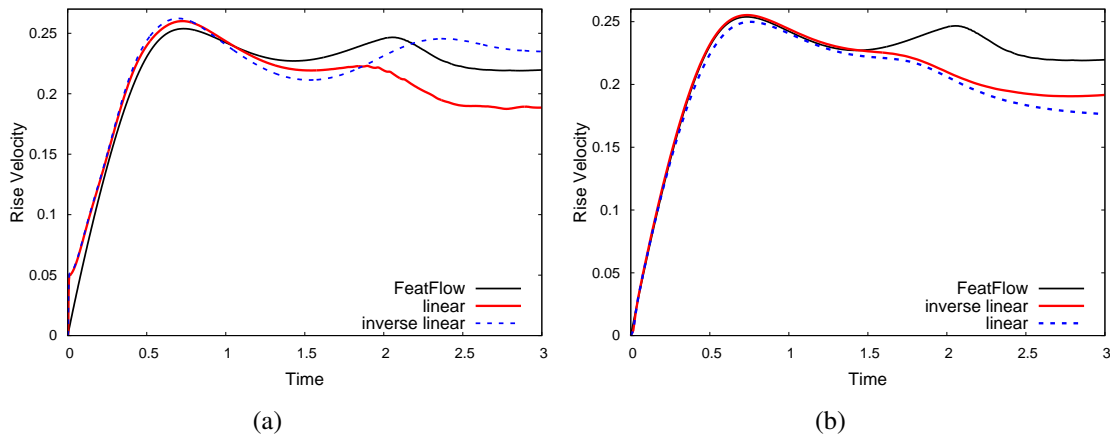


Figure 6.7: Effect of using different formulations for τ on the rise velocity for (a) one-fluid and (b) pressure evolution models.

as in the static case, is valid in the dynamic problem. In addition, under-relaxing the energy flux mode tends to slightly suppress the initial velocity fluctuations which are triggered as the bubble starts to rise through the surrounding stationary fluid. Nevertheless, further under-relaxation must be applied with care as it tries to underestimate the rise velocity during the rest of the rise process. With regards to the e^2 mode, experiments show that reducing s_3 brings no improvement to the

velocity evolution and choosing values smaller than 0.7 results in negative initial velocities and thus eventual divergence of the solution.

The above analysis for the four crucial moments ultimately leads to the following choice of the relaxation parameters in rising bubble test cases

$$\hat{\Lambda} = \text{Diag}\{0, 0.3, 1.0, 0, 0.7, 0, 0.7, 1/\tau, 1/\tau\}$$

for the D2Q9 stencil and the one-fluid scheme, whereas the choice for the pressure evolution scheme reads

$$\hat{\Lambda} = \text{Diag}\{0, 0.01, 1.0, 0, 0.05, 0, 0.05, 1/\tau, 1/\tau\}.$$

Extension of the above choice for the 3D discrete models for the pressure evolution method looks like the following

$$\hat{\Lambda} = \text{Diag}\{0, 0.01, 1.0, 0, 0.05, 0, 0.05, 0, 0.5, 1/\tau, 1/\tau, 1/\tau, 1/\tau, 1/\tau, 1/\tau\}$$

for the D3Q15 stencil and

$$\hat{\Lambda} = \text{Diag}\{0, 0.01, 1.0, 0, 0.01, 0, 0.01, 0, 0.05, 1/\tau, 1.0, 1/\tau, 1.0, 1/\tau, 1/\tau, 1/\tau, 1.0, 1.0, 1.0\}$$

for the D3Q19 stencil. Finally for the D3Q27 stencil the choice will be

$$\hat{\Lambda} = \text{Diag}\{0, 0, 0, 0.01, 1/\tau, 1/\tau, 0.05, 0.05, 1.0, , 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1/\tau, 1/\tau, 1/\tau, 0.05, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0\}.$$

Lastly, the choice between linear and inverse linear formulations for the derivation of $\tau = 1/s_v$ is investigated in figure 6.7. The results confirm the default selection by comparing the velocities to the benchmark values from Featflow [56]. In particular, the one-fluid model shows great deviations from the benchmark solution if an inverse linear correlation is used to evaluate τ .

6.3. 2D rising bubble benchmarking

This section elaborates on the eventual benchmarking of the bubble quantities using the present coupled LB-LS schemes against the reference solutions of the Navier-Stokes equations.

6.3.1. Test case 1

Based on the bubble shape map, the bubble gains an ellipsoidal shape by $T = 3.0$ in TC1 as depicted in figure 6.8 and 6.9 using a lattice resolution of $1/h = 320$. The two coupled models closely follow the pattern given by the finite element solution in [58], also obtained on a $1/h = 320$ grid, for this surface tension-dominated test case.

For the steady state solution at $T = 3.0$, figure 6.10 also shows the convergence trend in the bubble interface shape. Figures 6.11 to 6.16 show the temporal evolution of the bubble quantities on different lattice levels compared to the reference solution from [56]. Note that the bubble velocity reaches its maximum around $T = 1$ and decays afterwards. Such a trend is due to the reflected pressure waves from the upper wall which hit the bubble as they descend and decelerate the upward motion of the bubble. As the inertia effects diminish, the surface tension forces take their chance to recover the bubble's circular shape as evident by the increase in the circularity around $T = 2$. The large fluctuations in the circularity values for $1/h = 80$ produced by the one-fluid model in figure 6.13, are caused by the periodic reinitializations which tend to relocate the interface and hence yield non-monotonic variations in the circularity.

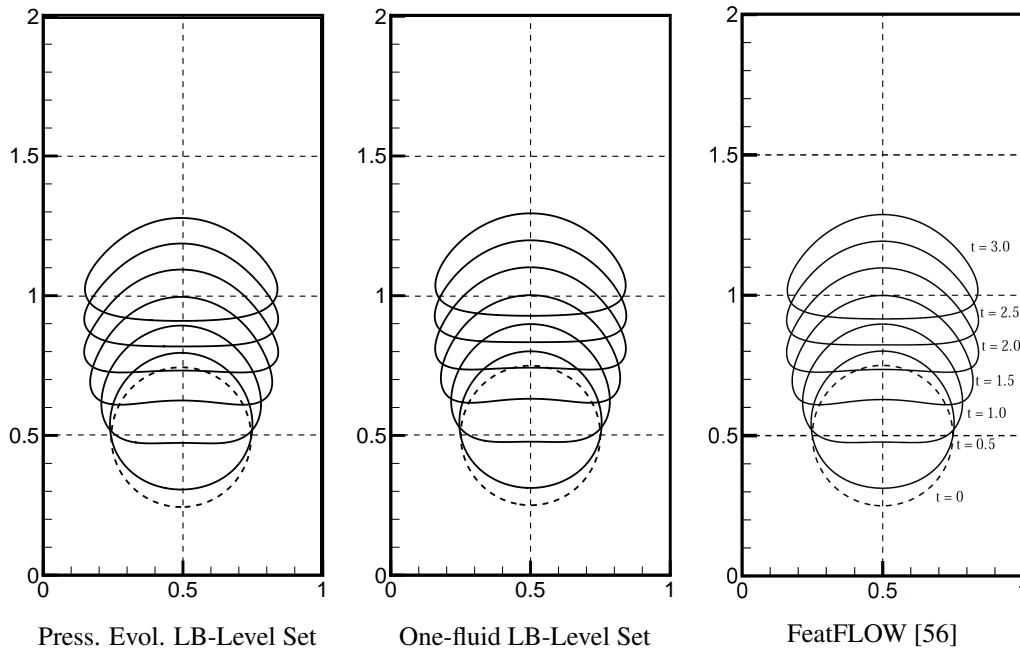


Figure 6.8: Time evolution of the bubble shape for TC1

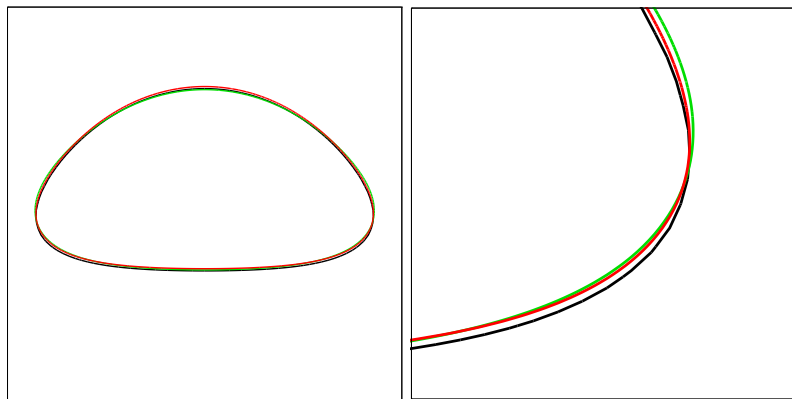


Figure 6.9: Terminal shape of the bubble at $T = 3$ for TC1 with the enlarged view (right), using one-fluid method (green), pressure evolution method (red) and FeatFlow (black).

The accuracy of the present coupled solutions based on relative errors and the corresponding rates of convergence (ROC) is studied in tables 6.3 and 6.4 for the one-fluid and pressure evolution schemes, respectively. Although the pressure evolution method does not produce fluctuations on coarse lattices, its dependence on the lattice resolution has clearly led to greater errors on $1/h = 80$ as compared to the one-fluid scheme. This is more evidently reflected in the large deviations in the temporal values of circularity on $1/h = 80$ in figure 6.16. Nevertheless, the rates of convergence for both schemes are mostly between 1.5 and 2.0, thus confirming the second order accuracy of the solutions. Deviations from the rate of 2 in a few cases are partly due to the low order of post processing calculations used for the interface reconstruction in both schemes together with the loss of mass due to the LS reinitialization in the one-fluid approach.

Figures 6.18 through 6.19 present a comparison between the converged bubble quantities obtained on $1/h = 320$ using the coupled LB-LS schemes and the reference values from [56]. As the

low density ratio poses no severe numerical difficulty, both sharp and diffuse CSF implementations of the surface tension forces are successful and discrepancies are quite negligible as compared to the direct numerical solution of the Navier-Stokes equation.

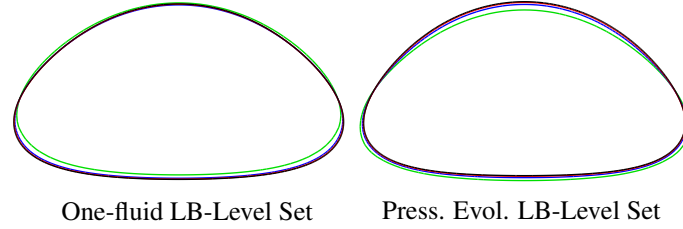


Figure 6.10: Bubble interface at time $T = 3$ for TC1 on different lattice resolutions; $1/h = 80$ (green), $1/h = 160$ (blue), $1/h = 320$ (red), $1/h = 640$ (black).

Table 6.3: Errors in bubble quantities for TC1 using the one-fluid coupled scheme

$1/h$	$\ e_1\ $	ROC_1	$\ e_2\ $	ROC_2	$\ e_\infty\ $	ROC_∞
Rise Velocity						
80	1.9e-2		3.3e-2		3.6e-2	
160	6.3e-3	1.65	1.0e-2	1.68	1.4e-2	1.3
320	1.9e-3	1.68	3.3e-3	1.66	3.5e-3	2.0
Center of mass						
80	7.0e-3		3.1e-2		1.3e-2	
160	2.0e-3	1.77	8.4e-3	1.91	3.0e-3	2.14
320	7.3e-4	1.48	2.9e-3	1.51	1.1e-3	1.45
Circularity						
80	1.5e-3		2.6e-3		3.1e-3	
160	5.1e-4	1.59	9.2e-4	1.51	1.2e-3	1.29
320	2.1e-4	1.27	3.8e-4	1.26	4.5e-4	1.46

Table 6.4: Errors in bubble quantities for TC1 using the pressure evolution coupled scheme

$1/h$	$\ e_1\ $	ROC_1	$\ e_2\ $	ROC_2	$\ e_\infty\ $	ROC_∞
Rise Velocity						
80	1.9e-2		3.4e-2		3.2e-2	
160	7.6e-3	1.36	1.3e-2	1.38	1.2e-2	1.44
320	2.1e-3	1.86	3.5e-3	1.91	2.9e-3	2.0
Center of mass						
80	1.3e-2		6.0e-2		2.2e-2	
160	4.9e-3	1.44	2.2e-2	1.42	8.2e-3	1.47
320	1.3e-3	1.86	5.9e-3	1.91	2.2e-3	1.9
Circularity						
80	5.8e-3		1.1e-2		1.3e-2	
160	2.3e-3	1.33	4.5e-3	1.34	5.5e-3	1.27
320	6.8e-4	1.75	1.3e-3	1.76	1.6e-3	1.74

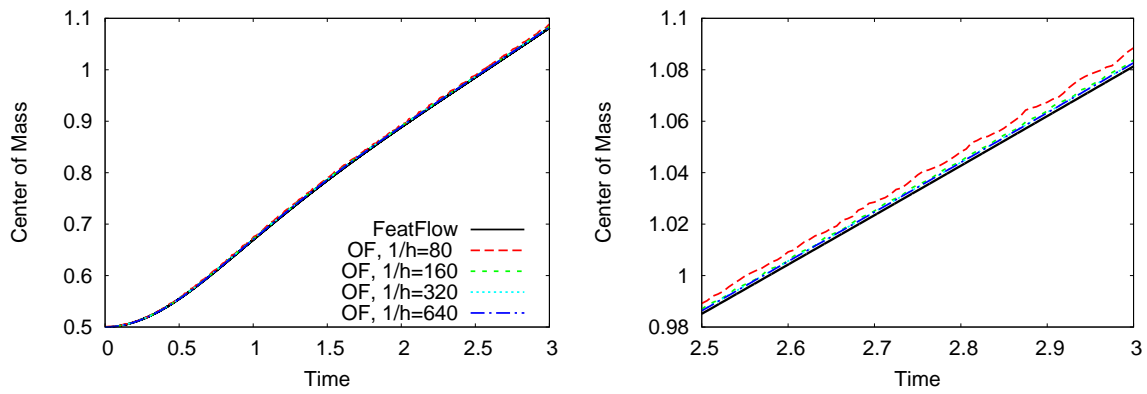


Figure 6.11: Time evolution of the center of mass for the 2D rising bubble TC1 using one-fluid coupled scheme (left) and the enlarged view (right).

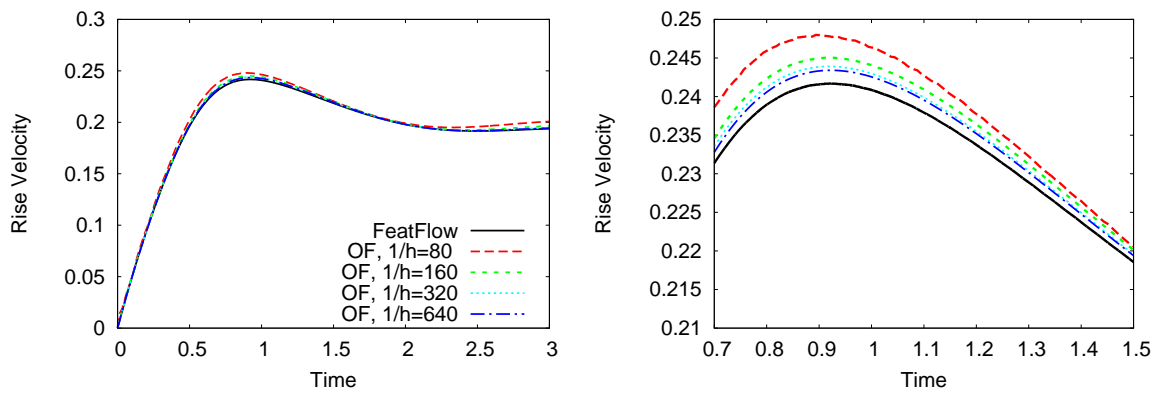


Figure 6.12: Time evolution of the rise velocity for the 2D rising bubble TC1 using one-fluid coupled scheme (left) and the enlarged view (right).

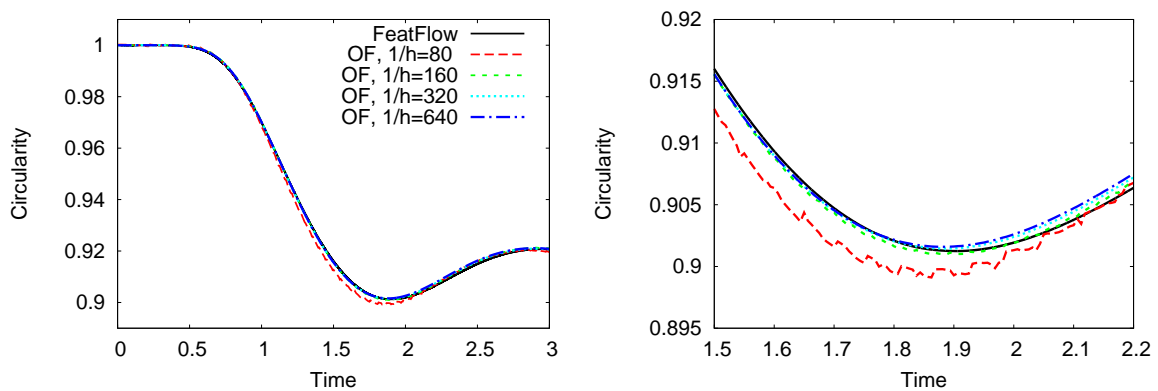


Figure 6.13: Time evolution of the circularity for the 2D rising bubble TC1 using one-fluid coupled scheme (left) and the enlarged view (right).

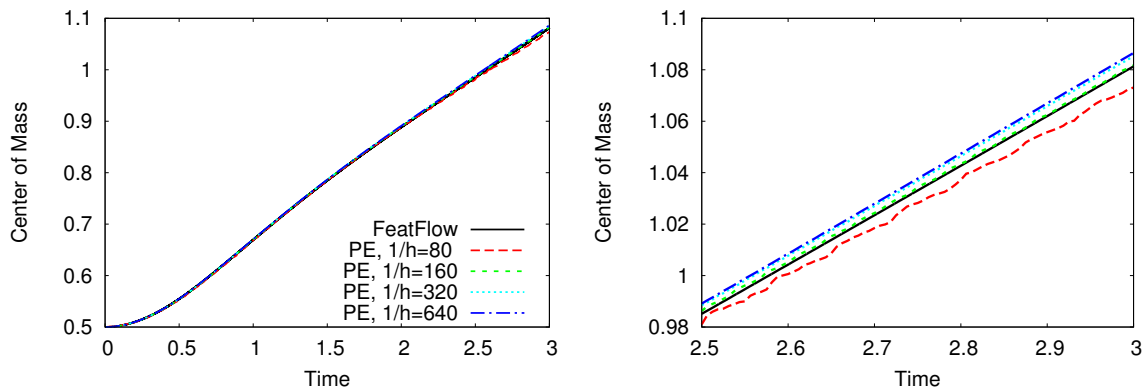


Figure 6.14: Time evolution of the center of mass for the 2D rising bubble TC1 using pressure evolution coupled scheme (left) and the enlarged view (right).

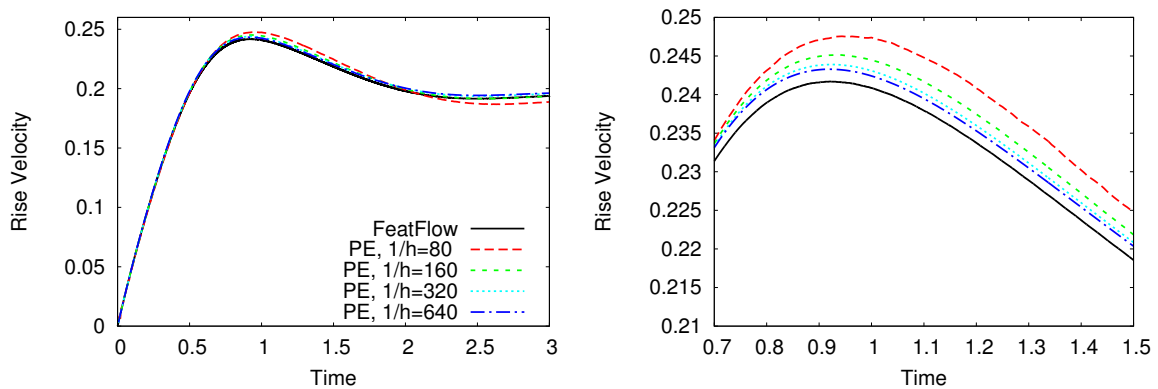


Figure 6.15: Time evolution of the rise velocity for the 2D rising bubble TC1 using pressure evolution coupled scheme (left) and the enlarged view (right).

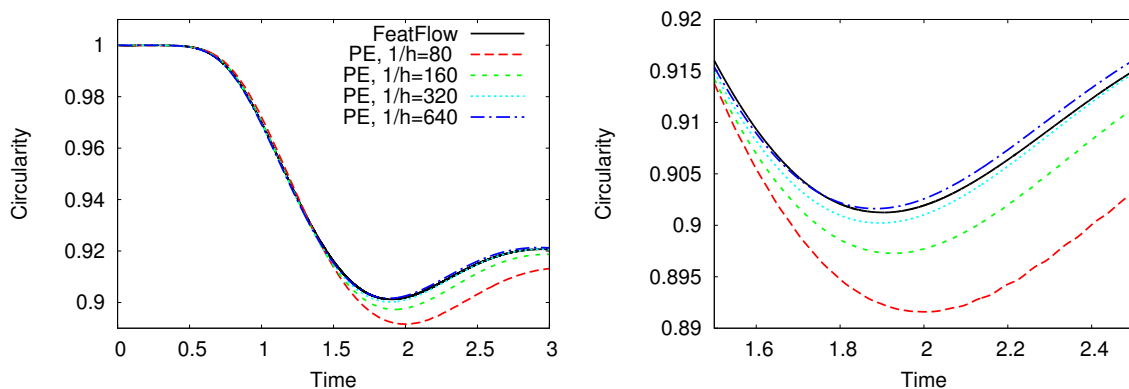


Figure 6.16: Time evolution of the circularity for the 2D rising bubble TC1 using pressure evolution coupled scheme (left) and the enlarged view (right).

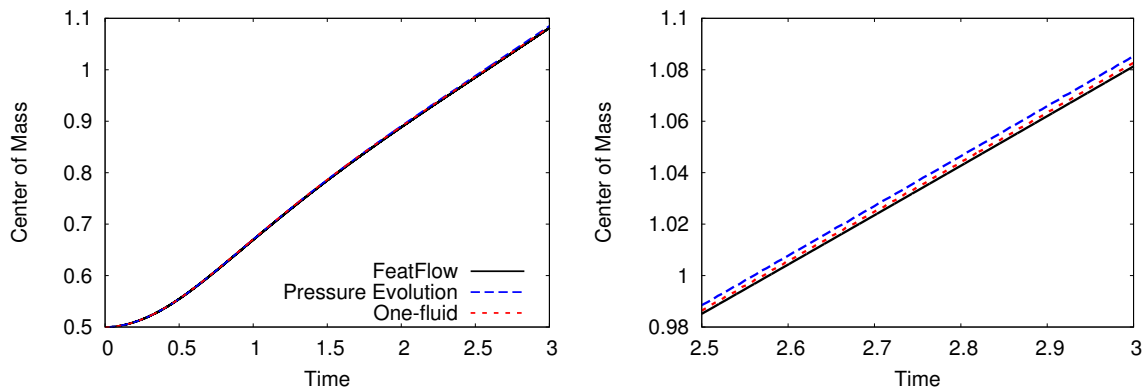


Figure 6.17: Time evolution of the center of mass for the 2D rising bubble TC1 using different schemes (left) and the enlarged view (right).

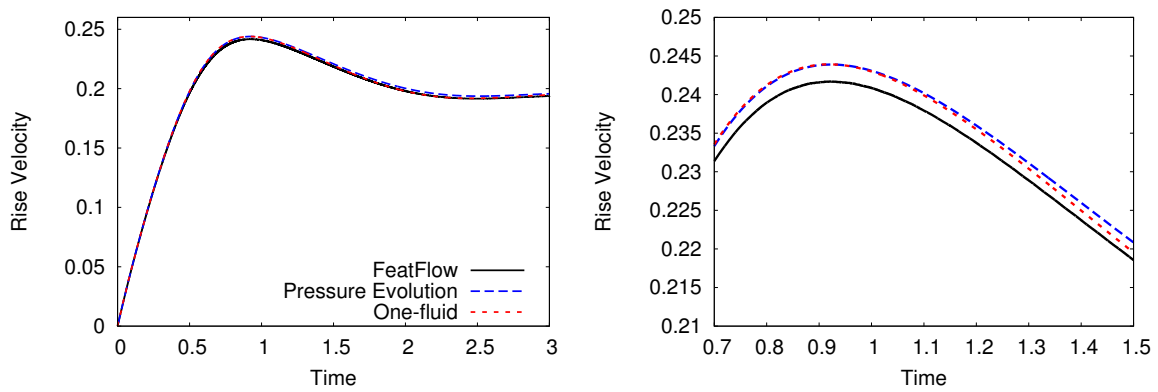


Figure 6.18: Time evolution of the rise velocity for the 2D rising bubble TC1 using different schemes (left) and the enlarged view (right).

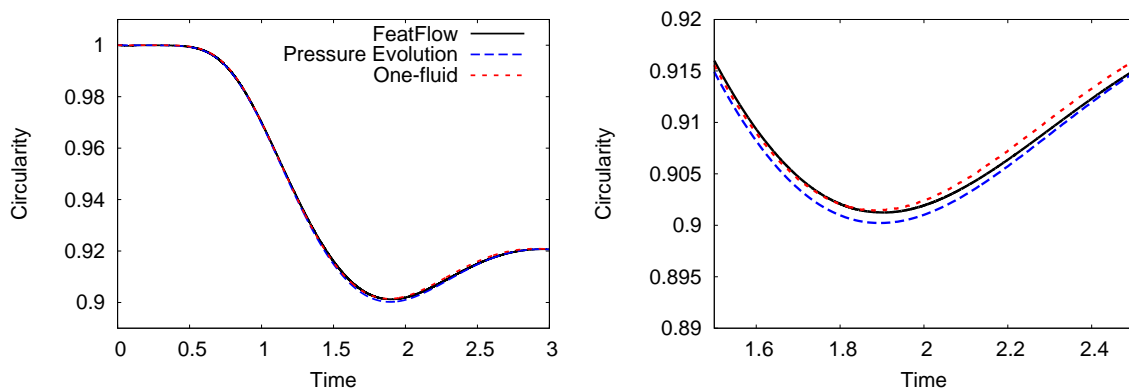


Figure 6.19: Time evolution of the circularity for the 2D rising bubble TC1 using different schemes (left) and the enlarged view (right).

6.3.2. Test case 2

By increasing the EO number in the second test case to $EO = 125$, the bubble is allowed to exhibit a more pronounced deformation since the inertial effects take over the surface tension forces which typically tend to retain the bubble shape. The general shape of the bubble predicted by all 2D solvers on $1/h = 640$, as illustrated in figure 6.20, is in accordance to the experimental classification for this range of Re and EO numbers given in figure 6.2.

Figure 6.21 illustrates the temporal evolution of the bubble interface as it rises, produced by both one-fluid and pressure evolution coupled schemes along with those obtained by FeatFlow in [56]. The terminal shapes are more closely compared in figure 6.22. All three simulations are carried out using a grid resolution of $1/h = 640$. One can see that both LB-based methods reproduce the general bubble deformation pattern which eventually gives rise to a cap-shaped bubble with two trailing filaments (in case of LBM) or satellite bubbles (in case of FeatFlow). As contrary to the pressure evolution method, the one-fluid approach which benefits from a sharp interface LSM produces much thinner trailing filaments. Moreover, the FeatFlow solution predicts the evolution of the trailing filaments into satellite bubbles. It must be emphasised that such a spin-off phenomena can not be physically verified since LS-based methods do not take into account the small-scale effects pertained to the chemical potential field in the vicinity of the interface. Consequently, non of the present data for $T > 2$ are considered as reference solutions.

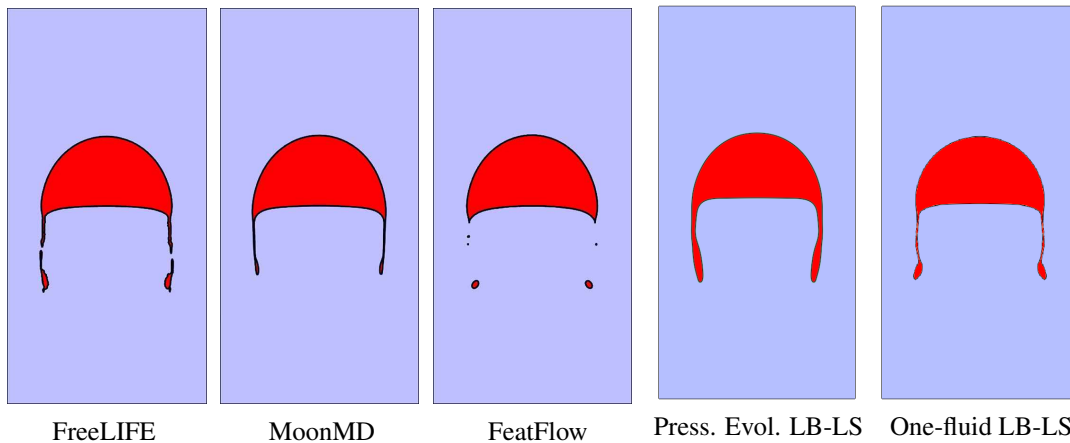


Figure 6.20: Terminal shape of the bubble at $T = 3$ for TC2.

Similar to TC1, a rather quantitative insight into the coupled LB-LS solutions is provided in figures 6.23 through 6.28 for bubble circularity, centroid position and rise velocity on different lattice levels. While prescribing the same Re number as in TC1 implies similar peak velocity and bubble position, the larger deformation of the bubble has resulted in circularity values as low as 0.48 in this test case. Note that the sudden jump of the circularity values given by FeatFlow refers to the spin-off moment, as it causes an abrupt reduction in the area of the main bubble.

A comparison of the rise velocities in figures 6.24 and 6.27 to those of TC1 reveals that the bubble velocity in TC2 attains two extreme values instead of one. The emerging of the second peak is believed to be due to the reduced drag force as the bubble sheds the trailing filaments and becomes more streamlined around $T = 2$. Consequently, the relatively stronger dependence of the diffused level set scheme on the lattice quality prevents the method to acquire the second peak even up to $1/h = 160$. Such an effect could be also noticed in figure 6.29 for the convergence of the terminal bubble shape, where the pressure evolution method requires a resolution of $1/h = 320$ to develop a deformation similar to that of a sharp interface solution. This also suggests that the

study of the converged solution may be performed using $1/h = 640$ rather than $1/h = 320$, so as to make sure the bubble will not deform any more by further refinement of the lattice.

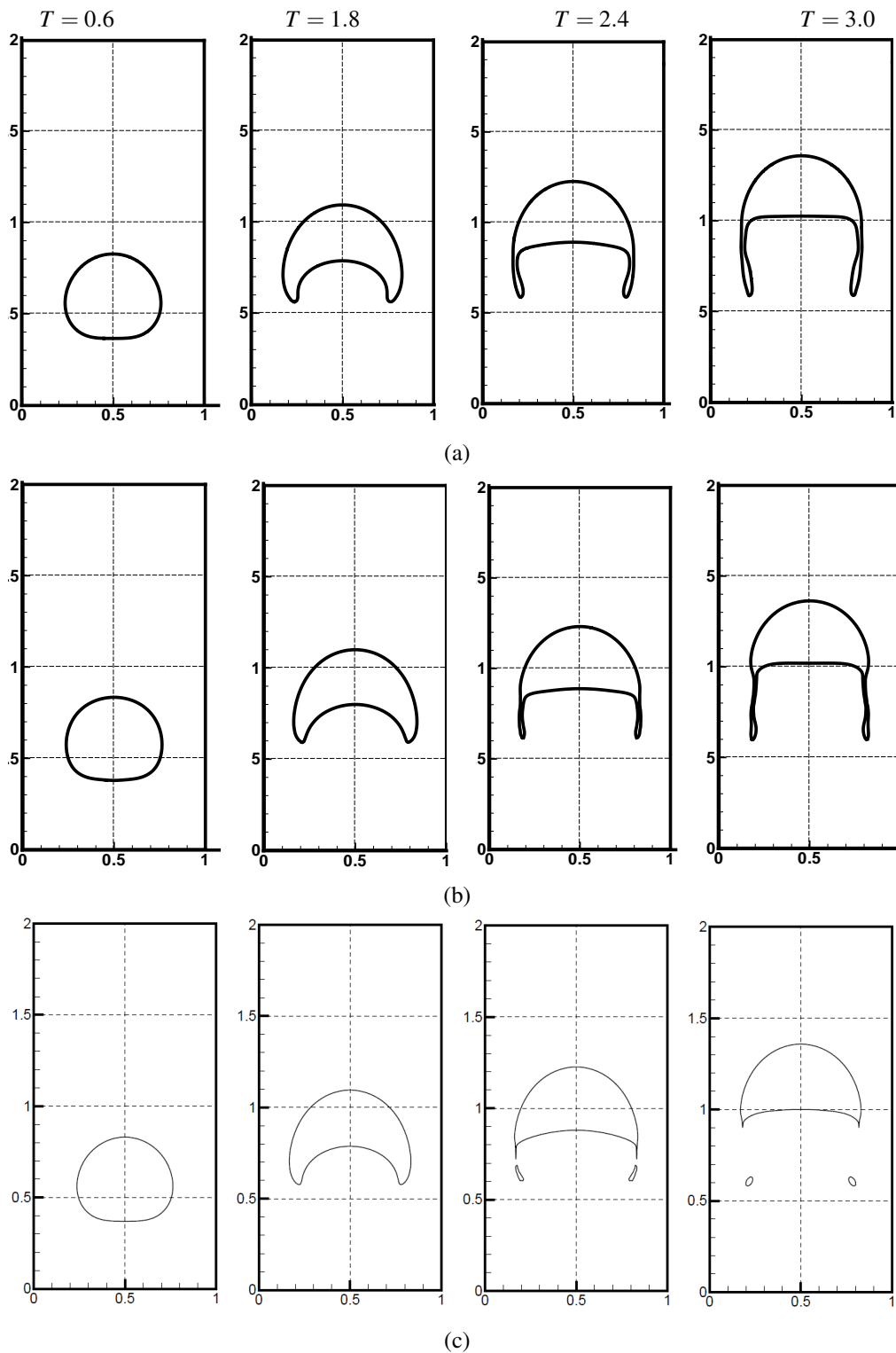


Figure 6.21: Time evolution of the bubble in test case 2 for $1/h = 640$ using (a) pressure evolution LB-levelset (b) one-fluid LB-level set (c) FeatFlow [58].

Another noteworthy observation in the obtained rise velocity results, is the initial spike in the

coarse grid solution of the one-fluid method shown in figure 6.24. The most likely reason behind this behaviour could be the weak approximation of the pressure through the linear EOS which is unable to resolve the large pressure gradients in the stagnation region right above the bubble. The issue with the pressure continues to affect the solution during the rest of the process as evident in the large deviations of the rise velocity from the converged solution obtained on coarse grids in figure 6.24. As the lattice resolution is increased and the gradients are more aptly treated, the spike is effectively eliminated and the solution follows that of FeatFlow. On the other hand, direct solution of the pressure in the pressure evolution method allows for recovering a more accurate pressure field. Consequently, there will be no overshoot in the initial velocities and the agreement with the reference solution up to $T < 1$ is fairly close even on the very coarse lattice of $1/h = 80$.

Comparison of the converged solutions of the present LB-based results against those of benchmark data is presented in figures 6.30 to 6.32 using $1/h = 640$. The sharp interface LSM used in FeatFlow, MoonMD and one-fluid computations has resulted in closer agreement between these three methods. On the other hand, the Lagrangian approach in FreeLIFE and the phase-field LSM used in the pressure evolution method predict generally lower values of circularity as seen in figure 6.32. For the rise velocity values, nevertheless, the results given by all 5 solvers closely follow each other up to $T = 2$. This shows that both present LB-based methods are able to virtually recover the Navier-Stokes solutions for the flow field provided that the underlying lattice is adequately refined.

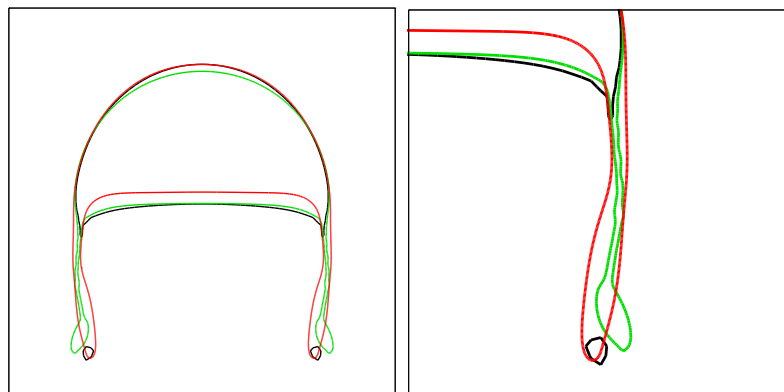


Figure 6.22: Terminal shape of the bubble at $T = 3$ for TC2 with the enlarged view (right), using one-fluid method (green), pressure evolution method (red) and FeatFlow (black).

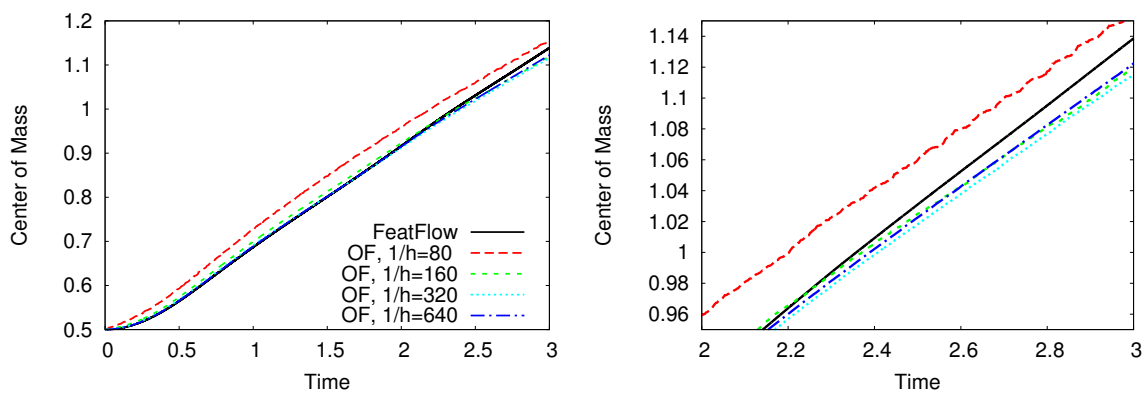


Figure 6.23: Time evolution of the center of mass for the 2D rising bubble TC2 using one-fluid coupled scheme (left) and the enlarged view (right).

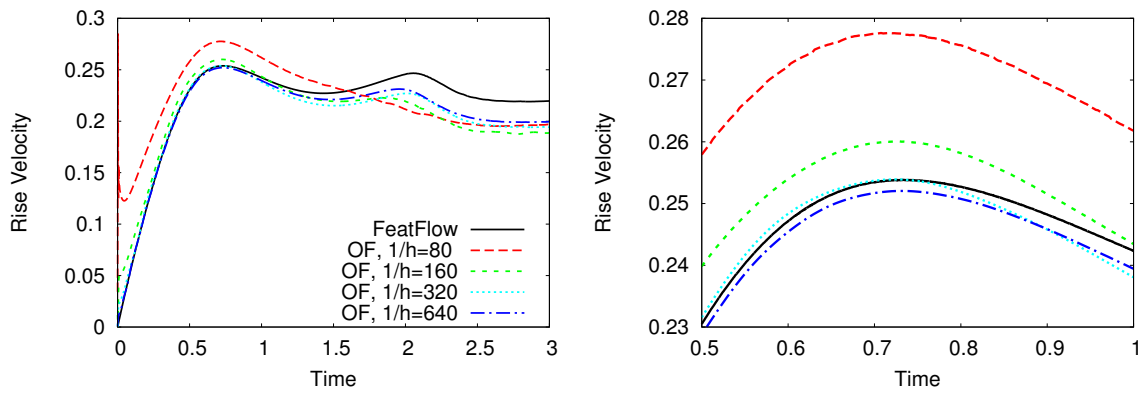


Figure 6.24: Time evolution of the rise velocity for the 2D rising bubble TC2 using one-fluid coupled scheme (left) and the enlarged view (right).

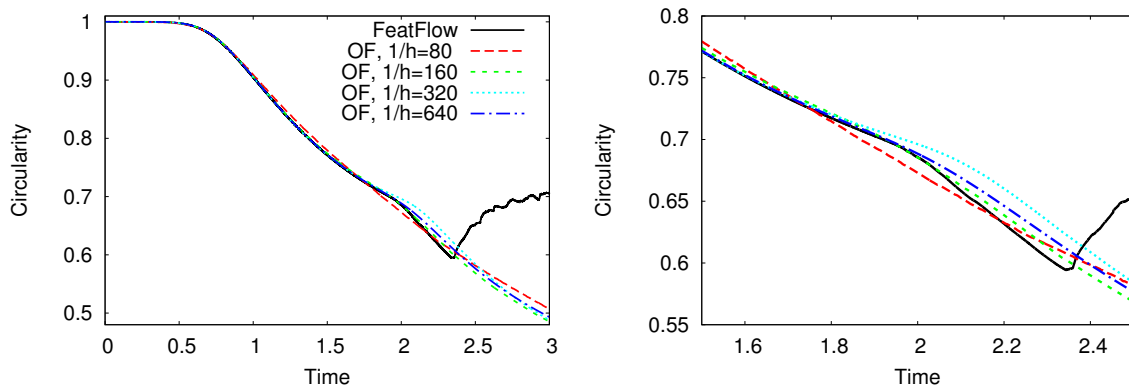


Figure 6.25: Time evolution of the circularity for the 2D rising bubble TC2 using one-fluid coupled scheme (left) and the enlarged view (right).

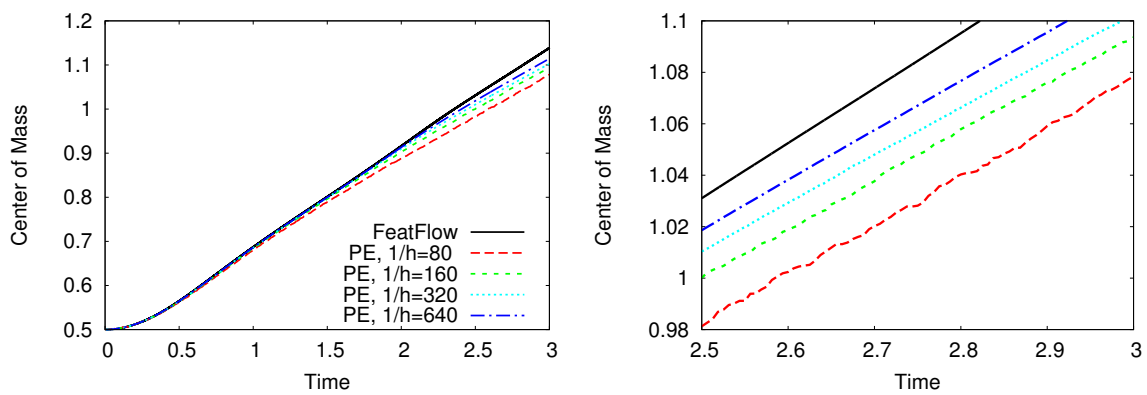


Figure 6.26: Time evolution of the center of mass for the 2D rising bubble TC2 using pressure evolution coupled scheme (left) and the enlarged view (right).

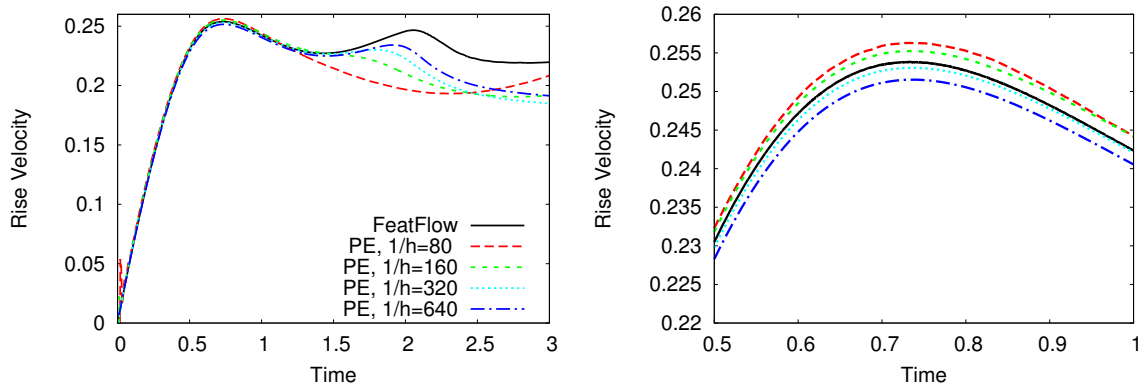


Figure 6.27: Time evolution of the rise velocity for the 2D rising bubble TC2 using pressure evolution coupled scheme (left) and the enlarged view (right).

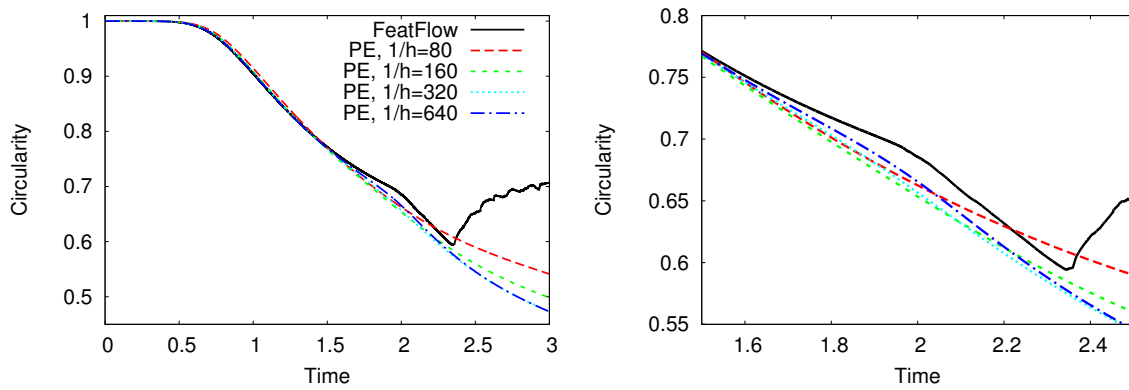


Figure 6.28: Time evolution of the circularity for the 2D rising bubble TC2 using pressure evolution coupled scheme (left) and the enlarged view (right).

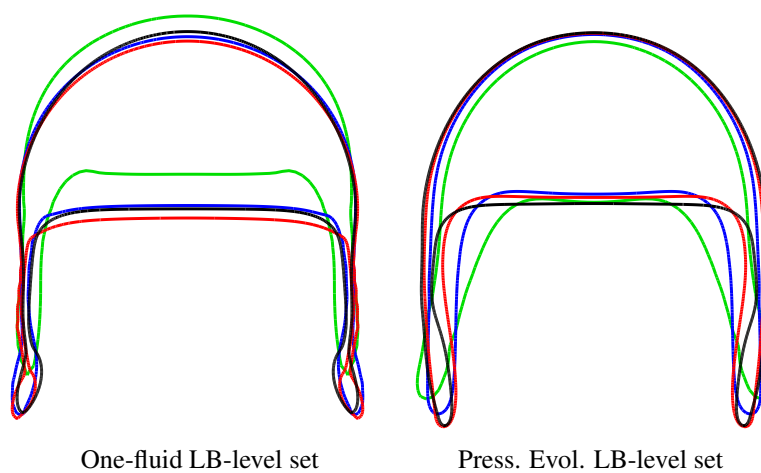


Figure 6.29: Bubble interface at time $T = 3$ for TC2 using $1/h = 80$ (green), $1/h = 160$ (blue), $1/h = 320$ (red), $1/h = 640$ (black).

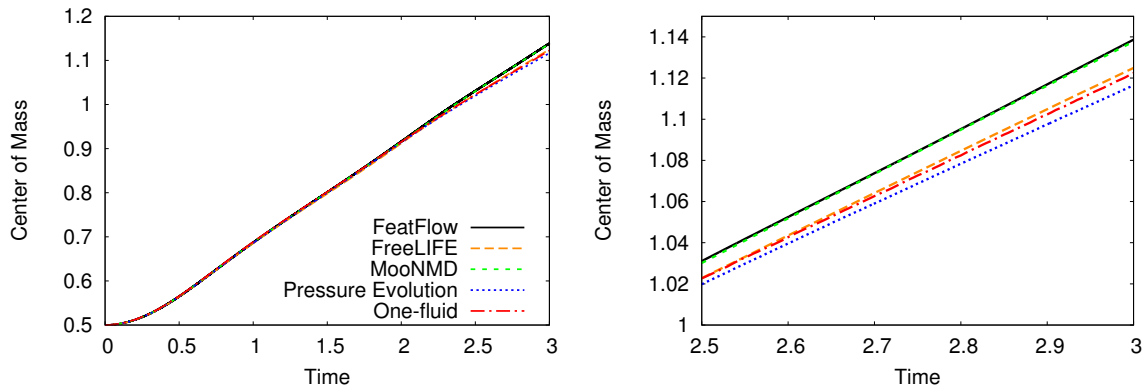


Figure 6.30: Time evolution of the center of mass for the 2D rising bubble TC2 using different schemes (left) and the enlarged view (right).

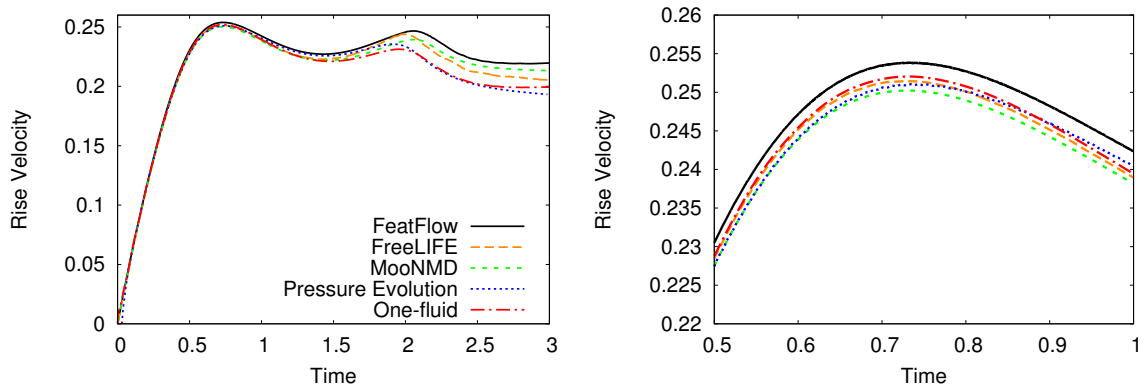


Figure 6.31: Time evolution of the rise velocity for the 2D rising bubble TC2 using different schemes (left) and the enlarged view (right).

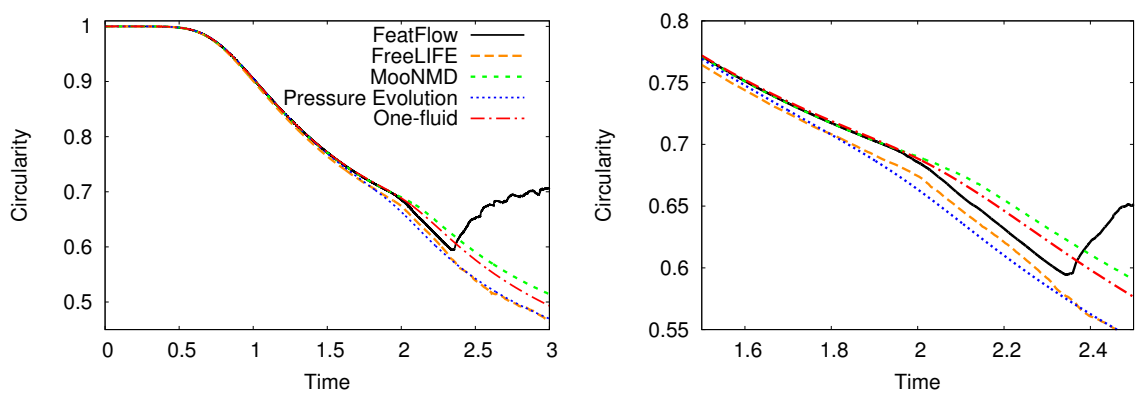


Figure 6.32: Time evolution of the circularity for the 2D rising bubble TC2 using different schemes (left) and the enlarged view (right).

6.3.3. Effect of interface thickness

In the previous section the interface thickness of $\varepsilon = 0.025$ was assumed as a constant for all lattice levels in the one-fluid approach, which ends up in an increasing number of grid points to resolve the interface on finer lattices. In contrary, a thinning interface trend was applied to the pressure evolution method through setting $m = \varepsilon/h = 4$, regardless of the resolution. These default selections were motivated by the discussions on pressure errors in the static bubble problem in section 5.4. In order to briefly investigate the effect of the interface thickness on the quality of dynamic problems, the evolution of the bubble velocity using different values of $m = \varepsilon/h$ is studied for TC2 in figures 6.33 and 6.34 for the one-fluid and pressure evolution methods, respectively. The lattice resolution is $1/h = 320$ for both coupled LB-LS schemes. The accuracy of the one-fluid scheme aggravates as the pressure gradients become sharper by reducing m from 8 to 4, as seen in the lowered values of the rise velocity in figure 6.33. Based on the discussions in chapter 4, applying a large value of $m = 8$ for the phase-field LSM in the pressure evolution method degrades the quality of the interface capturing, which causes the interface to become too much diffused and prevents it from capturing the second peak in the rise velocity shown in figure 6.34. The variations of the interface shapes in figure 6.35 support this later argument where the negligible change of the interface position and style in the one-fluid scheme is in contrast to the apparent sensitivity of the terminal bubble shape in the pressure evolution method.

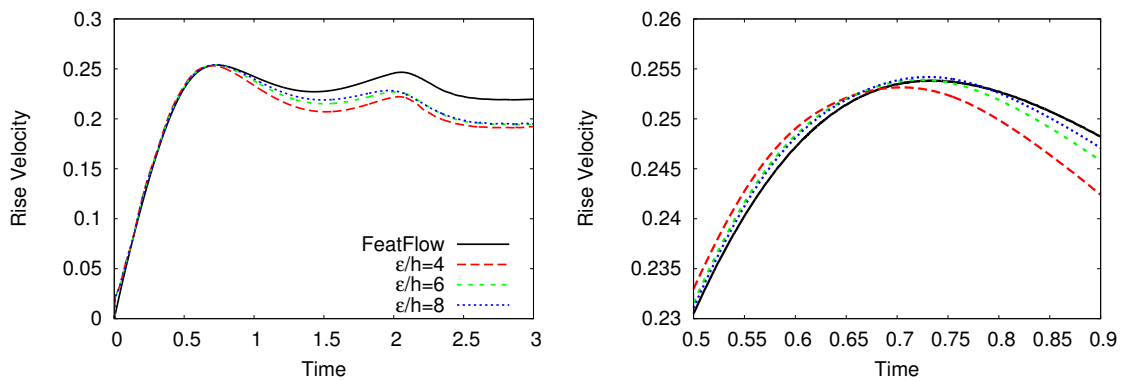


Figure 6.33: Effect of interface thickness on temporal values of rise velocity in TC2 using one-fluid coupled scheme (left) and the enlarged view (right).

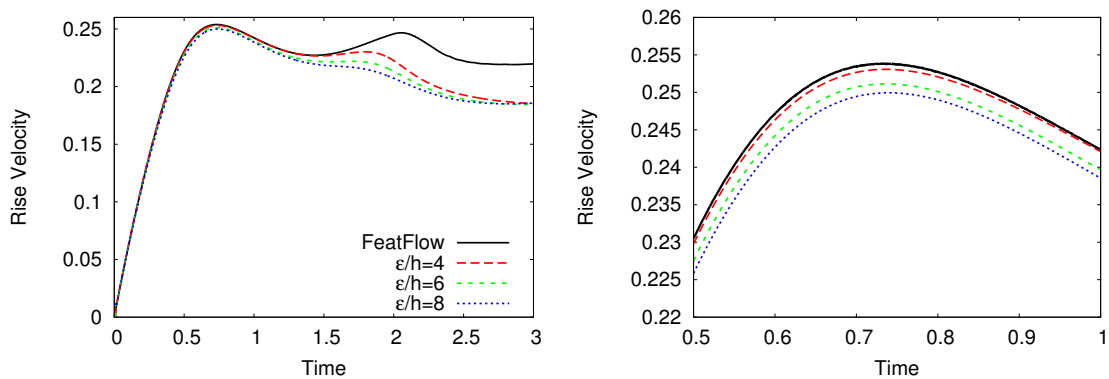


Figure 6.34: Effect of interface thickness on temporal values of rise velocity in TC2 using pressure evolution coupled scheme (left) and the enlarged view (right).

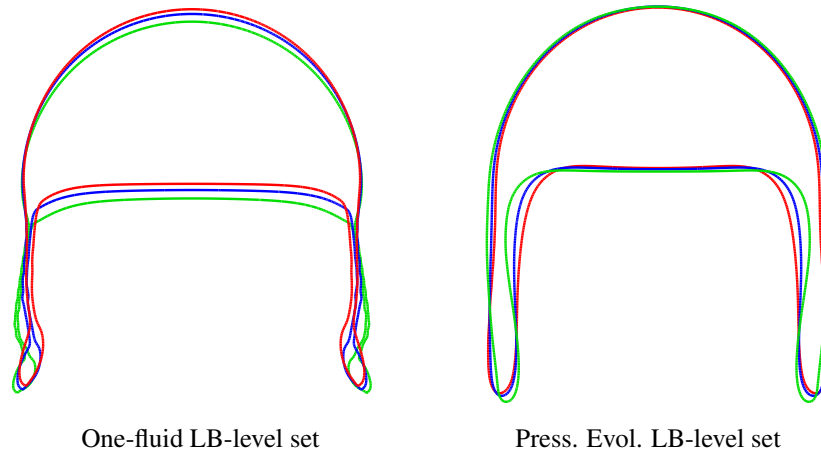


Figure 6.35: Bubble interface at time $T = 3$ using $\varepsilon/h = 4$ (green), $\varepsilon/h = 6$ (blue), $\varepsilon/h = 8$ (red).

6.4. Refined-grid level set

The favourable convergence of the solution by reducing the interface thickness up to $m = 4$ in the pressure evolution scheme, suggests to capture the interface on a finer grid as compared to that of the LB part, so as to achieve a sharper view of the interface without the need to refine the flow field. The *refined-grid level set* (RGLS) technique was first used in the coupled DNS approach by Herrmann [28], where local refinement of the signed distance LS function was applied to subdomains containing segments of the interface. Each subdomain was then assigned to an individual processor in a distributed-memory parallel implementation. In the present work, however, the emphasis is on studying whether or not a RGLS approach is compatible with the LB-based flow solver and local refinement is thus not pursued. To this end, a single refinement step, i. e., $h_{ls} = h_{lb}/2$ is applied to the entire LS grid, while the same value of time scale Δt is applied for both LB and RGLS. From the stability point of view, halving the length scale means that the diffusion number $D = \eta\Delta t/h_{ls}$ for the LSE is scaled by a factor of 2. Nevertheless, this is not an issue since the values of D are generally of order 10^{-5} in the actual non-refined simulations, which is far below the stability threshold of 0.15. In order to transfer the velocity values from the LB solution to the RGLS solver, a bi-cubic interpolation [97] is used. The obtained values of ψ on the refined grid are then restricted to the coarser grid via the nine-point full-weighting restriction rule [97, 104], and ported back to the LBM flow solver.

While the interface thickness $m = 4$ was considered for the non-refined solutions in the previous section, the present RGLS implementation uses $m_{ls} = 6$ which in turn gives $m_{lb} = 3$. This way, the interface becomes sharper by 50%, while at least 3 lattice cells will resolve the interface in the LB part which would have destabilized the solution in non-refined implementations. Note that a 100% reduction in the interface sharpness implies $m_{lb} = 2$ which triggers instabilities even using RGLS.

As the interface deformation in TC2 is more sensitive to the thickness, this test case is selected to evaluate the performance of the RGLS computations. Figures 6.36(a) and 6.36(b) show the bubble circularity on the lattice resolutions $1/h_{lb} = 80$ and $1/h_{lb} = 160$, respectively, using normal and refined grid LS. The results of non-refined LS on $1/h = 640$ are used as reference. One could see a slight improvement in the bubble circularity using the RGLS. The observation is advocated by the sharper representation of the interface compared to the converged solution in figure 6.37.

Another noteworthy consequence of using the RGLS scheme is to improve the mass conservation of the interface capturing tool as also pointed out in [28] for a signed distance LS function.

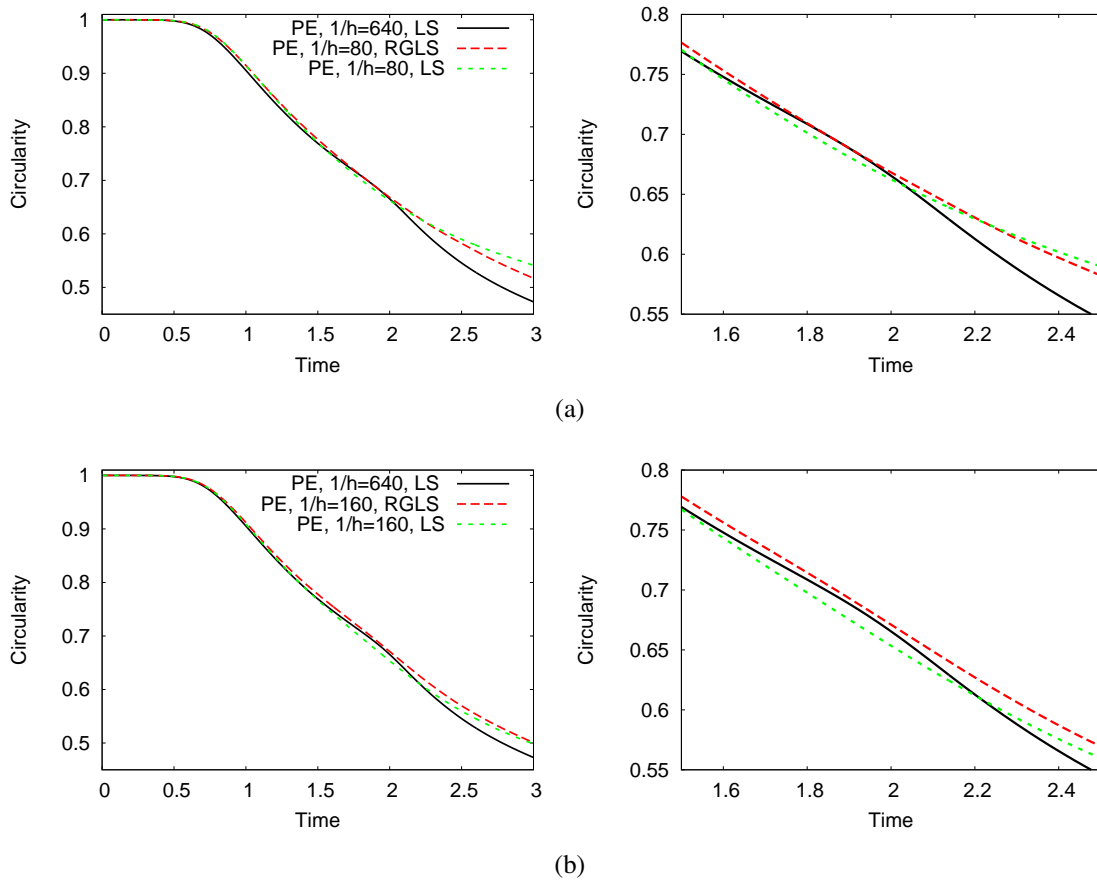


Figure 6.36: Effect of using RGLS on bubble circularity over time at $T = 3$ and (a) $1/h = 80$, (b) $1/h = 160$

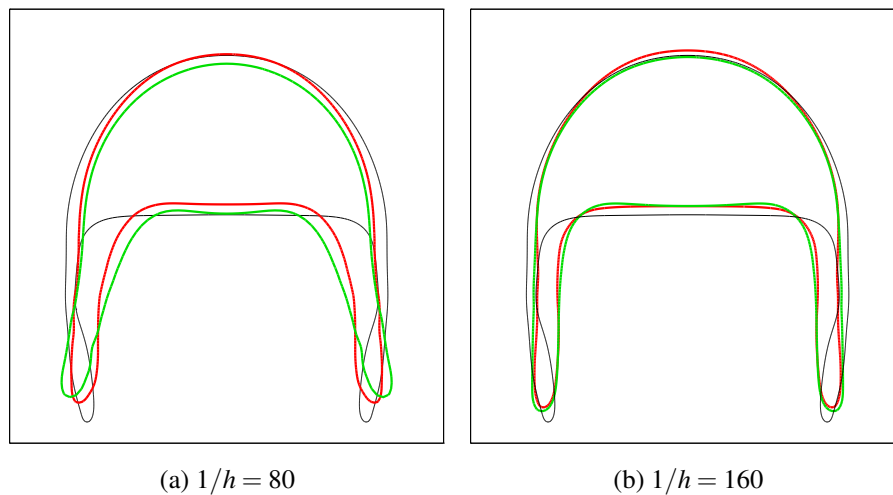


Figure 6.37: Bubble interface shape at $T = 3$ using RGLS (red), non-refined LS (green) and the reference non-refined LS on $1/h = 640$ (black).

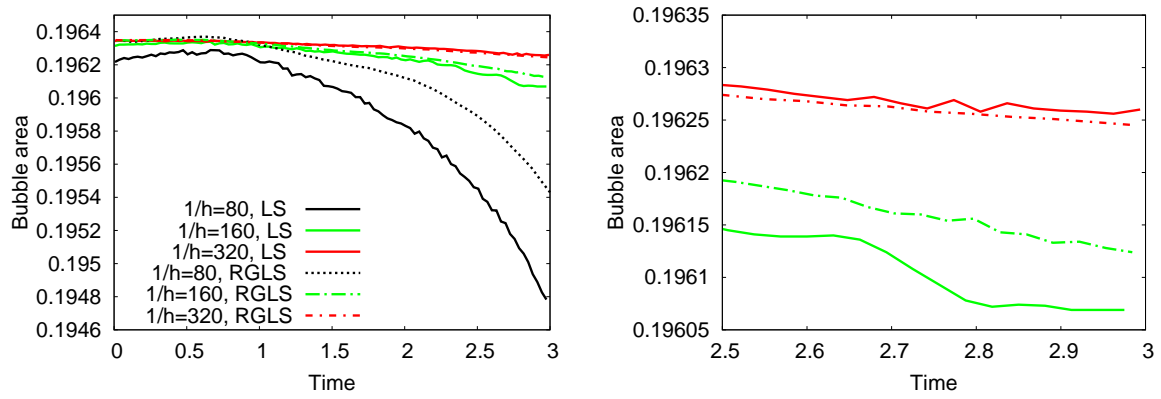


Figure 6.38: Effect of using RGLS on bubble area over time at $T = 3$ (a) and the enlarged view (b).

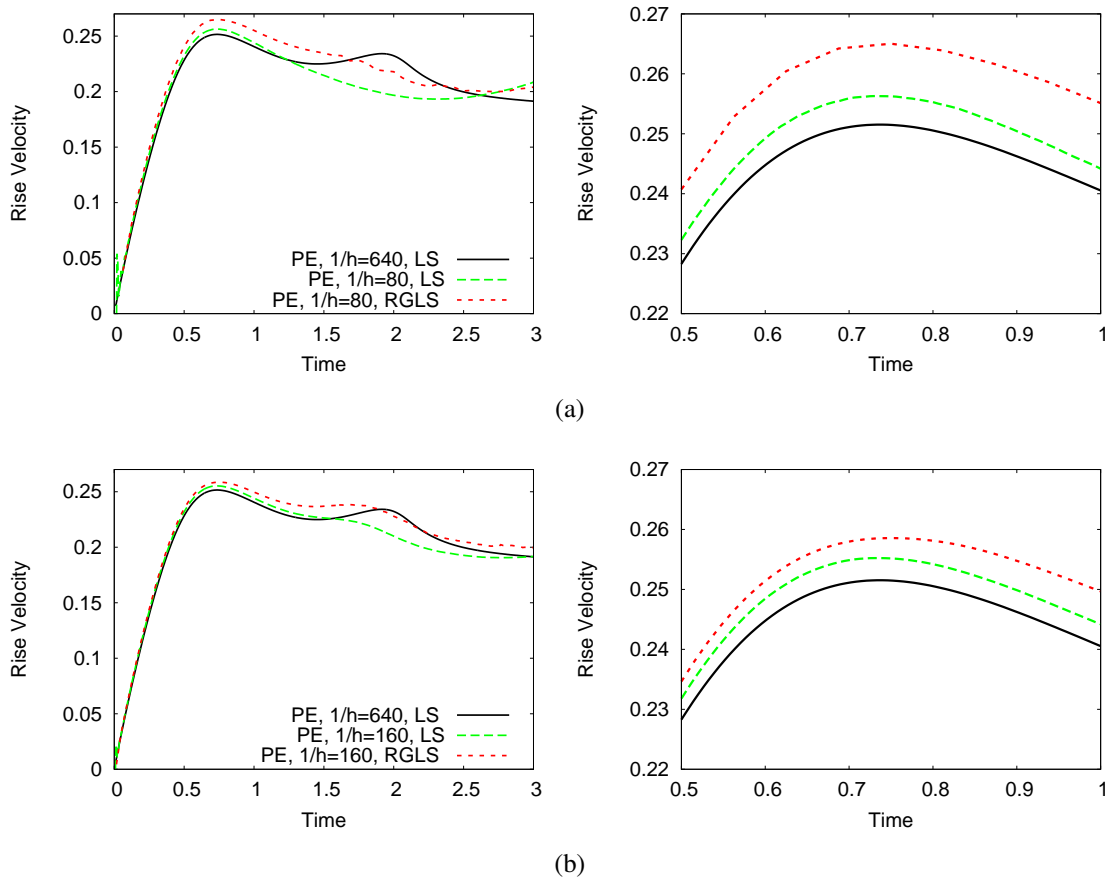


Figure 6.39: Effect of using RGLS on bubble rise velocity over time at $T = 3$ and (a) $1/h = 80$, (b) $1/h = 160$

The impact of refining the LS field on the mass conservation of the present phase-field scheme is evaluated in figure 6.38. For the coarse lattice of $1/h_{lb} = 80$ the mass loss will improve from 0.77% in the non-refined LS to 0.4% using RGLS. Nevertheless, the phase-field method is so

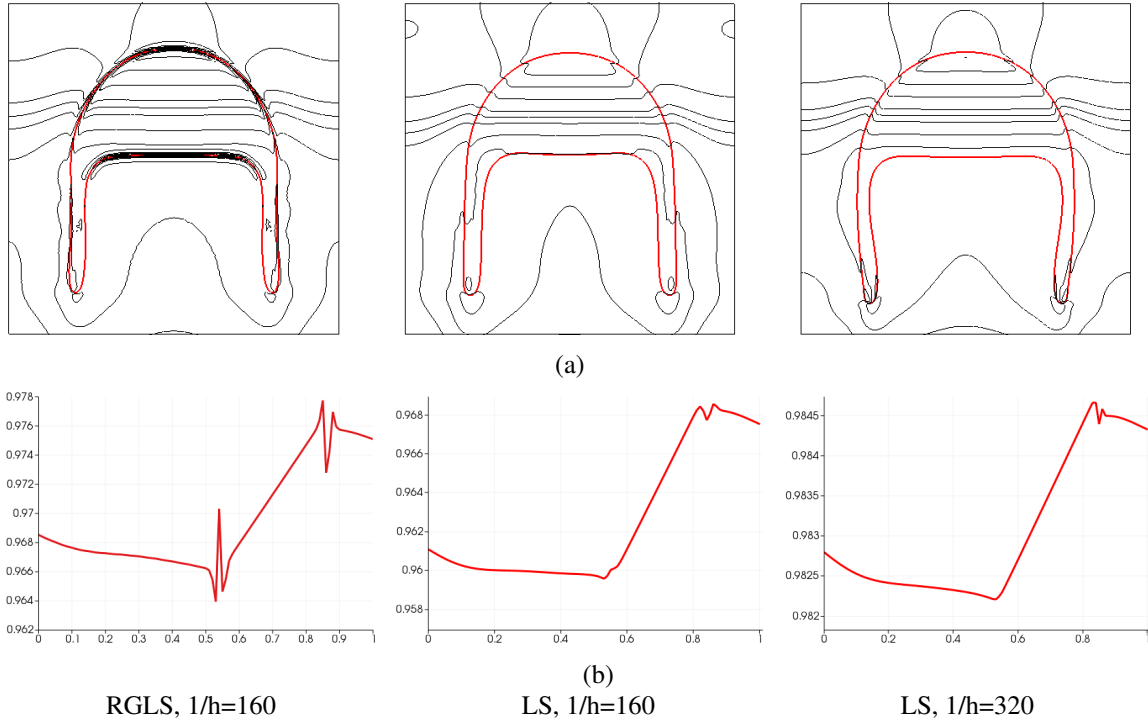


Figure 6.40: Pressure variations at $T = 3$ for TC2 shown through (a) contour lines, (b) pressure profiles along $x = 0.5, 0.5 < y < 1.5$.

effective in preserving the mass that the discrepancies in the the temporal mass becomes quite negligible for finer lattices, where the only advantage of RGLS will be to remove instantaneous mass fluctuations.

Investigating the rise velocity evolution in figure 6.39, however reveals the negative effect of shrinking the interface thickness to $m_{lb} = 3$ as the deviations from the reference velocities grow on RGLS. In fact, the velocities resemble the coarse grid behaviour of the one-fluid scheme, where the low accuracy of pressure solution would end up in large deviations in velocity even for $T < 1$. Such a destructive effect of enforcing $m_{lb} < 4$ for the LBE could be related to the non-smooth variations of the pressure shown in figure 6.40. It could be seen that the pressure changes abruptly over the interface at bottom and top of the bubble. In conclusion, despite having the advantage of realising a sharper view of the interface, reducing the mass loss on coarse grids and extending the stability limit of the solution to $m_{lb} = 3$, an inevitable outcome of the RGLS will be the lowered quality of the underlying pressure and subsequently velocity fields.

6.5. 3D rising bubble

The implementation of any 3D multiphase flow solver is inevitably involved with some practical considerations. The most prominent hurdles are the restrictions from the computational time and the required memory space. Thanks to the suitable work-flow of LBM for parallel computing, the first obstacle is considerably alleviated in this work, while the memory issue is eased through replacing the conventional kinetic-based interface capturing methods with the light-weight LSM. Nevertheless, the numerical experiments carried out for the 2D problems in the previous section leads to a number of conclusions which are crucial to 3D implementations. Firstly, the one-fluid scheme has difficulty in resolving the correct pressure field particularly for the high density ra-

tio problem. The issue also prevents the method to reduce ε with the increase in the resolution. Secondly, the sharp interface LSM used in the one-fluid scheme requires explicit reinitialization which in turn involves periodic reconstruction of the interface. While the first issue necessitates having a high lattice resolution to cope with the large pressure gradients, the second issue is contradictory, adding a great deal of code complexity and considerable computational overhead due to post-processing at high resolutions. It must be noted that unlike the 2D implementations, the large size of the post-processing data in 3D makes the hybrid CPU-GPGPU implementation a hardly affordable remedy, since the data to be transferred between CPU and GPGPU memories is scaled with N^2 in 3D problems. As such, the diffuse interface pressure evolution scheme is preferred as the method of choice for the 3D benchmark studies in this section.

As to the choice of the discrete velocity stencil, the D3Q19 stencil is followed as the default selection which is known to possess a higher isotropy than the D3Q15 stencil and impose only moderate increase in the computational workload. The choice is then verified through detailed comparisons of the results obtained by D3Q15, D3Q19 and D3Q27 velocity models against the benchmark data. Yet, regardless of the type of the lattice stencil in use and considering the 5.7 GB memory available on Kepler 20x GPGPU used for the present simulations, the maximum lattice resolution is selected to be $1/h = 256$ which for example occupies 3.9 GB of memory for 1/4 of the problem using the D3Q19 stencil. The memory size on GPGPU accounts for only those arrays which are essential for the core computations and could not be defined and freed on the fly. These include $2 \times q$ arrays for the pre- and post-collision distribution functions in D3Qq stencil, 8 arrays for velocity vector, pressure, viscous and pressure force vectors, a total of 10 arrays to solve the phase-field LS equation and 2 integer arrays for the boundary conditions.

A summary of the simulation statistics using different stencils is presented in table 6.5 for TC1, where the number of lattice nodes is denoted by NLN, the total memory size occupied on GPGPU in GB by MGPU, number of time steps by NTS and the total simulation time in seconds by TGPU. Note that the simulation times also include the intermediate post-processing periods and more precise data on the computational performance will be presented in chapter 7.

Table 6.5: Simulation statistics the 3D rising bubble using different lattice stencils

$1/h$	NLN	MGPU	NTS	TGPU	TGPU/NTS
D3Q15					
96	435456	0.17	26508	170	0.006
192	3511296	1.43	108300	4290	0.039
256	8339456	3.40	193548	17797	0.091
D3Q19					
96	435456	0.20	26508	207	0.007
192	3511296	1.65	108300	5329	0.049
256	8339456	3.93	193548	20774	0.107
D3Q27					
96	435456	0.26	26508	292	0.011
192	3511296	2.1	108300	7018	0.064
256	8339456	5.0	193548	29550	0.152

6.5.1. Test case 1

The evolution of the 3D bubble interface in TC1 up to $T = 3$ is presented in figure 6.41 using the pressure evolution scheme on a lattice of $1/h = 256$ and the reference shapes obtained by FeatFlow on a grid of $1/ = 128$. The bubble develops into an ellipsoidal shape as expected,

although it preserves its initial sphericity to a higher degree than the 2D bubble where almost no deformation occurs after $T = 2.0$.

The convergence of the interface shape is also studied in figure 6.42, followed by more detailed comparisons between the bubble quantities obtained by LBM NaSt3D, and the reference values of FeatFlow in figures 6.43 to 6.46. For the low density and viscosity ratio of TC1, the present maximum lattice resolution seems to be sufficient to repeat the reference data by FeatFlow. For the rise velocity, the discrepancies grow around the peak values although the difference between the LBM data on $1/h = 256$ and that of FeatFlow amounts to less than 0.5%. A similar behaviour is observed for the interface-related sphericity and diameter values around $T = 1.5$ where the overall convergence trend is towards the reference curves.

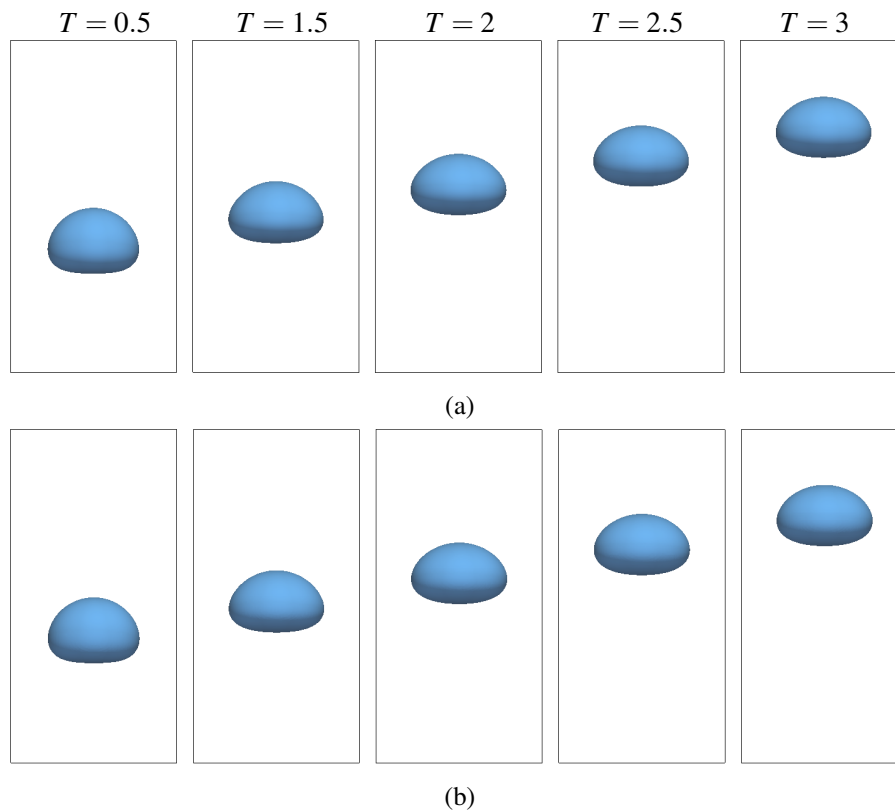


Figure 6.41: Time evolution of 3D bubble in TC1 (a) pressure evolution LB-levelset, (b) FeatFlow.

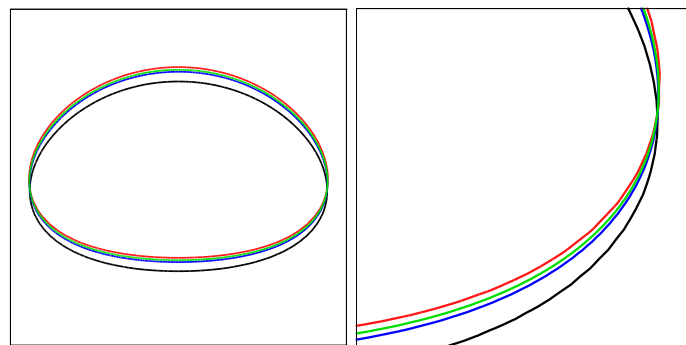


Figure 6.42: Terminal shape of 3D bubble at $T = 3$ for TC1 with enlarged view (right), on $1/h = 96$ (black), $1/h = 192$ (blue), $1/h = 256$ (green) and FeatFlow on $1/h = 128$ (red).

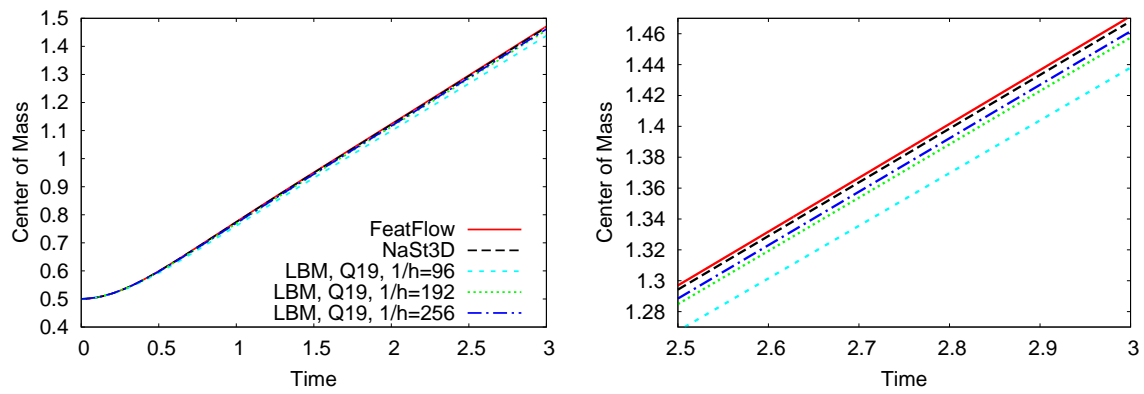


Figure 6.43: Time evolution of the center of mass for the 3D rising bubble TC1 (left), and the enlarged view (right).

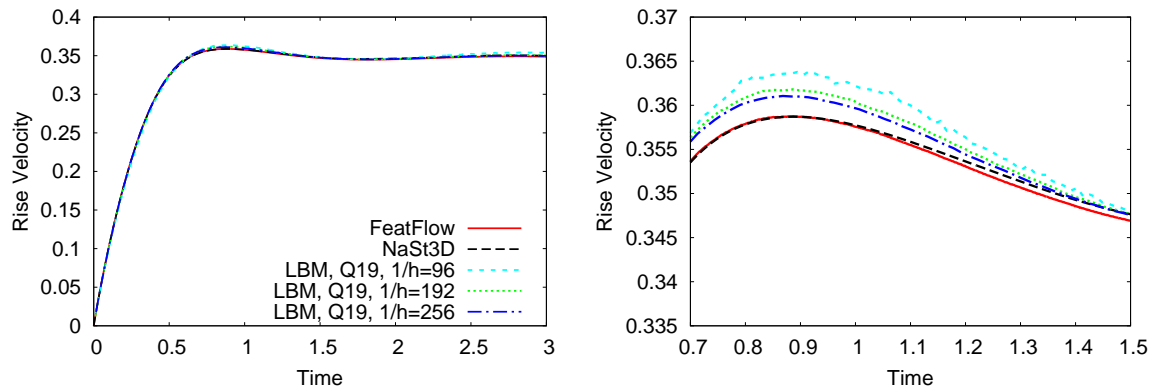


Figure 6.44: Time evolution of the rise velocity for the 3D rising bubble TC1 (left), and the enlarged view (right).

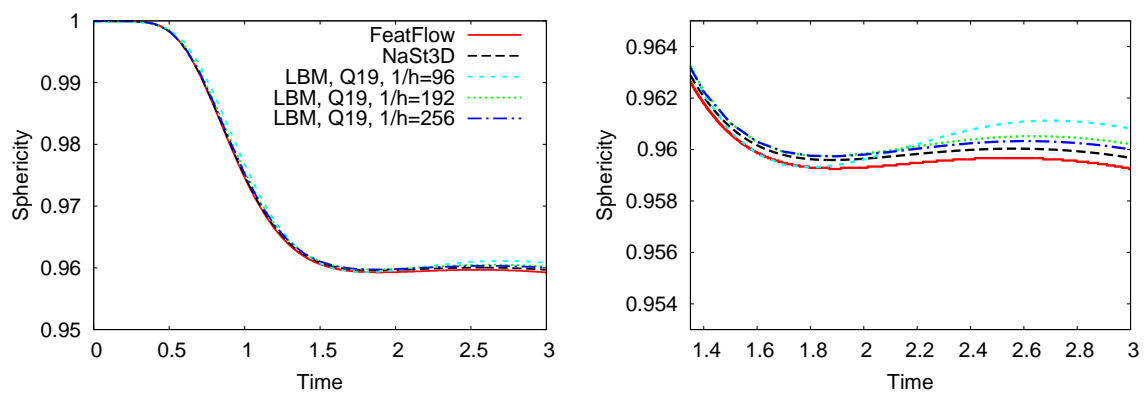


Figure 6.45: Time evolution of the circularity for the 3D rising bubble TC1 (left), and the enlarged view (right).

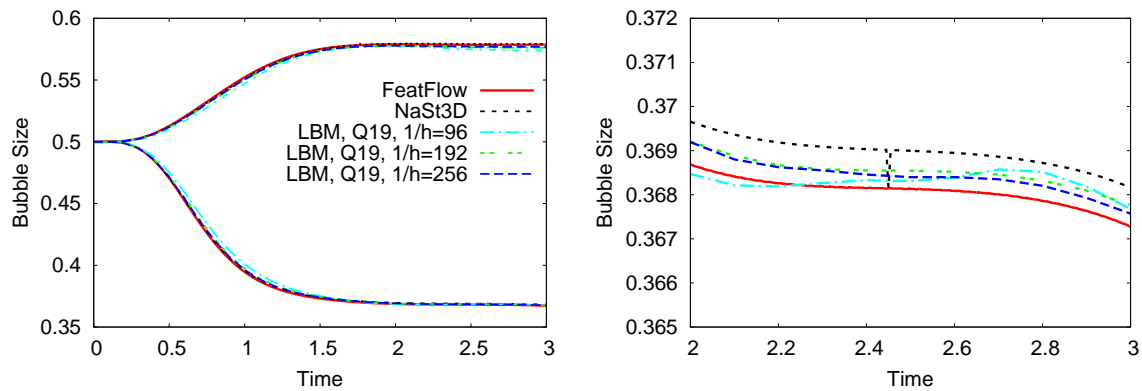


Figure 6.46: Time evolution of the diameters in horizontal and vertical directions for the 3D rising bubble TC1(left), and the enlarged view (right).

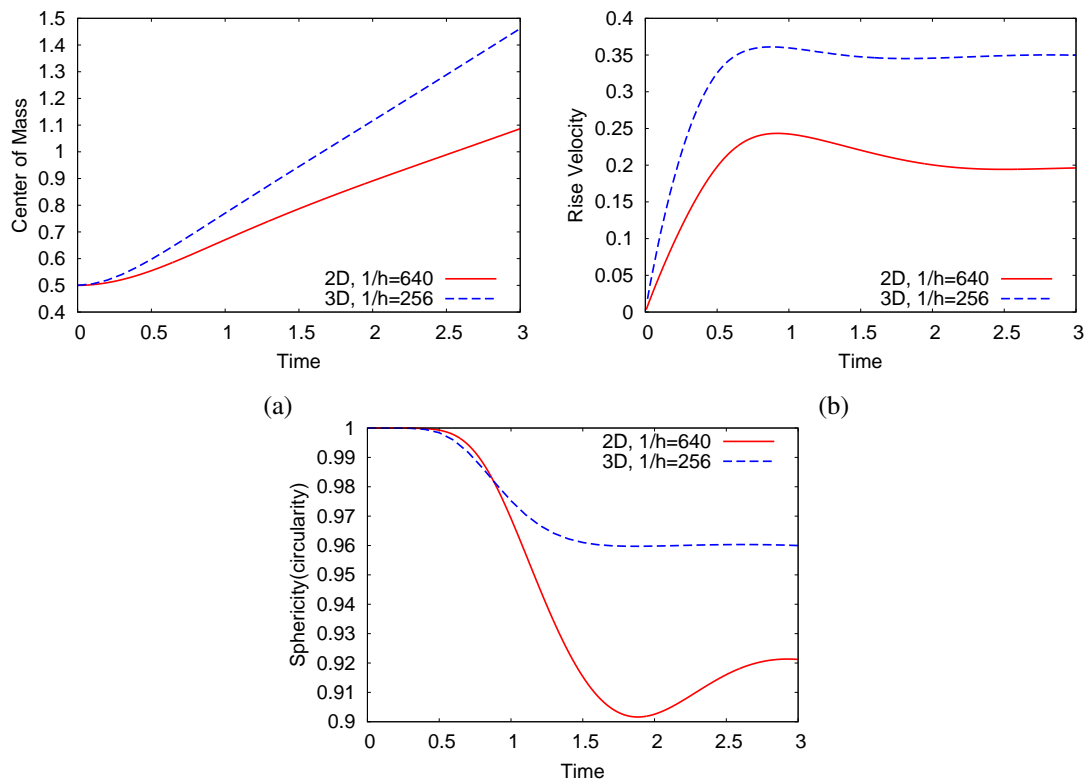


Figure 6.47: Time evolution of (a) center of mass, (b) rise velocity and (c) sphericity of the bubble in 2D and 3D problems of TC1.

Apart from the benchmarking, the temporal evolution of the bubble demonstrates a number of key distinctions from the 2D case. In order to quantify the differences, the bubble metrics for both problems are plotted together in figure 6.47. In addition to the rounder state of the terminal bubble shape which results in higher values of sphericity seen in figure 6.47(c), one also discerns that the bubble gains higher elevations and rise velocities. Such obvious discrepancies are to some extent due to the existence of solid walls on all sides in the 3D case, whereas the 2D case enjoyed free slip conditions on the side walls. The presence of the additional walls prevents high energy vortices around the bubble to extend into the far-field, thus letting the bubble to preserve its kinetic energy. On the other hand, the surface tension forces acting on a 2D bubble are in fact underestimated since

they lack the third derivative in the z direction in the calculation of the curvature $\kappa = \nabla \cdot \mathbf{n}$. The relatively larger values of κ then boost the surface tension forces which then resist against further deformation of the bubble. The rounder shape of the bubble in turn limits the effective frontal area of the bubble, thus helping the bubble to reduce the drag force and maintain its velocity after $T = 1$ as seen through an early steady state already reached around $T = 1.5$.

6.5.2. Test case 2

Analogous to the 2D problem of TC2, the 3D bubble shown in figure 6.48(a) also experiences strong deformations provoked by the weaker surface tension effects due to a larger EO number. The bubble first deforms into a dimpled cap and then proceeds to extend the edges so that the eventual cusp-shaped bubble is formed. Yet again, the diffuse nature of the pressure evolution scheme prevents the bubble to develop sharp edges, e. g., those predicted by Featflow in figure 6.48(a). Nevertheless, the picture norm admits the close relevance of the overall evolution trend in both schemes. Examining the convergence trend in the interface shape in figure 6.49 shows that LBM may eventually converge to the sharp interface solution of the two-phase Navier-Stokes equations, provided that the lattice will be sufficiently refined, as in the 2D problem of TC2 where a resolution of $1/h = 640$ was used to approach the sharp interface results.

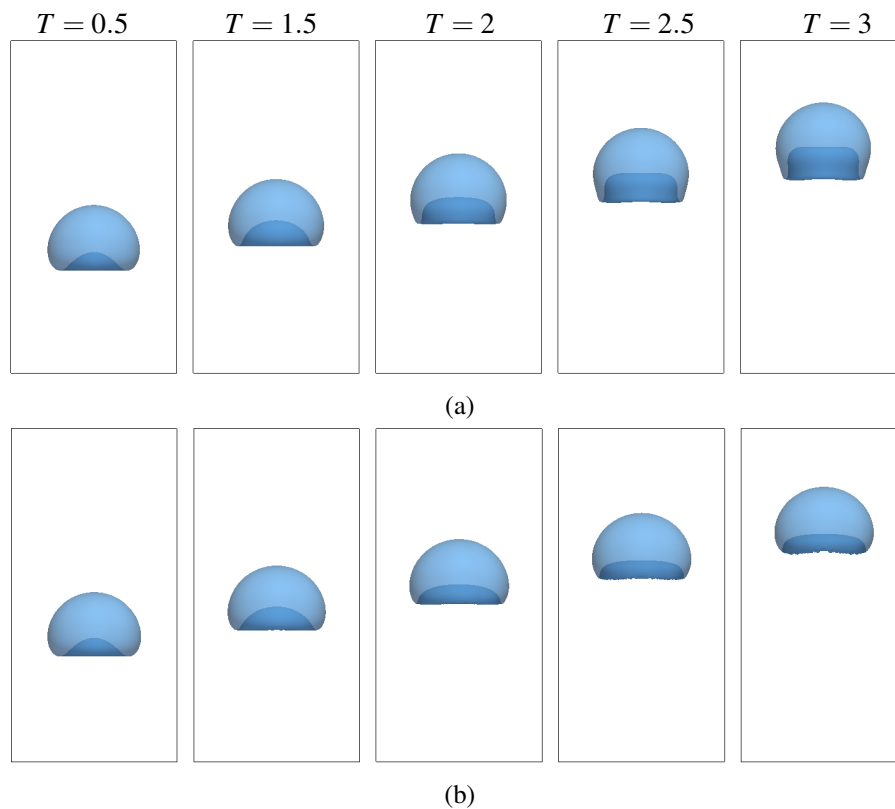


Figure 6.48: Time evolution of the bubble in 3D problem of TC2 using (a) pressure evolution LB-levelset (b) FeatFlow. The dark blue regions reflect the interior surface.

The convergence of the bubble quantities together with the NaSt3D data and the reference data of FeatFlow are collected in figures 6.50 to 6.53. The agreement between the rise velocities keeps favourably up to $T = 1.5$ which confirms the decent performance of the D3Q19 discrete model in recovering the correct pressure and velocity fields under high density and viscosity differences. For $T > 1.5$, as the bubble starts to develop the cusp shape, the phase-field LSM falls behind the

sharp LSM, and tries to keep the edges prolonged and smooth. In terms of bubble quantities, this results in growing deviations from the sharp interface data, which is more clearly seen in the bubble sphericity, diameter and centroid position. Similar to the 2D test case, the sharp interface methods tend to create kinks or teeth-like irregularities on the skirt of the bubble towards $T = 2$ which then makes the solution strongly dependent on the specific parametrizations used in different codes. Consequently, the sphericity and diameter values given by each of the solvers start to take different paths after $T = 2$ and hence the solutions could not be regarded as reference beyond this point.

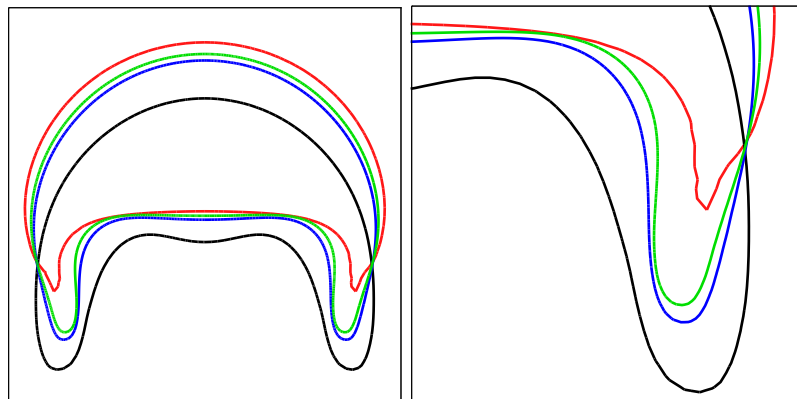


Figure 6.49: Terminal shape of 3D bubble at $T = 3$ for TC2 with enlarged view (right), on $1/h = 96$ (black), $1/h = 192$ (blue), $1/h = 256$ (green) and FeatFlow on $1/h = 128$ (red).

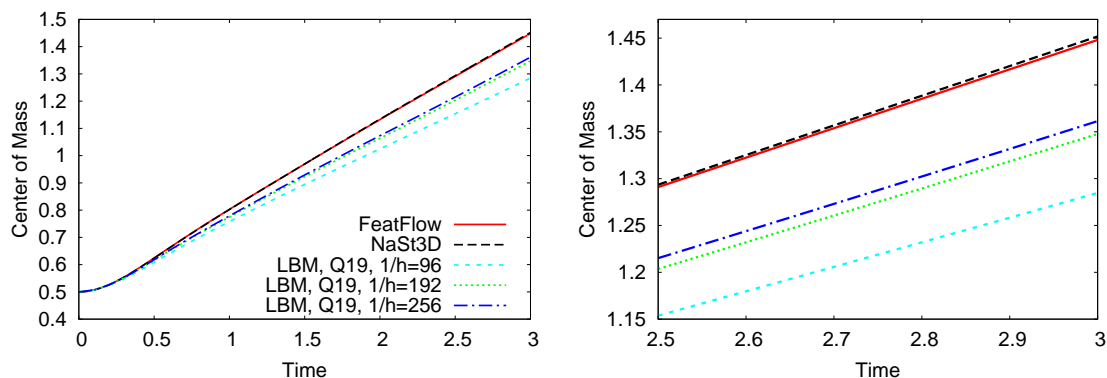


Figure 6.50: Time evolution of center of mass for 3D bubble TC2 (left), and enlarged view (right).

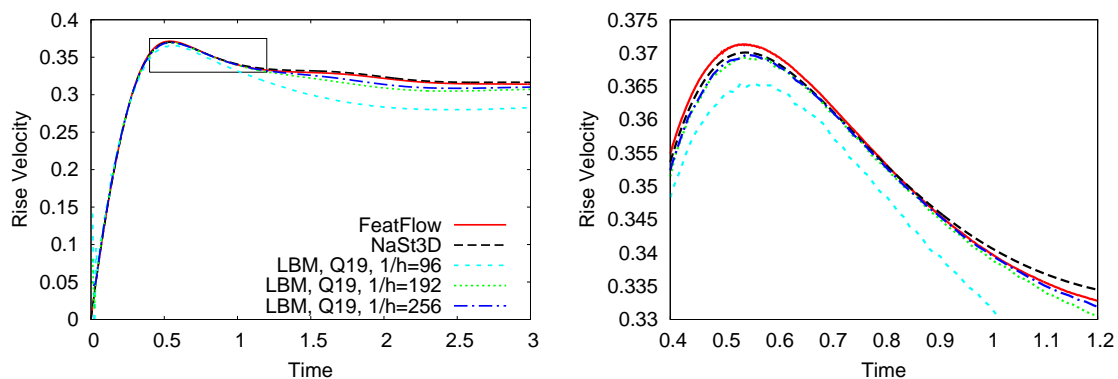


Figure 6.51: Time evolution of rise velocity for 3D bubble TC2 (up left), and enlarged view (up).

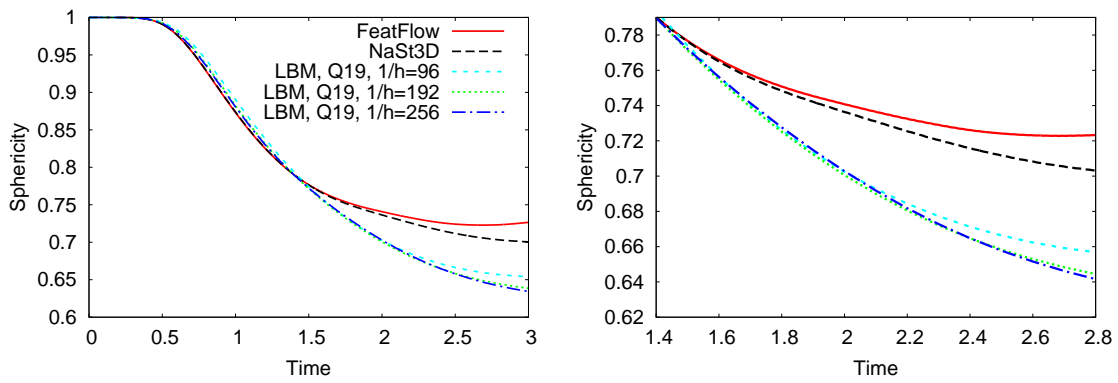


Figure 6.52: Time evolution of circularity for 3D bubble TC2 (left), and enlarged view (right).

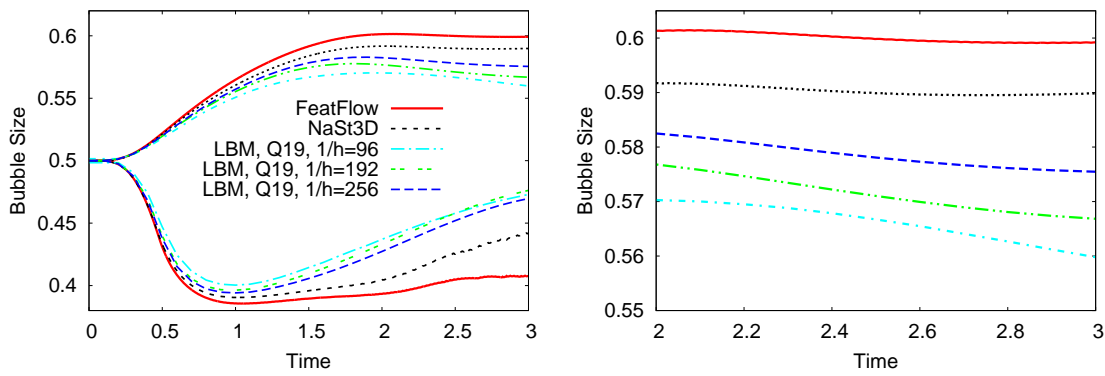


Figure 6.53: Time evolution of 3D bubble diameters for TC2 (left), and enlarged view (right).

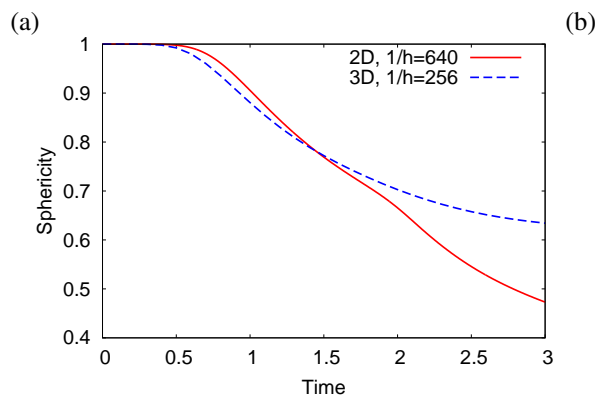
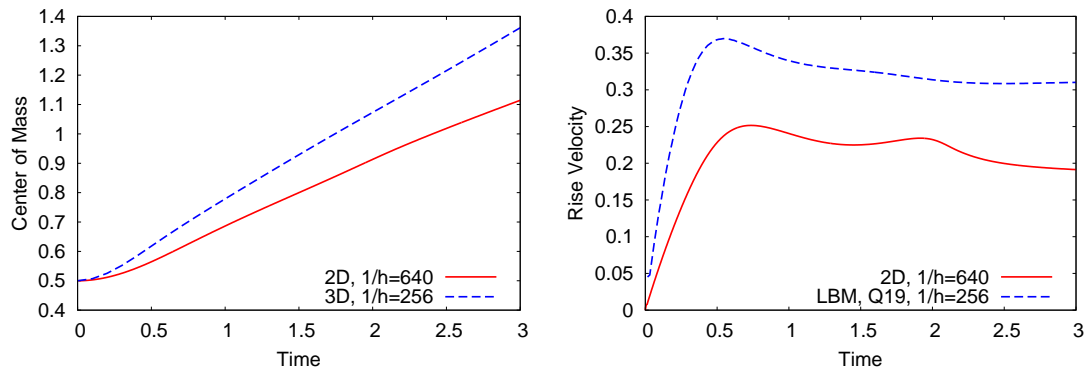


Figure 6.54: Time evolution of (a) center of mass, (b) rise velocity and (c) sphericity of the bubble

Although the bubble dynamics is quite different in TC2 than that of TC1, the differences from the corresponding 2D cases are of the same extent as depicted in figure 6.54 for bubble centroid position, sphericity and rise velocity. The proximity of the solid walls in the horizontal plane and the enhanced surface tension forces eventually result in higher rise velocity and centroid position, and the increased tendency of the bubble to preserve its sphericity to higher values. An interesting observation in this test case, however, is the transformation of the long trailing filaments predicted in the 2D case into a mere short skirt beneath the bubble due to the stronger surface tension effects in 3D. The overall obtained terminal shapes in 3D are nevertheless more close to the patterns reported in the experimental classification in [32].

6.5.3. Isotropy and the choice of the lattice stencil

The most prominent distinction between discrete velocity stencils, is their associated degrees of isotropy. This property could be best studied through examining the performance of different stencils for TC2, where the presence of four orthogonal walls and the large deformations of the bubble creates a prominent natural anisotropy in the terminal shape of the bubble as shown in figure 6.55. The interface contours for the plane perpendicular to the wall ($\theta = 0$) and diagonal plane ($\theta = 45$) are also depicted in figure 6.56. In the first look, the comparisons unveil the overestimated deformation produced by the D3Q15 model on the diagonal plane. The deviation is caused by the inherent lack of isotropy of the D3Q15 stencil in the diagonal directions on $x - y$, $x - z$, $y - z$ planes (see figure 3.3). Although one expects to gain the best relevance to the FeatFlow results using the D3Q27 stencil, it is the D3Q19 stencil which demonstrates the highest isotropy and closer similarity to the FeatFlow shape. Overlaying the interface lines of different solutions on the $\theta = 0$ plane shown in figure 6.57 also depicts that the interface given by the D3Q19 comes marginally closer to that of the FeatFlow.

The evolution of the quantities produced by the different stencils on a $1/h = 256$ lattice is presented in figures 6.58 to 6.61 and compared to the benchmark data. The enlarged views reveal greater deviations in the D3Q15 results and slightly better agreement of the D3Q19 results with those of the benchmarks. For the Bubble diameter which is more sensitive to minimal changes of the interface, the D3Q15 stencil shows a more steep return in the vertical diameter up to nearly $d_z = 0.5$ which is also seen by the interface lines in figure 6.56 where the strong anisotropy yields a longer skirt of the bubble in the diagonal plane.

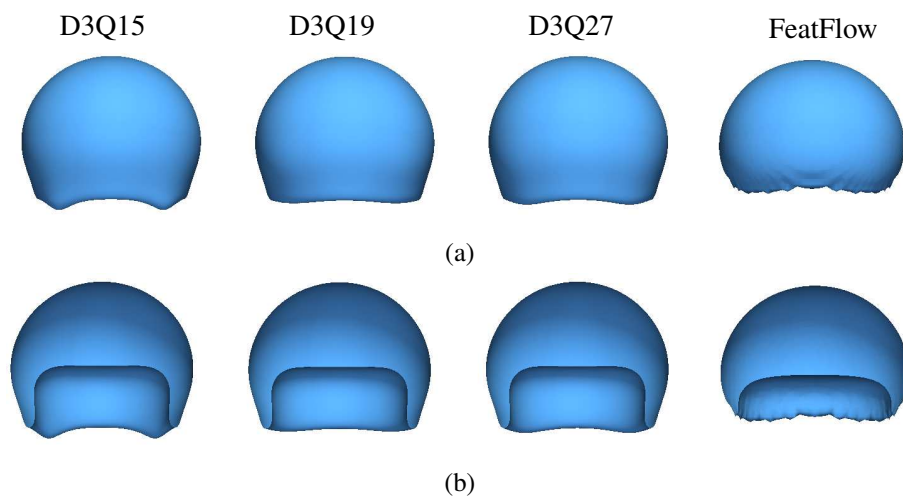


Figure 6.55: Bubble interface $T = 3.0$ in 3D TC2, (a) exterior view and (b) middle section view.

Finally, the reason behind the slightly superior accuracy and isotropy of the D3Q19 stencil over the D3Q27 discrete model is speculated to be the excessive spatial support of the D3Q27 stencil for this particular problem as it makes the directional derivatives for the pressure gradient more dependent on $(\pm 1, \pm 1, \pm 1)$ directions. Nevertheless, the above argument could not be firmly verified as almost all dynamic two-phase LB simulations in the literature are limited to D3Q15 or D3Q19 stencils for computational reasons [18, 122, 142, 143]. Lee and Liu [76] used the D3Q27 discrete model to simulate droplet impact on dry surfaces. Yet they brought no justification for this choice and provided no comparisons against the other discrete models.

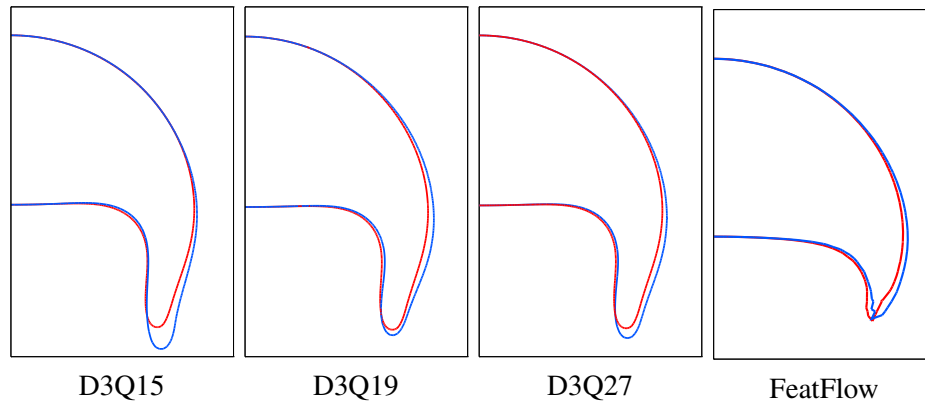


Figure 6.56: Bubble interface at $T = 3$ in 3D TC2 for $\theta = 0$ (red) and $\theta = 45$ (blue) planes.

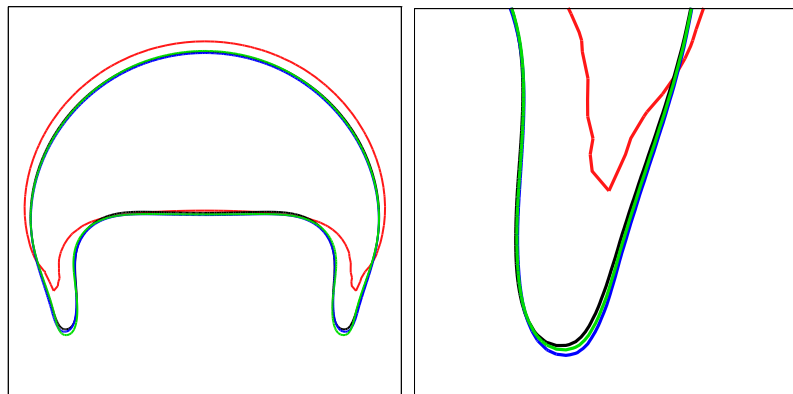


Figure 6.57: Terminal shape of bubble at $T = 3$ for 3D TC2 with enlarged view (right), using D3Q15 (green), D3Q19 (black), D3Q27 (blue) and FeatFlow (red) on $\theta = 0$ plane.

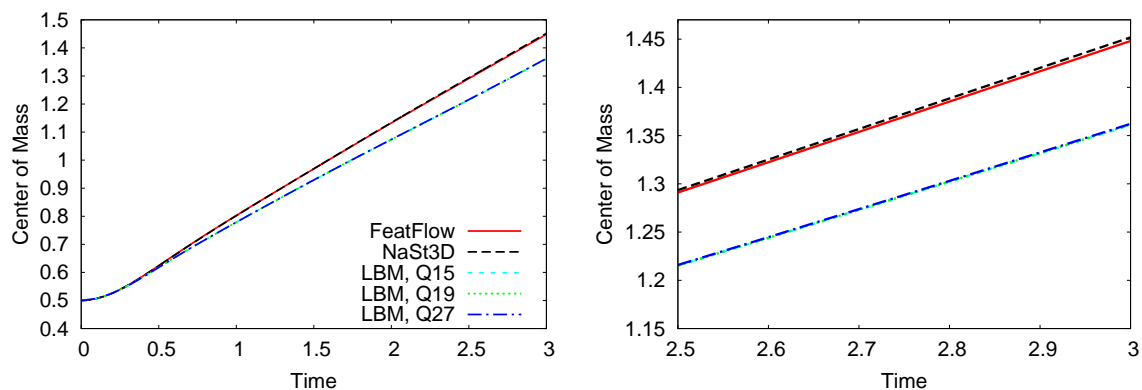


Figure 6.58: Time evolution of center of mass for 3D TC2 (left), and enlarged view (right).

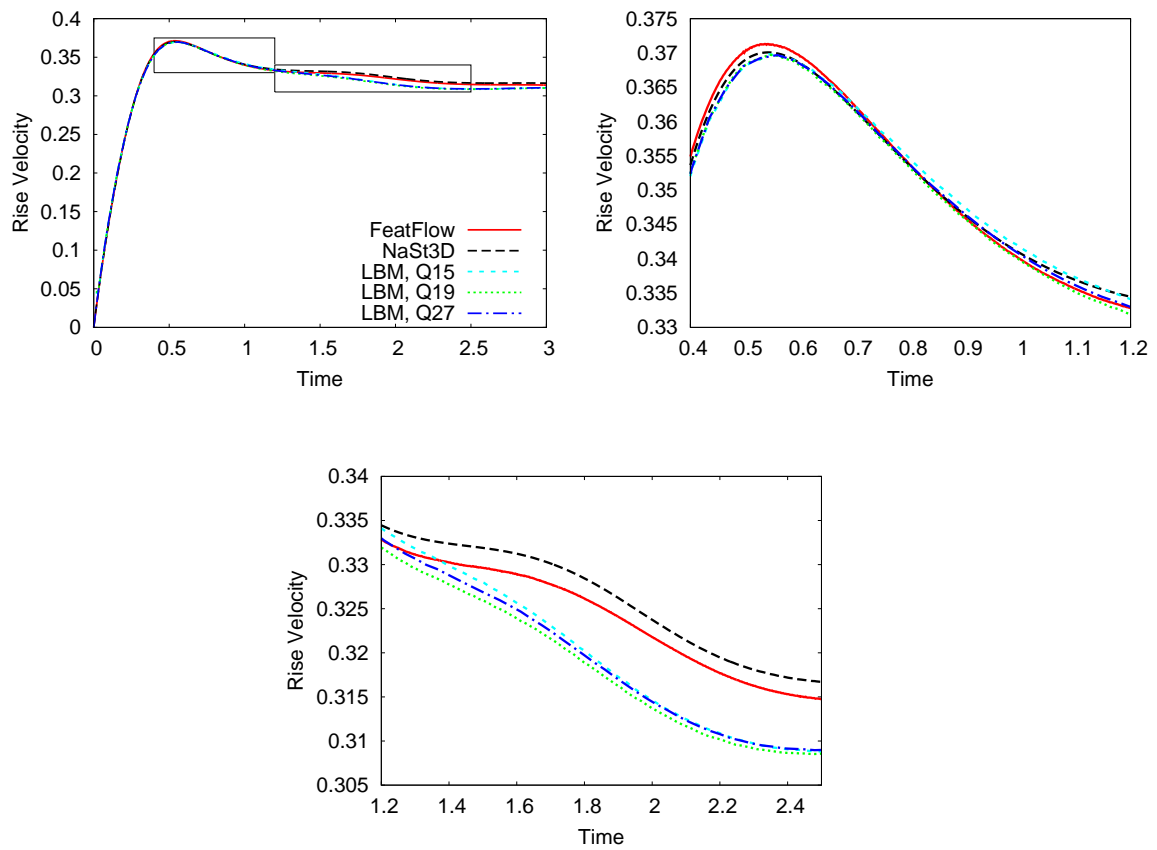


Figure 6.59: Time evolution of the rise velocity for the 3D rising bubble TC2 (up left), and the enlarged views (up right and down).

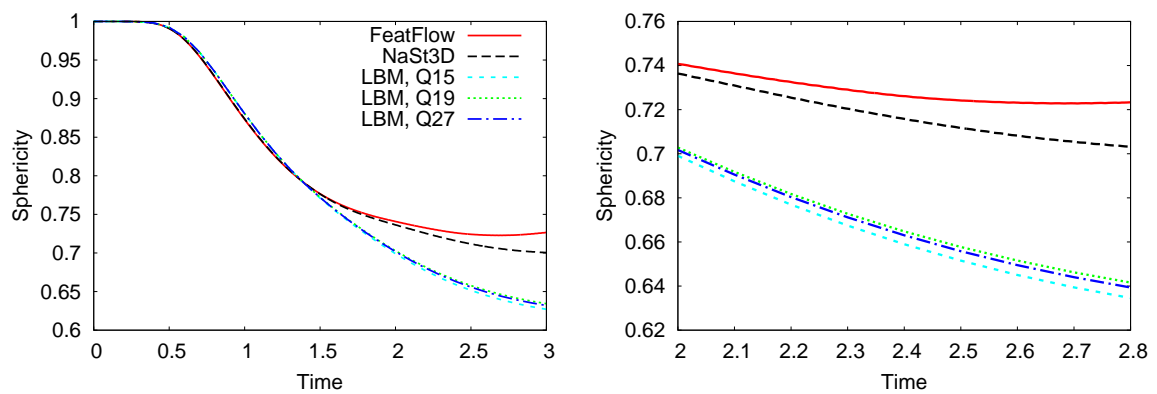


Figure 6.60: Time evolution of the circularity for the 3D rising bubble TC2 (left), and the enlarged view (right).

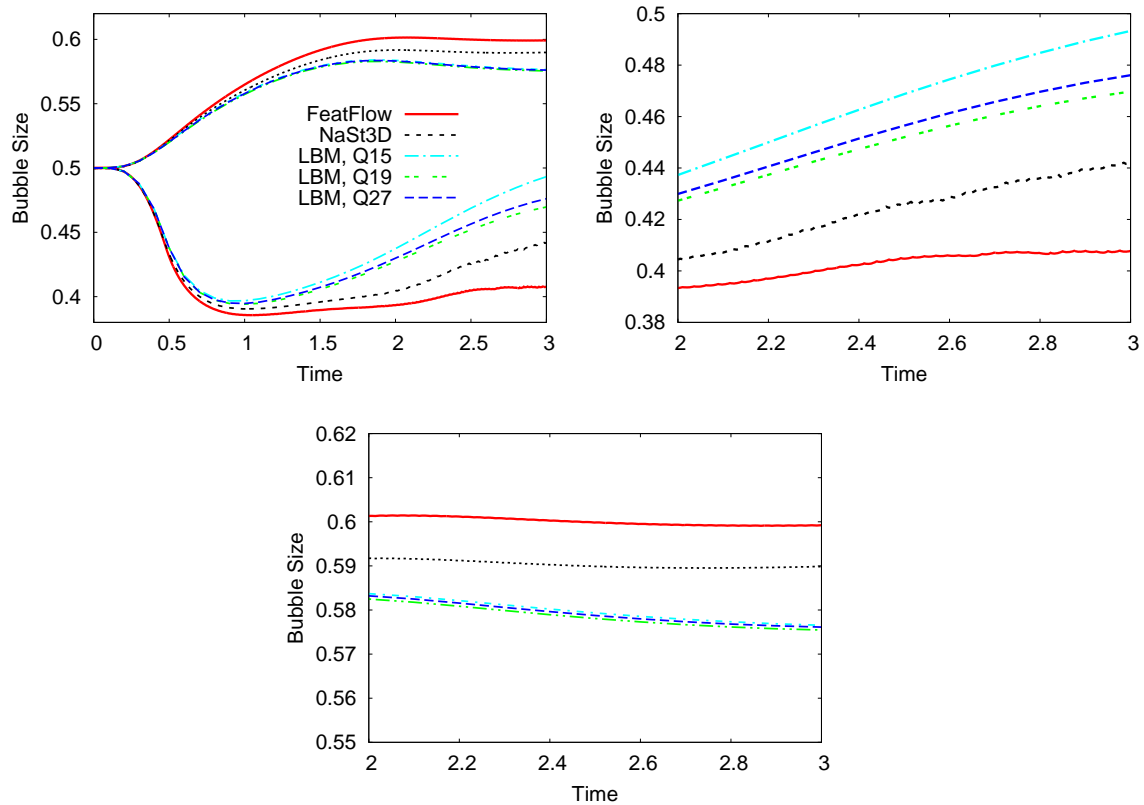


Figure 6.61: Time evolution of the diameters in horizontal and vertical directions for the 3D rising bubble TC2 (left), and the enlarged views (up right and bottom).

6.6. Effects of arithmetic precision

The LBM is known as a technique where unlike the numerical solution of the Navier-Stokes equations, convergence and accuracy of the scheme has no primary dependence on the choice of the arithmetic precision, i. e., single or double precision (henceforth denoted by SP and DP). Beside some brief implicit discussions, e. g., those in [69] and [89] for the 2D driven cavity problem, there has been no elaborate study on this property of LBM. Nevertheless, one can associate such a behaviour to the fact that the explicit nature of the LBM removes the need for solving a sparse linear system as often required, for example, for solving the pressure Poisson equation in the NS-based solutions. During such algorithms, the recurring round-off, and subsequently cancellation errors caused by the reduced mantissa size of the SP arithmetics can accumulate the computational error to such an extent where iterative or direct solutions eventually diverge.

On the other hand, the advantage of dooming the computations to SP, if ever feasible, becomes substantial when implementing the solutions on GPGPUs. In fact the rewarding implications are twofold. Firstly, in most modern GPGPUs the nominal SP floating point computing power is at least twice that of DP, with some gaming GPGPUs offering a SP to DP FLOPS ratio of 32. Secondly, as algorithms, i. e., LBM are mostly memory-bounded, they can already gain an almost free speedup of factor 2 since the size of a `float` value (4 bytes) is half of that for a `double` value (8 bytes), while they both enjoy the memory bandwidth.

For the multiphase simulations, the potential benefits of SP computations could be quite tempt-

ing as simulations are much more involved and thus time consuming as compared to single phase ones. Prior to follow this idea for real scale simulations, however, one has to examine whether the additional computations for two-phase forces and tracking the interface have brought along more dependence on the arithmetic precision. To this end, the accuracy of SP and DP implementations is studied for the 2D TC2, where the lattice resolution could be increased to challenge the precision without hitting the memory limits. Moreover, the high density ratio is generally more vulnerable to instabilities due to very low density and viscosities of the gas phase and high deformation rates, and thus can reveal early symptoms of precision-related errors.

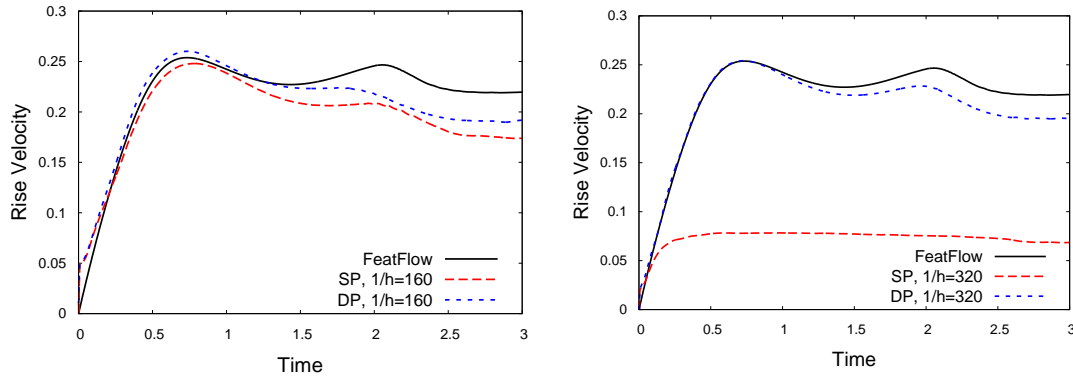


Figure 6.62: Time evolution of the rise velocity for 2D TC2 using the one-fluid coupled method under SP and DP arithmetics.

Figure 6.62 shows the evolution of velocity for the 2D TC2 using the one-fluid coupled approach where SP computations have an immediate destructive effect on the solution although not ending up in a complete divergence. The reduced rise velocities and the fact that bubble position and deformation would not be greatly affected, suggest that the pressure force $\nabla p(1 - 1/\rho)$ is being underestimated in SP. In fact a major source of round-off in this term could be found in the infinitesimal changes in the virtual density $\bar{\rho}$ which are supposed to eventually recover the pressure force via the EOS. The round-off is then magnified when multiplying ∇p by $1/\rho$ which was a result of deriving the force term in the one-fluid formulation.

Application of the SP computations to the pressure evolution scheme shows a stronger tolerance of the method against computational errors as seen in figures 6.63 and 6.64 for rise velocity and circularity. The deviations from the reference DP solution become apparent only for $1/h > 320$ and quite large for $1/h = 1280$. For $1/h = 2048$ both rise velocity and circularity may take completely erroneous patterns (not shown here), and the solution diverges around $T = 2.5$.

By assuming the q_{ref} in the error correlations of section 6.1.5, to be those obtained on the same lattice level using DP computations, one can quantify the errors induced by SP arithmetics as shown in figure 6.65. The error quantities show the negligible level of deviations for $1/h = 160$ and $1/h = 320$ with a growing order or ≈ 1 , which is followed by an almost 4th order growth in the errors for $1/h > 320$. In fact $1/h = 320$ acts like a threshold after which the round-off errors in SP become eventually significant, with the maximum errors as large as 1% for $1/h = 640$ and 5-10% for $1/h = 1280$.

The pressure contours for $1/h = 640$ and $1/h = 1280$ given in figure 6.66 provide a better insight on the errors in the hydrodynamics caused by SP implementation. For $1/h = 640$, noises in the pressure field become obvious in the wake behind the bubble, although not too strong to affect the velocity field and hence the interface shape. For $1/h = 1280$, however, the pressure noises are so prevalent that the velocity field becomes fairly distorted, causing false deformations

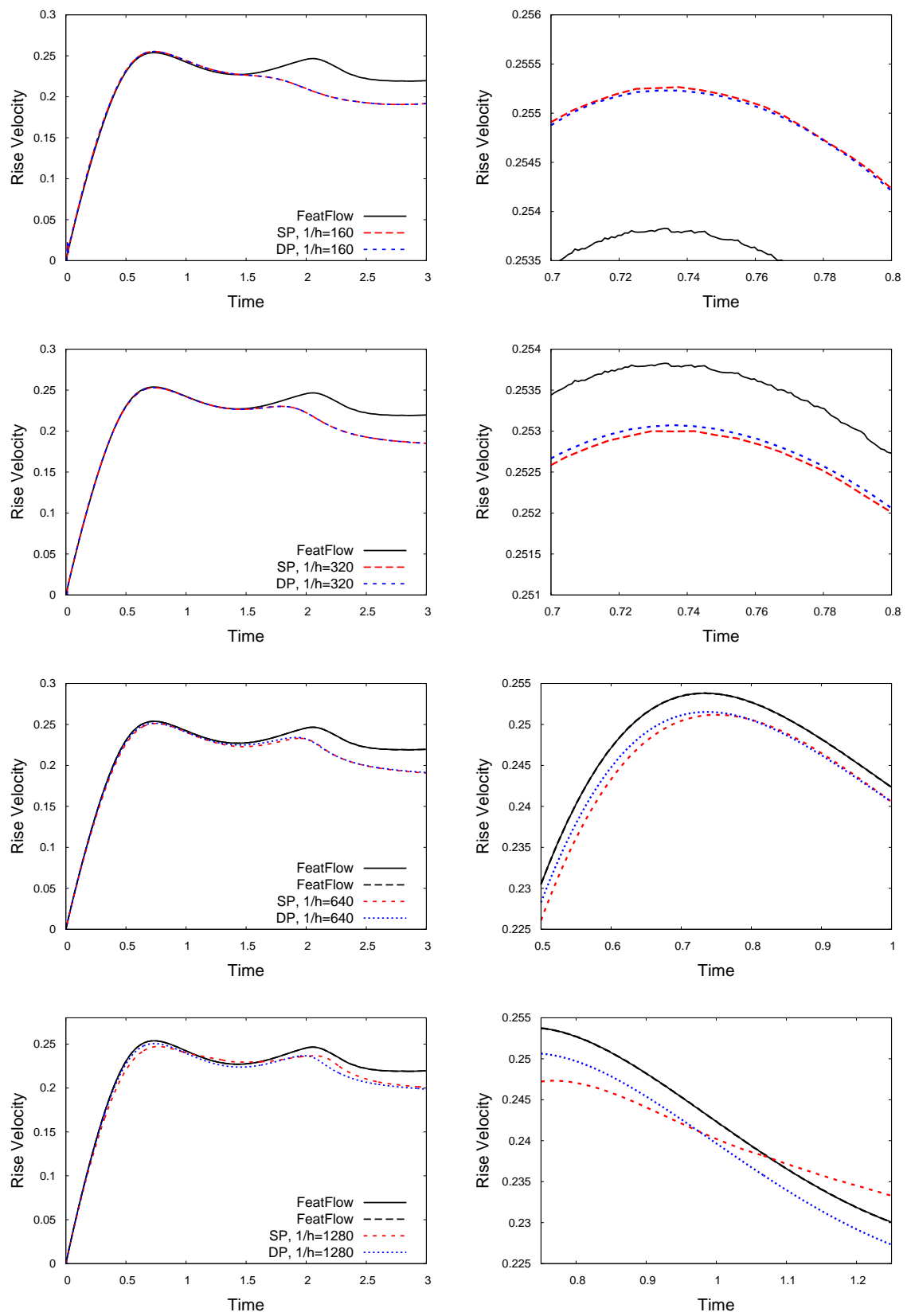


Figure 6.63: Time evolution of the rise velocity for 2D TC2 using the pressure evolution coupled method under SP and DP arithmetics.

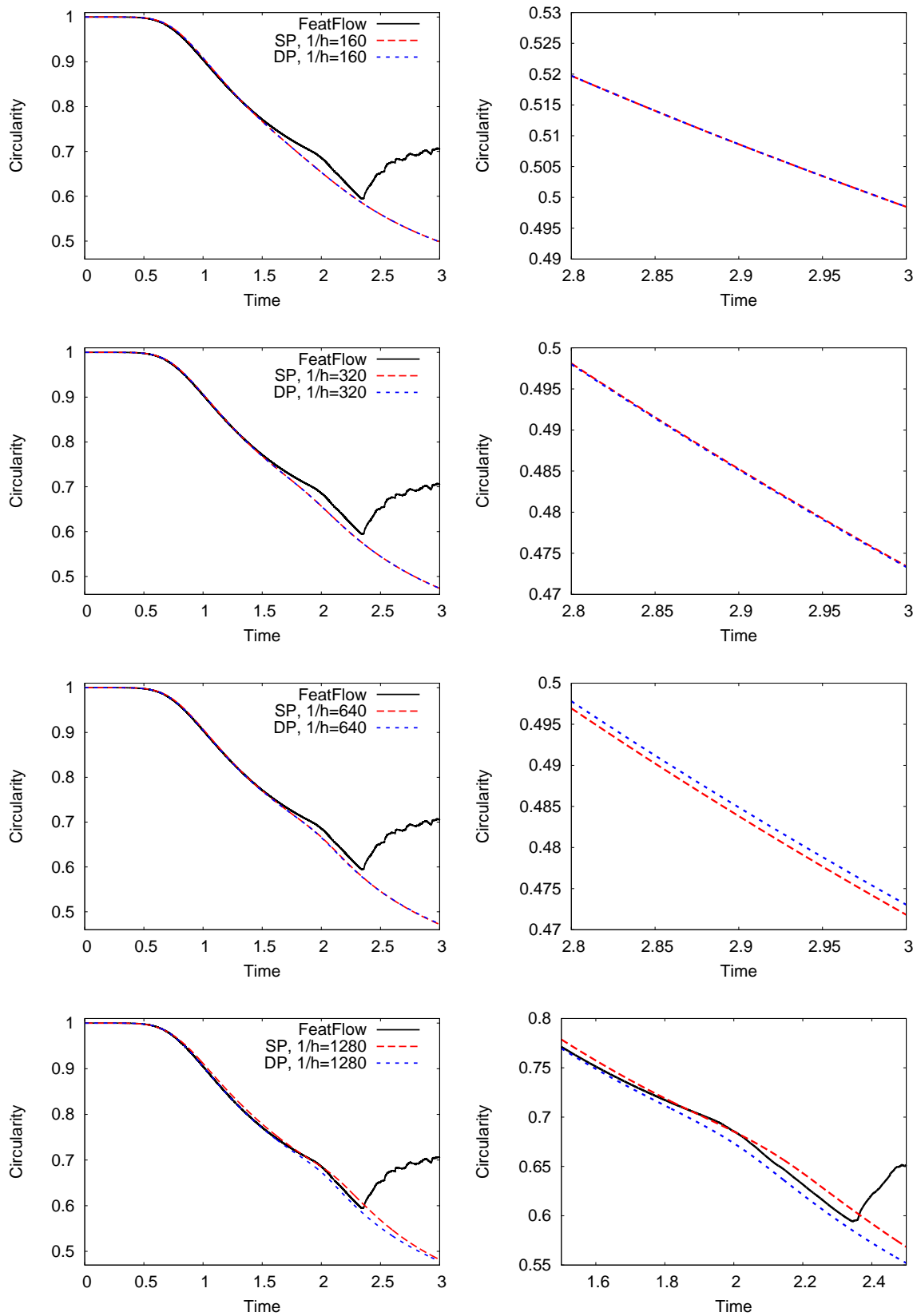


Figure 6.64: Time evolution of bubble circularity for 2D TC2 using the pressure evolution coupled method under SP and DP arithmetics.

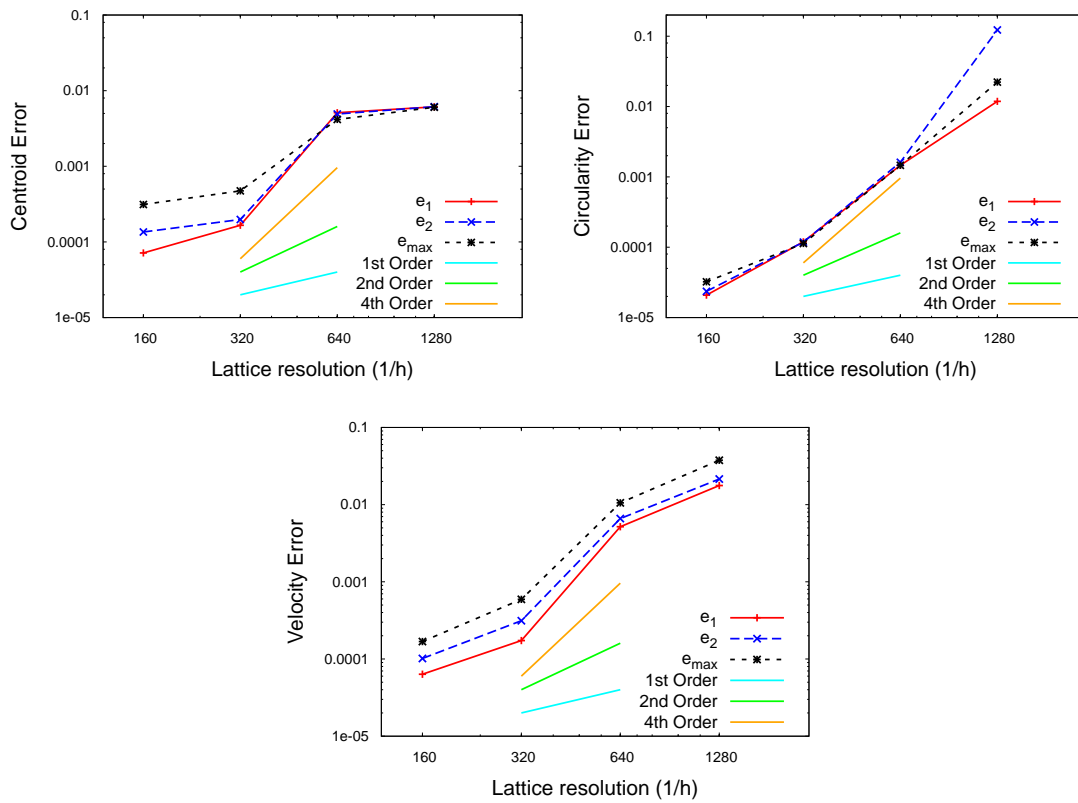


Figure 6.65: Errors in SP computations relative to DP ones for 2D TC2 using the pressure evolution coupled method.

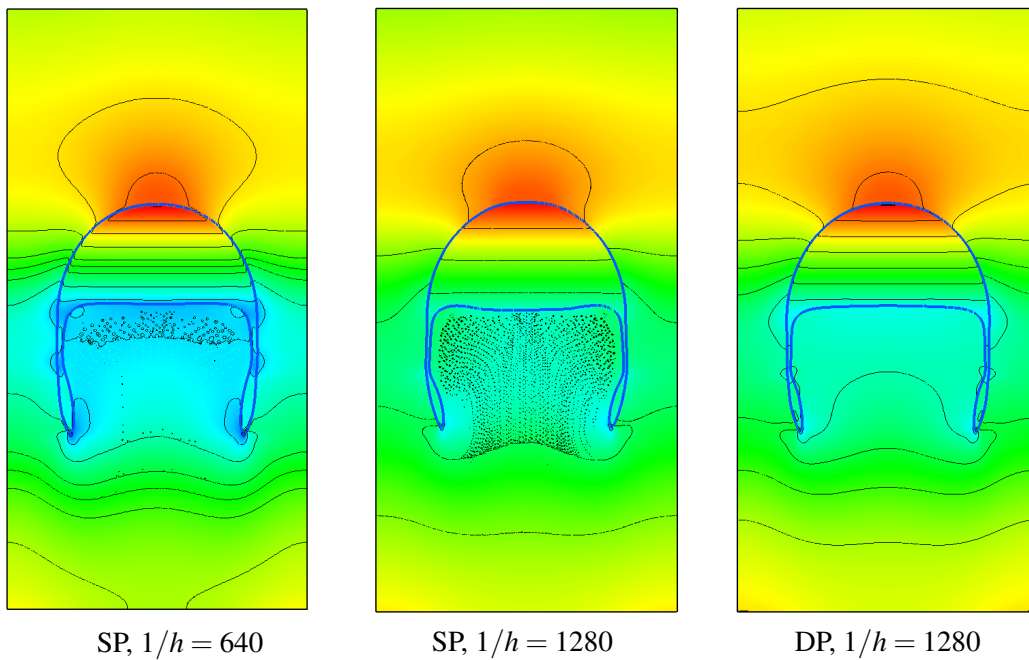


Figure 6.66: Pressure contours and the interface shape at $T = 3$ for the 2D TC2 using the pressure evolution coupled scheme under SP and DP arithmetics.

to creep in around the bottom of the bubble and along the filaments.

The above observations imply that exploiting SP computations could be legitimate in order to save time and memory for low density ratio problems, e. g., TC1 where the results obtained using $1/h = 320$ are already close enough to convergence as was seen in section 6.3.1. For problems involving high density ratio and large EO numbers, e. g., TC2 where resolving the pressure and tracking a highly deformed interface are crucial, using high resolution lattice becomes a compelling which may invalidate the results obtained in SP, depending on the desired level of accuracy.

6.7. Application to non-benchmark flows

In the previous sections, it was shown that the pressure evolution coupled scheme performs reliably for standard rising bubble test cases and convergence towards the direct solution of the Navier-Stokes equations is eventually attained by refining the underlying lattice. The aim of this section is to further examine the flexibility of this scheme in coping with complex topological changes which may include thinning filaments and/or merging and breakup of bubbles or droplets under pressing Re numbers and high deformation rates. For this purpose, three interesting problems are chosen; the Rayleigh-Taylor instability, the falling of a droplet on a thin liquid film and bursting of bubbles at the free surface. It must be noted that detailed validations against other numerical or experimental data is refrained for the forgoing problems as the numerical stability and accuracy has already been established for the present coupled schemes.

6.7.1. Rayleigh-Taylor instability

The Rayleigh-Taylor instability encompasses the mixing of two immiscible fluids, where the top heavy fluid starts to mix through the lighter one below due to the gravity force. The surface tension effects are assumed to be negligible and the problem is characterized with Reynolds and Atwood numbers

$$Re = \frac{\rho_1 \sqrt{g} (L_0)^{3/2}}{\mu_1}.$$

$$At = \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2}.$$

which together with setting a gravity parameter $\sqrt{L_0^* g}$, regulate the mixing intensity. Following the problem definition in [53], a 2D domain of size 1×4 is selected and periodic boundary conditions are applied to the side walls while the top and bottom boundaries are set to no-slip walls. The non-dimensional numbers are set to $Re = 256$ and $At = 0.5$ and the gravity satisfies $\sqrt{L_0^* g} = 0.04$.

The mixing is triggered via adding an initial perturbation of magnitude $0.1 \cos(2\pi x)$ along the $y = 2$ line. Moreover, the time is measured as $T = t_{lb} / \sqrt{L_0^* / g_{lb}}$. Figure 6.67 illustrates the results using a lattice resolution of $1/h = 200$ in the x direction. The front of the heavier fluid first takes a mushroom head and then sheds a pair of thinning spikes, clearly to enhance the mixing. A qualitative comparison to the similar data from [53] shows that the method is quite successful in capturing the eventual mixing effect and the complex deformations in the interface.

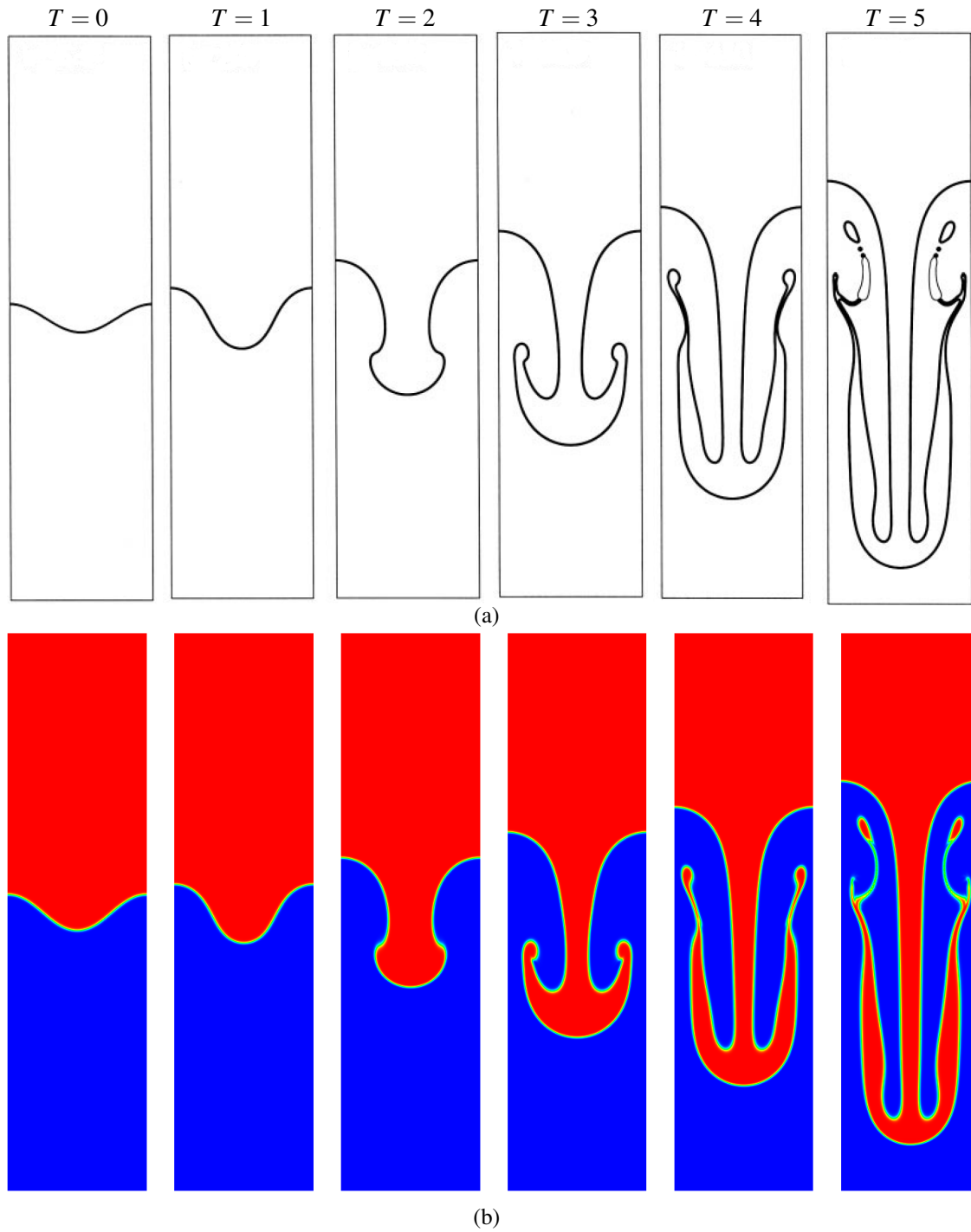


Figure 6.67: Mixing of liquids over time in Rayleigh-Taylor instability problem, (a) data from [53], (b) present coupled LB-LS scheme.

6.7.2. Droplet Splashing on thin liquid film

Splashing of a droplet over a thin liquid film is an intriguing problem for both physicists and mathematicians. In fact, the resulting splashing patterns could vary significantly with the falling velocity and the density of the droplet. On the other hand, the employed numerical technique has to resolve the bridging and then merging of the liquid interfaces under typically high impact speeds without causing the solver to suffer from the local singularities formed at contact spots.

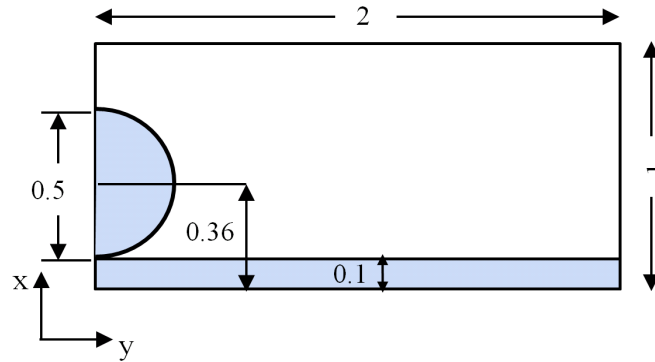


Figure 6.68: Problem definition for the 2D droplet splashing on a thin liquid film.

The different splashing regimes are classified using the Reynolds number Re as usual

$$Re = \frac{\rho_1 U_0 (2r_0)}{\mu_1}.$$

together with the Weber number We

$$We = \frac{\rho_1 U_0 (2r_0)^2}{\sigma}.$$

where U_0 is the velocity of the droplet at the impact moment. In order to address the problem numerically, the 2D configuration used in [75] is adopted here, which is also extended to 3D simulations. The 2D problem consists of a domain of size 2×1 where only half of the problem is considered and symmetry and periodic boundary conditions are used at $x = 0$ and $x = 2$ planes, respectively. The lower boundary is set to no-slip wall and a second order extrapolation is used to find the unknown distribution functions at the top boundary. A circular droplet with radius $r_0 = 0.25$ is initially located at $(x, y) = (0, 0.36)$, while the liquid film underneath has a depth of $H = 0.1$ as seen in figure 6.68.

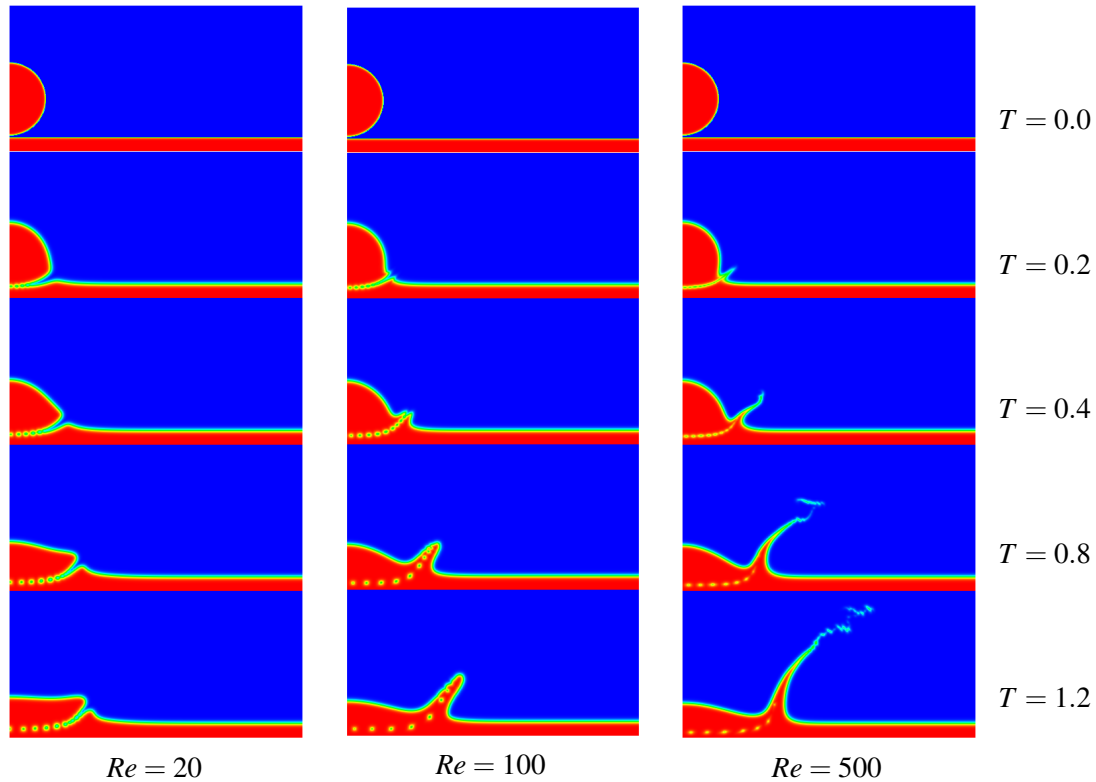


Figure 6.69: Temporal evolution of the 2D droplet splashing over a thin film under different values of Re number.

In the following numerical experiments, the density and viscosity ratios are set to 1000 and 40, respectively and the surface tension forces are realised via $We = 8000$ for all cases. Figure 6.69 shows the temporal pattern of the splashing phenomenon on a lattice resolution of 256×512 , where the time is measured based on $T = U_{0,ib}t/b/(2r_0^*)$. It can be seen that for $Re = 20$ which implies a small impact velocity, the droplet will be slowly deformed and flattened upon the film with no splashing effect developed. By increasing the Re number to 100 and 500, the splashing becomes fairly pronounced with a large rim and elevated fingers in the form of thin filaments.

To see how the 3D solution may affect the dynamics of the predicted splashing, an extension to 3D considered. The solution covers only 1/4 of the problem with symmetry boundary conditions on $x = 0$ and $y = 0$ planes, where the bubble hits the film along the z axis. The lattice resolution is $1/h = 128$, giving a domain size of $128 \times 256 \times 256$. Figures 6.70 and 6.71 show the 3D frames extended in the x direction for better visual observation. It could be seen that the edges of the ring become more straight while the ring itself attains a smaller radial extension as compared to the 2D case. Moreover, the spin-off of the satellite bubbles from the thinning rim for $Re = 500$ is more clearly captured and visible in 3D. Figure 6.72 presents a closer comparison between the 2D and 3D solutions on the cross section of the impact zone for different Re numbers. Analogous to the study of the 3D rising bubbles, a major source of difference between the 2D and 3D cases is speculated to come from the increased surface tension effects in 3D which try to resist the deformations. In addition, the present 3D problem allows for the distribution of the initial kinetic energy of the droplet over a broader area, causing the effective impact zone to become relatively smaller than that of the 2D case.

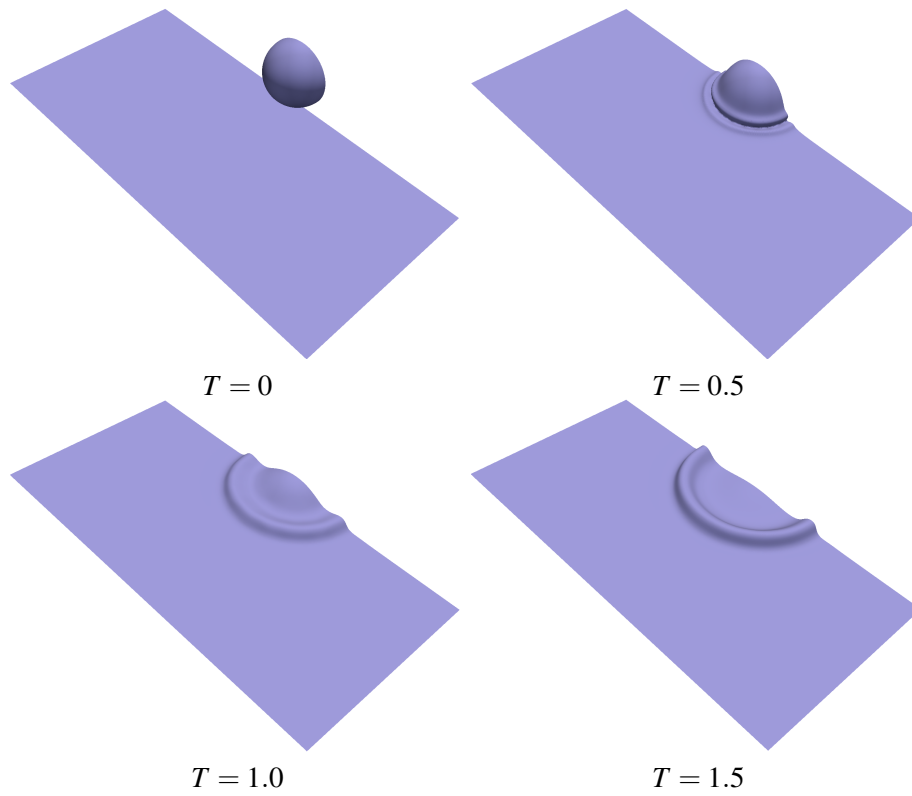


Figure 6.70: Temporal evolution of the 3D droplet splashing over a thin film for $Re = 100$.

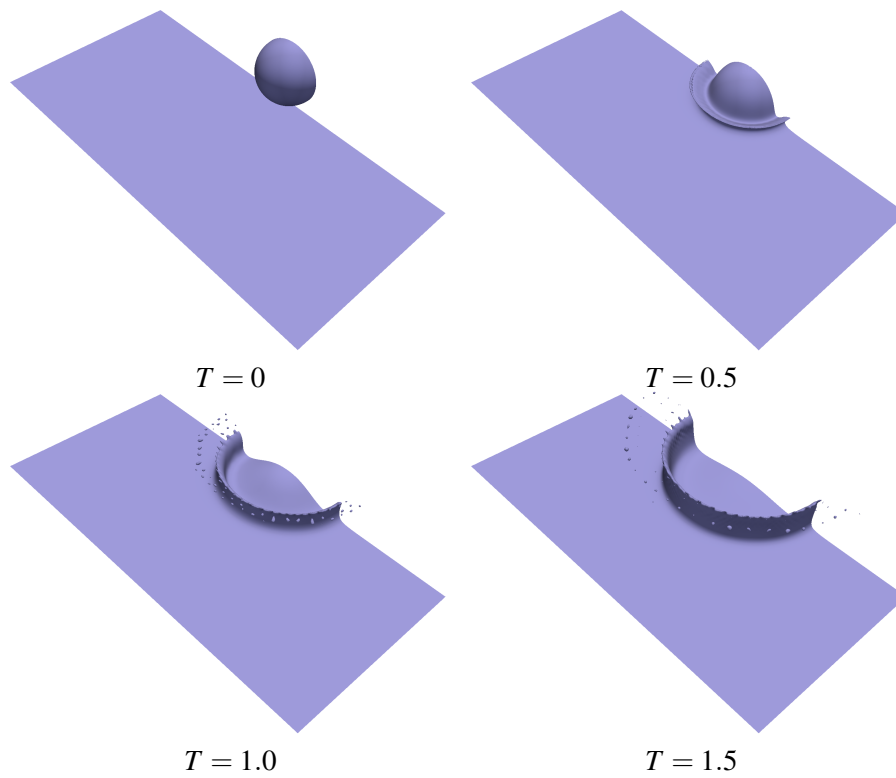


Figure 6.71: Temporal evolution of the 3D droplet splashing over a thin film for $Re = 500$.

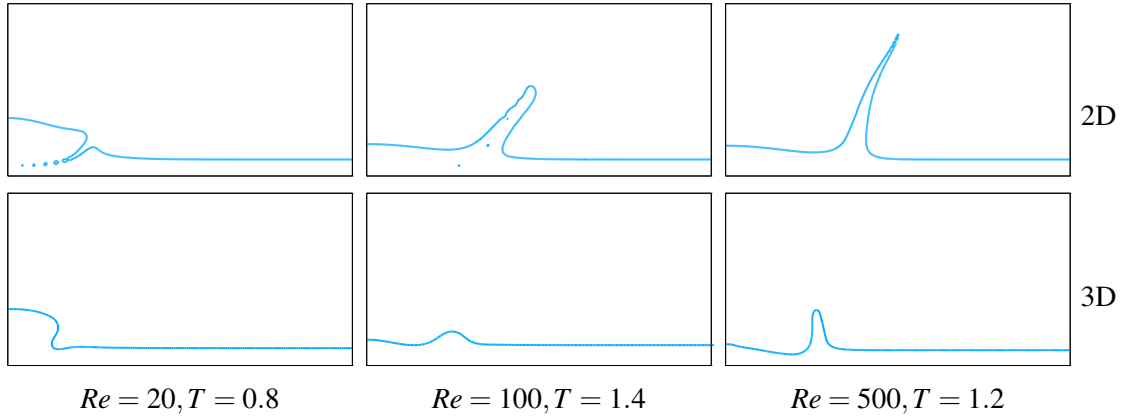


Figure 6.72: Cross section of the droplet impact zone in 2D and 3D problems and for different Re numbers.

6.7.3. Collision of binary droplets

The intriguing problem of binary droplet collision was first extensively studied in the experimental work of Ashghriz and Poo [15] where the evolution of the bubbles in different impact angles and arrangements was studied. Most importantly, they characterized the collisions using the Weber We and Reynolds Re numbers as in the case of droplet splashing, along with two additional parameters to specify the relative size and the impact angle of of the two droplets

$$\mathcal{D} = \frac{d_1}{d_2}.$$

$$B = \frac{2X}{d_1 + d_2}.$$

where d_1 and d_2 are the diameters of the two droplets and X is the distance between the center of one droplet to the relative velocity vector placed on the center of the other droplet [59]. In the present numerical experiments it is assumed that the droplets are of the same size, i. e., $\mathcal{D} = 1$ and $d_1 = d_2 = D$. The 3D domain is of size $3D \times 3D \times 6D$ and the droplets are initially placed $2D$ apart. The density and viscosity ratios are $\rho_2/\rho_1 = 1000$ and $\nu_2/\nu_1 = 100$, and the droplets collide with identical (and opposite) velocities of $U_0/2$. Periodic boundary conditions are considered on all sides. For the case of a *head-on collision*, i. e., $B = 0$, the symmetry allows to solve for only 1/4 of the problem. As outlined in [15], for $500 < Re < 4000$ the Reynolds number has no significant role in the dynamics of the impact and it is thus fixed at $Re = 500$ such that the velocity magnitudes will remain in the low Ma regime on moderately fine lattices. The progress in time is measured in the units of U_0/D .

Figure 6.73 shows the evolution of the colliding droplets for $We = 15, 25, 40$ using a lattice resolution of $1/h = 128$. For the high surface tension case of $We = 15$, the collision pattern falls into the *coalescence* regime, where the droplets merge and first form a circular disk. The disk then stretches out which is derived by the curvature difference between its exterior and the interior. This initiates an oscillatory motion which eventually ends up in a larger circular droplet as $t \rightarrow \infty$. By increasing the Weber number to $We = 25$ the weaker surface tension allows for a thinner and larger disk, a stronger stretching and finally the spin-off of the merging droplets, after which, the droplets travel away from each other. This regime is therefore called *reflective separation*. By further increasing the Weber number to $We = 40$ the enhanced stretching causes the entire system to extend more such that the two round ends will no longer hold attached to the middle section

and break-up, forming a third satellite bubble. The third bubble then remains in the middle and oscillates until becomes perfectly circular.

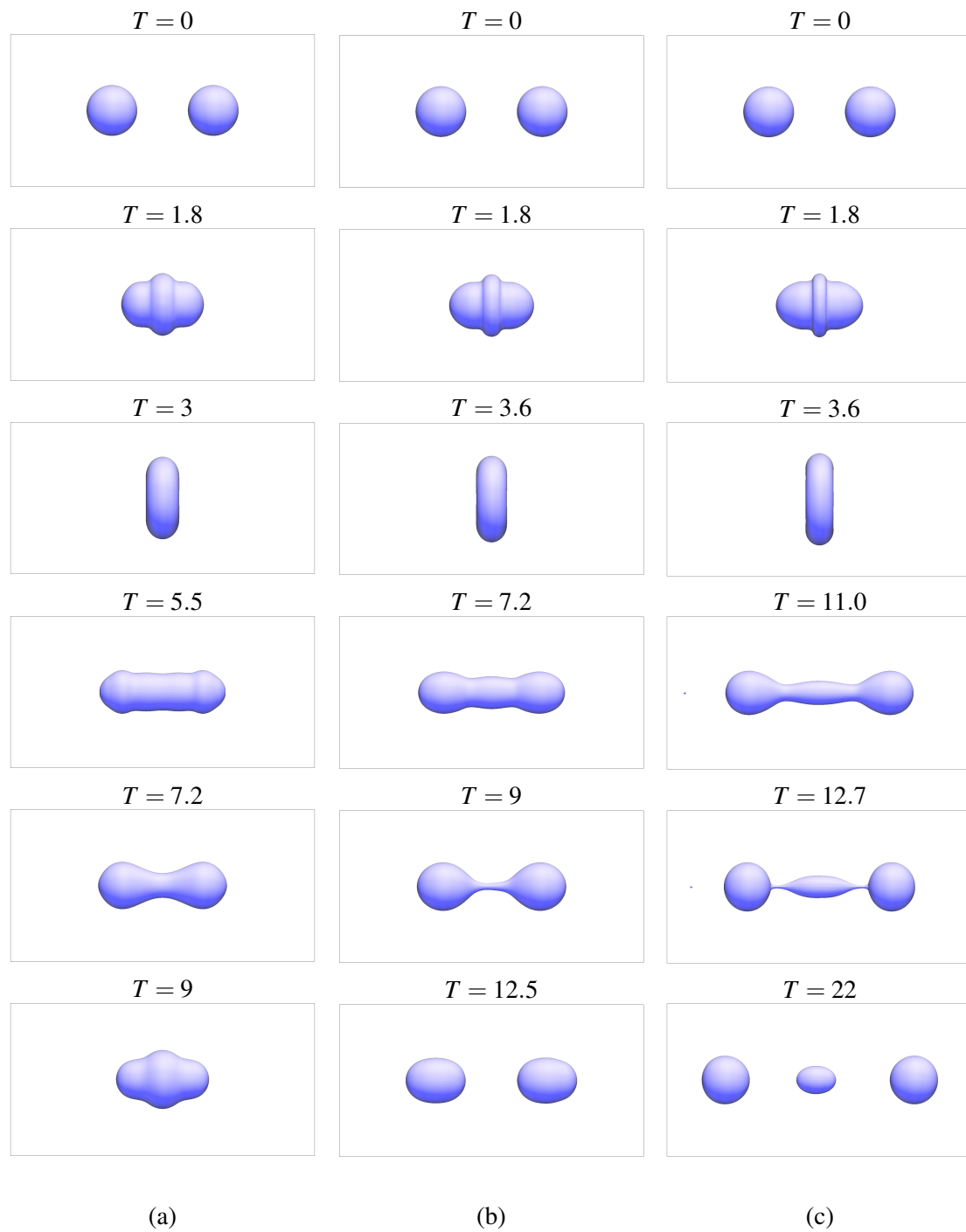


Figure 6.73: Head-on collision of micro-droplets with $B = 0$, $Re = 500$ and (a) $We = 15$, (b) $We = 25$, (c) $We = 40$.

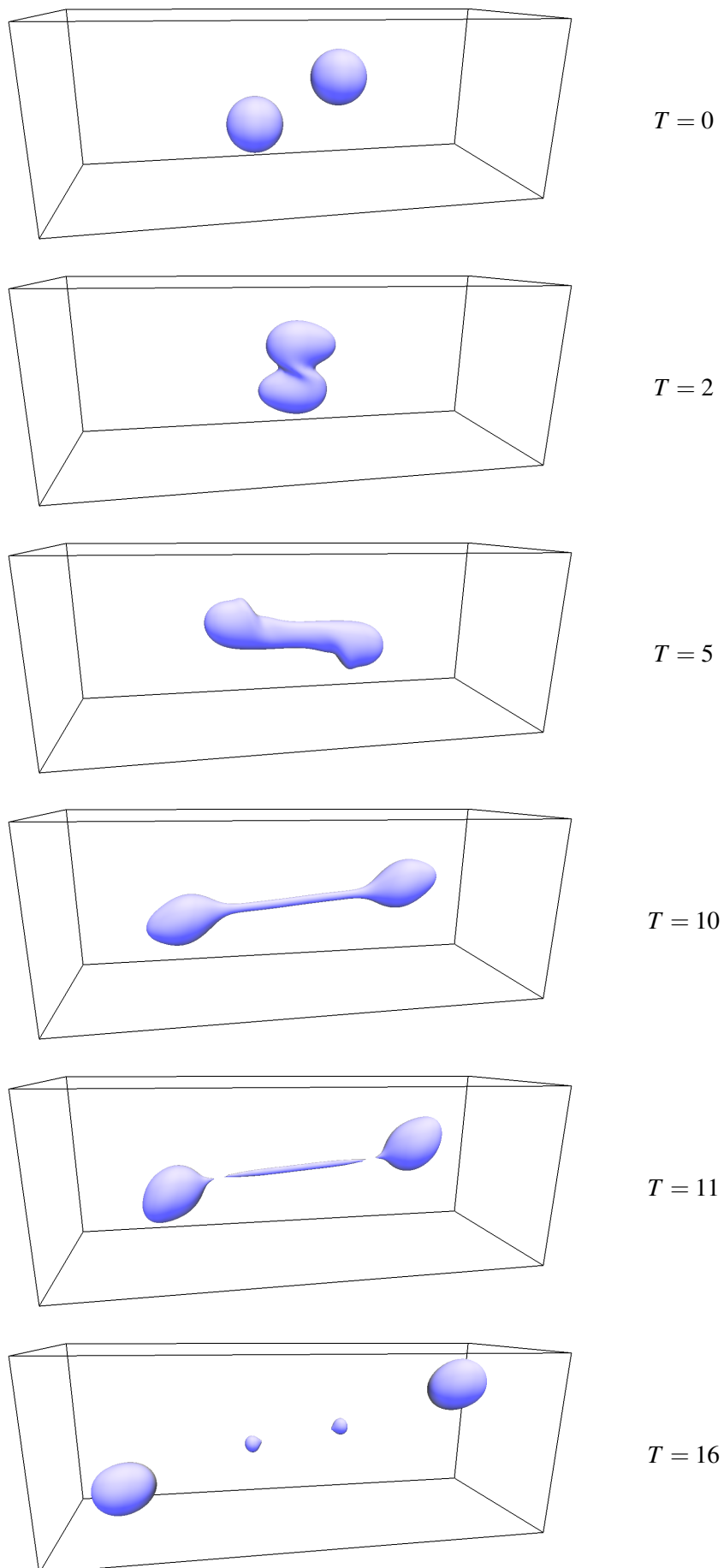


Figure 6.74: Oblique collision of micro-droplets at $Re = 500$, $We = 80$ and $B = 0.8$.

Figure 6.74 also shows the simulation result for the case of an oblique collision with $We = 80$ and $B = 0.8$. The droplets partially collide, but keep travelling close to their initial path. The oblique collision, however, causes a twist and rotary motion which creates a long connecting filament. The filament then separates from the main droplets and further breaks up into two satellite droplets. The obtained results for the head-on and oblique collisions closely repeat the patterns reported in [15]. Minor discrepancies, however, are expected since the boundary conditions and fluid properties might differ from those in real experiments and the LS method can not resolve the exact chemical potential field.

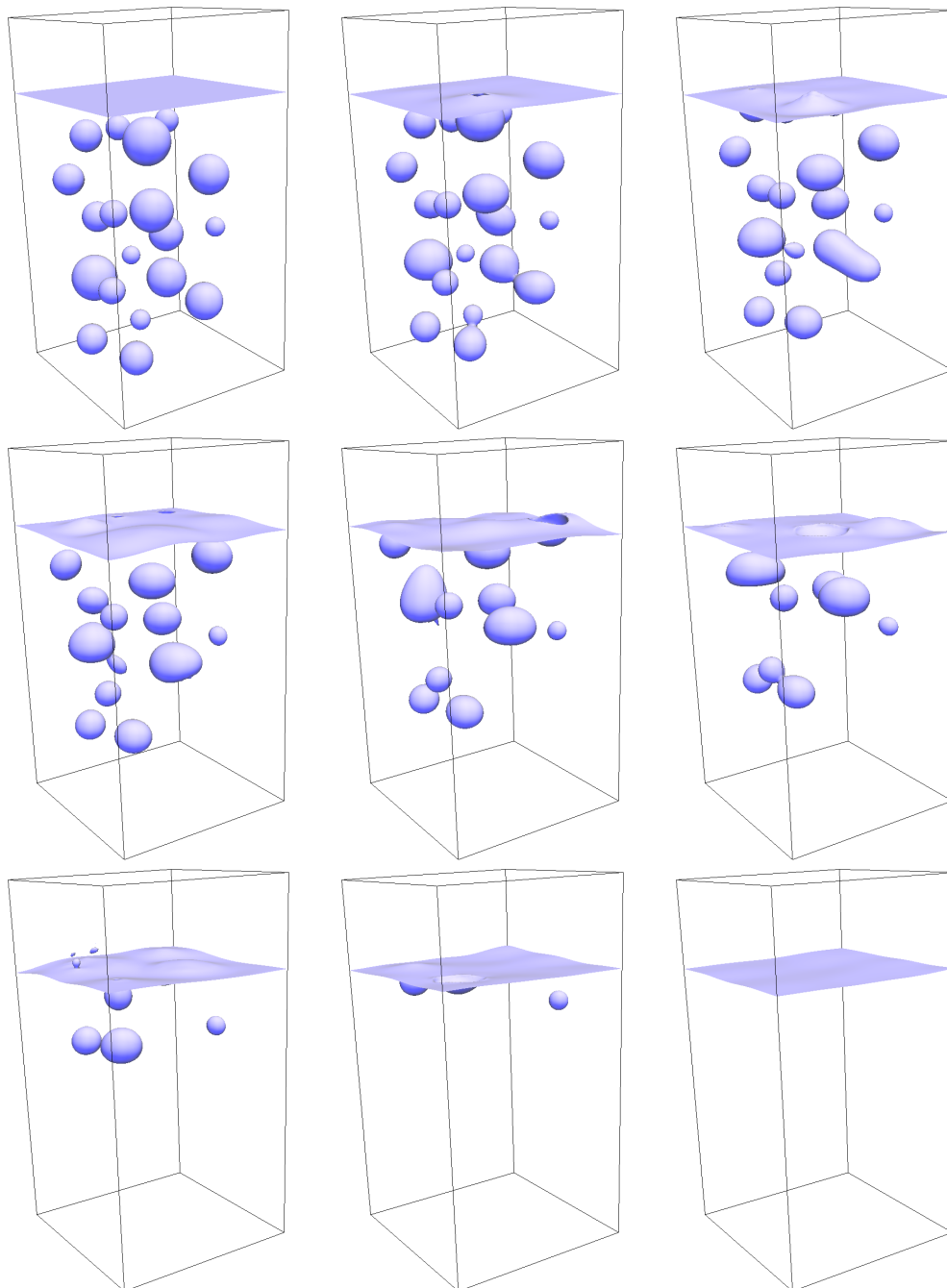


Figure 6.75: Merging and bursting of 3D bubbles at the free surface over time.

6.7.4. Bubbles bursting at free surface

The last test case is the problem of bubbles bursting at a free surface which is a popular phenomena in free-surface flow studies. It basically involves opposite effects as the droplet splashing problem, where the entities of the lighter fluid merge and eventually collapse as they reach the free surface. Figure 6.75 shows the evolution of a group of 20 randomly sized and placed bubbles as they ascend in a heavier liquid using a grid resolution of $1/h = 256$. The problem setting for density and viscosity ratios as well as for Re and EO numbers follows that of 2D TC1 in section 6.1.3 based on the largest bubble diameter, except that the density and viscosity ratios are set to $\rho_2/\rho_1 = 1000$ and $\nu_2/\nu_1 = 100$ and the lattice resolution is $1/h = 320$. Top and bottom boundaries are set to no-slip walls, while periodic boundaries are applied to the side walls. Some bubbles are first trapped inside the larger ones, forming new bubbles with increased sizes. Eventually the bubbles burst at the surface which entails the breakup of a very thin sheet and leaves singularities at the bursting ring. Nevertheless, the coupled LB-LS solver successfully handles the singularities and realises a smooth release of the bubbles into the bulk of the lighter fluid.

Parallel implementation and computational performance

Favourable parallel scalability is one of the most desired features of general single-phase LBM flow solvers and preserving this property for two-phase flow solutions is thus of a paramount importance. To acquire this goal, designing a nicely scalable algorithm is only the first. In fact the emergence of the additional elements for two-phase computations into the LBM and the subsequent coupling with the level set equation, bring along new implementation challenges with regards to parallel processing. This becomes quite imperative when developing codes for GPGPUs, as the programming model and hardware architecture of these many-core processors necessitates special measures to benefit from the typically vast amount of memory bandwidth and floating point throughput available on modern GPGPUs. Parallel implementation on GPGPUs is emphasized here since by the advent of these machines, extensive studies [42, 69, 89, 124] have revealed the unique relevance between LBM's solution algorithm and GPGPUs computational model, offering in turn remarkable parallel performances as compared to those obtained on CPU-based architectures. As such, this chapter first reviews the specific requirements of an efficient GPGPU implementation. The distribution of the computational workload for the present coupled LB-LS schemes is briefly analysed, which is followed by outlining a number of optimization strategies to approach the best parallel efficiency. The obtained computational performances are eventually analysed and parallel scalability is compared with those obtained on multi-core CPUs.

7.1. GPGPU programming model

From the hardware point of view, GPGPUs are designed in a way to maximize the distribution of the computational load among many cores which are grouped in the so called streaming multiprocessors (SM¹). Each SM has its own computational resources, e. g., texture and shared memory, L1 and L2 caches and finally registers which receive the data from the GPGPU's (henceforth called *device*) memory and feed them to a maximum of 192² micro-processor cores with typically low clock rates. The large number of processors together with a high memory bandwidth between the device global memory and the on-chip memories on each SM consequently results in computational powers as large as 4 Terra Floating Point Operation per Second (TFLOPS) in single precision and 1.3 TFLOPS in double precision on modern Kepler GPGPUs. These devices are so compact that can fit into a PC board and have prices and power consumptions typically comparable to high-end multi-core CPUs.

From the programming point of view, application programming interfaces (API), e. g., NVIDIA's CUDA C which comes with *nvcc* compiler, are designed to distribute the computations among as many compute *threads* as possible where each thread carries out a series of memory and com-

¹Also termed as SMX in NVIDIA's new generation Kepler chips

²for Kepler and Titan generation chips

pute operations in parallel with the other ones in the so-called single instruction multiple thread (SIMT) fashion. To keep the model compatible with the SM-based hardware design, threads are grouped in *thread blocks*, where at least one block is assigned to a SM at a time. On the lowest level, the group of blocks constitutes the *grid* of blocks which completes the distribution model. Similar to the shared memory programming model, the threads within each block can share SM's memory and compute resources among each other with the possibility of being synchronized. On the highest level, all the operations for a group of 32 threads in a block, called a *warp*, will be managed in parallel by the SM. Eventually, the obtained structure of threads, blocks and grids, may directly or indirectly represent the numerical data structure. The user has the control over designing this structure. The data is then allocated and initiated on CPU side (henceforth called *host*) and then copied to the GPGPU side (device). The program finally launches a special C-like function, called *kernel*, which is invoked under the hood for as many threads as launched within the grid. An illustration of the GPGPU's programming model is given in figure 7.1.

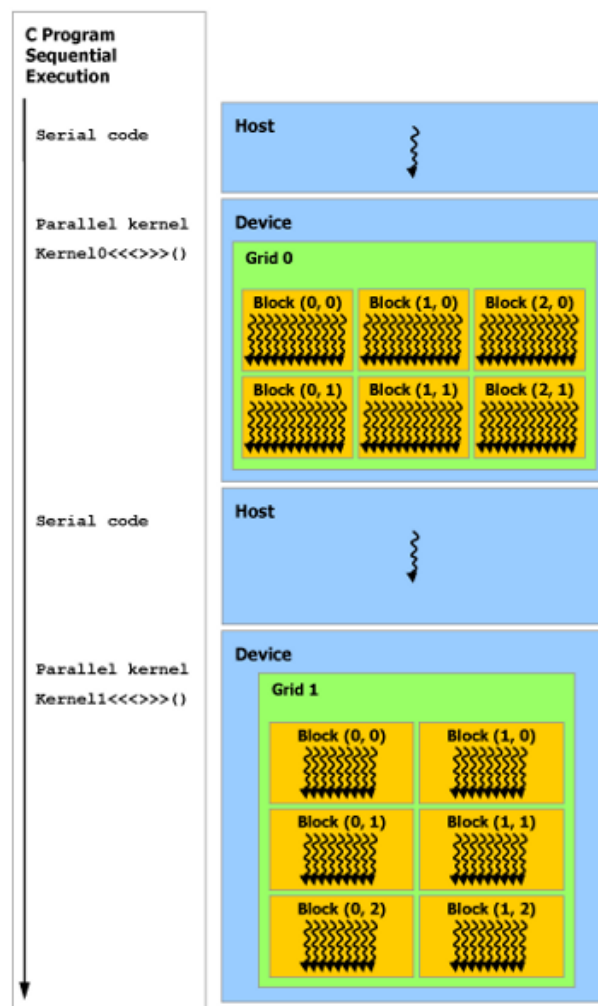


Figure 7.1: Illustration of the programming model on GPGPUs.

In most finite difference based discretizations, including LBM and the present LSM with structured grids, updating the values on each lattice node is independent from the neighbouring ones and the mapping between the computational grid and GPGPU grid is thus direct, i. e., each thread points to one lattice node.

7.1.1. Performance Optimization

In order to extract the highest achievable FLOPS and bandwidth on GPGPUs, a number of optimization guidelines must be always respected in the actual application programming. Elaborate discussions on performance optimization as well as detailed hardware descriptions are given by GPGPU manufacturers, e. g. in [1] and [2], for various hardware capabilities. A number of general guidelines though apply to every GPGPU application and could be listed as below,

Minimizing host-device transfers

Typically, data transfer between host and device has a very low bandwidth as compared to that between device memory and its processors. Moreover, such data transfers are synchronous with the kernel launches, i. e., computations could not be overlapped with memory transfers and thus slow down the simulation.³

Minimizing memory accesses in the device code

Internal data access on modern GPGPUs could benefit from a high global memory bandwidth. Nevertheless, the latency of memory accesses is still greater than that of computations and global memory requests are thus suggested to be reserved only for the main input and output data, as well as those intermediate data which have to be exchanged between the kernels.

Coalescing global memory requests

The data associated with the 32 threads in a warp are coalesced into a single memory access and are picked from the global memory and saved into the SM's cache lines of limited size⁴. If all data requested by a warp could be found in one cache line, the transfer rate would be the fastest, preparing the cache for the next warp and reaching the maximum transfer rate. In case the data access pattern is such that more than one cache line is to be loaded for a warp, the memory bandwidth is then reduced according to the number of extra cache lines. For example, if two cache lines must be loaded to find 32 SP values, the bandwidth efficiency simply drops to 50% as the time for processing one warp could have been used to serve two warps. The above analysis suggests that one has to keep the memory requests local, i. e., to refrain from high order spatial dependency in updating each node's values, so as to preserve the bandwidth efficiency.

Avoid warp divergences

If all the threads within the warp take the identical compute path, including memory access and computations, then all threads are treated in parallel by default. However, in case that one or more threads take different paths as they proceed, the warp diverges and the computations turn serialized from the divergence point onward, in the code. This severely degrades the instruction and memory throughput depending on how prevalent the divergence occurs in the algorithm. The general recommendation in such a case is then to minimize the divergences or to postpone them towards the end of kernel computations, where the consequences would be less critical. An example of such a situation could be found in the case where immersed boundary forces on the interface of a sharp interface multiphase flow solver must be applied only locally via a conditional statement which causes the associated warps to diverge.

Keep the SMs busy enough

One way to push the GPGPU further to its limits is to keep the SM occupancy at an acceptable

³An exception is the asynchronous data transfer which is possible through allocating the host data on the so-called page-locked memory but only applicable when the data being copied is not used by the next kernel

⁴For example on Tesla K20Xm the cache lines size is 128 bytes which comprise a total of 16 KB L1 cache per SM while the device has a total of 1.3 MB slower L2 cache available to all SMs

level. By definition, occupancy is the ratio of the threads that a SM receives in the actual implementation to the maximum number of threads that can reside simultaneously on the SM. A high occupancy means that the processors may not remain idle and could be reached by playing with the size and the number of thread blocks considering the specifications of the GPGPU in use. Nevertheless, care must be taken that the resources on each SM are already scarce and a too high occupancy may excessively reduce the amount of registers and shared memory allocated for each thread and thus slow down the computations.

7.2. Efficient implementations for coupled LB-LS schemes

Before proceeding with outlining the best practice guides for the present two-phase flow solvers, it makes an awful lot of sense to analyse the workload associated with the major components of the solvers so as to concentrate the fine-tuning efforts on most computationally demanding parts. A first version of the code could then be designed and developed according to this early study and then fine-tuned to approach a proper performance for the problems of interest.

7.2.1. Workload analysis

An a priori workload analysis of the present coupled schemes quickly shows that LB-related routines are the most memory and instruction intensive parts of the simulation. The computations fall into force calculation, collision, streaming, applying the boundary conditions and finally updating velocity and pressure fields. Here, the D2Q9 discrete velocity model and the pressure evolution coupled scheme is considered as a case under study. In this model, calculating the force needs to compute 9 directional central and 9 directional upwind derivatives for the force terms of the LBE, plus 2 force vectors for pressure S_{α}^p , and surface tension S_{α}^{σ} , in the principal directions used in updating the hydrodynamics. The procedure is heavily memory-intensive and involves relatively few additions and multiplications. The collision part of the LBE consists of calculating the non-linear equilibrium functions, constructing the final force term S_{α} and then performing two matrix-vector multiplications for transformation to the moment space and back to the discrete velocity space. Unlike the force calculation part, the collision part is strongly compute-intensive and optimizations must concentrate on accelerating the instructions. The streaming part receives the post-collision values in 9 directions and writes them to the shifted addresses along the discrete velocities. The process then involves only memory transfers and no computation. Applying bounce-back boundary conditions for the solid walls could be well integrated into the streaming step as explained in [124] and involves an inevitable warp divergence if the solid nodes fall at the beginning or end of the thread block. Nevertheless, application of other boundary conditions, e. g., interpolation or symmetry which depend on post streaming values of distribution functions (DF), has to be done synchronously after the streaming step.

The computational load on the LS side is more concentrated around updating the convective terms $\mathbf{u} \cdot \nabla \phi$ and $\mathbf{u} \cdot \nabla \psi$ for sharp and phase-field LS equations, respectively. The algorithm mostly consists of reading the LS values with a large spatial support up to $(x_i \pm 3\Delta x_i)$ and conditional statements required in the WENO scheme. It is thus strongly memory-intensive and may involve out of cache memory requests and warp divergences depending on the local interface shape and velocity field.

Reinitialization of the LS function is done in a monolithic fashion as explained in chapter 4 for the phase-field scheme. The procedure is again memory-intensive and requires different derivatives of the LS function. In this sense, the computations could be done alongside with calculating interface normal vectors and curvature so as to avoid reading the LS values twice.

Reinitialization of the sharp LS function requires periodic reconstruction of the interface. The

involved operations, however, comprise a compute-intensive algorithm where the reconstruction part is particularly hard to be implemented efficiently on GPGPUs. This issue will be addressed later for fine-tuning strategies.

7.2.2. Performance optimization for two-phase flows

In addition to the general optimization rules given in section 7.1.1 which are respected based on the implementation guidelines in [42, 69, 89, 124], the above analysis of instruction and memory workload implies a number of further optimization measures as follows:

- Generate and use the directional forces on-the-fly and refrain from saving and writing them to and from the global memory.
- To avoid unnecessary warp divergences, domain sizes are strictly assumed to be multiples of 32.
- Unroll the `for` loops used in the calculation of directional forces, equilibrium functions and MRT collision. Unrolling is observed to be particularly crucial to the performance of the compute-intensive 3D MRT codes.
- For the pressure evolution scheme, limit the calculations for directional force as well as for convective term $\mathbf{u} \cdot \nabla \psi$ to central type if $|\nabla \psi| < 10^{-9}$ for DP and $|\nabla \psi| < 10^{-6}$ for SP and hence avoid the large spatial support of upwind and WENO schemes where a large number of memory requests are involved. For the one-fluid scheme, the criterion can be chosen as $|\phi| < 2(\epsilon/h)$.
- In order to further ease the instruction bottleneck in 3D simulations caused by the matrix-vector operations of the MRT scheme, the MRT collision can be limited to $\psi > (1 - 10^{-9})$ and $\psi > (1 - 10^{-6})$ regions for DP and SP computations, respectively, leaving the rest of the domain to rely upon the low cost SRT model.

The last two points comprise an adaptive computation refinement approach which comes as an effective tool in GPGPU implementations. In fact, a first estimation of $|\phi|$ or $|\nabla \psi|$ may act as a feedback to save computational time on smooth regions where expensive differencing schemes are not required.

Moreover, the MRT scheme in principle improves the stability of the solution wherever large velocity and pressure gradients are present. In problems, e. g. the present 3D rising bubble flow, no strong flow-induced pressure gradients are present far from the bubble since the solid walls prevent the flow to affect the far field. As such, the only critical zone will be the two-phase zone including the interface and the lighter fluid which are well covered by the MRT collision in the adaptive implementation. Nevertheless, such an approach must be followed with care, as it depends on the specific dynamics of the problem under study.

It must be noted that for both adaptive techniques above, the warp divergence caused by applying the above conditions is only local and affects those thread blocks containing parts of the interface.

In order to verify whether adaptive computation approach is numerically legitimate for the present problems, the obtained rise velocities for 2D and 3D TC2 are compared against those of full computations in figure 7.2, where the adaptive MRT computations are limited to the 3D case only and the 2D cases run the full MRT computations. The almost seamless agreement between the full and adaptive results supports the above arguments.

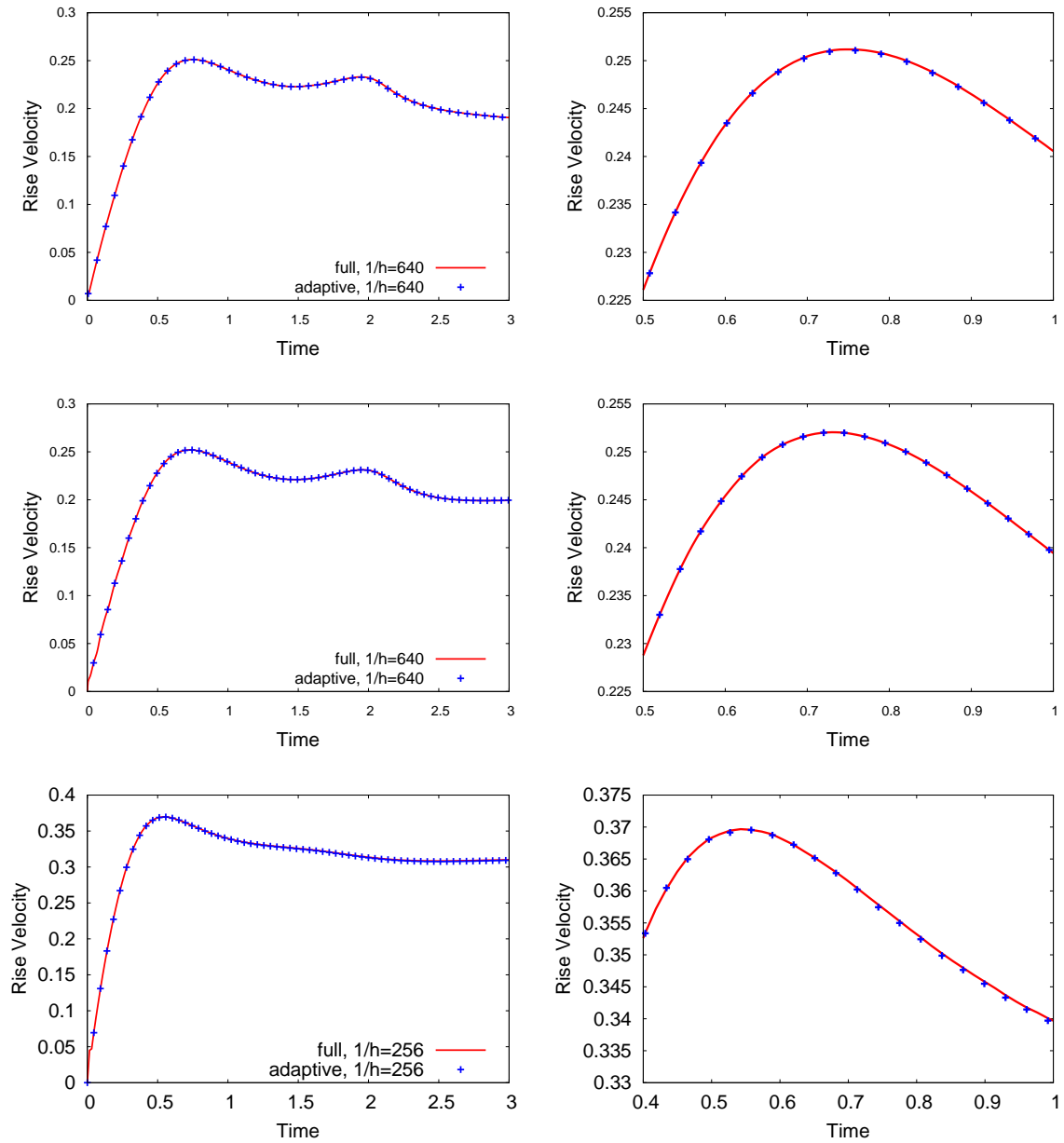


Figure 7.2: Time evolution of the rise velocity using full and adaptive computations for (a) 2D TC2 and pressure evolution scheme, (b) 2D TC2 and one-fluid scheme, (c) 3D TC2 using pressure evolution scheme and D3Q19 stencil.

7.2.3. Implementation outlines

In order to implement the actual code based on the aforementioned analysis, a group of GPGPU parallel kernels is defined as below:

- `kernel_collide_stream<<<>>>`

To avoid unnecessary memory accesses, force calculations, collision and streaming are carried out in a single kernel. The kernel reads the DFs, pressure, velocity, surface properties and LS function. It then calculates two-phase forces and equilibrium functions on-the-fly,

performs collision and bounce-back for fluid and solid nodes, respectively and conducts streaming along the discrete velocities. The streaming may include the periodic boundary condition as well. Post stream values of DFs as well as pressure and surface tension force vectors in the principal directions comprise the output of the kernel.

- `kernel_baoundary<<<>>>`
In case symmetry or extrapolation boundaries are involved, this kernel completes the streaming step by reading the known post-stream DFs and calculating the unknown ones.
- `kernel_properties<<<>>>`
Using the post-streaming DFs and the force vectors, this kernel updates the hydrodynamic properties.
- `kernel_calcNormals<<<>>>`
Based on the old values of the LS function, this kernel calculates the interface normal vectors. Moreover, the term $\Pi = \psi(1 - \psi)\mathbf{n}$ and $\nabla^2\psi$ used in the monolithic reinitialization of the phase-field LS scheme are also calculated as the by-products of this kernel.
- `kernel_calcCurvature<<<>>>`
Having the values of normal \mathbf{n} and Π over the entire domain, the kernel calculates the curvature $\kappa = \nabla \cdot \mathbf{n}$, as well as the compression term $\nabla \cdot \Pi$.
- `kernel_calcWENO<<<>>>`
All the computations for obtaining the convective terms $\mathbf{u} \cdot \nabla \phi$ and $\mathbf{u} \cdot \nabla \psi$ are carried out in this kernel.
- `kernel_solveLS<<<>>>`
Eventually, this kernel brings the LS function to the new time step using the convective and reinitialization terms (for the phase-field LS) on the right hand side of the LSE.

Hybrid CPU-GPGPU LS reinitialization

The reinitialization for the sharp interface LS function is hard to be entirely carried out on GPGPU. Indeed, it requires all the lattice cells to search for the interface segments and then reduce them into an array of $\mathcal{O}(1/h)$ size. As reduction algorithms are usually slow on GPGPUs, a coarse grained parallelization on a multi-core CPU could then be considered as an alternative. As such, a hybrid CPU-GPGPU reinitialization is proposed for the entire reinitialization procedure in 2D as follows

- At the beginning of each reinitialization step, the values of the LS function for the entire domain are copied from the device back to the host.
- The interface edges are then reconstructed on the host using a CPU-parallel code which conducts a search algorithm to interpolate and find the coordinates of the interface edges on several subdomains. The coordinates are then reduced into a single array on the host.
- An array of the size of order $1/h$, containing the interface data is copied back to the device.
- A kernel named `kernel_redistance<<<>>>` is assigned to find the minimum distance from each lattice node (corresponding to one thread) to the interface, thus creating a signed distance field. The kernel eventually updates the values of ϕ based on equation (4.3.11).

In fact, the overhead associated with device-host memory transfers is negligible provided that the algorithm is called only periodically. In addition, interface reconstruction search and reduction process is carried out efficiently in parallel on the CPU side, since the available cache space for CPU cores is by far larger than that for each SM on GPGPU.

7.3. Performance analysis

The performance of the GPGPU code developed according to the implementation outlines discussed above is presented in this section. In order to further examine the quality of the parallel scaling behaviour of the code on GPGPU, an optimized CPU parallelization based on the implementations in [105] is also performed for 2D codes.

Parallel computations are carried out on a high-end compute machine equipped with two Intel Xeon E5-2670 CPUs with a total of 32 processor cores at 2.60GHz, peak memory bandwidth of 51 GB/sec and peak performance of 0.332 TFLOPS in DP which is simply multiplied by 2 for SP computations. The machine also benefits from a Kepler K20Xm GPGPU, having a peak memory bandwidth of 250 GB/sec. The device has peak DP and SP performances of 1.32 and 3.95 TFLOPS, respectively. Nevertheless, this Tesla enabled machine is only used for DP implementations, while for pure SP codes, a compute machine equipped with GeForce GTX980 Ti gaming GPU with 5.6 TFLOPS in SP (and only 0.176 TFLOPS in DP), and 336 GB/sec of bandwidth is employed.

For a shared memory parallel implementation of the code on the multi-core CPU, OpenMP directives are used and the computational domain is divided into multiple subdomains along the y axis. For GPGPU implementations, NVIDIA's `nvcc` compiler for CUDA 7.5 is used where the OpenMP flag of `-xcompiler -fopenmp` is turned on for the interface reconstruction parts of the brute-force reinitialization on the CPU. Computations are done for 2D and 3D rising bubble test cases, while according to the discussions in section 6.6 and 6.5, 3D and SP implementations are restricted to the pressure evolution scheme only.

In order to verify the a priori workload analysis in section 7.2.3, a survey of profiling the GPGPU codes is presented in tables 7.1 to 7.5 using the K20Xm GPGPU and DP computations. In these tables, RT and WT denote the global memory read and write throughputs, respectively, and Occ denotes the achieved SM occupancy. The profiling is done by invoking the `nvvp` command provided as part of the CUDA toolkit.

By looking at the time contributions of each kernel, one readily notices that LBM related computations account for 75-90% of the simulation time, while the contribution of interface capturing kernels will be no more than 25%. This ensures that the present coupled schemes would be definitely less expensive than the entirely kinetic-based two-phase LB solvers, where the interface capturing LBE would be almost as expensive as the LBE for flow. Also, note that the absence of the monolithic reinitialization in the brute force scheme and extra computations for the viscous force term in the one-fluid scheme has led to a higher time contribution of the `kernel_collide_stream` in table 7.1 as compared to that in table 7.2 for the pressure evolution scheme.

The total RT+WT bandwidth usage is around 80% of the maximum bandwidth for pure memory-bound kernels, e.g., `kernel_properties`. The memory transfer quality is lower for compute-intensive kernels, e.g., `kernel_collide_stream` and keeps degrading as the lattice stencils become denser in 3D cases, also seen through the lowered occupancy of this kernel compared to 2D cases. This directly affects the cost of the `kernel_collide_stream` which grows from 58% for D3Q15 to 71% for D3Q27 stencil.

Table 7.1: Profiling survey of GPGPU implementation for the 2D one-fluid coupled scheme.

Kernel	%Total Time	RT (GB/s)	WT (GB/s)	Occ
kernel_collide_stream	67.3	70	34	0.20
kernel_properties	26.9	151	54	0.90
kernel_calcNormals	5.5	33	163	0.85
kernel_calcCurvature	2.7	128	64	0.85
kernel_calcWENO	6.9	84	55	0.64
kernel_solveLS	3.4	150	52	0.84
kernel_redistance	1.2	50	77	0.928

Table 7.2: Profiling survey of GPGPU implementation for the 2D pressure evolution coupled scheme.

Kernel	%Total Time	RT (GB/s)	WT (GB/s)	Occ
kernel_collide_stream	56.4	59	43	0.15
kernel_properties	16.1	156	45	0.86
kernel_calcNormals	7.8	35	161	0.86
kernel_calcCurvature	8.1	128	45	0.66
kernel_calcWENO	6.8	84	56	0.64
kernel_solveLS	4.8	163	38	0.86

Table 7.3: Profiling survey of GPGPU implementation for the 3D pressure evolution coupled scheme and D3Q15 velocity stencil.

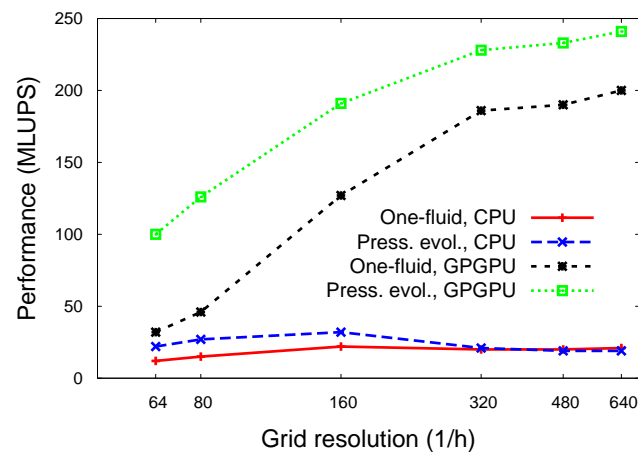
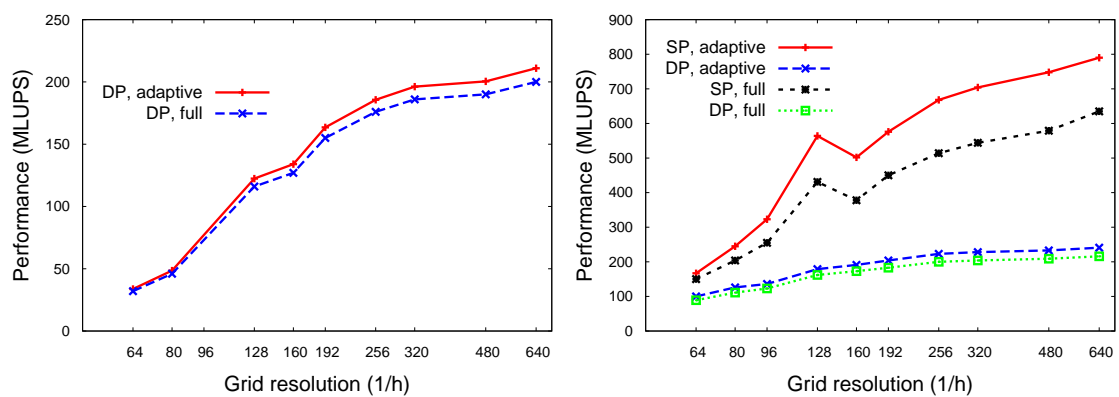
Kernel	%Total Time	RT (GB/s)	WT (GB/s)	Occ
kernel_collide_stream	58.7	55	41	0.12
kernel_properties	16.1	163	36	0.47
kernel_calcNormals	7.3	46	151	0.46
kernel_calcCurvature	7.4	166	29	0.66
kernel_calcWENO	5.7	140	58	0.64
kernel_solveLS	3.6	163	38	0.86
kernel_boundary	2.0	32	17	0.33

Table 7.4: Profiling survey of GPGPU implementation for the 3D pressure evolution coupled scheme and D3Q19 velocity stencil.

Kernel	%Total Time	RT (GB/s)	WT (GB/s)	Occ
kernel_collide_stream	63.3	52	39	0.12
kernel_properties	15.0	169	32	0.70
kernel_calcNormals	4.8	46	151	0.46
kernel_calcCurvature	6.3	166	29	0.66
kernel_calcWENO	6.2	140	58	0.64
kernel_solveLS	2.9	163	38	0.86
kernel_boundary	1.4	38	20	0.50

Table 7.5: Profiling survey of GPGPU implementation for the 3D pressure evolution coupled scheme and D3Q27 velocity stencil.

Kernel	%Total Time	RT (GB/s)	WT (GB/s)	Occ
kernel_collide_stream	71.1	42	40	0.11
kernel_properties	13.0	175	25	0.70
kernel_calcNormals	4.8	46	151	0.46
kernel_calcCurvature	6.3	166	29	0.66
kernel_calcWENO	6.2	140	58	0.64
kernel_solveLS	2.9	163	38	0.86
kernel_boundary	1.9	71	39	0.56

**Figure 7.3:** Weak scaling for the performance of the 2D two-phase flow solvers using DP arithmetics.**Figure 7.4:** Weak scaling for the performance of the 2D code using (a) one-fluid scheme, (b) pressure evolution scheme.

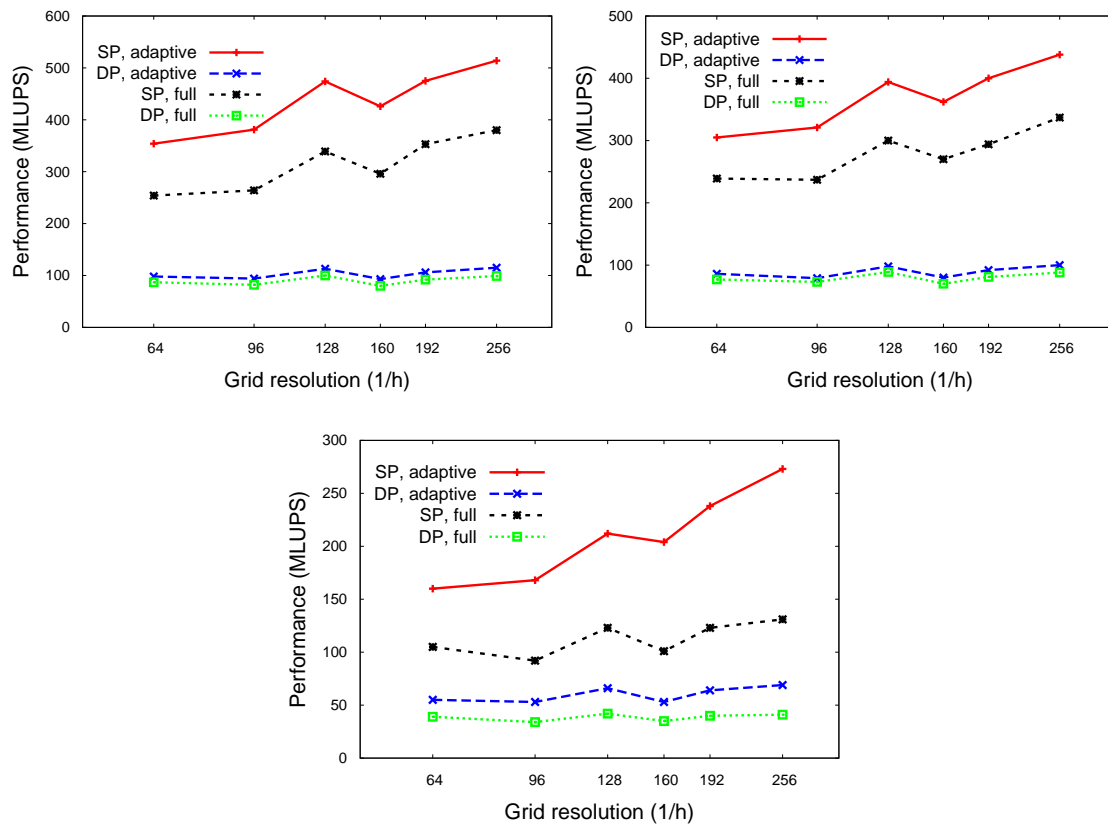


Figure 7.5: Errors in SP computations relative to DP ones for 2D TC2 using the pressure evolution coupled method.

Figures 7.3 to 7.5 show the weak scaling of the computational performance which is calculated using the conventional unit of *millions of lattice update per second* (MLUPS).

Figure 7.3 first shows the scaling of the 2D codes in DP on CPU and GPGPU architectures. The high bandwidth efficiency and low arithmetic workload of the 2D codes allows for an almost linear scaling of the GPGPU performance and an order of magnitude speedup over the parallel CPU codes. Nevertheless, the performance growth becomes sub-linear as the memory-bound code reaches the bandwidth limit. For parallel CPU implementations, the lower bandwidth of the machine causes the peak performance of 30 MLUPS to occur as early as for $1/h = 160$. The performance then slightly decreases for larger problem sizes as the CPU cache becomes already saturated and data has to be picked directly from the host DRAM.

Figures 7.4(a) and (b) show the effect of adaptive computations on the performance of the 2D codes. The high memory and instruction latency of the DP computations causes the adaptive computations to result in no more than 10% performance gain, while the adaptive SP implementation gives around 25% improvement in the performance of the pressure evolution code for large problems sizes. Eventually, note that both coupled schemes reach a similar performance level for 2D simulations, which means that the explicit interface reinitialization together with a more complex force term of the one-fluid scheme impose a similar overhead as that of the monolithic reinitialization of the phase-field LSM in the pressure evolution scheme.

The performance of the 3D simulations in figures 7.5 (a) to (c) reveals 35, 55 and 65% decrease in the peak MLUPS for D3Q15, D3Q19 and D3Q27, respectively, compared to the D2Q9 performance in both SP and DP. Again, the scaling is favourable for low latency SP computations and

becomes severely sub-linear for DP computations where the performance does not significantly increase with scaling the problem size. On the other hand, the effect of adaptive computations becomes quite apparent for the heavily compute-intensive 3D MRT implementations where the highest improvement of 100% is achieved for the SP implementation of the highly dense D3Q27 velocity stencil. Finally, note that for all 2D and 3D computations, the best SP performance is around 4 times higher than that for DP, which is in accordance with the factor of 4 difference between DP and SP peak FLOPS of K20Xm and GTX980 devices.

In order to compare the obtained performances with other LB-based two phase flow implementations on GPGPUs, one could refer to the recent works of Banari et al. in [17] and [18] for 2D and 3D simulations, respectively. Their model is based on the free-energy method of Inamuro et al. [59] which uses SRT collisions for both flow and interface capturing LBEs and updates the pressure via solving the pressure Poisson equation to enhance the stability at high Re numbers. They carried out DP computations on a Tesla C2070 GPGPU and reached 52 MLUPS for D2Q9, and 20 MLUPS for D3Q19 simulations and large problem sizes. In comparison, running the present codes in DP on Tesla C2070 device gives 105 MLUPS for D2Q9 and 30 MLUPS for the D3Q19 implementations and thus confirms at least 100% gain in 2D and 50% gain in 3D. Such an improvement is partly due to the specific techniques employed in the current work, but to a greater extent is a result of replacing the kinetic-based interface capturing method with the LSM. In addition, the largest possible 3D problem size in [18] occupied almost 6 GB of memory on the device, whereas the present code needs only 1.9 GB to handle the same problem size.

Conclusion and outlook

To close the thesis, a summary of the conducted studies as well as proposals for future works are presented in this chapter.

8.1. Summary and conclusions

Two coupled LB-LS approaches were proposed in this thesis and applied to a set of two-phase flow problems. The one-fluid approach was equipped with a sharp signed distance-based LS method for capturing the interface, while the interface was resolved in the pressure evolution scheme via a diffuse phase-field LS technique. Studying the accuracy and mass conservation of both interface capturing methods revealed that the phase-field approach is more flexible in realising complex topology changes and better preserves the mass. Nevertheless, the quality of the interface capturing strongly depends on the assumed interface thickness in the phase-field scheme.

On the flow solver side, consistent discretization and time integration schemes were investigated for both one-fluid and pressure evolution LBE through studying the static bubble test case. It was concluded that using the directional averaged differencing for both t and $t + \Delta t$ in the one-fluid formulation gives the best accuracy with regards to recovering the correct pressure and minimizing the spurious velocities. As compared to the pressure evolution scheme, it was observed that the pressure accuracy in one-fluid model may still suffer from relatively lower accuracy of the linear EOS. This implied that a larger number of lattice cells have to cover the interface in the one-fluid scheme to reach the same accuracy as in the pressure evolution scheme.

The two coupled approaches were finally applied to the rising bubble benchmarks in 2D and the results were compared to those from different solutions of two-phase Navier-Stokes equations. For low density ratio systems, both methods performed well as shown through examining flow quantities. Moreover, the convergence rates for the relative error norms confirmed the second order accuracy of both schemes. Examining the solutions under high density and viscosity ratios, showed relatively large errors in the velocity obtained by the one-fluid model on coarse lattices as the low accuracy of pressure was magnified by the large density ratios. Nevertheless, both schemes were quite successful in repeating the benchmark data and the overall interface shape provided that the underlying lattice is sufficiently refined.

Based on the 2D simulations, the effect of interface thickness was also investigated. The performed analysis showed that the one-fluid scheme requires $\varepsilon = \text{const.}$ regardless of the lattice resolution and the accuracy of pressure and velocity enhances with increasing $m = \varepsilon/h$. On the contrary, the accuracy of the pressure evolution scheme improves for $m = \text{const.}$, i. e., $\varepsilon \rightarrow 0$ with refining the lattice. Such a favourable property of the pressure evolution scheme suggested to develop a refine-grid level set model, by which a sharper resolution of the interface and an higher mass conservation was achieved for global coarse LB grids.

Studying the effect of the arithmetic precision revealed that, due to the particular form of the

extended force, the one-fluid model strictly requires DP arithmetics. Using SP computations for the pressure evolution, however, was shown to be legitimate up to a certain lattice resolution which is sufficient for solving low density ratio problems.

The pressure evolution two-phase method was then chosen for the 3D benchmarking studies where the method was quite successful in both qualitative and quantitative aspects compared to the reference solutions. Application of different discrete velocity models for the LBE revealed the lack of isotropy in the D3Q15 stencil as seen in the exaggerated deformations predicted in diagonal directions and relatively larger deviations in the bubble temporal quantities. Indeed, the D3Q19 stencil was shown to provide the required isotropy at a reasonable extra cost, while the D3Q27 stencil asks for more than 50% longer simulation times and no noticeable accuracy improvement upon the D3Q19 stencil.

Eventually, the parallel efficiency and scalability of the coupled schemes were examined through implementations on GPGPUs. Following the conventional optimization guidelines for GPGPU programming together with considering a number of adaptive computation techniques for the present two-phase solvers, an order of magnitude performance gain was achieved over optimized parallel CPU codes which is in the same range as for general single-phase LB simulations. The scalability is almost linear for 2D implementations where the pressure evolution scheme may gain slightly higher performances for sufficiently large problem sizes which is almost 100% more than similar fully kinetic 2D implementations on GPGPUs. 3D implementations using the pressure evolution approach though tend to make the solution more compute-intensive as seen by the low scalability of the obtained performances in DP. Nevertheless, the performance is only 55% less than that of the 2D code for the D3Q19 stencil, which is still 50% faster and asks for 60% less memory space than those of fully kinetic computations.

In conclusion, both presented coupled LB-LS schemes appeared to be promising tools with regards to accuracy for high density ratio and deformations rates. Their main advantage over conventional LB-based models, however, lies in the lower memory consumption and higher computational performance. In particular, the coupled pressure evolution approach was proved to be more robust than the one-fluid scheme with regards to accuracy, flexibility against topology changes and parallel efficiency. This confirms the suitability of the scheme to be integrated into commercial and academic CFD packages, e. g., FeatFLOW for large scale two-phase flow simulations. To this end, a number of key features could be envisaged to be included into the present solver as the future works, explained in the next section.

8.2. Outlook of the future works

8.2.1. Coupling with Cahn-Hilliard equation

A great deal of the recent efforts on developing LB-based two-phase flow solvers is focused on solving the CH equation via kinetic or finite difference techniques [17, 36, 76, 131]. The disadvantages of these implementations were discussed in chapter 3 in detail. However, in some situations, where deploying a more physically authentic solution is crucial to verify the delicate two-phase flow phenomena, e. g., merging, breakup or spreading on dry surfaces, the most reliable answer could be provided by solving the CH equation in its full, thermodynamically consistent form. This imperatively requires use of implicit time integration and high order space discretizations, e. g., iso-geometric finite elements [44], to cope with the strong non-linearity of the full CH equation.

In order to show how an explicit finite difference solution of the CH equation would behave, the LS equation is replaced with the CH equation (3.2.21) in the pressure evolution scheme and the resulting coupled LB-CH solution is applied to the 2D high density ratio problem of TC2. A constant mobility of $M = 0.001$ is chosen in accordance to implementations in [17] and [131] so as to ease the non-linearity. To keep the solution stable, the positivity of the concentration

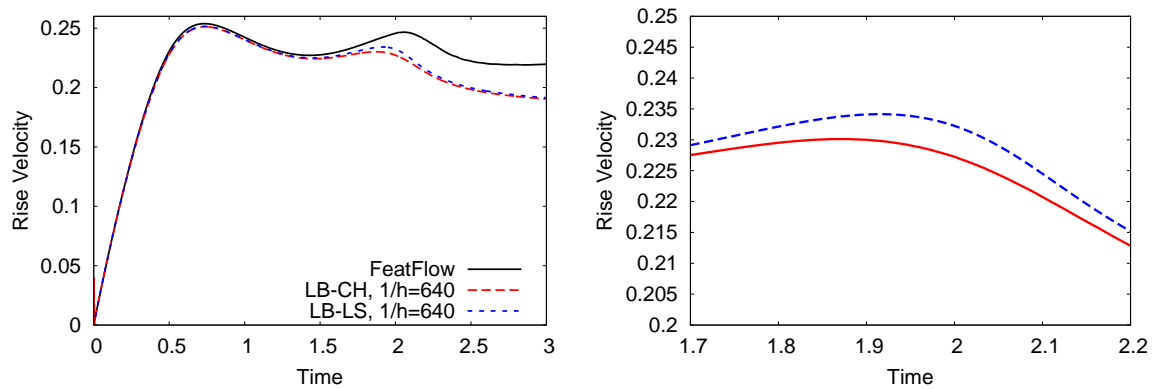


Figure 8.1: Time evolution of the rise velocity for the 2D rising bubble TC2 using Ls and CH interface capturing methods (left), and the enlarged view (right).

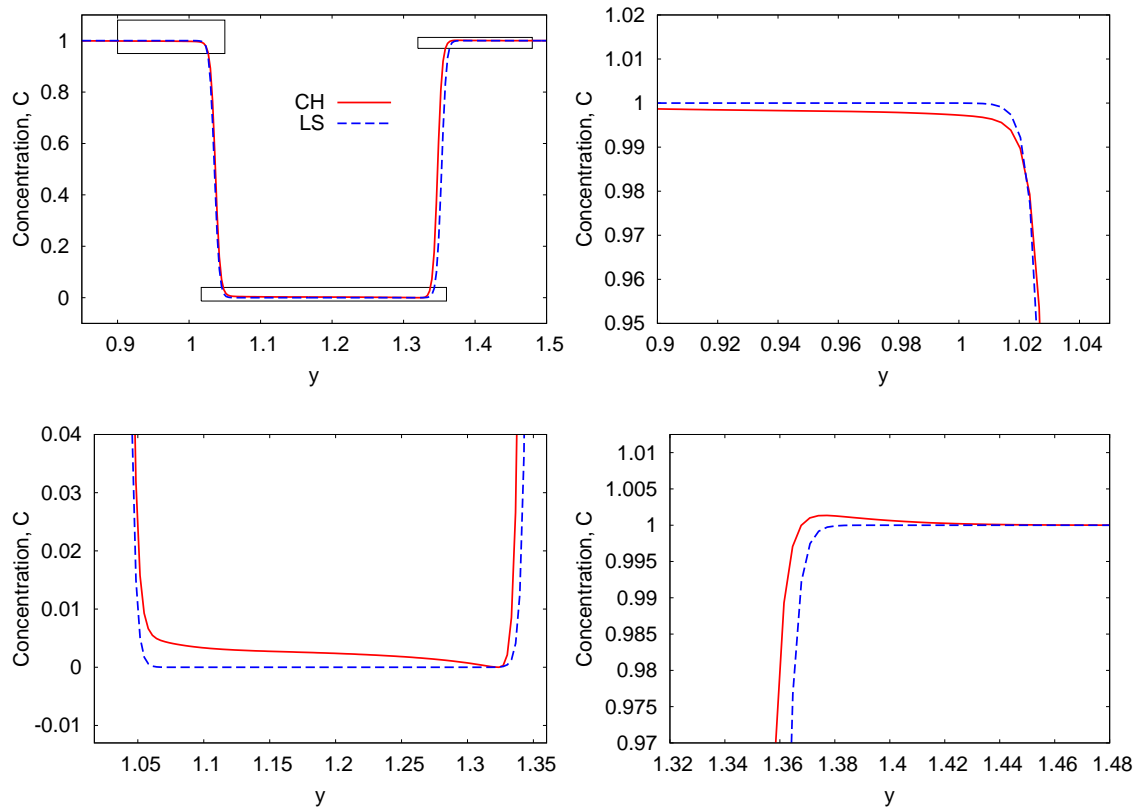


Figure 8.2: Concentration profile along the y axis for 2D TC2 using CH and Ls as interface capturing methods.

field C is constantly checked and an additional diffusion term is added wherever a negative C is detected. The rise velocities are presented in figure 8.1. Artificially controlling the concentration field eventually degrades the accuracy in capturing the second velocity peak. This further has led to non-physical variations of C as shown in figure 8.2 for the concentration profile along the y axis, where C gets values below and above 1 in the liquid and goes above 0 in the gas region.

8.2.2. Parallel AMR implementations

Although using a lower order surface tension and interface capturing formulations rectifies the need to have very high resolutions in the present scheme, a great deal of computational time and memory is still wasted on regions far from the interface where a coarse lattice resolution may suffice the smooth variations in LS function as well as pressure and velocity. An adaptive mesh refinement (AMR) technique was not carried out in this thesis so as to refrain from the complications it brings to the parallel processing. Nevertheless, efficient GPGPU implementations for AMR algorithm could be considered as a future work. In particular, NVIDIA's Kepler generation GPGPUs provide a so-called dynamic parallelism framework which are aimed at facilitating adaptive computations through launching multiple levels of nested kernels, each resolving one lattice level in the refinement hierarchy. Conventional multi-block LBE frameworks using variable time steps for each level as in [48], or those based on a single time step finite difference formulations [37, 38] could then be considered as candidates for such parallel implementations.

8.2.3. Multiple GPGPU implementations

GPGPU implementations in this thesis were designed to run on a single GPGPU, where the problem resolution was bounded by the limited on-board global memory of a single device, especially for 3D problems. Although the convergence trend in 3D benchmark flows proved that the results are approaching those of DNS for Navier-Stokes equations, multiple GPGPU implementations will indeed facilitate reaching the same accuracy level via high resolution simulations. In case a more memory efficient AMR programming is also pursued, such massively parallel implementations also allow to treat large real scale problems of interest for industrial applications within reasonable times and costs.

A

Appendix

1.1. Weighted essentially non-oscillatory (WENO) scheme

Before outlining the full fifth order WENO scheme, we shall first cover the its predecessor ENO scheme as described in [91]. The scheme is based on obtaining $\nabla\phi$ by differentiating Newton's polynomial interpolation, which gives the following form of polynomial for ϕ in the x direction

$$\phi(x) = Q_0(x) + Q_1(x) + Q_2(x) + Q_3(x). \quad (\text{A.1})$$

Based on Newton's interpolation method, the zeroth divided difference is

$$D_i^0\phi = \phi \quad (\text{A.2})$$

while the first divided difference could be the forward differencing

$$D_{i+\frac{1}{2}}^1\phi = \frac{D_{i+1}^0\phi - D_i^0\phi}{\Delta x} \quad (\text{A.3})$$

which is also denoted by $(D^+\phi)_i$, and accordingly $D_{i-\frac{1}{2}}^1\phi = (D^-\phi)_i$ for the backward differencing. The second and third divided differences are defined as

$$D_i^2\phi = \frac{D_{i+\frac{1}{2}}^1\phi - D_{i-\frac{1}{2}}^1\phi}{\Delta x} \quad (\text{A.4})$$

$$D_{i+\frac{1}{2}}^3\phi = \frac{D_{i+1}^2\phi - D_i^2\phi}{\Delta x}, \quad (\text{A.5})$$

respectively. Equation (A.1) could now be differentiated to derive $\phi_x(x)$

$$\phi_x(x_i) = Q_1'(x) + Q_2'(x) + Q_3'(x). \quad (\text{A.6})$$

Setting $k = i - 1$ gives $(\phi_x^-)_i$, and conversely $k = i$ gives $(\phi_x^+)_i$. Then, following the Newton polynomial convention we define

$$Q_1(x) = (D_{k+\frac{1}{2}}^1\phi)(x - x_i) \quad (\text{A.7})$$

such that

$$Q_1'(x_i) = D_{k+\frac{1}{2}}^1\phi. \quad (\text{A.8})$$

Up to this point, it is clear that the first order polynomial interpolation is equivalent to first order upwinding. Addition of $Q_2'(x)$ and $Q_3'(x)$ improves the approximation to second and third order, respectively. In order to construct

$$Q_2(x) = c(x - x_k)(x - x_{k+1}), \quad (\text{A.9})$$

two choices could be imagined: if $|D_k^2\phi| \leq |D_{k+1}^2\phi|$, then $c = D_k^2\phi$ and we set $k^* = k - 1$, otherwise $c = D_{k+1}^2\phi$ and $k^* = k$. In either case, differentiating equation (A.9) gives

$$Q_2'(x_i) = c(2(i - k) - 1)\Delta x. \quad (\text{A.10})$$

A similar procedure is followed for the third order term which is of the form

$$Q_3(x) = c^*(x - x_{k^*})(x - x_{k^*+1})(x - x_{k^*+2}) \quad (\text{A.11})$$

such that

$$Q_3'(x) = c^*(3(i - k^*) - 6(i - k^*) + 2)(\Delta x)^2. \quad (\text{A.12})$$

In order to choose c^* we compare $|D_{k^*+\frac{1}{2}}^3\phi|$ and $|D_{k^*+\frac{3}{2}}^3\phi|$. Again, if $|D_{k^*+\frac{1}{2}}^3\phi| \leq |D_{k^*+\frac{3}{2}}^3\phi|$ we set $c^* = D_{k^*+\frac{1}{2}}^3\phi$, otherwise $c^* = D_{k^*+\frac{3}{2}}^3\phi$.

Looking at the above third order approximation, one may find out that there are three different ENO approximations for example for $(\phi_x^-)_i$. By defining $v_1 = D^-\phi_{i-2}$, $v_2 = D^-\phi_{i-1}$, $v_3 = D^-\phi_i$, $v_4 = D^-\phi_{i+1}$ and $v_5 = D^-\phi_{i+2}$ we could have

$$\phi_x^1 = \frac{v_1}{3} - \frac{7v_2}{6} + \frac{11v_3}{6} \quad (\text{A.13})$$

$$\phi_x^2 = -\frac{v_2}{6} + \frac{5v_3}{6} + \frac{v_4}{3} \quad (\text{A.14})$$

and

$$\phi_x^3 = \frac{v_2}{3} + \frac{5v_4}{6} - \frac{v_5}{6} \quad (\text{A.15})$$

as three possible approximations to $(\phi_x^-)_i$. The idea behind the fifth order weighted ENO (WENO) is to use a convex combination of all these possible approximations. Proceeding along the lines of [91], in order to find the weights w_j , $j = 1, 2, 3$, the smoothness parameters are defined as

$$S_1 = \frac{13}{12}(v_1 2v_2 + v_3)^2 + \frac{1}{4}(v_1 4v_2 + 3v_3)^2 \quad (\text{A.16})$$

$$S_2 = \frac{13}{12}(v_2 2v_3 + v_4)^2 + \frac{1}{4}(v_2 - v_4)^2 \quad (\text{A.17})$$

$$S_3 = \frac{13}{12}(v_3 2v_4 + v_5)^2 + \frac{1}{4}(v_3 4v_4 + v_5)^2. \quad (\text{A.18})$$

Using these smoothness parameters we then define

$$\alpha_1 = \frac{0.1}{(S_1 + \varepsilon)^2} \quad (\text{A.19})$$

$$\alpha_2 = \frac{0.6}{(S_2 + \varepsilon)^2} \quad (\text{A.20})$$

and

$$\alpha_3 = \frac{0.3}{(S_3 + \varepsilon)^2} \quad (\text{A.21})$$

where the small value ε is obtained from

$$\varepsilon = 10^{-6} \max\{v_1^2 + v_2^+ v_3^2 + v_3^2 + v_4^2 + v_5^2\} + 10^{-99}. \quad (\text{A.22})$$

Eventually, the the weights are derived as

$$w_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3} \quad (\text{A.23})$$

$$w_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3} \quad (\text{A.24})$$

$$w_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3}. \quad (\text{A.25})$$

Although the above procedure seems to be computationally costly, it was shown in [62] that setting the weights to their optimal constant values of $w_1 = 0.1$, $w_2 = 0.6$ and $w_3 = 0.3$ still gives a fifth order accuracy in smooth regions of the problem, yet at a much lower cost.

B

Appendix

2.1. Transformation matrices for MRT collision

Based on the analysis in chapter 5 and performing the Gram-Schmidt procedure, the transformation matrices for $D3Q15$, $D3Q19$ and $D3Q27$ read as in equations (B.1), (B.2) and (B.3), respectively.

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 16 & -4 & -4 & -4 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & -4 & 4 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & -4 & 4 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -4 & 4 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 0 & 2 & 2 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (\text{B.1})$$

Bibliography

- [1] Cuda c programming guide, version 7.5, Sept. 2015. URL http://docs.nvidia.com/cuda/pdf/CUDA_C_Best_Practices_Guide.pdf.
- [2] Cuda c best practices guide, version 7.5, Sept. 2015. URL https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf.
- [3] Featflow software, 2015. URL <http://www.featflow.de/en>.
- [4] Exa powerflow software, 2015. URL <https://www.exa.com/powerflow.html>.
- [5] D. Adalsteinsson and J.A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148(1):2–22, 1999.
- [6] J Adelsberger, P. Esser, M. Griebel, S. Groß, M Klitz, and A. Rüttgers. 3d incompressible two-phase flow benchmark computations for rising droplets. pages 5274–5285, 2014.
- [7] B. Ahrenholz, J. Tölke, P. Lehmann, A. Peters, A. Kaestner, M. Krafczyk, and W. Durner. Prediction of capillary hysteresis in a porous material using lattice-boltzmann methods and comparison to experimental data and a morphological pore network model. *Advances in Water Resources*, 31(9):1151–1173, 2008. DOI: 10.1016/j.advwatres.2008.03.009.
- [8] F.J. Alexander, S. Chen, and D.W. Grunau. Hydrodynamic spinodal decomposition: Growth kinetics and scaling functions. *Physical Review B*, 48(1):634–637, 1993. DOI: 10.1103/PhysRevB.48.634.
- [9] L. Amaya-Bower and T. Lee. Single bubble rising dynamics for moderate reynolds number using lattice boltzmann method. *Computers and Fluids*, 39(7):1191–1207, 2010.
- [10] D. Anderl, S. Bogner, C. Rauh, U. Rade, and A. Delgado. Free surface lattice boltzmann with enhanced bubble model. *Computers and Mathematics with Applications*, 67(2):331–339, 2014. DOI: 10.1016/j.camwa.2013.06.007.
- [11] D.M. Anderson, G.B. McFadden, and A.A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Review of Fluid Mechanics*, 30:139–165, 1998.
- [12] S. Ansumali and I.V. Karlin. Entropy function approach to the lattice boltzmann method. *Journal of Statistical Physics*, 107(1-2):291–308, 2002. DOI: 10.1023/A:1014575024265.
- [13] S. Ansumali and I.V. Karlin. Single relaxation time model for entropic lattice boltzmann methods. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 65(5):056312/1–056312/9, 2002. DOI: 10.1103/PhysRevE.65.056312.

-
- [14] A.M. Artoli, A.G. Hoekstra, and P.M.A. Slood. Simulation of a systolic cycle in a realistic artery with the lattice boltzmann bgk method. *International Journal of Modern Physics B*, 17(1-2):95–98, 2003.
- [15] N. Ashgriz and J. Y. Poo. Coalescence and separation in binary collisions of liquid drops. *Journal of Fluid Mechanics*, 221:183–204, 12 1990. DOI: 10.1017/S0022112090003536.
- [16] L. Axner, A.G. Hoekstra, A. Jeays, P. Lawford, R. Hose, and P.M. Slood. Simulations of time harmonic blood flow in the mesenteric artery: comparing finite element and lattice boltzmann methods. *Biomedical engineering online*, 8:23, 2009.
- [17] A. Banari, C. Janßen, S.T. Grilli, and M. Krafczyk. Efficient gpgpu implementation of a lattice boltzmann model for multiphase flows with high density ratios. *Computers and Fluids*, 93:1–17, 2014.
- [18] A. Banari, C.F. Janßen, and S.T. Grilli. An efficient lattice boltzmann multiphase model for 3d flows with large density ratios at high reynolds numbers. *Computers and Mathematics with Applications*, 68(12):1819–1843, 2014. DOI: 10.1016/j.camwa.2014.10.009.
- [19] J. Bao and L. Schaefer. Lattice boltzmann equation model for multi-component multi-phase flow with high density ratios. *Applied Mathematical Modelling*, 37(4):1860–1871, 2013.
- [20] J. Becker, M. Junk, D. Kehrwald, G. Thömmes, and Z. Yang. A combined lattice bgk/level set method for immiscible two-phase flows. *Computers and Mathematics with Applications*, 58(5):950–964, 2009.
- [21] D. Bhaga and M.E. Weber. Bubbles in viscous liquids: Shapes, wakes and velocities. *Journal of Fluid Mechanics*, 105:61–85, 1981.
- [22] D. Bhaga and M.E. Weber. Bubbles in viscous liquids: Shapes, wakes and velocities. *Journal of Fluid Mechanics*, 105:61–85, 1981.
- [23] P.L. Bhatnagar, E.P. Gross, and M. Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Physical Review*, 94(3):511–525, 1954. DOI: 10.1103/PhysRev.94.511.
- [24] F. Boyer. A theoretical and numerical model for the study of incompressible mixture flows. *Computers and Fluids*, 31(1):41–68, 2002. DOI: 10.1016/S0045-7930(00)00031-1.
- [25] J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, 1992. DOI: 10.1016/0021-9991(92)90240-Y.
- [26] E. Burman and N. Parolini. Subgrid edge stabilization for transport equations. Technical report, 2005. EPFL-IACS report 09.2005.
- [27] E. Burman and N. Parolini. A new reinitialization procedure for the finite element approximation of the level set equation. Technical report, 2005. EPFL-IACS report 13.2005.
- [28] E. Burman and N. Parolini. Refined level set grid method for tracking interfaces. Technical report, 2005. Center for Turbulence research, Annual Research Briefs.
- [29] A. Capozzoli, C. Curcio, A. Liseno, and S. Savarese. A comparison of fast marching, fast sweeping and fast iterative methods for the solution of the eikonal equation. pages 685–688, 2013. DOI: 10.1109/TELFOR.2013.6716321.

- [30] S. Chen and G.D. Doolen. Lattice boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30:329–364, 1998.
- [31] M. Cheng, J. Hua, and J. Lou. Simulation of bubble-bubble interaction using a lattice boltzmann method. *Computers and Fluids*, 39(2):260–270, 2010. DOI: 10.1016/j.compfluid.2009.09.003.
- [32] Grace J. R. Clift, R. and M. E. Weber. *Bubbles, Drops and Particles*. Academic Press, New York, First edition, 1978.
- [33] P.J. Dellar. Nonhydrodynamic modes and a priori construction of shallow water lattice boltzmann equations. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 65(3):036309/1–036309/12, 2002. DOI: 10.1103/PhysRevE.65.036309.
- [34] D. D’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo. Multiple-relaxation-time lattice boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 360(1792):437–451, 2002. DOI: 10.1098/rsta.2001.0955.
- [35] B. Engquist, A.-K. Tornberg, and R. Tsai. Discretization of dirac delta functions in level set methods. *Journal of Computational Physics*, 207(1):28–51, 2005.
- [36] A. Fakhari and T. Lee. Multiple-relaxation-time lattice boltzmann method for immiscible fluids at high reynolds numbers. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 87(2), 2013.
- [37] A. Fakhari and T. Lee. Finite-difference lattice boltzmann method with a block-structured adaptive-mesh-refinement technique. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 89(3), 2014. DOI: 10.1103/PhysRevE.89.033310.
- [38] A. Fakhari and T. Lee. Numerics of the lattice boltzmann method on nonuniform grids: Standard lbm and finite-difference lbm. *Computers and Fluids*, 107:205–213, 2015. DOI: 10.1016/j.compfluid.2014.11.013.
- [39] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical Review Letters*, 56(14):1505–1508, 1986.
- [40] S. Ganesan. *Finite Element Methods on Moving Meshes for Free Surface and Interface Flows*. PhD thesis, Otto-von-Guericke-Universität, Fakultät für Mathematik, Magdeburg, 2006. published as book by docupoint Verlag Magdeburg.
- [41] M. Geier, A. Fakhari, and T. Lee. Conservative phase-field lattice boltzmann model for interface tracking equation. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 91(6), 2015. DOI: 10.1103/PhysRevE.91.063309.
- [42] M. Geveler, D. Ribbrock, D. Göddeke, and S. Turek. Lattice-boltzmann simulation of the shallow-water equations with fluid-structure interaction on multi- and manycore processors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6310 LNCS:92–104, 2010.
- [43] V. Girault and P. A. Raviart. *Finite Element Methods for Navier-Stokes equations*. Springer-Verlag, 1986.

-
- [44] H. Gomez, V.M. Calo, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of the cahn-hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4333–4352, 2008. DOI: 10.1016/j.cma.2008.05.003.
- [45] A.K. Gunstensen, D.H. Rothman, S. Zaleski, and G. Zanetti. Lattice boltzmann model of immiscible fluids. *Physical Review A*, 43(8):4320–4327, 1991. DOI: 10.1103/PhysRevA.43.4320.
- [46] Z. Guo and C. Shu. *Lattice Boltzmann Method and its Applications in Engineering*. World Scientific, Singapore, 2013.
- [47] Z. Guo and T.S. Zhao. Lattice boltzmann model for incompressible flows through porous media. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 66(3):036304/1–036304/9, 2002. DOI: 10.1103/PhysRevE.66.036304.
- [48] Z. Guo, C. Zheng, and T.S. Zhao. A lattice bgk scheme with general propagation. *Journal of Scientific Computing*, 16(4):569–585, 2001.
- [49] Z. Guo, C. Zheng, and B. Shi. Discrete lattice effects on the forcing term in the lattice boltzmann method. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 65(4):046308/1–046308/6, 2002. DOI: 10.1103/PhysRevE.65.046308.
- [50] M.E. Gurtin, D. Polignone, and J. Vinals. Two-phase binary fluids and immiscible fluids described by an order parameter. *Mathematical Models and Methods in Applied Sciences*, 6(6):815–831, 1996.
- [51] R. A. Hartunian and W. R. Sears. On the instability of small gas bubbles moving uniformly in various liquids. *Journal of Fluid Mechanics*, 3:27–47, 10 1957. ISSN 1469-7645. DOI: 10.1017/S0022112057000464.
- [52] X. He, X. Shan, and G.D. Doolen. Discrete boltzmann equation model for nonideal gases. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 57(1):R13–R16, 1998.
- [53] X. He, S. Chen, and R. Zhang. A lattice boltzmann scheme for incompressible multiphase flow and its application in simulation of rayleigh-taylor instability. *Journal of Computational Physics*, 152(2):642–663, 1999.
- [54] H. Huang, M. Sukop, and X. Lu. *Multiphase Lattice Boltzmann Methods: Theory and Application*. Wiley-Blackwell, July 2015.
- [55] J.S.-R. Hysing and S. Turek. Evaluation of commercial and academic cfd codes for a two-phase flow benchmark test case. *International Journal of Computational Science and Engineering*, 10(4):387–394, 2015. DOI: 10.1504/IJCSE.2015.070993.
- [56] S. Hysing. *Numerical simulation of immiscible fluids with FEM level set techniques*. PhD thesis, Technische Universität Dortmund, December 2007.
- [57] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska. Proposal for quantitative benchmark computations of bubble dynamics. Technical report, Fakultät für Mathematik, TU Dortmund, August 2007. Ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 351.

- [58] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska. Quantitative benchmark computations of two-dimensional bubble dynamics. *International Journal for Numerical Methods in Fluids*, 60(11):1259–1288, 2009.
- [59] T. Inamuro, T. Ogata, S. Tajima, and N. Konishi. A lattice boltzmann method for incompressible two-phase flows with large density differences. *Journal of Computational Physics*, 198(2):628–644, 2004. DOI: 10.1016/j.jcp.2004.01.019.
- [60] C. Janssen and M. Krafczyk. A lattice boltzmann approach for free-surface-flow simulations on non-uniform block-structured grids. *Computers and Mathematics with Applications*, 59(7):2215–2235, 2010. DOI: 10.1016/j.camwa.2009.08.064.
- [61] C.F. Janßen, S.T. Grilli, and M. Krafczyk. On enhanced non-linear free surface flow simulations with a hybrid lbm-vof model. *Computers and Mathematics with Applications*, 65(2):211–229, 2013.
- [62] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted eno schemes. *Journal of Computational Physics*, 126(1):202–228, 1996. DOI: 10.1006/jcph.1996.0130.
- [63] V. John and G. Matthies. Moonmd - a program package based on mapped finite element methods. *Computing and Visualization in Science*, 6(2-3):163–170, 2004. DOI: 10.1007/s00791-003-0120-1.
- [64] I.V. Karlin, A. Ferrante, and H.C. Öttinger. Perfect entropy functions of the lattice boltzmann method. *Europhysics Letters*, 47(2):182–188, 1999. DOI: 10.1209/epl/i1999-00370-1.
- [65] J. Kim. A diffuse-interface model for axisymmetric immiscible two-phase flow. *Applied Mathematics and Computation*, 160(2):589–606, 2005. DOI: 10.1016/j.amc.2003.11.020.
- [66] L.S. Kim, H.K. Jeong, M.Y. Ha, and K.C. Kim. Numerical simulation of droplet formation in a micro-channel using the lattice boltzmann method. *Journal of Mechanical Science and Technology*, 22(4):770–779, 2008. DOI: 10.1007/s12206-007-1201-8.
- [67] A. Koelman. A simple lattice boltzmann scheme for navier-stokes fluid flow. *EPL (Europhysics Letters)*, 15(6):603, 1991.
- [68] C. Körner, M. Thies, T. Hofmann, N. Thörey, and U. Rädde. Lattice boltzmann model for free surface flow for modeling foaming. *Journal of Statistical Physics*, 121(1-2):179–196, 2005. DOI: 10.1007/s10955-005-8879-8.
- [69] F. Kuznik, C. Obrecht, G. Rusaouen, and J.-J. Roux. Lbm based flow simulation using gpu computing processor. *Computers and Mathematics with Applications*, 59(7):2380–2392, 2010.
- [70] F. Kuznik, L.-S. Luo, and M. Krafczyk. Mesoscopic methods in engineering and science. *Computers and Mathematics with Applications*, 65(6):813–814, 2013.
- [71] P. Lallemand and L.-S. Luo. Theory of the lattice boltzmann method: Dispersion, dissipation, isotropy, galilean invariance, and stability. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 61(6 B):6546–6562, 2000.
- [72] P. Lallemand and L.-S. Luo. Theory of the lattice boltzmann method: Acoustic and thermal properties in two and three dimensions. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 68(3 2):367061–3670625, 2003.

-
- [73] P. Lallemand, D. D'Humières, L.-S. Luo, and R. Rubinstein. Theory of the lattice boltzmann method: Three-dimensional model for linear viscoelastic fluids. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 67(2 1):212031–2120319, 2003.
- [74] T. Lee and P.F. Fischer. Eliminating parasitic currents in the lattice boltzmann equation method for nonideal gases. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 74(4), 2006.
- [75] T. Lee and C.-L. Lin. A stable discretization of the lattice boltzmann equation for simulation of incompressible two-phase flows at high density ratio. *Journal of Computational Physics*, 206(1):16–47, 2005.
- [76] T. Lee and L. Liu. Lattice boltzmann simulations of micron-scale drop impact on dry surfaces. *Journal of Computational Physics*, 229(20):8045–8063, 2010.
- [77] Q. Li, K. H. Luo, and X. J. Li. Forcing scheme in pseudopotential lattice boltzmann model for multiphase flows. *Phys. Rev. E*, 86:016709, Jul 2012. DOI: 10.1103/PhysRevE.86.016709.
- [78] Q. Li, C. Zhong, K. Li, G. Zhang, X. Lu, Q. Zhang, K. Zhao, and X. Chu. A parallel lattice boltzmann method for large eddy simulation on multiple gpus. *Computing*, 96(6):479–501, 2014. DOI: 10.1007/s00607-013-0356-7.
- [79] H. Liang, B.C. Shi, Z.L. Guo, and Z.H. Chai. Phase-field-based multiple-relaxation-time lattice boltzmann model for incompressible multiphase flows. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 89(5), 2014. DOI: 10.1103/PhysRevE.89.053320.
- [80] J. Linxweiler, M. Krafczyk, and J. Tölke. Highly interactive computational steering for coupled 3d flow problems utilizing multiple gpus : Towards intuitive desktop environments for interactive 3d fluid structure interaction. *Computing and Visualization in Science*, 13(7):299–314, 2010. DOI: 10.1007/s00791-010-0151-3.
- [81] X. D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212, 1994. DOI: 10.1006/jcph.1994.1187.
- [82] J. Lowengrub and L. Truskinovsky. Quasi-incompressible cahn-hilliard fluids and topological transitions. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 454(1978):2617–2654, 1998.
- [83] L.-S. Luo. Unified theory of lattice boltzmann models for nonideal gases. *Physical Review Letters*, 81(8):1618–1621, 1998.
- [84] G.R. McNamara and G. Zanetti. Use of the boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, 61(20):2332–2335, 1988.
- [85] M. Mehravaran and S. Kazemzadeh Hannani. Simulation of buoyant bubble motion in viscous flows employing lattice boltzmann and level set methods. *Scientia Iranica*, 18(2): 231 – 240, 2011. ISSN 1026-3098. DOI: <http://dx.doi.org/10.1016/j.scient.2011.03.018>.
- [86] M. Mehravaran and S.K. Hannani. Simulation of incompressible two-phase flows with large density differences employing lattice boltzmann and level set methods. *Computer Methods in Applied Mechanics and Engineering*, 198(2):223–233, 2008.

- [87] S. Mukherjee and J. Abraham. A pressure-evolution-based multi-relaxation-time high-density-ratio two-phase lattice-boltzmann model. *Computers and Fluids*, 36(6):1149–1158, 2007.
- [88] B. D. Nichols. Methods for calculating multidimensional, transient free surface flows past bodies. Technical report, 1975.
- [89] C. Obrecht, F. Kuznik, B. Tourancheau, and J.-J. Roux. A new approach to the lattice boltzmann method for graphics processing units. *Computers and Mathematics with Applications*, 61(12):3628–3638, 2011.
- [90] E. Olsson and G. Kreiss. A conservative level set method for two phase flow. *Journal of Computational Physics*, 210(1):225–246, 2005.
- [91] Fedkiw R. Osher, S. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, New York, First edition, 2003.
- [92] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988. DOI: 10.1016/0021-9991(88)90002-2.
- [93] N. Parolini. *Computational Fluid Dynamics for Naval Engineering Problems*. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), 2004.
- [94] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A pde-based fast local level set method. *Journal of Computational Physics*, 155(2):410–438, 1999. DOI: 10.1006/jcph.1999.6345.
- [95] C.S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977. DOI: 10.1016/0021-9991(77)90100-0.
- [96] K.N. Premnath and J. Abraham. Three-dimensional multi-relaxation time (mrt) lattice-boltzmann models for multiphase flow. *Journal of Computational Physics*, 224(2):539–559, 2007.
- [97] W. H. Press, S. A. Teukolsky, and B. P. Vetterling, W. T. Flannery. *Numerical Recipes, The Art of Scientific Computing*. Cambridge University Press, New York, 2007.
- [98] A. Prosperetti and G. Tryggvason. *Computational Methods for Multiphase Flow*. Cambridge University Press, New York, First edition, 2007.
- [99] Y. H. Qian, D. D’Humières, and Lallemand P. Lattice bgk models for navier-stokes equation. *EPL (Europhysics Letters)*, 17(6):479, 1992.
- [100] Rolf Rannacher. *Incompressible Viscous Flows*. John Wiley & Sons, Ltd. ISBN 9780470091357. DOI: 10.1002/0470091355.ecm056.
- [101] Y. Renardy and M. Renardy. Prost: A parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of Computational Physics*, 183(2):400–421, 2002. DOI: 10.1006/jcph.2002.7190.
- [102] W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 141(2):112–152, 1998. DOI: 10.1006/jcph.1998.5906.
- [103] D.H. Rothman and J.M. Keller. Immiscible cellular-automaton fluids. *Journal of Statistical Physics*, 52(3-4):1119–1127, 1988. DOI: 10.1007/BF01019743.

-
- [104] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, Second edition, 2003.
- [105] A. Safi and M. Ashrafizaadeh. Entropic Lattice Boltzmann simulation of three dimensional binary gas mixture flow in packed beds using graphics processors. Technical report, Fakultät für Mathematik, TU Dortmund, September 2014. Ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 504.
- [106] M. Sbragaglia, R. Benzi, L. Biferale, S. Succi, K. Sugiyama, and F. Toschi. Generalized lattice boltzmann method with multirange pseudopotential. *Phys. Rev. E*, 75:026702, Feb 2007. DOI: 10.1103/PhysRevE.75.026702.
- [107] Robert Sedgewick. *Algorithms*. Addison-Wesley, 1983. ISBN 0-201-06672-6.
- [108] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4):1591–1595, 1996. DOI: 10.1073/pnas.93.4.1591.
- [109] J.A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [110] Xiaowen Shan. Analysis and reduction of the spurious current in a class of multiphase lattice boltzmann models. *Phys. Rev. E*, 73:047701, Apr 2006.
- [111] Xiaowen Shan. Pressure tensor calculation in a class of nonideal gas lattice boltzmann models. *Phys. Rev. E*, 77:066702.
- [112] Xiaowen Shan and Hudong Chen. Lattice boltzmann model for simulating flows with multiple phases and components. *Phys. Rev. E*, 47:1815–1819, 1993.
- [113] A. Smolianski. *Numerical Modeling of Two-Fluid Interfacial Flows*. PhD thesis, University of Jyväskylä, 2001.
- [114] V. Sofonea and R.F. Sekerka. Boundary conditions for the upwind finite difference lattice boltzmann model: Evidence of slip velocity in micro-channel flow. *Journal of Computational Physics*, 207(2):639–659, 2005. DOI: 10.1016/j.jcp.2005.02.003.
- [115] J.D. Sterling and S. Chen. Stability analysis of lattice boltzmann methods. *Journal of Computational Physics*, 123(1):196–206, 1996. DOI: 10.1006/jcph.1996.0016.
- [116] S. Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Oxford University Press, New York, 2001.
- [117] K. Suga, Y. Kuwata, K. Takashima, and R. Chikasue. A d3q27 multiple-relaxation-time lattice boltzmann method for turbulent flows. *Computers and Mathematics with Applications*, 69(6):518–529, 2015. DOI: 10.1016/j.camwa.2015.01.010.
- [118] M. Sussman. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, 1994.
- [119] M. Sussman and E. Fatemi. Efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing*, 20(4):1165–1191, 1999. DOI: 10.1137/S1064827596298245.
- [120] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An improved level set method for incompressible two-phase flows. *Computers and Fluids*, 27(5-6):663–680, 1998.

- [121] M.R. Swift, E. Orlandini, W.R. Osborn, and J.M. Yeomans. Lattice boltzmann simulations of liquid-gas and binary fluid systems. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 54(5):5041–5052, 1996.
- [122] N. Takada, M. Misawa, A. Tomiyama, and S. Fujiwara. Numerical simulation of two- and three-dimensional two-phase fluid motion by lattice boltzmann method. *Computer Physics Communications*, 129(1):233–246, 2000. DOI: 10.1016/S0010-4655(00)00110-7.
- [123] G. Thommes, J. Becker, M. Junk, A.K. Vaikuntam, D. Kehrwald, A. Klar, K. Steiner, and A. Wiegmann. Numerical investigation of a combined lattice boltzmann-level set method for three-dimensional multiphase flow. *International Journal of Computational Fluid Dynamics*, 23(10):687–697, 2009.
- [124] J. Tolke and M. Krafczyk. Teraflop computing on a desktop pc with gpus for 3d cfd. *Int. J. Comput. Fluid Dyn.*, 22(7):443–456, 2008. ISSN 1061-8562.
- [125] J. Töllke, M. Krafczyk, M. Schulz, and E. Rank. Lattice boltzmann simulations of binary fluid flow through porous media. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 360(1792):535–545, 2002. DOI: 10.1098/rsta.2001.0944.
- [126] A.-K. Tornberg. *Interface Tracking Methods with Application to Multiphase Flows*. PhD thesis, NADA, KTH, Stockholm, Sweden, 2000.
- [127] G. Tryggvason, M. Sussman, and M. Y. Hussaini. Immersed boundary methods for fluid interfaces. In Andrea Prosperetti and Grétar Tryggvason, editors, *Computational Methods for Multiphase Flow*, pages 37–77. Cambridge University Press, 2007. ISBN 9780511607486. DOI: 10.1017/CBO9780511607486.004.
- [128] S. Turek, O. Mierka, S. Hysing, and D. Kuzmin. Numerical study of a high order 3D FEM–level set approach for immiscible flow simulation. In S. Repin, T. Tiihonen, and T. Tuovinen, editors, *Numerical methods for differential equations, optimization, and technological problems*, Computational Methods in Applied Sciences, Vol. 27, pages 65–70. Springer, 2012.
- [129] Hakon Wadell. Volume, shape, and roundness of quartz particles. *The Journal of Geology*, 43(3):pp. 250–280, 1935.
- [130] X. Wang, Y. Shangguan, N. Onodera, H. Kobayashi, and T. Aoki. Direct numerical simulation and large eddy simulation on a turbulent wall-bounded flow using lattice boltzmann method and multiple gpus. *Mathematical Problems in Engineering*, 2014, 2014. DOI: 10.1155/2014/742432.
- [131] Y. Wang, C. Shu, H.B. Huang, and C.J. Teo. Multiphase lattice boltzmann flux solver for incompressible multiphase flows with large density ratio. *Journal of Computational Physics*, 280:404–423, 2015. DOI: 10.1016/j.jcp.2014.09.035.
- [132] Y. Wang, C. Shu, J.Y. Shao, J. Wu, and X.D. Niu. A mass-conserved diffuse interface method and its application for incompressible multiphase flows with large density ratio. *Journal of Computational Physics*, 290:336–351, 2015. DOI: 10.1016/j.jcp.2015.03.005.
- [133] D. Wolf-Gladrow. *Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An Introduction*. Springer, Berlin, 2005.

-
- [134] W.-A. Yong and L.-S. Luo. Nonexistence of h theorems for the athermal lattice boltzmann models with polynomial equilibria. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 67(5 1):051105/1–051105/4, 2003.
- [135] W.-A. Yong and L.-S. Luo. Nonexistence of h theorem for some lattice boltzmann models. *Journal of Statistical Physics*, 121(1-2):91–103, 2005. DOI: 10.1007/s10955-005-5958-9.
- [136] D. L. Youngs. An interface tracking method for a 3d eulerian hydrodynamics code. Technical report, 44/92/35, AWRE, 1984.
- [137] D. Yu, R. Mei, L.-S. Luo, and W. Shyy. Viscous flow computations with the method of lattice boltzmann equation. *Progress in Aerospace Sciences*, 39(5):329–367, 2003.
- [138] Y. Yu, L. Chen, J. Lu, and G. Hou. A coupled lattice boltzmann and particle level set method for free-surface flows. *ScienceAsia*, 40(3):238–247, 2014. DOI: 10.2306/scienceasia1513-1874.2014.40.238.
- [139] Zhao Yu and Liang-Shih Fan. Multirelaxation-time interaction-potential-based lattice boltzmann model for two-phase flow. *Phys. Rev. E*, 82:046708, 2010.
- [140] Zhao Yu and Liang-Shih Fan. Multirelaxation-time interaction-potential-based lattice boltzmann model for two-phase flow. *Phys. Rev. E*, 82:046708, Oct 2010. DOI: 10.1103/PhysRevE.82.046708.
- [141] P. Yuan and L. Schaefer. Equations of state in a lattice boltzmann model. *Physics of Fluids*, 18(4), 2006.
- [142] D. Zhang, K. Papadikis, and S. Gu. Three-dimensional multi-relaxation time lattice-boltzmann model for the drop impact on a dry surface at large density ratio. *International Journal of Multiphase Flow*, 64:11–18, 2014. DOI: 10.1016/j.ijmultiphaseflow.2014.04.005.
- [143] H.W. Zheng, C. Shu, and Y.T. Chew. A lattice boltzmann model for multiphase flows with large density ratio. *Journal of Computational Physics*, 218(1):353–371, 2006. DOI: 10.1016/j.jcp.2006.02.015.
- [144] J.G. Zhou. A lattice boltzmann model for the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 191(32):3527–3539, 2002. DOI: 10.1016/S0045-7825(02)00291-8.