CHAPTER 1
Introduction

Multiphase fluid phenomena and flows occur when two or more fluids that do not readily mix (such as air and water) share an interface. Multiphase fluid interactions are nearly ubiquitous in natural and industrial processes. Multiphase phenomena and flows can involve single component multiphase fluids, e.g., water and its own vapor, and multi-component multiphase fluids, e.g., oil/water. Some practical examples of multiphase fluid problems are the recovery and enhanced recovery of petroleum resources from reservoirs, non-aqueous phase liquid contamination of groundwater, soil water behavior, surface wetting phenomena, fuel cell operation, and the movement and evolution of clouds.

Computational fluid dynamics (CFD) has become very important in fluid flow studies. The Lattice Boltzmann method (LBM) has developed very quickly in the last two decades and has become a novel and powerful CFD tool – particularly for multiphase flows. The LBM has some major advantages compared to traditional CFD methods. First, it originates from Boltzmann's kinetic molecular dynamics – a more foundational level than normal continuum approaches. The LBM is able to recover the traditional macroscopic scale continuity and Navier–Stokes (N–S) equations, which are discretized and solved numerically in the common CFD methods. In the LBM, the more fundamental Boltzmann equation is directly discretized. Alternatively, the LBM can be viewed from its discrete-particle, more molecular-dynamics-like lattice gas origins. Second, in the LBM the pressure is usually related to the density through an ideal gas equation of state (for single-phase flow) or through a non-ideal van der Waals-like equation of state for some types of complex multiphase fluids. The pressure fields can be obtained directly once the density field is known. Hence, the Poisson equation – which can be computationally expensive – does not have to be solved in the LBM. The third advantage of the LBM is that the method is easy to parallelize due to the locality of much of the computation. Finally, no-slip boundary condition can be easily handled by simple bounce-back scheme.

The LBM has had great success in studies of single-phase flows, with commercial software known as POWERFLOW (Exa Corporation, https://www.exa.com/), based on the LBM, appearing about ten years ago. In contrast, multiphase LBMs

Multiphase Lattice Boltzmann Methods: Theory and Application, First Edition.
Haibo Huang, Michael C. Sukop and Xi-Yun Lu.

Companion Website: www.wiley.com/go/huang/boltzmann

are still undergoing development and there are many multiphase Lattice Boltzmann models available.

1.1 History of the Lattice Boltzmann method

LBMs trace their roots to cellular automata, which were originally conceived by Stanislaw Ulam and John von Neumann in the 1940s. Cellular automata consist of a discretization of space on which individual cells exist in a particular state (say 0 or 1), and update their state at each time step according to a rule that takes as input the states of some set of the cell's neighbors. Sukop and Thorne (2006) provide an introduction to cellular automata. Wolfram (1983, 2002) studied simple cellular automata systematically and inspired some of the earliest application to fluids, leading to the first paper to propose a lattice gas cellular automaton (LGCA) for the N–S equations (Frisch et al. 1986). The use of a triangular grid restored some of the symmetry required to properly simulate fluids. Rothman and Zaleski (1997), Wolf-Gladrow (2000), Succi (2001), and Sukop and Thorne (2006) all provide instructive information on this model and the extensions that appeared. All of the LGCA models suffer from inherent defects, however, in particular the lack of Galilean invariance for fast flows and statistical noise (Qian et al. 1992, Wolf-Gladrow 2000). These are explicit particle-based Boolean models that include the random fluctuations that one would expect at a molecular level of gas simulation and hence required extensive averaging to recover the smooth behavior expected at macroscopic scales.

A second major step towards the modern LBM was taken by McNamara and Zanetti (1988), who dispensed with the individual particles of the LGCAs and replaced them with an averaged but still directionally discrete distribution function. This completely eliminated the statistical noise of the LGCA. A major simplification was introduced by Qian et al. (1992): the collision matrix of Higuera et al. (1989) is replaced by a single relaxation time, leading to the Bhatnagar, Gross, and Krook (BGK) model. After that, the LBM developed very quickly. Sukop and Thorne (2006) showed that there were fewer than 20 papers on the topic in 1992; more than 600 were published in 2013.

Later Lallemand and Luo (2000) and Luo (1998) showed that the LBM can be derived from the continuous Boltzmann equation (Boltzmann 1964/1995). Hence, it can be considered as a special discretized form of the Boltzmann equation (Nourgaliev et al. 2003). From the Chapman–Enskog expansion (Wolf-Gladrow 2000), the governing continuity and N–S equations can be recovered from the LBM. Without solving Poisson's equation, the pressure field can be obtained directly from the density distributions.

Today, the use of LBM spans a broad variety of disciplines. For example, an overview of the LBM for material science and engineering can be found in

Raabe (2004). Application of the LBM to biophysics can be found in Boyd et al. (2005) and Sun et al. (2003).

1.2 The Lattice Boltzmann method

The LBM can be derived from the BGK approximation of the Boltzmann equation (He and Luo 1997),

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla f + \mathbf{F} \cdot \nabla_{\boldsymbol{\xi}} f = -\frac{f - f^{eq}}{\tau}, \tag{1.1}$$

where $f(\mathbf{x}, \boldsymbol{\xi}, t)$ is the single-particle distribution function in the phase space $(\mathbf{x}, \boldsymbol{\xi})$, and $f^{eq}(\mathbf{x}, \boldsymbol{\xi})$ is the Maxwell–Boltzmann distribution function. $\mathbf{x}$ is the position vector, $\boldsymbol{\xi}$ is the microscopic velocity, $\mathbf{F}(\mathbf{x},t)$ is a body force, and τ is the relaxation time, which determines the kinematic viscosity.

In the lattice BGK method, a discrete distribution function f_i is introduced to represent the fluid. This distribution function satisfies the following Lattice Boltzmann equation (He and Luo 1997):

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau}(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)) + S_i(\mathbf{x}, t), \tag{1.2}$$

where $f_i(\mathbf{x}, t)$ is the density distribution function related to the discrete velocity direction i and τ is a relaxation time, which is related to the kinematic viscosity by $\nu = c_s^2(\tau - 0.5)\Delta t$, where c_s is the sound speed. $S_i(\mathbf{x}, t)$ is the source term added into the standard Lattice Boltzmann equation. The equilibrium distribution function $f_i^{eq}(\mathbf{x}, t)$ can be calculated as (Luo 1998)

$$f_i^{eq}(\mathbf{x}, t) = w_i \rho \left[1 + \frac{\mathbf{e}_i \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{(\mathbf{u})^2}{2c_s^2}\right]. \tag{1.3}$$

In Eqs (1.2) and (1.3) the $\mathbf{e}_i$ are the discrete velocities, as defined below, and w_is are weights, as given in Table 1.1. ρ is the macroscopic density and $\mathbf{u}$ is the macroscopic velocity vector. Discrete velocity models are usually specified as DnQm, where n is the space dimension and m is the number of velocities.

Table 1.1 Overview of the weighting coefficients and sound speeds.

Model	w_i	c_s^2
D2Q7	$\frac{1}{2}$ $(i = 0)$, $\frac{1}{12}$ $(i = 1, \ldots, 6)$,	$c^2/4$
D2Q9	$\frac{4}{9}$ $(i = 0)$, $\frac{1}{9}$ $(i = 1, 2, 3, 4)$, $\frac{1}{36}$ $(i = 5, 6, 7, 8)$	$c^2/3$
D3Q15	$\frac{2}{9}$ $(i = 0)$, $\frac{1}{9}$ $(i = 1, \ldots, 6)$, $\frac{1}{72}$ $(i = 7, \ldots, 14)$	$c^2/3$
D3Q19	$\frac{1}{3}$ $(i = 0)$, $\frac{1}{18}$ $(i = 1, \ldots, 6)$, $\frac{1}{36}$ $(i = 7, \ldots, 18)$	$c^2/3$

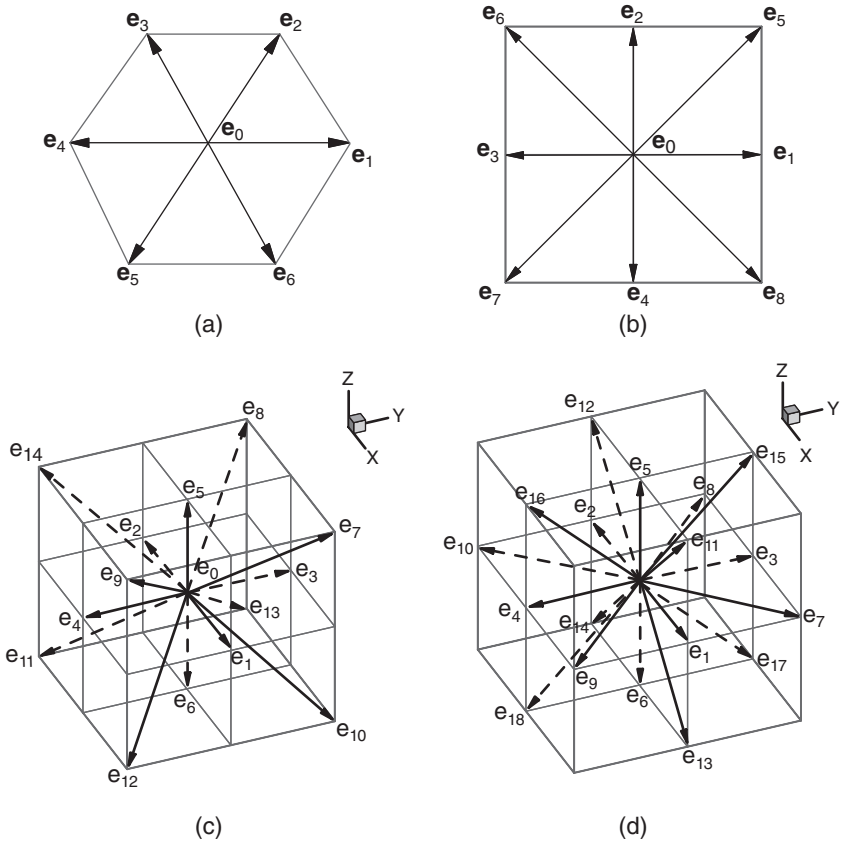

Figure 1.1 Discrete velocity models (a) D2Q7, (b) D2Q9, (c) D3Q15, and (d) D3Q19.

The popular 2D and 3D discrete velocity models are D2Q7, D2Q9, D3Q15, and D3Q19, which are shown in Figure 1.1.

For the D2Q7 model (Frisch et al. 1986), the discrete velocities are

$$[\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_5, \mathbf{e}_6] = c\begin{bmatrix} 0 & 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}.$$

For the D2Q9 model, the discrete velocities are given by (Qian et al. 1992)

$$[\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_5, \mathbf{e}_6, \mathbf{e}_7, \mathbf{e}_8] = c\begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}.$$

For the D3Q15 model (Wolf-Gladrow 2000), the velocities are

$$[\mathbf{e}_0,\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3,\mathbf{e}_4,\mathbf{e}_5,\mathbf{e}_6,\mathbf{e}_7,\mathbf{e}_8,\mathbf{e}_9,\mathbf{e}_{10},\mathbf{e}_{11},\mathbf{e}_{12},\mathbf{e}_{13},\mathbf{e}_{14}] = c\begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}.$$

For the D3Q19 model (Wolf-Gladrow 2000), they are

$$[\mathbf{e}_0,\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3,\mathbf{e}_4,\mathbf{e}_5,\mathbf{e}_6,\mathbf{e}_7,\mathbf{e}_8,\mathbf{e}_9,\mathbf{e}_{10},\mathbf{e}_{11},\mathbf{e}_{12},\mathbf{e}_{13},\mathbf{e}_{14},\mathbf{e}_{15},\mathbf{e}_{16},\mathbf{e}_{17},\mathbf{e}_{18}] = c\begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}.$$

In the above equations, c is the lattice speed and is defined as $c = \frac{\Delta x}{\Delta t}$. Here, we define 1 lattice unit (Δ x) as 1 lu, 1 time step (Δt) as 1 ts, and 1 mass unit as 1 mu. There are other velocity models available, for example the D3Q27 model (He and Luo 1997), but we do not use them in simulations in this book.

In Eq. (1.3) w_is are weighting coefficients that can be derived theoretically (He and Luo 1997). c_s^2 can be derived from

$$c_s^2\delta_{\alpha\beta} = \sum_i w_i e_{i\alpha} e_{i\beta}, \tag{1.4}$$

where $\delta_{\alpha\beta} = 1$ when $\alpha = \beta$, otherwise $\delta_{\alpha\beta} = 0$ and we use the Einstein summation convention as detailed in the appendix to this chapter. Hence, $c_s^2 = \sum_i w_i e_{ix} e_{ix}$ or $c_s^2 = \sum_i w_i e_{iy} e_{iy}$. As a detailed example, the computation of c_s^2 for the D2Q9 model is given in the following (calculation of each term from $i = 0$ to $i = 8$ is shown):

$$\begin{aligned}\sum_{i=0}^{8} w_i e_{ix} e_{ix} &= 0 + \frac{1}{9}c^2 + 0 + \frac{1}{9}c^2 + 0 \\ &\quad + \frac{1}{36}c^2 + \frac{1}{36}c^2 + \frac{1}{36}c^2 + \frac{1}{36}c^2 = \frac{1}{3}c^2 = c_s^2,\end{aligned} \tag{1.5}$$

while the contribution from $\alpha \neq \beta$ ($x \neq y$) is

$$\begin{aligned}\sum_{i=0}^{8} w_i e_{ix} e_{iy} &= 0 + \frac{1}{9}c \times 0 + \frac{1}{9}0 \times c + \frac{1}{9}(-c) \times 0 + \frac{1}{9}0 \times (-c) \\ &\quad + \frac{1}{36}c^2 - \frac{1}{36}c^2 + \frac{1}{36}c^2 - \frac{1}{36}c^2 = 0.\end{aligned} \tag{1.6}$$

In Eq. (1.3) ρ is the density of the fluid, which can be obtained from:

$$\rho = \sum_i f_i. \tag{1.7}$$

This is simply the sum of the f_i, revealing them as portions of the overall density associated with one of the discrete velocity directions. For $S_i = 0$, the macroscopic fluid velocity is given by

$$\mathbf{u} = \frac{1}{\rho} \sum_i f_i \mathbf{e}_i, \tag{1.8}$$

or in terms of the vector components of $\mathbf{u}$ as

$$u_\alpha = \frac{1}{\rho} \sum_i f_i e_{i\alpha}, \tag{1.9}$$

which means the discrete velocities weighted by the directional densities. Application examples for viscous single-phase flow can be found in Yu et al. (2003), Dünweg and Ladd (2009), Aidun and Clausen (2010), and many others. A discussion on the H theorem in the context of the LBM can be found in Succi et al. (2002).

1.3 Multiphase LBM

Numerous macroscopic numerical methods have been developed for solving the two-phase N–S equations (Scardovelli and Zaleski 1999), such as the front-tracking method, the volume-of-fluid (VOF) method, the level set method, and so on. The first three methods are the most popular ones. However, the front-tracking method is usually not able to simulate interface coalescence or break-up (Liu et al. 2012; Scardovelli and Zaleski 1999). In the VOF and level set methods an interface reconstruction step or interface reinitialization is usually required, which may be non-physical or complex to implement (Liu et al. 2012). In addition, numerical instability may appear when the VOF and level set methods are applied to simulate surface-tension-dominated flows in complex geometries (Scardovelli and Zaleski 1999).

Compared to common CFD methods, the LBM has many advantages (Chen and Doolen 1998). First, it is based on the molecular kinetic theory (Luo 1998). At the macroscopic scale it is able to recover N–S equations. Second, for single-phase flow simulations it usually involves an ideal-gas equation of state. Hence, it is not necessary to solve a Poisson equation for the pressure in the LBM. This saves significant computer central processing unit (CPU) time compared with common CFD methods. Third, it is easy to program and parallelize with much of the computational burden local to a node.

In the last decade LBM has become a numerically robust and efficient technique for simulating both single-phase and multiphase fluids (Chen and Doolen 1998; Guo and Shu 2013; He et al. 1999; Lee and Lin 2005; Rothman and Keller 1988; Shan and Chen 1993; Swift et al. 1995). Compared with conventional methods for multiphase flows, LBM usually automatically maintains sharp interfaces, and explicit interface tracking is not needed (Házi et al. 2002; Inamuro et al. 2004; Sankaranarayanan et al. 2003).

There are several popular multiphase LBM models. The earliest one is the color-gradient model proposed by Gunstensen et al. (1991), which is based on the Rothman–Keller (RK) multiphase lattice gas model (Rothman and Keller 1988). The Shan–Chen (SC) model (Shan and Chen 1993) appeared soon after and is based on incorporation of an attractive or repulsive force, which leads to phase separation. The free-energy (FE) model was proposed by Swift et al. (1995), then the He-Chen-Zhang (HCZ) model (He et al. 1999) was proposed. Coupling with common CFD techniques, some other less popular multiphase LBMs have also been proposed, such as symmetric free-energy-based multi-component LBM (Li and Wagner 2007), the front-tracking LBM (Lallemand et al. 2007), finite-difference LBM for binary fluid (Xu 2005), the total variation diminishing LBM (Teng et al. 2000), the model of Nourgaliev et al. (2002), etc.

We provide an introduction to each of these popular models below and the models are examined in detail in the chapters that follow.

1.3.1 Color-gradient model

In the two-component model, one component is red-colored fluid and the other is blue-colored fluid. Two distribution functions are used to represent the two fluids. In addition to the common collision step in the LBM, there is an extra collision term in the model (Latva-Kokko and Rothman 2005a). There is also a re-coloring step in the model. Grunau et al. (1993) modified the model to handle binary fluids with different density and viscosity ratios. More recently, Ahrenholz et al. (2008) improved the RK model and used a multiple relaxation time (MRT) LBM to handle cases of higher viscosity ratios and lower capillary numbers. One advantage of the RK model is that the surface tension and the ratio of viscosities can be adjusted independently (Ahrenholz et al. 2008). Huang et al. (2013) confirmed that although the RK model is able to correctly simulate density-matched cases, it is usually unable to handle high-density ratio cases. The possible reasons are given in Huang et al. (2013). For cases of density ratio of order O(10), a scheme to improve the RK model is suggested (Huang et al. 2013).

1.3.2 Shan–Chen model

The second type of multiphase LBM model is the SC model (Shan and Chen 1993, Shan and Chen 1994, Sukop and Thorne 2006). In the single-component multiphase (SCMP) SC model, incorporating a forcing term into the corresponding Lattice Boltzmann equation replaces the ideal gas equation of state (EOS) in single-phase LBMs by a non-ideal non-monotonic EOS (Shan and Chen 1993). In the multi-component multiphase (MCMP) SC model, each component is represented by its own distribution function (Shan and Doolen 1995).

The SC SCMP model works well with density ratios of O(10) (Huang et al. 2011a), but the surface tension and the ratio of densities and viscosities cannot be adjusted independently. Some parameters have to be determined through

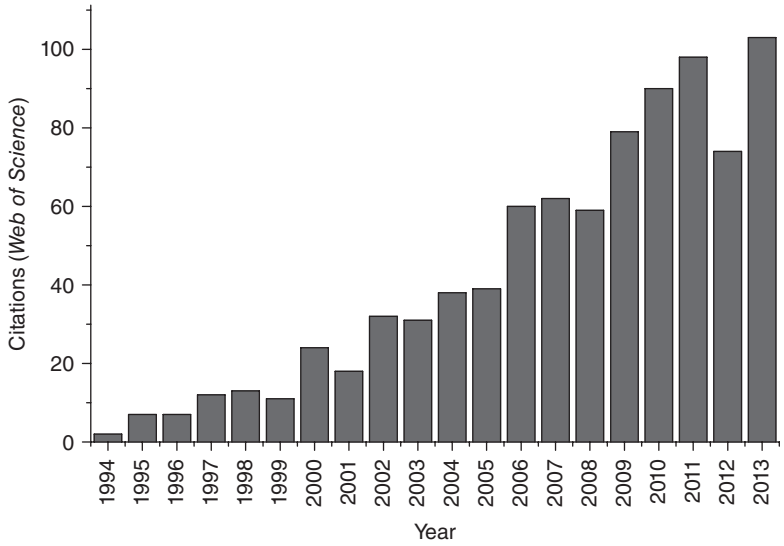

Figure 1.2 Citations in each year for the SC model (Shan and Chen 1993) (data from *Web of Science*).

numerical experiments (Ahrenholz et al. 2008). Falcucci et al. (2010) and Sbragaglia et al. (2007) argue that by adopting a multirange pseudopotential form (including interactions beyond nearest neighbors), the ratio of densities and the surface tension can be adjusted. Shan (2008) outlined a general approach for calculating the pressure tensor in the SC LBM with interactions beyond nearest neighbors. The extension of the interaction beyond the range of nearest neighbors is able to eliminate the spurious currents (Shan 2006, Wagner 2003), which are small-amplitude artificial and unphysical velocity fields near the interface. This finding may expand the possible applications of the SC model. However, some studies (Huang et al. 2011a; Shan et al. 2006) indicate that there is a defect in the original forcing strategy of the SC model and suggest a correct one.

Web of Science citations of the first SC model paper (Shan and Chen 1993) are illustrated in Figure 1.2. From the figure we can see that the citations have increased with time. Study and/or application of the model are very active (approximately 100 citations per year). It is worth mentioning that Shan and Chen (1993) is first in the list of most-cited articles that have been published in *Physical Review E* since 1993. To the end of 2013, the total number of citations is about 860 in the *Web of Science*. Refer to http://pre.aps.org/ for additional information.

1.3.3 Free-energy model

The third type of multiphase LBM model is the FE LBM (Swift et al. 1996, 1995). In this model the thermodynamic issue of the non-monotonic EOS is

Table 1.2 Citations of main articles about Lattice Boltzmann multiphase models.

Model	Article	Total citations through 2013	Citations in 2013
RK	Rothman and Keller (1988)	314	12
	Gunstensen et al. (1991)	468	43
SC	Shan and Chen (1993)	860	103
	Shan and Chen (1994)	398	40
FE	Swift et al. (1995)	468	38
	Swift et al. (1996)	498	50
HCZ	He et al. (1999)	277	38
	Lee and Lin (2005)	146	31

incorporated into the pressure tensor in the N–S equations and the normal equilibrium distribution function is revised (Swift et al. 1995). However, the original FE model (Swift et al. 1995) is not Galilean invariant for the viscous terms in the N–S equation (Luo 1998; Swift et al. 1995). Holdych et al. (1998) improved the model by redefining the stress tensor and Galilean invariance was recovered to $O(u^2)$, which is consistent with the spirit of the LBM.

Inamuro et al. (2004) achieved a high-density ratio through improving Swift's FE model (Swift et al. 1995), but the model has to solve a Poisson equation, which decreased the simplicity of the usual LBM. Zheng et al. (2006) proposed a Galilean-invariant FE LBM model. This model is simpler than that of Inamuro et al., but only valid for density-matched cases (Fakhari and Rahimian 2010).

1.3.4 Interface tracking model

The fourth type of multiphase LBM model is the interface tracking model proposed by He et al. (1999) (the HCZ model). In the HCZ model, two distribution functions and two corresponding LBEs are used. Macroscopically, the Cahn–Hilliard interface tracking equation and the N–S equations can be recovered from the Lattice Boltzmann equations. Based on this model, many models have been developed to access higher density ratios (Amaya-Bower and Lee 2010; Lee and Fischer 2006; Lee and Lin 2005; Lee and Liu 2010) or to enhance numerical stability by extending into an MRT version (McCracken and Abraham 2005).

All the above Lattice Boltzmann multiphase models are under active development. The citation trends are similar to those of the SC model. In Table 1.2, the total citations garnered by each model (based on only the most representative articles) are listed.

1.4 Comparison of models

There are some theoretical analyses (He and Doolen 2002; Luo 1998; Swift et al. 1995) of the RK, SC, and FE models, and a few numerical analyses (Hou et al.

Table 1.3 Comparison of Lattice Boltzmann multiphase models.

Model	Maximum density ratio	Convenient to specify wetting condition?	Efficiency	Accuracy
RK	$O(10)^*$	Yes	Not so efficient	Accurate
SC SCMP	$O(10^2)$	Yes	Very efficient	Less accurate
SC MCMP	$O(1)$	Yes	Very efficient	Less accurate
FE	$O(10)$	No, density gradient required	Not so efficient	Accurate
HCZ	$O(10)$	Yes	Efficient	Accurate
Lee–Lin	$O(10^3)$	No, density gradient required	Efficient	Accurate

* The revised RK (Huang et al. 2013). For the RK model (Latva-Kokko and Rothman 2005a), only density-matched cases can be simulated correctly.

1997; Huang et al. 2011b). Hou et al. (1997) compared the SC and RK models, and focused on drop/bubble simulation. However, there are no quantitative comparisons with other available analytical solutions in that work. Huang et al. (2011b) evaluated the performance of the RK, SC, and FE models for multi-component flow in porous media.

In this book, all of these popular models, the RK model (Grunau et al. 1993; Gunstensen et al. 1991; Rothman and Keller 1988), the SC model (Shan and Chen 1993; Shan and Doolen 1995), the FE model (Inamuro et al. 2004; Swift et al. 1995), and the HCZ model (He et al. 1999; Lee and Lin 2005), will be evaluated in detail. The emphasis is on the strengths and weakness of each model.

The models are compared in Table 1.3. For the RK model (Latva-Kokko and Rothman 2005a), only density-matched cases can be simulated correctly. Huang et al. (2013) extended the RK model to handle higher density ratios. According to our experience, the SC model is very efficient but less accurate. The FE model is as efficient as the RK model. Potentially, there are some similarities between them (Huang et al. 2011b). For the original HCZ model (He et al. 1999), the density ratio is about $O(10)$. Later the HCZ model was extended by Lee and Lin (2005) (Lee–Lin model) to handle high-density ratios. How to specify the wetting condition is an important topic in multiphase flow problems, especially for flows in porous media. In the RK, SC, and HCZ models, a "wall density" can be specified to obtain the desired contact angles (Huang et al. 2014b; Huang and Lu 2009) or a fluid-surface force can be incorporated (Huang et al. 2007; Martys and Chen 1996; Sukop and Thorne 2006). In the FE and Lee–Lin models, the wetting condition can only be implemented by specifying the density gradient on the wall (Liu et al. 2013). Specifying "wall density" or force is more convenient than the density gradient scheme (Liu et al. 2013). For more details readers should refer to the corresponding chapters.

The structure of the book is shown in Figure 1.3 and it is organized as follows. In each chapter, first the model will be introduced briefly. Second, the relevant

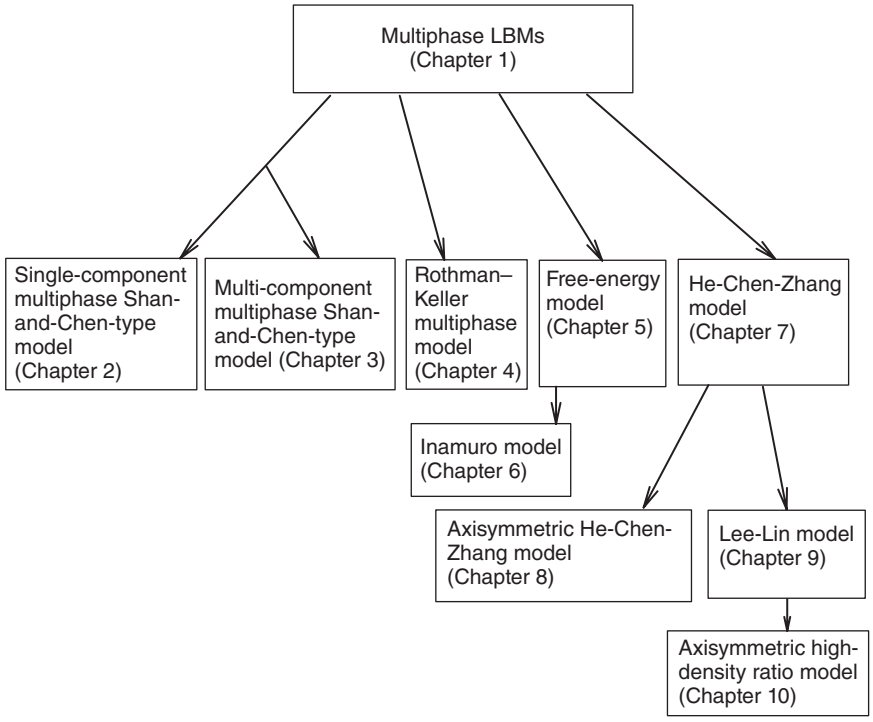

Figure 1.3 Content of the book.

numerical analysis, such as the Chapman–Enskog expansion, and other important formula derivations relevant to the model are presented. Then applications of the model are given, such as contact angles, bubble rise, and multiphase flows in porous media.

1.5 Units in this book and parameter conversion

In this book, if the units of a variable are not specified, the units are in lattice units. In other words, the units are combinations of the basic units listed in Table 1.4. For example, velocity is given in lu/ts, density in mu/ lu^3, pressure in mu/(lu ts^2), surface tension in mu/ ts^2, etc.

Usually, parameter conversion can be performed through non-dimensional parameters, e.g., Reynolds number *Re*, Weber number *We*, or capillary number *Ca*. For single-component flows, the parameter conversion is easier than for multiphase flows because often only *Re* is considered. Hilpert (2011), Sukop and Thorne (2006) provide a procedure for *Re*-matching for gravity-driven flow in a slit. As another example, in a lid-driven cavity flow, the fluid is water at 20° C ($\nu = 10^{-6}$ m^2/s), cavity dimension (characteristic length) is $L = 0.01$ m, and the

Table 1.4 Units in Lattice Boltzmann methods.

	Unit
Mass	mu (mass unit)
Length	lu (lattice unit)
Time	ts (time step)
Temperature	tu (temperature unit)

lid velocity is $U = 0.01$ m/s. The Reynolds number in this flow is

$$Re = \frac{U_{phys} L_{phys}}{\nu_{phys}} = \frac{0.01 \text{ m/s} \times 0.01 \text{ m}}{10^{-6} \text{ m}^2/\text{s}} = 100 = \frac{U_{LBM} L_{LBM}}{\nu_{LBM}}. \tag{1.10}$$

One can first choose the LBM length L, pick an LBM kinematic viscosity ν, and then compute the LBM lid velocity U. Change parameters if U is too big (> 0.1 lu/ts). Alternatively, if the dimension of the cavity and the velocity in LBM are assumed to be $L = 100$ lu and $U = 0.1$ lu/ts, the kinematic viscosity can be calculated from $\nu_{LBM} = \frac{U_{LBM} L_{LBM}}{Re} = 0.1 \text{ lu}^2/\text{ts}$. Then the relaxation time can be obtained through $\tau = \frac{\nu}{c_s^2 \Delta t} + 0.5 = 0.8$, where τ is supposed to be a non-dimensional parameter. The non-dimensional time step is

$$\Delta t^* = \frac{\Delta t_{LBM} U_{LBM}}{L_{LBM}} = \frac{\Delta t_{phys} U_{phys}}{L_{phys}}. \tag{1.11}$$

Hence, in this case, $\Delta t^* = \frac{U_{LBM} \Delta t_{LBM}}{L_{LBM}} = 10^{-3}$. In the physical situation, the time step is Δt_{phys}, which corresponds to the time step in the LBM $\Delta t_{LBM} == \frac{\Delta t^* \times L_{LBM}}{U_{LBM}} = \frac{10^{-3} \times 100 \text{ lu}}{0.1 \text{ lu/ts}} = 1$.

From the above calculation procedure, we can see that if U is chosen to be smaller, say $U = 0.001$ and L is fixed, the calculated $\tau = 0.503$, which is very close to 0.5. That means the non-dimensional time step $\Delta t^* = \frac{U \Delta t}{L} = 10^{-5}$ is smaller. For unsteady flow problems, not only grid-independence studies but also time step-independence studies should be performed in CFD simulations. For unsteady flow problems, a smaller non-dimensional time step (Δt^*) is preferred, which means that a smaller τ should be adopted.

For the LBM, the *CFL* (Courant–Friedrichs–Lewy) number is $CFL = \frac{e_\alpha \Delta t}{\Delta x_\alpha} = 1$, which is fixed to be unity. A 'predictor–corrector' algorithm was introduced in Nourgaliev et al. (2003) to relax constraints imposed by the *CFL* condition.

For the cases of multiphase flows, here an example about oil–water flow in porous media is used to illustrate how to convert the parameters between LBM and reality. The density and viscosity of oil (dyed PCE) and water are shown in Table 1.5 (Pan et al. 2004). The subscripts n and w denote non-wetting and wetting fluid, respectively.

To make the simulation analogous to reality, the following non-dimensional parameters (capillary number *Ca*, dynamic viscosity ratio *M*, and Bond

Table 1.5 Properties of an experimental multiphase system.

NWP (dyed PCE) density ρ_n	1.6×10^3 kg/ m^3
WP (water) density ρ_w	1.0×10^3 kg/ m^3
NWP dynamic viscosity η_n	1.844×10^{-3} Pa·s
WP dynamic viscosity η_w	1.0×10^{-3} Pa·s
Interfacial tension σ	3.623×10^{-2} (kg/ s^2)
Dynamic viscosity ratio M	1.844
Bond number Bo	8.9×10^{-5}

NWP, non-wetting phase; PCS, tetrachloroethylene; WP, wetting phase.

number Bo) should be identical in both the simulations and reality:

$$Ca = \frac{u_w \eta_w}{\sigma},$$
$$M = \frac{\eta_n}{\eta_w},$$
$$Bo = g_0(\rho_n - \rho_w)\frac{R^2}{\sigma}, \tag{1.12}$$

where u_w is the wetting phase Darcy velocity, g_0 is the gravitational constant, and R is the mean pore radius.

Using the SC model, Pan et al. (2004) proposed two ways to achieve the desired dynamic viscosity ratio in the above flow system (Table 1.5). One approach matches both the density and kinematic viscosity ratios (Case A) and the other approach assumes that the densities for both fluids are identical and matches the dynamic viscosity ratio by changing the kinematic viscosity of each fluid (Case B). Obviously, in Case B the body force effect is neglected because the Bond number $Bo = 0$ due to $\rho_n = \rho_w$. As mentioned in Pan et al. (2004) in their work, capillary force dominates and $Bo = 0$ is acceptable.

Here we discuss Case A. If the densities of the water and oil (PCE) are set to be 1 mu/ lu^3 and 1.6 mu/ lu^3, respectively, then the density ratio matches that in reality. The surface tension is related to the interaction force between the two components in the SC model (Chapter 3). After the strength of the fluid/fluid interaction (a parameter in the interaction formula) is set, the surface tension is determined. Suppose the relaxation time of the wetting fluid is unity ($\tau_w = 1$) because the dynamic viscosity ratio is

$$\frac{\eta_{n,phys}}{\eta_{w,phys}} = \frac{\eta_{n,LBM}}{\eta_{w,LBM}} = \frac{\rho_n c_s^2(\tau_n - 0.5)}{\rho_w c_s^2(\tau_w - 0.5)} \tag{1.13}$$

and by rearrangement the relaxation time of the non-wetting fluid is

$$\tau_n = \frac{\eta_n}{\eta_w}\frac{\rho_w}{\rho_n}(\tau_w - 0.5) + 0.5 = \frac{1.844}{1.6}(1 - 0.5) + 0.5 = 1.08. \tag{1.14}$$

Through the definition of Ca in Eq. (1.12), we have

$$Ca = \frac{u_{w,phys}\eta_{w,phys}}{\sigma_{phys}} = \frac{u_{w,LBM}\eta_{w,LBM}}{\sigma_{LBM}}. \tag{1.15}$$

Hence, to match a case with the physical $Ca = 10^{-3}$, if $\sigma_{LBM} = 0.1$ mu/ ts^2 (possibly determined by fitting the Laplace law to drops and bubbles), displacement velocity $u_{w,LBM}$ can be calculated in lattice units as follows:

$$\begin{aligned} u_{w,LBM} &= \frac{\sigma_{LBM}Ca}{\eta_{w,LBM}} = \frac{\sigma_{LBM}Ca}{\rho_w c_s^2(\tau_{w,LBM} - 0.5)\Delta t} \\ &= \frac{0.1 \text{ mu/ts}^2 \times 10^{-3}}{1 \text{ mu/lu}^3 \times \frac{1}{3} \text{ lu}^2/\text{ ts}^2 \times (1.0 - 0.5) \times 1 \text{ ts}} = 6 \times 10^{-4} \text{lu/ts}. \end{aligned} \tag{1.16}$$

This $u_{w,LBM}$ can be used to specify the inlet velocity of the displacement to match the case with $Ca = 10^{-3}$. For the gravity effect, if in our LBM simulation $R = 10$ lu, then g_0 in lattice units can be calculated from the definition of Bo:

$$Bo = g_{0,phys}(\rho_{n,phys} - \rho_{w,phys})\frac{R_{phys}^2}{\sigma_{phys}} = g_{0,LBM}(\rho_{n,LBM} - \rho_{w,LBM})\frac{R_{LBM}^2}{\sigma_{LBM}}. \tag{1.17}$$

That is

$$\begin{aligned} g_{0,LBM} &= \frac{\sigma_{LBM}Bo}{R_{LBM}^2(\rho_{n,LBM} - \rho_{w,LBM})} \\ &= \frac{0.1 \text{ mu/ts}^2 \times 8.9 \times 10^{-5}}{100 \text{ lu}^2 \times (1.6 - 1.0) \text{ mu/lu}^3} = 1.48 \times 10^{-7} \text{ lu/ts}^2. \end{aligned} \tag{1.18}$$

For more examples of parameter conversion between LBM simulations and physical reality, please refer to Chapter 2 (Section 2.6), Chapter 7 (Sections 7.6 and 7.7), and Chapter 8 (Sections 8.6 and 8.8).

1.6 Appendix: Einstein summation convention

In this book, the subscripts α, β, and γ denote Cartesian components (for 2D cases, it is x or y coordinates; for 3D cases, it is x, y, or z coordinates).

According to the Einstein summation convention, when an index variable (the subscript) appears twice in a single term it implies summation of that term over all the values of the index. For example, the kinematic energy E in 2D cases is

$$E = \frac{1}{2}m(u_x^2 + u_y^2), \tag{1.19}$$

where m is mass of an object. Obviously, the indices (subscripts) can range over the set x, y for 2D cases. Eq. (1.19) can be reduced by the convention to

$$E = \frac{1}{2}mu_\alpha u_\alpha. \tag{1.20}$$

For 3D cases, the equation (1.20) means $E = \frac{1}{2}m(u_x^2 + u_y^2 + u_z^2)$.

The index that is summed over is a summation index, in this case α. It is also called a dummy index since any symbol can replace α without changing the meaning of the expression, provided that it does not collide with index symbols in the same term (Hazewinkel 1993). Note that dummy indices do not appear in the 'answer', e.g., $I = \int f(\theta)d\theta = \int f(x)dx$, where θ and x are dummy variables.

An index that is not summed over is a free index and should be found in each term of the equation or formula if it appears in any term (Hazewinkel 1993). The connection between the boldface notation and Einstein summation can be illustrated in the following examples. The boldface notation $\mathbf{v} \cdot \mathbf{w}$, $\nabla \cdot \vec{\vec{\tau}}$, and $\nabla^2 s$ can be expressed as $v_i w_i$, $\partial_j \tau_{ji}$, and $\partial_i \partial_i s = \partial_i^2 s$, respectively. Here $\mathbf{v}$, $\mathbf{w}$ are vectors, $\vec{\vec{\tau}}$ is a second-order tensor, and s is a scaler.

1.6.1 Kronecker δ function

When the subscripts in $\delta_{\alpha\beta}$ are identical, for example $\alpha = \beta = x$ (or $\alpha = \beta = y$ in 2D cases), then $\delta_{\alpha\beta} = 1$, otherwise $\delta_{\alpha\beta} = 0$. Hence, we have $u_\alpha u_\beta \delta_{\alpha\beta} = u_\alpha u_\alpha = u_x^2 + u_y^2$ for 2D cases.

In the following, two additional examples demonstrate the operation of the δ function. For 2D cases ((x, y) coordinates), $\delta_{\alpha\beta}\delta_{\alpha\beta} = 2$ because according to the summation rule

$$\begin{aligned}\delta_{\alpha\beta}\delta_{\alpha\beta} &= (\delta_{\alpha x}\delta_{\alpha x}) + (\delta_{\alpha y}\delta_{\alpha y}) = (\delta_{xx}\delta_{xx} + \delta_{yx}\delta_{yx}) + (\delta_{xy}\delta_{xy} + \delta_{yy}\delta_{yy}) \\ &= \delta_{xx}\delta_{xx} + \delta_{yy}\delta_{yy} = 2. \end{aligned} \tag{1.21}$$

It is easy to derive $\delta_{\alpha\beta}\delta_{\alpha\beta} = 3$ for 3D cases.

For 2D cases ((x, y) coordinates), $\delta_{\alpha\beta}\delta_{\alpha\gamma} = \delta_{\beta\gamma}$ because

$$\delta_{\alpha\beta}\delta_{\alpha\gamma} = \delta_{x\beta}\delta_{x\gamma} + \delta_{y\beta}\delta_{y\gamma}. \tag{1.22}$$

It is obvious that providing $\beta = \gamma$ (no matter $\beta = \gamma = x$ or $\beta = \gamma = y$), $\delta_{\alpha\beta}\delta_{\alpha\gamma} = 1$ (otherwise $\delta_{\alpha\beta}\delta_{\alpha\gamma} = 0$), which exactly means $\delta_{\alpha\beta}\delta_{\alpha\gamma} = \delta_{\beta\gamma}$.

1.6.2 Lattice tensors

For the first- and third-order lattice tensors in the D2Q9 model, because of the symmetry of the velocities in the velocity model we have

$$\sum_{i=0}^{8} e_{i\alpha} = 0 \tag{1.23}$$

and

$$\sum_{i=0}^{8} e_{i\alpha}e_{i\beta}e_{i\gamma} = 0. \tag{1.24}$$

For the second-order lattice tensor in the D2Q9 model we have:

$$\sum_{i=1}^{4} e_{i\alpha}e_{i\beta} = 2c^2\delta_{\alpha\beta} \quad \text{and}$$

$$\sum_{i=5}^{8} e_{i\alpha}e_{i\beta} = 4c^2\delta_{\alpha\beta}. \tag{1.25}$$

For the fourth-order lattice tensor in the D2Q9 model, the formula is a little bit more complex:

$$\sum_{i=1}^{4} e_{i\alpha}e_{i\beta}e_{i\gamma}e_{i\delta} = 2c^4\delta_{\alpha\beta\gamma\delta} \qquad \text{and}$$

$$\sum_{i=5}^{8} e_{i\alpha}e_{i\beta}e_{i\gamma}e_{i\delta} = 4c^4(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) - 8c^4\delta_{\alpha\beta\gamma\delta} \tag{1.26}$$

If the weighting factor w_i in the equilibrium distribution function (Eq. (1.3)) is added into the above tensors, we have

$$\sum_{i=0}^{8} w_i e_{i\alpha} = 0, \tag{1.27}$$

$$\sum_{i=0}^{8} w_i e_{i\alpha}e_{i\beta} = \frac{1}{3}c^2\delta_{\alpha\beta}, \tag{1.28}$$

$$\sum_{i=0}^{8} w_i e_{i\alpha}e_{i\beta}e_{i\gamma} = 0, \tag{1.29}$$

and

$$\sum_{i=0}^{8} w_i e_{i\alpha}e_{i\beta}e_{i\gamma}e_{i\delta} = c_s^4(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}). \tag{1.30}$$

In the following chapters, in the Chapman–Enskog expansion analysis the above formulae are used extensively.

1.7 Use of the Fortran code in the book

The codes provided with this book were originally written and edited using Compaq Visual Fortran version 6.5.0. The format is strictly based on the fixed format (Fortan77): before each line, there are seven blank spaces. For a long line exceeding the fixed line length of 80 columns, "&", "*" or "%" at the beginning of the next line can be used to extend the content onto that line.

Lines "c==========" in the book's listing of the code are used to separate the different files only. Please do not include these lines in the separate Fortran files that will be needed to

compile the code. Lines "c- - - - - - - - - - - - - - - - - " in the code are used to separate small segments inside a subroutine.

If Compaq Visual Fortran is used to compile the files, choose Win32 Release in the menu Build-> Set Active Configuration. Executable files compiled using Win32 Release are much more efficient than those compiled using the Win32 Debug option. If you encounter an error "forrt1: severe <170> : Program Exception – stack overflow", modify the following option and compile again. In "Project->Setting->Link->Output->Reserve", set the reserve value to be at least 200,000,000. This means you reserve more memory when the executable file runs.

Create a new folder "out" in the working directory before running the code and all output files will be written into the subfolder. The codes given in this book have not been optimized for memory or efficiency.

We also provide instructions designed to make the codes compatible with gfortran when directly copied from the PDF. This is necessary due to the fact that directly copying from the PDF changes much of the spacing and involves characters that are not compilable by gfortran. We suggest not using Adobe reader for directly copying and pasting due to the program not working well. We suggest using other alternative PDF viewer/editors such as Foxit, which do not have these problems.

The code should be compiled using the command "gfortran -ffree-form *.for" in order to compile all the files at once. Each file should be tested individually using gfortran in order to catch errors. All c's used to comment should be changed to !. All continuation characters, including &, *, and %, should be changed to & and placed at the end of the preceding line, not at the beginning of the new line.

When copying directly from a PDF to a text editor, the system uses * as a linebreak separating code. If this occurs, then the previous line must be joined with the next line with the * in between. This is only in the case when * is used as an operator and not as a replacement for &.

A line of - - - - - - - - - ending in - tends to pull the next line up. The next line should be moved back down where it is part of the code.

Thick apostrophes are not readable by the compiler and should be replaced with a standard'.

Finally, we have made a repository of codes in both the Compaq and gfortan forms available on the internet (www.wiley.com/go/huang/boltzmann).