Lattice Boltzmann Framework for Multiphase Flows by Eulerian–Eulerian Navier–Stokes Equations

computation

MDPI

Article

Lattice Boltzmann Framework for Multiphase Flows by Eulerian–Eulerian Navier–Stokes Equations

Matteo Maria Piredda [1] and Pietro Asinari [1,2,*]

[1] Dipartimento Energia, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, TO, Italy; matteo.piredda@polito.it
[2] Istituto Nazionale di Ricerca Metrologica, Strada delle Cacce 91, 10135 Turin, TO, Italy
* Correspondence: pietro.asinari@polito.it; Tel.: +39-011-0904434

Abstract

Although the lattice Boltzmann method (LBM) is relatively straightforward, it demands a well-crafted framework to handle the complex partial differential equations involved in multiphase flow simulations. For the first time to our knowledge, this work proposes a novel LBM framework to solve Eulerian–Eulerian multiphase flow equations without any finite difference correction, including very-large-density ratios and also a realistic relation for the drag coefficient. The proposed methodology and all reported LBM formulas can be applied to any dimension. This opens a promising venue for simulating multiphase flows in large High Performance Computing (HPC) facilities and on novel parallel hardware. This LBM framework consists of six coupled LBM schemes—running on the same lattice—ensuring an efficient implementation in large codes with minimum effort. The preliminary numeral results agree in an excellent way with the reference numerical solution obtained by a traditional finite difference solver.

Keywords: computational fluid dynamics; multiphase flows; Eulerian–Eulerian Navier–Stokes equations; Lattice Boltzmann method

Academic Editor: Sergey A. Karabasov

Received: 11 June 2025
Revised: 3 July 2025
Accepted: 6 July 2025
Published: 9 July 2025

Citation: Piredda, M.M.; Asinari, P. Lattice Boltzmann Framework for Multiphase Flows by Eulerian–Eulerian Navier–Stokes Equations. *Computation* **2025**, *13*, 164. https://doi.org/10.3390/computation13070164

1. Introduction

The computational fluid dynamics (CFD) of multiphase flows is crucial in the energy sector, particularly in oil and gas, as it enables the detailed simulation and optimization of complex fluid interactions, such as those in bubble column reactors. These reactors are used extensively in refining processes, gas treatment, and chemical synthesis, where accurate modeling of gas–liquid interactions can lead to improved efficiency, reduced energy consumption, and enhanced safety. Using CFD, industry can better understand the flow dynamics, optimize reactor designs, and ultimately reduce costs and emissions, making it a vital tool for the advancement of sustainable energy solutions.

A multiphase flow system consists of the simultaneous flow of materials with different phases or states of matter in regions bounded by moving interfaces. The lattice Boltzmann method (LBM) has garnered significant interest for modeling multiphase flows, particularly because its inherent parallelism makes it ideal for high-performance computing applications when dealing with large domains. Notably, diffuse interface approaches have become well-established for simulating such flows without relying on Lagrangian meshes and immersed boundaries [1]. Among these, "color gradient" LBM techniques have proven to be particularly effective for modeling multiphase flows in porous media and microchannels [2]. The color-gradient method assigns different "colors" to each fluid

and uses recoloring algorithms to maintain immiscibility and model surface tension. It is known for robustness and numerical stability, accommodating moderate-density ratios (up to ∼100). Various strategies have been developed to incorporate surface tension into LBMs. For example, pseudo-potential models, originally introduced by Shan and Chen, simulate surface tension through pairwise molecular interactions [3,4]. They introduce phase separation through a density-dependent interparticle force. They are widely used due to their simplicity and and work best for moderate-density ratios (typically up to ∼100, or ∼1000 with advanced forcing schemes). The free-energy method employs a Ginzburg–Landau-type free energy functional to model interfacial dynamics and control surface tension precisely. It is suitable for capturing diffuse interfaces but is typically limited to lower-density ratios (∼10–20) [5]. Furthermore, phase-field models solve a coupled Cahn–Hilliard equation to evolve an order parameter representing phase concentration. It shares many features with the free-energy approach, including good interface capturing and limited-density-ratio capabilities [6]. However, all these lattice Boltzmann methods are generally formulated to solve a single-fluid set of equations with a diffuse interface or interfacial force rather than separate momentum equations for each phase. This means they do not directly solve two separate Eulerian momentum equations coupled with drag. Instead, they treat the multiphase system as a single effective fluid with an internal interface. Accurately capturing phase interfaces becomes particularly difficult in complex systems, such as a swarm of bubbles like those seen in the bubble column reactors. Furthermore, these models often introduce an artificial dependence of surface tension on viscosity, leading to numerical instabilities and artifacts at fluid interfaces [7]. A possible approach to modeling multiphase flows without simulating the interface is the mixture approach, which essentially treats the multiphase system as a single mixture with properties that are weighted averages of the different phases [8]. Instead of solving separate momentum equations for each phase, a single momentum equation is solved for the mixture. For this reason, it is a simple and computationally convenient approach but with a lower detail level because a proper model must be provided for the velocity difference between the two phases. It needs small relative velocities, so it is more suitable for closely coupled phases moving together (slurry, sediment) [9].

This work, on the other hand, focuses on developing an Eulerian–Eulerian lattice Boltzmann method (LBM) to simulate multiphase flows. It has the advantage of not solving the interface between the phases, simulating the two fluids as if they interpenetrate and solving separate volume-averaged Navier–Stokes equations for each phase. This frees its applicability from any extensional constraints regarding the number of dispersed phase elements, making it optimal for cases where there are swarms of bubbles/particles. This implies the application of a volume-averaging procedure to both the continuity and momentum equation for every phase, and it requires us to define a volume fraction as follows:

$$\alpha_\varphi = \frac{V_\varphi}{V}, \qquad \sum_{\varphi=1}^{P} \alpha_\varphi = 1, \tag{1}$$

where V is the volume under consideration, the subscript φ stays for a generic phase, and P is used for the total number of phases. In the following, for the sake of simplicity and without loss of generality, let us focus on two phases, namely, $P = 2$, where two phases are identified by l for the liquid phase and g for the gas phase, respectively. In case $P = 2$, if φ stays for a generic phase, then $\overline{\varphi}$ stays for the other one. Writing down the governing equation for both phases constitutes the Eulerian–Eulerian approach (or two-fluid approach) and is a framework used to model multiphase flows where each phase is treated as a continuum, and separate sets of equations are solved for each phase in the same Eulerian (fixed) reference frame. They are used in fluid dynamics to describe the motion of

a fluid while accounting for the effects of spatial variations over a finite region [10]. Let us start by recalling the original Eulerian–Eulerian Navier–Stokes equations for multiphase flows in the isothermal limiting case, used in traditional CFD for both phases [11]:

$$\frac{\partial(\rho_l \alpha_l)}{\partial t} + \nabla \cdot (\alpha_l \rho_l \boldsymbol{u}_l) = 0, \tag{2}$$

$$\frac{\partial(\rho_l \alpha_l \boldsymbol{u}_l)}{\partial t} + \nabla \cdot (\alpha_l \rho_l \boldsymbol{u}_l \boldsymbol{u}_l) = -\alpha_l \nabla p + \rho_l \alpha_l \boldsymbol{g} + \nabla \cdot (\alpha_l \rho_l \boldsymbol{\sigma}_l) + \boldsymbol{F}_{lg} + \boldsymbol{F}_l, \tag{3}$$

$$\frac{\partial(\rho_g \alpha_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g \boldsymbol{u}_g) = 0, \tag{4}$$

$$\frac{\partial(\rho_g \alpha_g \boldsymbol{u}_g)}{\partial t} + \nabla \cdot (\alpha_g \rho_g \boldsymbol{u}_g \boldsymbol{u}_g) = -\alpha_g \nabla p + \rho_g \alpha_g \boldsymbol{g} + \nabla \cdot (\alpha_g \rho_g \boldsymbol{\sigma}_g) + \boldsymbol{F}_{gl} + \boldsymbol{F}_g. \tag{5}$$

For each phase (the subscript φ can be either l or g), α_φ is the previously mentioned volume fraction, ρ_φ is the density, $\boldsymbol{u}_\varphi$ the velocity vector, $\boldsymbol{\sigma}_\varphi$ the viscous stress tensor per unit volume, p the common pressure, $\boldsymbol{F}_{gl} = -\boldsymbol{F}_{lg}$ is the interphase momentum exchange term, which is a cumulative force resulting from the summation of the single interfacial forces acting between the two phases (drag force, lift force, wall lubrication, virtual mass, etc.) per unit volume, and $\boldsymbol{F}_\varphi$ is the force per unit volume of the specific phase φ. Concerning the interphase momentum exchange, let us limit the following discussion to the drag contribution only, which can be formulated as

$$\boldsymbol{F}_{gl} = \rho_g \, K_I \, \|\boldsymbol{u}_l - \boldsymbol{u}_g\| \, (\boldsymbol{u}_l - \boldsymbol{u}_g) = -\boldsymbol{F}_{lg}, \tag{6}$$

where K_I is the effective drag coefficient and is evaluated from empirical or semi-empirical correlations. The viscous stress tensor per unit volume is defined as follows:

$$\boldsymbol{\sigma}_\varphi = \nu_\varphi \left(\nabla \boldsymbol{u}_\varphi + \nabla \boldsymbol{u}_\varphi^T \right) + \left(\xi_\varphi - \frac{2}{3} \nu_\varphi \right) (\nabla \cdot \boldsymbol{u}_\varphi) \, \boldsymbol{I}, \tag{7}$$

where ν_φ is the effective kinematic viscosity for the generic phase, defined as the ratio between dynamic viscosity and density, i.e., $\nu_\varphi = \mu_\varphi / \rho_\varphi$, and ξ_φ is the effective kinematic bulk viscosity. In general, the effective kinematic viscosity should also include turbulent effects. It is important to realize that in multiphase flows, even under the incompressible limit which we will discuss in the following, $\nabla \cdot \boldsymbol{u}_\varphi$ for the individual phase can be different from zero in general.

The previous equations define a proper system of equations for the following variables: α_g, p, $\boldsymbol{u}_g$ and $\boldsymbol{u}_l$, as far as the equation of state for the dispersed phase, i.e., $\rho_g = \rho_g(p)$, and the equation of state for the liquid phase, i.e., $\rho_l = \rho_l(p)$ in the isothermal limiting case, are provided.

In case of the incompressible limit (i.e., low Mach number limit), one can assume the following equations of state: $\rho_g = \rho_g(p) = \rho_g^0$ and $\rho_l = \rho_l(p) = \rho_l^0$, where ρ_g^0, and ρ_l^0 are proper constants consisting in the average density of each phase. In this case, the previous equations reduce to the following:

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{u}_l) = 0, \tag{8}$$

$$\frac{\partial(\alpha_l \boldsymbol{u}_l)}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{u}_l \boldsymbol{u}_l) = -\frac{\alpha_l}{\rho_l^0} \nabla p + \alpha_l \boldsymbol{g} + \nabla \cdot (\alpha_l \boldsymbol{\sigma}_l) + \frac{1}{\rho_l^0} \left(\boldsymbol{F}_{lg} + \boldsymbol{F}_l \right), \tag{9}$$

$$\frac{\partial \alpha_g}{\partial t} + \nabla \cdot (\alpha_g \boldsymbol{u}_g) = 0, \tag{10}$$

$$\frac{\partial(\alpha_g \boldsymbol{u}_g)}{\partial t} + \nabla \cdot (\alpha_g \boldsymbol{u}_g \boldsymbol{u}_g) = -\frac{\alpha_g}{\rho_g^0}\nabla p + \alpha_g \boldsymbol{g} + \nabla \cdot (\alpha_g \boldsymbol{\sigma}_g) + \frac{1}{\rho_g^0}\Big(\boldsymbol{F}_{gl} + \boldsymbol{F}_g\Big). \tag{11}$$

From the numerical point of view, the previous formulation is not convenient and it is better to derive an equivalent system of equations. Summing up continuity equations for both phases in the incompressible limit yields

$$\nabla \cdot (\alpha_g \boldsymbol{u}_g + \alpha_l \boldsymbol{u}_l) = 0. \tag{12}$$

Summing up momentum equations for both phases in the incompressible limit yields

$$\begin{aligned}\frac{\partial(\alpha_g \boldsymbol{u}_g + \alpha_l \boldsymbol{u}_l)}{\partial t} + \nabla \cdot (\alpha_g \boldsymbol{u}_g \boldsymbol{u}_g + \alpha_l \boldsymbol{u}_l \boldsymbol{u}_l) = -\left(\frac{\alpha_g}{\rho_g^0} + \frac{\alpha_l}{\rho_l^0}\right)\nabla p + \boldsymbol{g} + \dots \\ \cdots + \nabla \cdot (\alpha_g \boldsymbol{\sigma}_g) + \nabla \cdot (\alpha_l \boldsymbol{\sigma}_l) + \frac{1}{\rho_g^0}\Big(\boldsymbol{F}_{gl} + \boldsymbol{F}_g\Big) + \frac{1}{\rho_l^0}\Big(\boldsymbol{F}_{lg} + \boldsymbol{F}_l\Big).\end{aligned} \tag{13}$$

Applying the divergence operator to both sides of the equation and swapping the order of the derivatives in the first term yields

$$\begin{aligned}\frac{\partial}{\partial t}\left[\nabla \cdot (\alpha_g \boldsymbol{u}_g + \alpha_l \boldsymbol{u}_l)\right] + \nabla \cdot \nabla \cdot (\alpha_g \boldsymbol{u}_g \boldsymbol{u}_g + \alpha_l \boldsymbol{u}_l \boldsymbol{u}_l) = -\nabla \cdot \left[\left(\frac{\alpha_g}{\rho_g^0} + \frac{\alpha_l}{\rho_l^0}\right)\nabla p\right] \dots \\ + \nabla \cdot \left[\nabla \cdot (\alpha_g \boldsymbol{\sigma}_g) + \nabla \cdot (\alpha_l \boldsymbol{\sigma}_l) + \frac{1}{\rho_g^0}\Big(\boldsymbol{F}_{gl} + \boldsymbol{F}_g\Big) + \frac{1}{\rho_l^0}\Big(\boldsymbol{F}_{lg} + \boldsymbol{F}_l\Big)\right].\end{aligned} \tag{14}$$

Applying Equation (12), it is possible to derive the Poisson equation, which is typically solved for computing the common pressure field p, namely,

$$\begin{aligned}\nabla \cdot \nabla \cdot (\alpha_g \boldsymbol{u}_g \boldsymbol{u}_g + \alpha_l \boldsymbol{u}_l \boldsymbol{u}_l) = -\nabla \cdot \left[\left(\frac{\alpha_g}{\rho_g^0} + \frac{\alpha_l}{\rho_l^0}\right)\nabla p\right] \dots \\ + \nabla \cdot \left[\nabla \cdot (\alpha_g \boldsymbol{\sigma}_g) + \nabla \cdot (\alpha_l \boldsymbol{\sigma}_l) + \frac{1}{\rho_g^0}\Big(\boldsymbol{F}_{gl} + \boldsymbol{F}_g\Big) + \frac{1}{\rho_l^0}\Big(\boldsymbol{F}_{lg} + \boldsymbol{F}_l\Big)\right].\end{aligned} \tag{15}$$

From the numerical point of view, it is better to reformulate the momentum equations in the incompressible limit. This is performed by expanding the derivatives in the left-hand side of the equations and applying Equations (8) and (10) to the momentum equation of the liquid and dispersed phases, respectively:

$$\frac{\partial \boldsymbol{u}_g}{\partial t} + \boldsymbol{u}_g \cdot \nabla \boldsymbol{u}_g = -\frac{1}{\rho_g^0}\nabla p + \boldsymbol{g} + \frac{1}{\alpha_g}\nabla \cdot (\alpha_g \boldsymbol{\sigma}_g) + \frac{1}{\alpha_g \rho_g^0}\Big(\boldsymbol{F}_{gl} + \boldsymbol{F}_g\Big), \tag{16}$$

$$\frac{\partial \boldsymbol{u}_l}{\partial t} + \boldsymbol{u}_l \cdot \nabla \boldsymbol{u}_l = -\frac{1}{\rho_l^0}\nabla p + \boldsymbol{g} + \frac{1}{\alpha_l}\nabla \cdot (\alpha_l \boldsymbol{\sigma}_l) + \frac{1}{\alpha_l \rho_l^0}\Big(\boldsymbol{F}_{lg} + \boldsymbol{F}_l\Big). \tag{17}$$

The singularity of the forces depend on the terms $\nabla \ln(\alpha_\varphi)$ (where φ can be either g or l). There are well-established techniques in standard computational fluid dynamics for multiphase flows to handle exactly this kind of term [12]. For this term to be well-behaved as $\alpha_\varphi \to 0$, it is necessary for the gradient $\nabla \alpha_\varphi$ to approach zero faster than α_φ. Numerically, in standard CFD methods, it is straightforward to discretize this term in a way that prevents division by zero: this can be achieved by representing α_φ in the denominator using a proper volumetric average and/or by applying a proper slope limiter [13].

The system of equations defined by Equations (10) and (15)–(17) in terms of quantities α_g, p, $\boldsymbol{u}_g$, and $\boldsymbol{u}_l$ represents a promising starting point for numerics in most of the

existing software for solving multiphase flows by the Eulerian–Eulerian approach (e.g., OpenFOAM).

In spite of the existence of feasible numerical methods, there are remaining complexities of the Eulerian–Eulerian approach which must be faced. The solution of the multifluid set of equations presents many challenges:

- Possible singularities of the phase momentum equations;
- Coupling between the phases, which could lead to instabilities in the numerical procedure;
- Phase volume fraction needs to be bounded between 0 and 1;
- Sharp profiles of phase volume fractions;
- Extension of iterative solution procedures to the co-located grid arrangement for avoiding checkerboard instability patterns.

For example, the open-source code OpenFOAM implements a numerical iterative solution procedure proposed by Passalacqua et al. [14], originally developed for fluid-particle flows. It consists in a finite volume method (FVM) discretizing the Navier–Stokes equations Equations (10) and (15)–(17). It uses face fluxes and velocity fluxes to overcome the problems described above along with deriving the pressure equation and the dispersed phase continuity equation. Despite its effectiveness in FVM, this procedure is clearly unfeasible for LBM, e.g., because LBM cannot easily solve Equation (15) and cannot straightforwardly impose the same pressure gradient ∇p to both phases.

2. Materials and Methods: Eulerian–Eulerian Lattice Boltzmann Method (LBM) to Simulate Multiphase Flows

The lattice Boltzmann method (LBM) is highly promising for simulating multiphase flows on High Performance Computing (HPC) systems due to its unique computational structure, which is naturally parallel and localized. Unlike conventional CFD methods that rely on solving the Navier–Stokes equations through techniques like finite difference (FD), the finite volume method (FVM), or the finite element method (FEM), LBM simulates fluid flows by modeling the fluid as discrete particles moving and colliding on a lattice [15]. This approach enables LBM to perform computations locally at each grid point, making it well-suited for parallel execution across HPC platforms.

For LBM to fully exploit HPC capabilities, it is essential to maintain the standard, unmodified LBM formulation, avoiding FD corrections or other modifications that introduce non-local dependencies. These corrections, sometimes used to address stability or accuracy, create dependencies that disrupt the purely local and independent computations that are a key advantage of LBM. Such non-local adjustments increase computational costs, reduce parallel efficiency, and complicate memory access patterns, which diminishes the performance of LBM on large HPC systems.

Therefore, adhering closely to the standard LBM formulation preserves its localized computation advantage, reducing communication overhead and allowing efficient scaling across numerous computational nodes. This efficiency enables the simulation of high-resolution, large-domain multiphase flows crucial for energy sector applications, like bubble column reactors in oil and gas, where detailed fluid dynamics insights are essential for optimizing processes.

This paper aims to derive an LBM framework to simulate Eulerian–Eulerian equations for multiphase flows. The overall scheme consists of six LBM schemes for the two phases, as follows:

- Two LBM schemes for artificially compressible continuity equations and momentum equations, in terms of particle distribution functions f_g and f_l;
- Two LBM schemes for phase volume fractions, in terms of $f_{\alpha g}$ and $f_{\alpha l}$;
- Two LBM schemes for phase continuity sources, in terms of $f_{\beta g}$ and $f_{\beta l}$.

The proposed LBM framework is derived and analyzed in the rest of this section in a general, way which does not pose any constraint on the physical dimensionality, while the following section reports a preliminary numerical validation.

2.1. The Key Point: Artificially Compressible Continuity Equation and Momentum Equation for Each Phase

The LBM uses the artificial compressibility concept [16] and hence it cannot solve Equation (15) directly. Hence a pseudo-compressible system of equations for each phase is needed as a starting point for deriving the Eulerian–Eulerian LBM framework to simulate multiphase flows. Applying the artificial compressibility concept requires a system of equations made of (a) a momentum equation and (b) an artificially compressible continuity equation, which is obviously missing in the incompressible formulation, for each phase. Let us start with the dispersed phase and let us start by modifying Equation (12) in the following way:

$$\frac{\partial \epsilon_g}{\partial t} + \nabla \cdot (\alpha_g \boldsymbol{u}_g + \alpha_l \boldsymbol{u}_l) = 0, \tag{18}$$

where ϵ_g is a function computed by flow quantities. If the artificial compressibility is used as a shortcut to find out only the steady-state solution, then it is enough that $\partial \epsilon_g / \partial t$ is small enough with regard to the other terms. However, in this work, the idea is to recover the right incompressible dynamics as well, and hence, $\partial \epsilon_g / \partial t$ must be asymptotically small during the entire dynamics, according to the order of convergence of the numerical scheme. For example, this term could be $\partial \epsilon_g / \partial t = O(h^2)$ in the case of a second-order method, where h is the mesh spacing. The last condition can be recovered by a proper scaling, namely, a proper choice of the simulation parameters during mesh refinement. We will enforce this condition in the following, after discussing the asymptotic analysis of the proposed methodology. Let us manipulate the previous equation as follows:

$$\frac{\partial \epsilon_g}{\partial t} + \nabla \cdot \boldsymbol{u}_g = \nabla \cdot [\alpha_l (\boldsymbol{u}_g - \mathbf{u}_l)] \equiv S_g. \tag{19}$$

The previous equation is an artificially compressible continuity equation designed for the dispersed-phase velocity field $\mathbf{u}_g$. We have to derive a similar equation for the liquid-phase velocity field $\mathbf{u}_l$. Hence, this time, let us modify Equation (12) as follows:

$$\frac{\partial \epsilon_l}{\partial t} + \nabla \cdot \boldsymbol{u}_l = \nabla \cdot [\alpha_g (\boldsymbol{u}_l - \boldsymbol{u}_g)] \equiv S_l. \tag{20}$$

Different strategies are possible in choosing the pair of functions ϵ_g and ϵ_l:

- Because both previous equations are derived from the same Equation (12), then one could assume $\epsilon_g = \epsilon_l = \epsilon(p)$. The simplest choice would be $\epsilon(p) = p$, or by looking at the Poisson equation for the mixture velocity given by Equation (15), one could use instead $\epsilon(p) = (\alpha_g/\rho_g^0 + \alpha_l/\rho_l^0)\, p$, introducing a further dependence on the volume fraction in the pseudo-compressibility.
- The problem with the previous approach is that it forces both phases not only to have the same asymptotic target given by Equation (12) but also to have the same approaching dynamics. Because the two phases are subject to different forces, this could lead to over-constraining. Hence, one could assume instead two independent functions $\epsilon_g = \epsilon_g(p)$ and $\epsilon_l = \epsilon_l(p)$. In this work, we choose this second approach. The explicit expressions of these functions will be provided in the following.

Concerning the momentum equations, let us manipulate Equations (16) and (17) as follows:

$$\frac{\partial \boldsymbol{u}_g}{\partial t} + \nabla \cdot (\boldsymbol{u}_g \boldsymbol{u}_g) - \boldsymbol{u}_g \nabla \cdot \boldsymbol{u}_g = -\frac{1}{\rho_g^0} \nabla p + \boldsymbol{g} + \frac{1}{\alpha_g} \nabla \cdot (\alpha_g \boldsymbol{\sigma}_g) + \frac{1}{\alpha_g \rho_g^0} \Big(\boldsymbol{F}_{gl} + \boldsymbol{F}_g \Big), \tag{21}$$

$$\frac{\partial \boldsymbol{u}_l}{\partial t} + \nabla \cdot (\boldsymbol{u}_l \boldsymbol{u}_l) - \boldsymbol{u}_l \nabla \cdot \boldsymbol{u}_l = -\frac{1}{\rho_l^0} \nabla p + \boldsymbol{g} + \frac{1}{\alpha_l} \nabla \cdot (\alpha_l \boldsymbol{\sigma}_l) + \frac{1}{\alpha_l \rho_l^0} \Big(\boldsymbol{F}_{lg} + \boldsymbol{F}_l \Big). \tag{22}$$

Using Equations (19) and (20) coherently with the artificial compressibility approach, $\nabla \cdot \boldsymbol{u}_g = S_g + O(h^2)$ and $\nabla \cdot \boldsymbol{u}_l = S_l + O(h^2)$. Let us assume that $\nabla \cdot \boldsymbol{u}_g \approx S_g$ and $\nabla \cdot \boldsymbol{u}_l \approx S_l$, which can be used to simplify the previous equations. For the dispersed phase, this yields

$$\frac{\partial \boldsymbol{u}_g}{\partial t} + \nabla \cdot (\boldsymbol{u}_g \boldsymbol{u}_g) = -\nabla \left(\frac{p}{\rho_g^0} \right) + \nabla \cdot \boldsymbol{\sigma}_g + \boldsymbol{G}_g, \tag{23}$$

where

$$\boldsymbol{G}_g \equiv S_g \boldsymbol{u}_g + \frac{1}{\alpha_g} \boldsymbol{\sigma}_g \cdot \nabla \alpha_g + \boldsymbol{g} + \frac{1}{\alpha_g \rho_g^0} (\boldsymbol{F}_{gl} + \boldsymbol{F}_g). \tag{24}$$

Similarly, it holds that

$$\frac{\partial \boldsymbol{u}_l}{\partial t} + \nabla \cdot (\boldsymbol{u}_l \boldsymbol{u}_l) = -\nabla \left(\frac{p}{\rho_l^0} \right) + \nabla \cdot \boldsymbol{\sigma}_l + \boldsymbol{G}_l, \tag{25}$$

where

$$\boldsymbol{G}_l \equiv S_l \boldsymbol{u}_l + \frac{1}{\alpha_l} \boldsymbol{\sigma}_l \cdot \nabla \alpha_l + \boldsymbol{g} + \frac{1}{\alpha_l \rho_l^0} (\boldsymbol{F}_{lg} + \boldsymbol{F}_l). \tag{26}$$

For a given dispersed-phase volume fraction α_g, Equations (19) and (23) may be target equations for an LBM scheme for the dispersed phase. Similarly, Equations (20) and (25) may be target equations for an LBM scheme for the liquid phase. However, it is important to highlight the following peculiarity: Equations (19) and (20) are driven by the same pressure time derivative because $\partial_t \epsilon_g = (\partial \epsilon_g / \partial p)\, \partial_t p$ and $\partial_t \epsilon_l = (\partial \epsilon_l / \partial p)\, \partial_t p$ (time pressure coupling), and moreover, Equations (23) and (25) are coupled by the same pressure gradient ∇p (space pressure coupling). It is important to remind at this point that the system of equations for both phases can be closed only by providing the additional equation for α_g (which is discussed in the following sections).

Before entering into the details of the LBM schemes, it is worth discussing first both the time and the space pressure coupling, mentioned above, in the LBM context. Let us recall first some preliminaries about LBM. Let us define ϵ as the zeroth order moment of the distribution function, namely, $\epsilon = \sum_q f(q)$, where q identifies the probability distribution function $f(q)$ corresponding to the lattice velocity $v_q \in \mathbb{L}$ in the velocity lattice/set $\mathbb{L}$, designed for the D-dimensional physical space and for the Q-dimensional velocity space. In the LBM framework, as will be clearer in the following, it is possible to impose an equation of state where the pressure is $p = c_s^2 \phi \rho \epsilon$, where $c_s \sqrt{\phi}$ is the (artificial) sound speed and ϕ is a tunable function (close to 1 for stability reasons). Let us now assume that Equations (19) and (23) are solved by an LBM scheme in terms of f_g, and for the sake of simplicity, let us assume $\phi_g = 1$. This allows one to identify the pressure p as

$$p \equiv c_s^2 \rho_g^0 \epsilon_g, \tag{27}$$

where $\epsilon_g = \sum_q f_g(q)$ or equivalently

$$\epsilon_g(p) = \frac{p}{c_s^2 \rho_g^0}, \tag{28}$$

which identifies the first function appearing in Equation (19).

Let us now assume that Equations (20) and (25) are solved by another LBM scheme in terms of f_l and that this time $\phi_l \neq 1$, which requires a specific equilibrium distribution function. Because Equations (23) and (25) are coupled, one has to ensure that the same pressure gradient (space pressure coupling) will drive also the evolution of this second LBM scheme for the liquid phase as well. The target Equation (25) involves the following term, which is ruled by the generalized equation of state, namely,

$$\nabla\left(\frac{p}{\rho_l^0}\right) = \nabla\left(\phi_l\, c_s^2 \epsilon_l\right), \tag{29}$$

where $\epsilon_l = \sum_l f_l(q)$. Substituting Equation (27) into the previous one yields

$$\nabla(\phi_l\, \epsilon_l) = \frac{1}{R}\, \nabla \epsilon_g, \tag{30}$$

where $R = \rho_l^0/\rho_g^0 > 1$. Solving the previous equations yields

$$\phi_l\, \epsilon_l = \frac{1}{R}\, \epsilon_g + k, \tag{31}$$

where k is an arbitrary constant. In order to choose the proper constant k, let us remember that $\epsilon_\varphi = \sum_q f_\varphi(q)$ is close to 1 in the incompressible limit and that one wants ϕ_l to be close to 1 for stability reasons. Assuming $k = -1/R + 1$ yields

$$\phi_l = \phi \equiv \frac{1}{\epsilon_l}\left[1 + \frac{1}{R}\,(\epsilon_g - 1)\right], \tag{32}$$

or equivalently

$$\epsilon_l(p) = \frac{1}{\phi}\left[1 + \frac{1}{R}\left(\frac{p}{c_s^2 \rho_g^0} - 1\right)\right], \tag{33}$$

which identifies the second function appearing in Equation (20). Clearly, $\epsilon_g(p)$ given by Equation (28) and $\epsilon_l(p)$ given by Equation (33) are driven by the same pressure dynamics (time pressure coupling), but they are different functions.

Let us recap the work performed so far. Equation (19) with source S_g and Equation (23) with force $\boldsymbol{G}_g$ can be solved by an LBM scheme in terms of f_g with a standard equation of state given by Equation (27). On the other hand, Equation (20) with source S_l and Equation (25) with force $\boldsymbol{G}_l$ can be solved by an LBM scheme in terms of f_l with a generalized equation of state, i.e., with ϕ_l given by Equation (32). In the next section, we will discuss the complete LBM schemes proposed for solving these equations.

2.1.1. LBM Schemes for Solving the Key Equations

Let us consider the following two LBM schemes for solving the artificially compressible continuity equations and the momentum equations for the two phases, formulated in terms of the corresponding particle distribution functions f_g and f_l, namely,

$$f_g(\hat{\boldsymbol{x}} + \boldsymbol{v}_q, \hat{t} + 1, q) = f_g(q) + \hat{\Omega}_g(q) + \hat{\Omega}_g^F(q), \tag{34}$$

$$f_l(\hat{\boldsymbol{x}} + \boldsymbol{v}_q, \hat{t} + 1, q) = f_l(q) + \hat{\Omega}_l(q) + \hat{\Omega}_l^F(q), \tag{35}$$

where all functions in the right-hand side are computed in $(\hat{\boldsymbol{x}}, \hat{t})$ locally, $\hat{\boldsymbol{x}}$ is the space coordinate divided by the distance λ between two neighboring lattices nodes (mean free path), $\hat{t}$ is the physical time divided by the time τ between two consecutive lattice collisions

(mean collision time), and $\boldsymbol{v}_q$ is the generic particle velocity divided by the average particle velocity $c = \lambda/\tau$ in the velocity lattice/set $\mathbb{L}$ designed for the D-dimensional physical space and for the Q-dimensional velocity space. These normalizations are consistent with the usual Boltzmann scaling adopted in the LBM numerical codes, which will be denoted by the hat notation $\hat{\cdot}$ from here on in order to simplify the discussion about how to tune the simulation parameters for solving the target equations. In the following, we discuss the terms $\hat{\Omega}_\varphi$ and $\hat{\Omega}^F_\varphi$.

The term $\hat{\Omega}_\varphi$ is the collisional operator, defined as

$$\hat{\Omega}_\varphi(q) = \omega_\varphi \Big[f^{eq}_\varphi(q) - f_\varphi(q) \Big], \tag{36}$$

where ω_φ is the relaxation frequency divided by the frequency $1/\tau$ for the phase φ, which drives the relaxation of the distribution function towards the equilibrium f^{eq}_φ. The equilibrium f^{eq}_φ is defined by means of the so-called incompressible equilibrium $f^e_I(\phi, \hat{\epsilon}, \hat{\boldsymbol{u}})$. This equilibrium is called incompressible because all terms of the equilibrium moments which depend on the velocity $\hat{\boldsymbol{u}}$ do not depend on the zeroth order moment $\hat{\epsilon}$, e.g., $\sum_q \boldsymbol{v}_q f^e_I(\phi, \hat{\epsilon}, \hat{\boldsymbol{u}}, q) = \hat{\boldsymbol{u}}$. The equilibrium distribution is different for the two phases even though it is formulated by means of the same functional form in the incompressible limit [1], namely,

$$f^{eq}_I(\phi, \hat{\epsilon}, \hat{\boldsymbol{u}}, q) = w_q \left[\hat{\epsilon}\, \eta_q(\phi) + \frac{\boldsymbol{v}_q \cdot \hat{\boldsymbol{u}}}{c_s^2} + \frac{(\boldsymbol{v}_q \boldsymbol{v}_q - c_s^2\, \boldsymbol{I}) : \hat{\boldsymbol{u}}\hat{\boldsymbol{u}}}{2c_s^4} \right], \tag{37}$$

where

$$\eta_q(\phi) = \delta_q \left[1 + \frac{(1 - w_0)}{w_0} (1 - \phi) \right] + (1 - \delta_q)\, \phi, \tag{38}$$

and

$$\delta_q = \begin{cases} 1, & \text{if } q = 0, \\ 0, & \text{if } q \neq 0. \end{cases} \tag{39}$$

The two equilibrium distributions differ from each other because $f^{eq}_g \equiv f^{eq}_I(1, \hat{\epsilon}_g, \hat{\boldsymbol{u}}_g)$ for the dispersed phase, while $f^{eq}_l \equiv f^{eq}_I(\phi, \hat{\epsilon}_l, \hat{\boldsymbol{u}}_l)$ for the liquid phase, where ϕ is given by Equation (32). The quantities $\hat{\epsilon}_\varphi$ and $\hat{\boldsymbol{u}}_\varphi$ can be computed by the function f_φ, which acts as an auxiliary vector, namely,

$$\hat{\epsilon}_\varphi = \sum_{q=0}^{Q-1} f_\varphi(q), \tag{40}$$

$$\hat{\boldsymbol{u}}_\varphi = \sum_{q=0}^{Q-1} \boldsymbol{v}_q f_\varphi(q). \tag{41}$$

It is important to highlight the difference between $\hat{\boldsymbol{u}}_\varphi$ and $\boldsymbol{u}_\varphi$ discussed in the previous sections. The former is the velocity field normalized by the characteristic flow speed $U = L/T$, where $L \gg \lambda$ is the characteristic length scale of the flow field and $T \gg \tau$ is the characteristic time scale of the flow field. Hence, the following relation holds: $\hat{\boldsymbol{u}}_\varphi = \boldsymbol{u}_\varphi\, (L/\lambda)\, (\tau/T)$. Hence it is not granted that the code output $\hat{\boldsymbol{u}}_\varphi$ converges to the physical solution $\boldsymbol{u}_\varphi$ when the mesh is refined, namely, when $\lambda/L \to 0$. The strategy for tuning the code parameters such that this convergence is enforced is called the scaling. Typically, the acoustic scaling, i.e., $\tau/T = \lambda/L$, and the diffusive scaling, i.e., $\tau/T = (\lambda/L)^2$, are the two most popular examples. Similar considerations hold also for $\hat{\epsilon}_\varphi$, but they require us to proceed with the formal asymptotic expansion, which is performed in the following.

The term $\hat{\Omega}_{\varphi}^{F}$ is the forcing operator and it can be defined by the same linearized functional form, namely,

$$\hat{\Omega}_{\varphi}^{F}(\psi_{\varphi}, \hat{S}_{\varphi}, \hat{\boldsymbol{G}}_{\varphi}, q) \equiv f_{L}^{eq}(\psi_{\varphi}, \hat{S}_{\varphi}, \hat{\boldsymbol{G}}_{\varphi}, q) = w_{q}\left[\hat{S}_{\varphi}\,\eta(\psi_{\varphi}) + \frac{\boldsymbol{v}_{q}\cdot\hat{\boldsymbol{G}}_{\varphi}}{c_{s}^{2}}\right]. \tag{42}$$

In general, ψ_{g} is different from ψ_{l} and is different from ϕ. Intuitively, $\hat{S}_{\varphi}$ and $\hat{\boldsymbol{G}}_{\varphi}$ are the expressions computed using $\hat{\boldsymbol{u}}_{\varphi}$ (Boltzmann scaling) in the code.

2.1.2. Asymptotic Analysis by the Equivalent Moment System

Equations (34) and (35) do not solve directly the target fluid equations, in the sense that the computed quantities, indicated with the hat notation $\hat{}$, do not converge automatically to the target fluid quantities if a proper scaling is not provided. The scaling is the set of rules used to update the input parameters once the mesh is refined. Hence, an asymptotic analysis is needed in order to find out under which conditions these equations at least approximate the target fluid equations. Here we use the asymptotic method based on the equivalent moment system [17], inspired by the moment method, which was first introduced to gas kinetic theory by H. Grad. The previous equations share the same structure, namely,

$$f_{\varphi}(\hat{\boldsymbol{x}} + \boldsymbol{v}_{q}, \hat{t} + 1, q) = f_{\varphi}(q) + \hat{\Omega}_{\varphi}(q) + \hat{\Omega}_{\varphi}^{F}(q). \tag{43}$$

Let us apply a Taylor expansion, namely,

$$f_{\varphi}(\hat{\boldsymbol{x}} + \boldsymbol{v}_{q}, \hat{t} + 1, q) - f_{\varphi}(\hat{\boldsymbol{x}}, \hat{t}, q) = \frac{\partial f_{\varphi}}{\partial \hat{t}} + \boldsymbol{v}_{q}\cdot\hat{\nabla} f_{\varphi} + \frac{1}{2}\,(\boldsymbol{v}_{q}\cdot\hat{\nabla})^{2} f_{\varphi} + \frac{1}{2}\,\frac{\partial^{2} f_{\varphi}}{\partial \hat{t}^{2}} + \ldots. \tag{44}$$

In the previous equation, the unit of the space coordinate is the distance λ between two neighboring lattice nodes (mean free path) and the unit of the time evolution is the time τ between two consecutive lattice collisions (mean collision time). Obviously, they are not appropriate as the characteristic scales for the flow field in the continuum limit, namely, L and T. Let us introduce the parameter $\varepsilon = \lambda/L \ll 1$ (mesh spacing) and let us reduce the corresponding time step quadratically, namely, $\tau/T = \varepsilon^{2}$ (diffusive scaling), which increases quadratically the number of time steps, as well as the computational time needed to solve the problem. By using these assumptions, the relations $\hat{t} = t/\varepsilon^{2}$ and $\hat{\boldsymbol{x}} = x/\varepsilon$ hold, which lead to

$$f_{\varphi}(\hat{\boldsymbol{x}} + \boldsymbol{v}_{q}, \hat{t} + 1, q) - f_{\varphi}(\hat{\boldsymbol{x}}, \hat{t}, q) = \varepsilon\,\boldsymbol{v}_{q}\cdot\nabla f_{\varphi} + \frac{\varepsilon^{2}}{2}\,(\boldsymbol{v}_{q}\cdot\nabla)^{2} f_{\varphi} + \varepsilon^{2}\,\frac{\partial f_{\varphi}}{\partial t} + O(\varepsilon^{3}). \tag{45}$$

Neglecting terms $O(\varepsilon^{3})$ yields

$$\varepsilon^{2}\,\frac{\partial f_{\varphi}}{\partial t} + \varepsilon\,\boldsymbol{v}_{q}\cdot\nabla f_{\varphi} + \frac{\varepsilon^{2}}{2}\,(\boldsymbol{v}_{q}\cdot\nabla)^{2} f_{\varphi} = \hat{\Omega}_{\varphi}(q) + \hat{\Omega}_{\varphi}^{F}(q). \tag{46}$$

Taking the zeroth-order, first-order, and second-order moments of the previous equations yields

$$\varepsilon^{2}\,\frac{\partial \hat{\epsilon}_{\varphi}}{\partial t} + \varepsilon\,\nabla\cdot\hat{\boldsymbol{u}}_{\varphi} + \frac{\varepsilon^{2}}{2}\,\nabla\cdot\nabla\cdot\hat{\boldsymbol{\Pi}}_{\varphi} = \hat{S}_{\varphi}, \tag{47}$$

$$\varepsilon^{2}\,\frac{\partial \hat{\boldsymbol{u}}_{\varphi}}{\partial t} + \varepsilon\,\nabla\cdot\hat{\boldsymbol{\Pi}}_{\varphi} + \frac{\varepsilon^{2}}{2}\,\nabla\cdot\nabla\cdot\hat{\boldsymbol{\Phi}}_{\varphi} = \hat{\boldsymbol{G}}_{\varphi}, \tag{48}$$

$$\varepsilon^{2}\,\frac{\partial \hat{\boldsymbol{\Pi}}_{\varphi}}{\partial t} + \varepsilon\,\nabla\cdot\hat{\boldsymbol{\Phi}}_{\varphi} + \frac{\varepsilon^{2}}{2}\,\nabla\cdot\nabla\cdot\sum_{q}\boldsymbol{v}_{q}\boldsymbol{v}_{q}\boldsymbol{v}_{q}\boldsymbol{v}_{q} f_{\varphi}(q) = \omega_{\varphi}(\hat{\boldsymbol{\Pi}}_{\varphi}^{eq} - \hat{\boldsymbol{\Pi}}_{\varphi}) + \psi_{\varphi} c_{s}^{2}\,\hat{S}_{\varphi}\,\boldsymbol{I}, \tag{49}$$

where $\hat{\mathbf{\Pi}}_\varphi = \sum_{q=0}^{Q-1} \boldsymbol{v}_q \boldsymbol{v}_q f_\varphi(q)$ is the second-order tensor of the distribution function and $\hat{\mathbf{\Phi}}_\varphi = \sum_{q=0}^{Q-1} \boldsymbol{v}_q \boldsymbol{v}_q \boldsymbol{v}_q f_\varphi(q)$ is the third-order tensor. Moreover, by the definition given by Equation (37), it is possible to compute

$$\hat{\mathbf{\Pi}}_\varphi^{eq} = \hat{\phi}_\varphi c_s^2 \, \hat{\epsilon}_\varphi \, \boldsymbol{I} + \hat{\boldsymbol{u}}_\varphi \hat{\boldsymbol{u}}_\varphi, \tag{50}$$

and

$$(\hat{\mathbf{\Phi}}_\varphi^{eq})_{ijk} = c_s^2 \Big(\hat{u}_{\varphi\, i} \delta_{jk} + \hat{u}_{\varphi\, j} \delta_{ik} + \hat{u}_{\varphi\, k} \delta_{ij} \Big). \tag{51}$$

In order to analyze this system of equations, it is now time to understand the impact of the previous assumptions on the scaling of the moments, namely, how changing the mesh spacing impacts the numerical values of the lattice moments (in Boltzmann scaling) computed by the code. The relation $\hat{\boldsymbol{u}}_\varphi = \boldsymbol{u}_\varphi \, (L/\lambda) \, (\tau/T)$ becomes $\hat{\boldsymbol{u}}_\varphi = \varepsilon \, \boldsymbol{u}_\varphi$. Consequently, $\hat{S}_\varphi = \varepsilon^2 \, S_\varphi$ because the source is the divergence of a velocity vector combination. Recalling the definitions given by Equations (24) and (26), there are four terms in $\hat{\boldsymbol{G}}_\varphi$. According to the previous scaling, $\hat{S}_\varphi \hat{\boldsymbol{u}}_\varphi$ and $\hat{\sigma}_\varphi \cdot \hat{\nabla} \alpha_g$ are automatically $O(\varepsilon^3)$. Hence it makes sense to scale in the same way also the two remaining terms. Adopting $\hat{\boldsymbol{g}} = \varepsilon^3 \boldsymbol{g}$ means that if ε becomes half, then $\hat{\boldsymbol{g}}$ must become $1/2^3$ of the original value. Similarly, we adopt $\hat{\boldsymbol{F}}_{\varphi\overline{\varphi}} = \varepsilon^3 \boldsymbol{F}_{\varphi\overline{\varphi}}$, where $\overline{\varphi}$ is the other phase in relation with phase φ, and $\hat{\boldsymbol{F}}_\varphi = \varepsilon^3 \boldsymbol{F}_\varphi$. The first decision implies $\hat{K}_I = \varepsilon \, K_I$ because in Equation (6) there is already a quadratic dependence on some velocity. Putting together all these assumptions yields $\hat{\boldsymbol{G}}_\varphi = \varepsilon^3 \, \boldsymbol{G}_\varphi$, which is consistent with the fact that a force induces an acceleration, namely, $\hat{\boldsymbol{G}}_\varphi \propto \partial \hat{\boldsymbol{u}}_\varphi / \partial \hat{t} = \varepsilon^3 \, \partial \boldsymbol{u}_\varphi / \partial t$.

A system of moments can be truncated if we have some expectations about the high-order moments. Actually, on a discrete lattice, the system of moments is automatically truncated because of the limited number of independent degrees of freedoms (typically up to some components of the fourth-order moment) [17]. Let us now imagine the equation for the third-order moment $\hat{\mathbf{\Phi}}_\varphi$, which is very similar to the previous equations. On the left-hand side of this equation, there is a term which is proportional to ε^2 multiplied by the double divergence of the fifth-order moment, which is "odd" with regard to the power of $\boldsymbol{v}_q$, and hence it scales as $\hat{\boldsymbol{u}}_\varphi = \varepsilon \boldsymbol{u}_\varphi$: altogether this term on the left-hand side scales as $O(\varepsilon^3)$. On the right-hand side, we have $\hat{\mathbf{\Phi}}_\varphi^{eq} - \hat{\mathbf{\Phi}}_\varphi$ and the third-order moment of the forcing operator, which is proportional to $\varepsilon^3 \, \boldsymbol{G}_\varphi$ and hence again $O(\varepsilon^3)$. Putting the pieces together, the equation for the third-order moment looks like $\hat{\mathbf{\Phi}}_\varphi = \hat{\mathbf{\Phi}}_\varphi^{eq} + O(\varepsilon^3)$. It is clear from Equation (51) that $\hat{\mathbf{\Phi}}_\varphi^{eq} = \varepsilon \mathbf{\Phi}_\varphi^{eq}$ and consequently

$$\mathbf{\Phi}_\varphi = \mathbf{\Phi}_\varphi^{eq} + O(\varepsilon^2). \tag{52}$$

Substituting these considerations in Equation (48) yields

$$\varepsilon^3 \, \frac{\partial \boldsymbol{u}_\varphi}{\partial t} + \varepsilon \, \nabla \cdot \hat{\mathbf{\Pi}}_\varphi + \frac{\varepsilon^3}{2} \, \nabla \cdot \nabla \cdot \mathbf{\Phi}_\varphi^{eq} = \varepsilon^3 \, \boldsymbol{G}_\varphi + O(\varepsilon^5), \tag{53}$$

which implies $\nabla \cdot \hat{\mathbf{\Pi}}_\varphi = \varepsilon^2 \, \nabla \cdot \mathbf{\Pi}_\varphi$. Consequently,

$$\frac{\partial \boldsymbol{u}_\varphi}{\partial t} + \nabla \cdot \mathbf{\Pi}_\varphi + \frac{1}{2} \, \nabla \cdot \nabla \cdot \mathbf{\Phi}_\varphi^{eq} = \boldsymbol{G}_\varphi + O(\varepsilon^2). \tag{54}$$

Similarly, the equation for the second-order moment looks like $\hat{\mathbf{\Pi}}_\varphi = \hat{\mathbf{\Pi}}_\varphi^{eq} + O(\varepsilon^2) = \hat{\phi}_\varphi c_s^2 \, \hat{\epsilon}_\varphi \, \boldsymbol{I} + O(\varepsilon^2)$. Using again the last relation implies $\nabla \cdot \hat{\mathbf{\Pi}}_\varphi = \varepsilon^2 \, \nabla \cdot \mathbf{\Pi}_\varphi = O(\varepsilon^2)$ and then $\nabla(\hat{\phi}_\varphi c_s^2 \, \hat{\epsilon}_\varphi) = O(\varepsilon^2)$ and equivalently $\nabla(\hat{\phi}_\varphi \hat{\epsilon}_\varphi) = O(\varepsilon^2)$. Moreover, let us assume that in some portion of the domain boundary (typically the outflow), namely, for $x \in \Omega_B$, the following boundary conditions hold:

$\hat{\epsilon}_g|_B = 1$, $\hat{\epsilon}_l|_B = 1$ and consequently $\hat{\phi}|_B = 1$. These boundary conditions allow one to integrate the previous relation, which yields

$$\hat{\epsilon}_g = 1 + \varepsilon^2 \epsilon_g, \tag{55}$$

$$\hat{\phi}\,\hat{\epsilon}_l = 1 + (1/R)\left(\hat{\epsilon}_g - 1\right) = 1 + \varepsilon^2 \epsilon_g / R. \tag{56}$$

Assuming $\phi_g = 1$ and $\phi_l = \left(\epsilon_g/\epsilon_l\right)/R$, the relations given by Equations (55) and (56) can be expressed by one unique formula, namely,

$$\hat{\phi}_\varphi \hat{\epsilon}_\varphi = 1 + \varepsilon^2 \phi_\varphi \epsilon_\varphi. \tag{57}$$

Concerning $\hat{\epsilon}_l$, introducing the relation $\nabla \cdot \hat{\mathbf{\Pi}}_l = O(\varepsilon^2)$ into Equation (47) implies that $\partial\hat{\epsilon}_l/\partial t = O(1)$ or $\partial\hat{\epsilon}_l/\partial t = O(\varepsilon^2)$. Let us focus on the second case by assuming that the boundary conditions change smoothly in time. Taking into account the boundary condition $\hat{\epsilon}_l|_B = 1$ and integrating the relation $\partial\hat{\epsilon}_l/\partial t = O(\varepsilon^2)$ yields

$$\hat{\epsilon}_l = 1 + \varepsilon^2 \epsilon_l, \tag{58}$$

and consequently

$$\hat{\phi} = \frac{1 + \varepsilon^2 \epsilon_g/R}{1 + \varepsilon^2 \epsilon_l} = 1 + \varepsilon^2 \left(\epsilon_g/R - \epsilon_l\right) + O(\varepsilon^4). \tag{59}$$

Introducing Equations (55), (57) and (58) into Equation (47) yields

$$\nabla \cdot \boldsymbol{u}_\varphi = S_\varphi + O(\varepsilon^2). \tag{60}$$

Equations (54) and (60) prove that the proposed scaling converges asymptotically with the second order of convergence to some equations, but we still have to prove that they coincide with the target equations. First of all, we search for an approximation of $\mathbf{\Pi}_\varphi$. Recalling Equation (49), taking into account that the term with a sequence of divergences of the fourth-order moment scales as $\varepsilon^2 \nabla(\hat{\phi}_\varphi \hat{\epsilon}_\varphi) = O(\varepsilon^4)$, because of Equation (57) and recalling Equation (60), it follows that

$$\varepsilon^2 \nabla \cdot \mathbf{\Phi}_\varphi^{eq} = \omega_\varphi(\hat{\mathbf{\Pi}}_\varphi^{eq} - \hat{\mathbf{\Pi}}_\varphi) + \varepsilon^2 \psi_\varphi c_s^2 \nabla \cdot \boldsymbol{u}_\varphi \boldsymbol{I} + O(\varepsilon^4). \tag{61}$$

Recalling that $\nabla \cdot \hat{\mathbf{\Pi}}_\varphi = \varepsilon^2 \nabla \cdot \mathbf{\Pi}_\varphi$ yields

$$\nabla \cdot \mathbf{\Pi}_\varphi = \nabla \cdot \mathbf{\Pi}_\varphi^{eq} - \nabla \cdot \left[\frac{1}{\omega_\varphi} \nabla \cdot \mathbf{\Phi}_\varphi^{eq} - \frac{c_s^2 \psi_\varphi}{\omega_\varphi} \nabla \cdot \boldsymbol{u}_\varphi \boldsymbol{I}\right] + O(\varepsilon^2). \tag{62}$$

Substituting Equation (62) into Equation (54) yields

$$\frac{\partial \boldsymbol{u}_\varphi}{\partial t} + \nabla \cdot \mathbf{\Pi}_\varphi^{eq} = \nabla \cdot \left[\left(\frac{1}{\omega_\varphi} - \frac{1}{2}\right) \nabla \cdot \mathbf{\Phi}_\varphi^{eq} - \frac{c_s^2 \psi_\varphi}{\omega_\varphi} \nabla \cdot \boldsymbol{u}_\varphi \boldsymbol{I}\right] + \boldsymbol{G}_\varphi + O(\varepsilon^2). \tag{63}$$

From the definition given by Equation (51), it is possible to prove that

$$\nabla \cdot \mathbf{\Phi}_\varphi^{eq} = c_s^2 \left(\nabla \boldsymbol{u}_\varphi + \nabla \boldsymbol{u}_\varphi^T + \nabla \cdot \boldsymbol{u}_\varphi \boldsymbol{I}\right). \tag{64}$$

Let us add another ingredient by defining the kinematic viscosity as

$$\nu_\varphi = c_s^2\left(\frac{1}{\omega_\varphi} - \frac{1}{2}\right), \tag{65}$$

and by choosing the tunable parameter ψ_φ in order to recover the right kinematic bulk viscosity ξ_φ as

$$\psi_\varphi = \frac{\omega_\varphi}{c_s^2}\left(\frac{5}{3}\,\nu_\varphi - \xi_\varphi\right). \tag{66}$$

It is important to note that both previous viscosity coefficients are normalized by the characteristic scales for the flow field in the continuum limit, namely, by L^2/T (in other words, the viscosity coefficients are the inverses of the corresponding Reynolds numbers). Substituting the previous expression into Equation (63) yields

$$\frac{\partial \boldsymbol{u}_\varphi}{\partial t} + \nabla\cdot\boldsymbol{\Pi}_\varphi^{eq} = \nabla\cdot\boldsymbol{\sigma}_\varphi + \boldsymbol{G}_\varphi + O(\varepsilon^2). \tag{67}$$

Clearly, Equations (60) and (67) approach the solution of the target equations in the asymptotic limit of $\varepsilon \to 0$ with second-order accuracy, thanks to the adopted diffusive scaling.

These LBM schemes are not only useful to recover an approximated solution of the target equations but they can also be used to compute relevant quantities which will be used in the following. For example, it is possible to compute the viscous stress tensor per unit volume without further computational efforts. Comparing Equations (63) and (67), it is possible to derive

$$\boldsymbol{\sigma}_\varphi = \left(\frac{1}{\omega_\varphi} - \frac{1}{2}\right)\nabla\cdot\boldsymbol{\Phi}_\varphi^{eq} - \frac{c_s^2\psi_\varphi}{\omega_\varphi}\,\nabla\cdot\boldsymbol{u}_\varphi\,\boldsymbol{I} + O(\varepsilon^2). \tag{68}$$

From the numerical point of view, it is possible to compute $\nabla\cdot\boldsymbol{\Phi}_\varphi^{eq}$ by Equation (61) and $\nabla\cdot\boldsymbol{u}_\varphi$ by Equation (60), namely,

$$\boldsymbol{\sigma}_\varphi = \left(\frac{1}{\omega_\varphi} - \frac{1}{2}\right)\left[\omega_\varphi\left(\boldsymbol{\Pi}_\varphi^{eq} - \boldsymbol{\Pi}_\varphi\right) + \psi_\varphi\, c_s^2\, S_\varphi\,\boldsymbol{I}\right] - \frac{c_s^2\psi_\varphi}{\omega_\varphi}\,S_\varphi\,\boldsymbol{I} + O(\varepsilon^2), \tag{69}$$

or equivalently,

$$\boldsymbol{\sigma}_\varphi = \left(\frac{1}{\omega_\varphi} - \frac{1}{2}\right)\omega_\varphi\left(\boldsymbol{\Pi}_\varphi^{eq} - \boldsymbol{\Pi}_\varphi\right) - \frac{c_s^2\psi_\varphi}{2}\,S_\varphi\,\boldsymbol{I} + O(\varepsilon^2), \tag{70}$$

which do not require us to compute explicitly additional space derivatives.

2.2. LBM Schemes for Solving the Volume Fraction Equation for Each Phase and Their Asymptotic Analysis

In addition of the two LBM schemes discussed above for the momentum equations, one needs to solve the equation for the volume fraction but must ensure that the volume fraction is bounded between zero and one, namely, $0 \le \alpha_\varphi \le 1$. There are many advanced numerical techniques to ensure this condition. One simple alternative may be that proposed by Spalding [18]. Essentially, the idea is to solve both Equations (8) and (10) for α_g and α_l, and then perform a proper renormalization. In the LBM context, let us consider the following two LBM schemes for solving the volume fraction equations given by Equations (8) and (10), formulated in terms of the corresponding particle distribution functions $f_{\alpha g}$ and $f_{\alpha l}$, namely,

$$f_{\alpha\varphi}(\hat{\boldsymbol{x}} + \boldsymbol{v}_q, \hat{t} + 1, q) = f_{\alpha\varphi}(q) + \hat{\omega}_{\alpha\varphi}\left[f_{\alpha\varphi}^{eq}(q) - f_{\alpha\varphi}\right], \tag{71}$$

where $\hat{\omega}_{\alpha\varphi}$ (for the volume phase fraction α_φ) is the relaxation frequency, divided by the frequency $1/\tau$, which drives the relaxation of the distribution function towards the equilibrium $f^{eq}_{\alpha\varphi}$. The fact that we report the relaxation frequency $\hat{\omega}_{\alpha\varphi}$ with the hat notation $\hat{}$ means that we allow the possibility of adjusting the relaxation frequency during the mesh refinement according the adopted scaling. The equilibrium distribution is formulated by means of the same functional form, namely, $f^{eq}_{\alpha\varphi} \equiv f^{eq}(\hat{\alpha}_\varphi, \hat{\boldsymbol{u}}_\varphi)$, where

$$f^{eq}(\hat{\alpha}, \hat{\boldsymbol{u}}, q) = w_q\, \hat{\alpha} \left[1 + \frac{\boldsymbol{v}_q \cdot \hat{\boldsymbol{u}}}{c_s^2} + \frac{(\boldsymbol{v}_q \boldsymbol{v}_q - c_s^2\, \boldsymbol{I}) : \hat{\boldsymbol{u}}\hat{\boldsymbol{u}}}{2c_s^4}\right], \tag{72}$$

and

$$\hat{\alpha}_\varphi = \sum_{q=0}^{Q-1} f_{\alpha\varphi}(q). \tag{73}$$

It is very important to highlight that the velocity field $\hat{\boldsymbol{u}}_\varphi$ is computed by the LBM schemes discussed in the previous section, and hence the first-order moment $\hat{\alpha}_\varphi \hat{\boldsymbol{v}}_\varphi$ of the distribution $f_{\alpha\varphi}$ is not conserved, namely,

$$\hat{\boldsymbol{u}}_\varphi = \sum_{q=0}^{Q-1} \boldsymbol{v}_q f_\varphi(q) \neq \sum_{q=0}^{Q-1} \boldsymbol{v}_q f_{\alpha\varphi}(q) = \hat{\alpha}_\varphi \hat{\boldsymbol{v}}_\varphi. \tag{74}$$

As discussed in the previous section, an asymptotic analysis is needed in order to find out under which conditions these equations at least approximate the target volume fraction equations. Again we use here the asymptotic method based on the equivalent moment system [17]. Let us apply a Taylor expansion and let us assume a diffusive scaling, namely, $\hat{t} = t/\varepsilon^2$ and $\hat{\boldsymbol{x}} = x/\varepsilon$, which leads to

$$f_{\alpha\varphi}(\hat{\boldsymbol{x}} + \boldsymbol{v}_q, \hat{t} + 1, q) - f_{\alpha\varphi}(\hat{\boldsymbol{x}}, \hat{t}, q) = \varepsilon\, \boldsymbol{v}_q \cdot \nabla f_{\alpha\varphi} + \frac{\varepsilon^2}{2} (\boldsymbol{v}_q \cdot \nabla)^2 f_{\alpha\varphi} + \varepsilon^2 \frac{\partial f_{\alpha\varphi}}{\partial t} + O(\varepsilon^3). \tag{75}$$

Neglecting terms $O(\varepsilon^3)$ yields

$$\varepsilon^2 \frac{\partial f_{\alpha\varphi}}{\partial t} + \varepsilon\, \boldsymbol{v}_q \cdot \nabla f_{\alpha\varphi} + \frac{\varepsilon^2}{2} (\boldsymbol{v}_q \cdot \nabla)^2 f_{\alpha\varphi} = \hat{\omega}_{\alpha\varphi} \left[f^{eq}_{\alpha\varphi}(q) - f_{\alpha\varphi}\right]. \tag{76}$$

Taking the zeroth-order, first-order, and second-order moments of the previous equations yields

$$\varepsilon^2 \frac{\partial \hat{\alpha}_\varphi}{\partial t} + \varepsilon \nabla \cdot \left(\hat{\alpha}_\varphi \hat{\boldsymbol{v}}_\varphi\right) + \frac{\varepsilon^2}{2} \nabla \cdot \nabla \cdot \hat{\boldsymbol{\Pi}}_{\alpha\varphi} = 0, \tag{77}$$

$$\varepsilon^2 \frac{\partial \left(\hat{\alpha}_\varphi \hat{\boldsymbol{v}}_\varphi\right)}{\partial t} + \varepsilon \nabla \cdot \hat{\boldsymbol{\Pi}}_{\alpha\varphi} + \frac{\varepsilon^2}{2} \nabla \cdot \nabla \cdot \hat{\boldsymbol{\Phi}}_{\alpha\varphi} = \hat{\omega}_{\alpha\varphi}\, \hat{\alpha}_\varphi \left(\hat{\boldsymbol{u}}_\varphi - \hat{\boldsymbol{v}}_\varphi\right), \tag{78}$$

$$\varepsilon^2 \frac{\partial \hat{\boldsymbol{\Pi}}_{\alpha\varphi}}{\partial t} + \varepsilon \nabla \cdot \hat{\boldsymbol{\Phi}}_{\alpha\varphi} + O(\varepsilon^2) = \hat{\omega}_{\alpha\varphi} (\hat{\boldsymbol{\Pi}}^{eq}_{\alpha\varphi} - \hat{\boldsymbol{\Pi}}_{\alpha\varphi}), \tag{79}$$

where $\hat{\boldsymbol{\Pi}}_{\alpha\varphi} = \sum_{q=0}^{Q-1} \boldsymbol{v}_q \boldsymbol{v}_q f_{\alpha\varphi}(q)$ is the second-order tensor of the distribution function and $\hat{\boldsymbol{\Phi}}_{\alpha\varphi} = \sum_{q=0}^{Q-1} \boldsymbol{v}_q \boldsymbol{v}_q \boldsymbol{v}_q f_{\alpha\varphi}(q)$ is the third-order tensor. Moreover, by the definition given by Equation (72), it is possible to compute

$$\hat{\boldsymbol{\Pi}}^{eq}_{\alpha\varphi} = c_s^2\, \hat{\alpha}_\varphi\, \boldsymbol{I} + \hat{\alpha}_\varphi\, \hat{\boldsymbol{u}}_\varphi \hat{\boldsymbol{u}}_\varphi, \tag{80}$$

and

$$(\hat{\boldsymbol{\Phi}}^{eq}_{\alpha\varphi})_{ijk} = c_s^2\, \hat{\alpha}_\varphi \left(\hat{u}_{\varphi\, i}\delta_{jk} + \hat{u}_{\varphi\, j}\delta_{ik} + \hat{u}_{\varphi\, k}\delta_{ij}\right). \tag{81}$$

As discussed in the previous section, in order to analyze this system of equations, one needs to understand the impact of the diffusive scaling on the scaling of the moments. Let us suppose we do not scale the volume fraction, namely, $\hat{\alpha}_\varphi = \alpha_\varphi$, which has an impact on all even moments of the distribution function. On the other hand, let us keep the same scaling as before for the odd moments, because they depend on the imposed velocity $\hat{\boldsymbol{u}}_\varphi = \varepsilon\, \boldsymbol{u}_\varphi$. This implies

$$\hat{\boldsymbol{\Pi}}^{eq}_{\alpha\varphi} = c_s^2\, \alpha_\varphi\, \boldsymbol{I} + \varepsilon^2 \alpha_\varphi\, \boldsymbol{u}_\varphi \boldsymbol{u}_\varphi, \tag{82}$$

and

$$(\hat{\boldsymbol{\Phi}}^{eq}_{\alpha\varphi})_{ijk} = \varepsilon\, c_s^2\, \alpha_\varphi \left(u_{\varphi\, i}\delta_{jk} + u_{\varphi\, j}\delta_{ik} + u_{\varphi\, k}\delta_{ij} \right). \tag{83}$$

A system of moments can be truncated if we have some expectations about the high-order moments. Let us now imagine the equation for the third-order moment $\hat{\boldsymbol{\Phi}}_{\alpha\varphi}$, which can be analyzed with very similar arguments discussed in the previous section. These arguments lead to conclude that the equation for the third-order moment looks like $\hat{\boldsymbol{\Phi}}_{\alpha\varphi} = \hat{\boldsymbol{\Phi}}^{eq}_{\alpha\varphi} + O(\varepsilon^3)$ and consequently $\hat{\boldsymbol{\Phi}}_{\alpha\varphi} = \varepsilon \boldsymbol{\Phi}^{eq}_{\alpha\varphi} + O(\varepsilon^3)$, because of Equation (83). Substituting the last result in Equation (79) and taking into account Equation (82) yields

$$O(\varepsilon^2) - \hat{\omega}_{\alpha\varphi}\varepsilon^2 \alpha_\varphi\, \boldsymbol{u}_\varphi \boldsymbol{u}_\varphi = \hat{\omega}_{\alpha\varphi} \left(c_s^2\, \alpha_\varphi\, \boldsymbol{I} - \hat{\boldsymbol{\Pi}}_{\alpha\varphi} \right), \tag{84}$$

or equivalently, $\hat{\boldsymbol{\Pi}}_{\alpha\varphi} = c_s^2\, \alpha_\varphi\, \boldsymbol{I} + O(\varepsilon^2)$. Substituting this result in Equation (77) yields

$$\varepsilon^2 \frac{\partial \hat{\alpha}_\varphi}{\partial t} + \varepsilon\, \nabla \cdot \left(\hat{\alpha}_\varphi \hat{\boldsymbol{v}}_\varphi \right) + \frac{\varepsilon^2}{2} c_s^2\, \nabla^2 \alpha_\varphi = O(\varepsilon^4), \tag{85}$$

which proves that $\hat{\boldsymbol{v}}_\varphi = \varepsilon \boldsymbol{v}_\varphi$. Substituting this result into Equation (78) yields

$$c_s^2\, \nabla \alpha_\varphi = \hat{\omega}_{\alpha\varphi}\, \alpha_\varphi \left(\boldsymbol{u}_\varphi - \boldsymbol{v}_\varphi \right) + O(\varepsilon^2). \tag{86}$$

Substituting the previous equation into Equation (85) yields

$$\frac{\partial \alpha_\varphi}{\partial t} + \nabla \cdot \left(\alpha_\varphi \boldsymbol{u}_\varphi \right) = \hat{\chi}_{\alpha\varphi}\, \nabla^2 \alpha_\varphi + O(\varepsilon^2), \tag{87}$$

where

$$\hat{\chi}_{\alpha\varphi} = c_s^2 \left(\frac{1}{\hat{\omega}_{\alpha\varphi}} - \frac{1}{2} \right). \tag{88}$$

Adopting the following scaling

$$\hat{\omega}_{\alpha\varphi} = \frac{2}{1 + 2\, \varepsilon^2\, \chi_{\alpha\varphi} / c_s^2}, \tag{89}$$

implies $\hat{\chi}_{\alpha\varphi} = \varepsilon^2\, \chi_{\alpha\varphi}$ and consequently

$$\frac{\partial \alpha_\varphi}{\partial t} + \nabla \cdot \left(\alpha_\varphi \boldsymbol{u}_\varphi \right) = O(\varepsilon^2). \tag{90}$$

Clearly, the previous equation approaches the solution of the target equations given by Equations (8) and (10) in the asymptotic limit of $\varepsilon \to 0$ with second-order accuracy, thanks to the adopted diffusive scaling.

These LBM schemes are not only useful to recover an approximated solution of the target equations but they can also be used to compute other relevant quantities. For example, it is possible to compute the volume fraction gradient without further computational efforts using Equation (86), namely,

$$\frac{1}{\alpha_\varphi}\nabla\alpha_\varphi = \frac{\hat{\omega}_{\alpha\varphi}}{c_s^2}\left(\boldsymbol{u}_\varphi - \boldsymbol{v}_\varphi\right) + O(\varepsilon^2). \tag{91}$$

In particular, Equations (70) and (91) can be used to compute the second term of the force $\boldsymbol{G}_g$ given by Equation (24) as well as the second term of the force $\boldsymbol{G}_l$ given by Equation (26), namely,

$$\frac{1}{\alpha_\varphi}\,\boldsymbol{\sigma}_\varphi \cdot \nabla\alpha_\varphi = \frac{\hat{\omega}_{\alpha\varphi}}{c_s^2}\left[\left(1-\frac{\omega_\varphi}{2}\right)\left(\boldsymbol{\Pi}_\varphi^{eq} - \boldsymbol{\Pi}_\varphi\right) - \frac{c_s^2\psi_\varphi}{2}\,S_\varphi\,\boldsymbol{I}\right]\cdot\left(\boldsymbol{u}_\varphi - \boldsymbol{v}_\varphi\right). \tag{92}$$

One last remark is about the need for ensuring that volume fraction α_φ is bounded between zero and one, namely, $0 \leq \alpha_\varphi \leq 1$. Among many available numerical techniques to ensure this condition, we follow here that proposed by Spalding [18]. Essentially, the idea is to solve both Equations (8) and (10) by the discussed LBM schemes for $f_{\alpha\varphi}$, compute the raw volume fraction $\alpha_\varphi = \sum_{q=0}^{Q-1} f_{\alpha\varphi}(q)$, and then perform the following renormalization at the beginning of every collision and streaming cycle:

$$\alpha_\varphi^B \equiv \frac{B(\alpha_\varphi)}{B(\alpha_g) + B(\alpha_l)}, \tag{93}$$

where $B(x) = x\,H(x)\,H(1-x) + H(x-1)$ and $H(x)$ is the Heaviside step function. Of course, $\alpha_l^B = 1 - \alpha_g^B$.

Similarly, we propose a second correction for ensuring that $\nabla\alpha_g = -\nabla\alpha_l$, when using Equation (91). Let us suppose that the vector $\boldsymbol{\delta} = \nabla\alpha_g + \nabla\alpha_l$ has a non-zero modulus, i.e., $\|\boldsymbol{\delta}\| \neq 0$. In this case, the second Spalding-like correction is

$$\nabla\alpha_\varphi^G \equiv \nabla\alpha_\varphi - \boldsymbol{\delta}/2. \tag{94}$$

Of course, $\nabla\alpha_l^G + \nabla\alpha_g^G = 0$. In the following, we will drop the superscript B in Equation (93) as well as the superscript G in Equation (94) for the sake of simplicity and without risk of confusion.

2.3. LBM Schemes for Solving the Phase Continuity Source for Each Phase and Their Asymptotic Analysis

In addition to the two LBM schemes discussed above for the momentum equations and the two LBM schemes for the volume fractions, one needs to compute the phase continuity sources, namely, S_g in Equation (19) and S_l in Equation (20). Let us consider the following two schemes for computing the phase continuity sources, formulated in terms of the corresponding particle distribution functions $f_{\beta g}$ and $f_{\beta l}$, namely,

$$f_{\beta\varphi}(\hat{\boldsymbol{x}} + \boldsymbol{v}_q, \hat{t}+1, q) = f_{\beta\varphi}^{eq}(\hat{\boldsymbol{x}}, \hat{t}, q), \tag{95}$$

where $f_{\beta\varphi}^{eq} \equiv f^{eq}(\hat{\beta}_\varphi, \hat{\boldsymbol{c}}_\varphi)$ and f^{eq} is the functional form given by Equation (72). The zero-order moment $\hat{\beta}_\varphi$ is given by

$$\hat{\beta}_\varphi = \sum_{q=0}^{Q-1} f_{\beta\varphi}(q) = \sum_{q=0}^{Q-1} f_{\beta\varphi}^{eq}(q), \tag{96}$$

and the imposed velocity field is given by

$$\hat{\boldsymbol{c}}_\varphi = \frac{1}{\hat{\beta}_\varphi}\,\hat{\alpha}_{\overline{\varphi}}\left(\hat{\boldsymbol{u}}_\varphi - \hat{\boldsymbol{u}}_{\overline{\varphi}}\right). \tag{97}$$

The velocity field $\hat{\boldsymbol{c}}_\varphi$ is computed by the LBM schemes discussed in the previous sections and, by definition, the following relation holds:

$$\sum_{q=0}^{Q-1} \boldsymbol{v}_q f^{eq}_{\alpha\varphi}(q) = \hat{\beta}_\varphi \hat{\boldsymbol{c}}_\varphi = \hat{\alpha}_{\overline{\varphi}} \left(\hat{\boldsymbol{u}}_\varphi - \hat{\boldsymbol{u}}_{\overline{\varphi}} \right). \tag{98}$$

It is worth noting that the continuity sources which we are trying to compute are related to the fluxes $\hat{\beta}_\varphi \hat{\boldsymbol{c}}_\varphi$, namely, $\hat{S}_\varphi = \hat{\nabla} \cdot (\hat{\beta}_\varphi \hat{\boldsymbol{c}}_\varphi) = \hat{\nabla} \cdot \left[\hat{\alpha}_{\overline{\varphi}} \left(\hat{\boldsymbol{u}}_\varphi - \hat{\boldsymbol{u}}_{\overline{\varphi}} \right) \right]$.

Equation (95) is the lattice kinetic scheme (LKS) [19] and is a particular case of the link-wise artificial compressibility method (link-wise ACM) [20]. As in LBM schemes, a link-wise ACM scheme also proceeds by a sequence of collision and streaming cycles. Equation (95) assumes collision first and then streaming, but the scheme can also be rationalized by inverting this order, namely,

$$f_{\beta\varphi}(\hat{\boldsymbol{x}}, \hat{t} + 1, q) = f^{eq}_{\beta\varphi}(\hat{\boldsymbol{x}} - \boldsymbol{v}_q, \hat{t}, q). \tag{99}$$

Let us apply the Taylor expansion to the previous expression, namely,

$$f_{\beta\varphi}(\hat{\boldsymbol{x}}, \hat{t} + 1, q) - f^{eq}_{\beta\varphi}(\hat{\boldsymbol{x}}, \hat{t}, q) = -\varepsilon \, \boldsymbol{v}_q \cdot \nabla f^{eq}_{\beta\varphi} + \frac{\varepsilon^2}{2} \left(\boldsymbol{v}_q \cdot \nabla \right)^2 f^{eq}_{\beta\varphi} - \frac{\varepsilon^3}{6} \left(\boldsymbol{v}_q \cdot \nabla \right)^3 f^{eq}_{\beta\varphi} + O(\varepsilon^4). \tag{100}$$

Computing the zeroth-order moment of the previous expression and applying the scaling to the known quantities yields

$$\hat{\beta}_\varphi(\hat{\boldsymbol{x}}, \hat{t} + 1) - \hat{\beta}_\varphi(\hat{\boldsymbol{x}}, \hat{t}) = -\varepsilon^2 \, S_\varphi + \frac{\varepsilon^2}{2} \, \nabla \cdot \nabla \cdot \hat{\boldsymbol{\Pi}}^{eq}_{\beta\varphi} - \frac{\varepsilon^3}{6} \, \nabla \cdot \nabla \cdot \nabla \cdot \hat{\boldsymbol{\Phi}}^{eq}_{\beta\varphi} + O(\varepsilon^4), \tag{101}$$

where

$$\hat{\boldsymbol{\Pi}}^{eq}_{\beta\varphi} = \hat{\beta}_\varphi \left(c_s^2 \, \boldsymbol{I} + \varepsilon^2 \, \boldsymbol{u}_\varphi \boldsymbol{u}_\varphi \right) = \hat{\beta}_\varphi \, c_s^2 \, \boldsymbol{I} + O(\varepsilon^2), \tag{102}$$

and

$$(\hat{\boldsymbol{\Phi}}^{eq}_{\beta\varphi})_{ijk} = \varepsilon \, c_s^2 \, \hat{\beta}_\varphi \Big(u_{\varphi\, i} \delta_{jk} + u_{\varphi\, j} \delta_{ik} + u_{\varphi\, k} \delta_{ij} \Big). \tag{103}$$

Substituting the previous expressions into Equation (101) yields

$$\hat{\beta}_\varphi(\hat{\boldsymbol{x}}, \hat{t} + 1) - \hat{\beta}_\varphi(\hat{\boldsymbol{x}}, \hat{t}) = -\varepsilon^2 \, S_\varphi + \varepsilon^2 \, \frac{c_s^2}{2} \, \nabla^2 \hat{\beta}_\varphi(\hat{\boldsymbol{x}}, \hat{t}) + O(\varepsilon^4). \tag{104}$$

It is important to remind that $\hat{\beta}_\varphi$ are just auxiliary functions without physical interest, with the exception of their derivatives, as will be clarified in the following. Assuming $\hat{\beta}_\varphi(\hat{\boldsymbol{x}}, \hat{t}) = 1$ at the beginning of every time step yields

$$\hat{\beta}_\varphi(\hat{\boldsymbol{x}}, \hat{t} + 1) - 1 = -\varepsilon^2 \, S_\varphi + O(\varepsilon^4), \tag{105}$$

or equivalently,

$$\frac{\partial \hat{\beta}_\varphi}{\partial t} = -S_\varphi + O(\varepsilon^2). \tag{106}$$

In particular, Equation (105) can be reformulated as

$$\hat{S}_\varphi = 1 - \hat{\beta}_\varphi(\hat{\boldsymbol{x}}, \hat{t} + 1) + O(\varepsilon^4), \tag{107}$$

which is particularly useful in this context because it allows one to compute $\hat{S}_\varphi$ as coded into the LBM schemes, without explicit finite difference formulas.

3. Results: Numerical Validation

3.1. One-Dimensional Test Case

We consider a one-dimensional vertical tube with a generic two-phase flow as a test case to validate the proposed computational methodology. Although simplified, the one-dimensional vertical tube offers a foundational model for analyzing multiphase flow behavior in systems such as bubble column reactors, where gas–liquid interactions are key to reactor performance. Additionally, this configuration is pertinent for studying flow dynamics in natural circulation loops driven by gas injection, a mechanism critical in systems like molten salt nuclear reactors. This test case provides a controlled yet insightful context, enabling us to investigate flow patterns, pressure gradients, and phase interactions relevant to both chemical and nuclear reactor environments [21].

Let us suppose that the vertical tube is aligned with the axis identified by the unit vector $\boldsymbol{e}_x$, where $\|\boldsymbol{e}_x\| = 1$. In this case, the gravitational acceleration field is $\boldsymbol{g} = -g\,\boldsymbol{e}_x$, where g is the standard acceleration of gravity. It is sometimes convenient to remove the hydrostatic pressure from the pressure in order to make the buoyancy effect explicit by defining a new quantity

$$p' = p + \rho_l^0\, g\, x, \tag{108}$$

where x is the coordinate along the unit vector $\boldsymbol{e}_x$. Let us introduce the kinematic pressure p_k, defined as

$$p_k = \frac{p'}{\rho_g^0} = \frac{p + \rho_l^0\, g\, x}{\rho_g^0} = \frac{p}{\rho_g^0} + R\, g\, x, \tag{109}$$

and consequently

$$\frac{1}{R}\, p_k = \frac{p}{\rho_l^0} + g\, x. \tag{110}$$

The previous definitions lead to $-\partial_x(p/\rho_g^0) - g = -\partial_x p_k + (R-1)\, g$ for the dispersed phase and $-\partial_x(p/\rho_l^0) - g = -(1/R)\,\partial_x p_k$ for the liquid phase. Hence, using the kinematic pressure p_k, instead of the original pressure p, allows one to consider only one gravitational acceleration in the dispersed phase, i.e., the term $(R-1)\, g$, which is the buoyancy acceleration.

Reformulating the artificially compressible equations in terms of the kinematic pressure implies that Equation (28) must be redefined as

$$\epsilon_g(p) = \frac{p'}{c_s^2 \rho_g^0} = \frac{p_k}{c_s^2}, \tag{111}$$

without modifying the proposed methodology. As already pointed out, for Equation (19) to be valid also during the dynamics, the following condition must hold: $\partial_t \epsilon_g = O(h^2)$. This means that the first derivative in Equation (19) is not mesh-independent. Assuming $h \equiv \varepsilon$, it is possible to make explicit such dependence on the mesh spacing by setting $\partial_t \epsilon_g$ more precisely as $\partial_t \hat{\epsilon}_g$ in Equation (19), since $\partial_t \hat{\epsilon}_g = \varepsilon^2\, \partial_t \epsilon_g$ according to Equation (55). Recalling that $\hat{\epsilon}_g = \hat{p}_k / c_s^2$ where $\hat{p}_k = c_s^2 + \varepsilon^2\, p_k$, Equation (19), in the one-dimensional case, becomes

$$\frac{\varepsilon^2}{c_s^2}\,\frac{\partial p_k}{\partial t} + \frac{\partial u_g}{\partial x} = \frac{\partial}{\partial x}[\alpha_l (u_g - u_l)] = S_g. \tag{112}$$

Substituting the previous quantities into Equations (23)–(26) in the one-dimensional case yields

$$\frac{\partial u_g}{\partial t} + \frac{\partial}{\partial x}(u_g u_g) = -\frac{\partial p_k}{\partial x} + \frac{\partial \sigma_g}{\partial x} + G_g, \tag{113}$$

where

$$G_g = S_g\, u_g + \frac{1}{\alpha_g}\,\sigma_g \frac{\partial \alpha_g}{\partial x} + (R-1)\, g + \frac{1}{\alpha_g \rho_g^0}(F_{gl} + F_g). \tag{114}$$

Similarly, it holds that

$$\frac{\partial u_l}{\partial t} + \frac{\partial}{\partial x}(u_l u_l) = -\frac{1}{R}\,\frac{\partial p_k}{\partial x} + \frac{\partial \sigma_l}{\partial x} + G_l, \tag{115}$$

where

$$G_l = S_l\, u_l + \frac{1}{\alpha_l}\,\sigma_l \frac{\partial \alpha_l}{\partial x} + \frac{1}{\alpha_l \rho_l^0}(F_{lg} + F_l). \tag{116}$$

In the one-dimensional case, the viscous stress tensor per unit volume defined by Equation (7), assuming $\xi_\varphi = -(1/3)\,\nu_\varphi$ (which is a clear indication of the degeneracy of the one-dimensional case because it implies negative kinematic bulk viscosity), becomes

$$\sigma_\varphi = \left(\frac{4}{3}\,\nu_\varphi + \xi_\varphi\right)\frac{\partial u_\varphi}{\partial x} = \nu_\varphi\,\frac{\partial u_\varphi}{\partial x}. \tag{117}$$

Concerning the interphase momentum exchange, in the one-dimensional case, Equation (6) becomes

$$F_{gl} = \rho_g^0\, K_I\, |u_l - u_g|(u_l - u_g) = -F_{lg}. \tag{118}$$

Concerning the momentum exchange with the wall, which cannot be properly modeled in the one-dimensional case because we have no curvature of the velocity field in axes other than $\boldsymbol{e}_x$, let us assume $F_g = 0$ and

$$F_l = \rho_g^0\, K_W\, |-u_l|(-u_l). \tag{119}$$

The analogy is clear between Equations (118) and (119), where the latter assumes zero wall velocity. Finally, the dispersed phase volume fraction equation given by Equation (10) in the one-dimensional case becomes

$$\frac{\partial \alpha_g}{\partial t} + \frac{\partial}{\partial x}(\alpha_g u_g) = 0, \tag{120}$$

The artificially compressible continuity equation given by Equation (112), together with Equations (113), (115) and (120), define a proper system of equations in the one-dimensional case for p_k, u_g, u_l, and α_g.

Before proceeding to the numerical results, let us search for a special analytical solution indicated by the notation $\bar{\cdot}$. Let us consider a flow regime where all flow quantities have zero gradients at steady state but the gradient of the kinematic pressure is constant, namely,

$$0 = -\partial_x \overline{p}_k + (R-1)\, g + \frac{K_I}{\overline{\alpha}_g}\,|\overline{u}_l - \overline{u}_g|(\overline{u}_l - \overline{u}_g), \tag{121}$$

$$0 = -\frac{1}{R}\,\partial_x \overline{p}_k + \frac{1}{\overline{\alpha}_l R}\left[K_I\,|\overline{u}_g - \overline{u}_l|(\overline{u}_g - \overline{u}_l) - K_W\,|\overline{u}_l|\overline{u}_l\right]. \tag{122}$$

Eliminating the kinematic pressure gradient from the previous equations yields

$$\frac{K_I}{\overline{\alpha}_g\,\overline{\alpha}_l}\,|\overline{u}_l - \overline{u}_g|(\overline{u}_l - \overline{u}_g) + \frac{K_W}{\overline{\alpha}_l}\,|\overline{u}_l|\overline{u}_l + (R-1)\, g = 0, \tag{123}$$

which can be interpreted as a condition for $\overline{u}_l$ as a function of $\overline{u}_g$ in this flow regime. In particular, let us search for $\overline{u}_g^0$ such that $\overline{u}_l^0 = 0$, namely,

$$(R-1)\,g = \frac{K_I}{\overline{\alpha}_g\,\overline{\alpha}_l}\,|\overline{u}_g^0|\overline{u}_g^0. \tag{124}$$

Substituting the previous expression back into the original equation yields

$$\frac{K_W}{\overline{\alpha}_l}\,|\overline{u}_l|\overline{u}_l = \frac{K_I}{\overline{\alpha}_g\,\overline{\alpha}_l}\,|\overline{u}_g - \overline{u}_l|\left(\overline{u}_g - \overline{u}_l\right) - \frac{K_I}{\overline{\alpha}_g\,\overline{\alpha}_l}\,|\overline{u}_g^0|\overline{u}_g^0. \tag{125}$$

Let us assume $\overline{u}_l < \overline{u}_g$ (which corresponds to the natural circulation loops driven by gas injection), then the previous condition becomes

$$r(\overline{u}_l)^2 = \left(\overline{u}_g - \overline{u}_l\right)^2 - (\overline{u}_g^0)^2, \tag{126}$$

where $r = (K_W/K_I)\,\overline{\alpha}_g$, which admits the following analytical solution:

$$\overline{u}_l = \frac{\overline{u}_g - \sqrt{(\overline{u}_g)^2 - (1-r)[(\overline{u}_g)^2 - (\overline{u}_g^0)^2]}}{1-r}. \tag{127}$$

It is worth discussing a couple of limiting cases. If $r = 0$, then the only possible coupling between the two phases is by the pressure gradient $\partial_x \overline{p}_k$: the relation becomes $\overline{u}_l = \overline{u}_g - \overline{u}_g^0$. If $r = 1$, then the coupling between the phases is ruled by the same proportionality that is found between the liquid and the wall; in this case, the relation becomes $\overline{u}_l = (\overline{u}_g - \overline{u}_g^0)\,(1 + \overline{u}_g^0/\overline{u}_g)/2$.

3.2. Very-Large-Density Ratios

In the limit of very-large-density ratios, namely, $R \gg 1$, some terms proportional to $1/R$ in the momentum equation of the liquid phase become very small and comparable with numerical errors, leading to numerical instabilities. Hence, special numerical ingredients are needed to overcome these instabilities. In this section, we analyze two of such ingredients.

In Reference [16], Ohwada suggested a simple but effective ingredient for the suppression of the spurious acoustic mode in the context of a single-phase artificial compressibility method. The basic idea here is the introduction of a similar dissipation term into the continuity equation for the gas, given by Equation (112), but we also automatically ensure the right asymptotic target equation in the context of the LBM. Using again the usual Boltzmann scaling adopted in the LBM numerical codes, this idea reads

$$\frac{\partial \hat{\epsilon}_g}{\partial \hat{t}} + \frac{\partial \hat{u}_g}{\partial \hat{x}} = \hat{S}_g - \gamma\,\hat{u}_g^2\,(\hat{\epsilon}_g - 1) \equiv \hat{S}_g', \tag{128}$$

where γ is a tunable constant. From a coding point of view, the previous ingredient is equivalent to using the source term $\hat{S}_g'$ instead of $\hat{S}_g$. The previous equation can be reformulated as

$$\left(\frac{\partial}{\partial \hat{t}} + \gamma\,\hat{u}_g^2\right)(\hat{\epsilon}_g - 1) + \frac{\partial \hat{u}_g}{\partial \hat{x}} = \hat{S}_g, \tag{129}$$

or equivalently

$$\frac{\varepsilon^2}{c_s^2}\left(\frac{\partial p_k}{\partial t} + \gamma\,u_g^2\,p_k\right) + \frac{\partial u_g}{\partial x} = S_g, \tag{130}$$

where it is clear that the term multiplied by γ acts as a dashpot in a simple mechanical oscillation system. Concerning the continuity equation for the liquid phase, some options are possible, which can be discussed by the following generic expression:

$$\frac{\partial \hat{\epsilon}_l}{\partial \hat{t}} + \frac{\partial \hat{u}_l}{\partial \hat{x}} = \hat{S}_l - \gamma\, \hat{u}_l^2\, \hat{\Gamma}_l \equiv \hat{S}'_l, \tag{131}$$

where $\hat{\Gamma}_l = \epsilon^2\, \Gamma_l$ is a function to be specified, or equivalently

$$\varepsilon^2 \left(\frac{\partial \epsilon_l}{\partial t} + \gamma\, u_l^2\, \Gamma_l \right) + \frac{\partial u_l}{\partial x} = S_l. \tag{132}$$

Different strategies are possible in choosing the function $\Gamma_l = \Gamma_l(p_k)$:

- The strategy most similar to the one adopted for the gas phase implies $\hat{\Gamma}_l = \hat{\epsilon}_l - 1$ or equivalently $\Gamma_l = \epsilon_l$ because of Equation (58), which would ensure a proper dashpot for the liquid phase as well. The problem is that ϵ_l is not simply related to p_k and therefore, for the finite difference engine (considered for comparison), this strategy would require solving an additional equation (precisely for ϵ_l).
- The second strategy is based on assuming that p_k is finally what really matters for both phases and therefore it assumes $\hat{\Gamma}_l = \hat{\epsilon}_g - 1$ for the liquid phase as well, which implies $\Gamma_l(p_k) = \epsilon_g = p_k / c_s^2$ because of Equation (55). This strategy definitively has the advantage of simplicity, but there is the drawback that different quantities are involved in the first two terms of Equation (132).
- The third strategy aims to improve the consistency by using liquid-phase quantities in computing $\hat{\Gamma}_l$ by including also $\hat{\phi}$, namely, $\hat{\Gamma}_l = \hat{\phi}\, \hat{\epsilon}_l - 1$, which implies $\Gamma_l(p_k) = \epsilon_g / R = (p_k / c_s^2) / R$ because of Equation (56). The advantage is that p_k / R is exactly the effective pressure acting on the liquid phase, which makes sense to use in the design of the liquid dashpot as well. On the other hand, the first two terms of Equation (132) are now formulated for ϵ_l and p_k, respectively, which is different from what happens in canonical modeling of a mechanical dashpot.

As far as the LBM is concerned, these three strategies are substantially equivalent. In this work, for the above reasons, we choose to adopt the third strategy.

The second ingredient for dealing with very-large-density ratios consists in updating the sources of the continuity equations only once in a while to let the fluid equations gradually accommodate the changes. In particular, it consists in updating them only at times $\hat{t}_\gamma$ which are multiples of the natural number n_γ, namely,

$$\hat{S}''_\varphi(\hat{x}, \hat{t}) = \hat{S}'_\varphi(\hat{x}, \hat{t}_\gamma), \tag{133}$$

where $\hat{t}_\gamma$ is the highest multiple of n_γ but still smaller than $\hat{t}$, namely, $\hat{t}_\gamma = r_\gamma\, n_\gamma$ where r_γ is another natural number and $r_\gamma\, n_\gamma \leq \hat{t} < (r_\gamma + 1)\, n_\gamma$. Of course n_γ is a free tunable parameter such that $n_\gamma \ll N_t$, where N_t is the total number of time steps. It is worth the effort to estimate the error introduced by the approximation given by Equation (133), which is a means of estimating the difference $|\hat{S}'_\varphi(\hat{x}, \hat{t}) - \hat{S}'_\varphi(\hat{x}, \hat{t}_\gamma)|$ or equivalently

$$\left| \hat{S}'_\varphi(\hat{x}, \hat{t}) - \hat{S}''_\varphi(\hat{x}, \hat{t}) \right| \approx (\hat{t} - \hat{t}_\gamma) \left| \frac{\partial \hat{S}'_\varphi}{\partial \hat{t}} \right| < n_\gamma \left| \frac{\partial \hat{S}'_\varphi}{\partial \hat{t}} \right| = O(\epsilon^4). \tag{134}$$

This proves that the approximation given by Equation (133) does not spoil the order of convergence of the proposed methodology. Even though this ingredient is simple and computationally very cheap, it ensures excellent stability; however, it smooths out the fastest dynamics (which is not compatible anyway with the incompressible limit).

3.3. Realistic Phenomenological Relation for the Drag Force

In order to better analyze the proposed methodology in practical applications, let us consider a more realistic phenomenological relation for the drag force. In particular, let us consider the model by Clift, Grace, and Weber (1978) [22]. In this model, the effective drag coefficient K_I in Equation (118) of the interphase momentum exchange can be expressed as $K_I = \kappa_I \, \alpha^2 \, \rho^0 / \rho_g^0$, where $\kappa_I = (3\, C_d)/(4\, d)$, C_d is the drag coefficient which depends on the relative Reynolds number, d is the characteristic size of the individual entity of the dispersed phase, $\alpha = \sqrt{\alpha_g \, \alpha_l}$, and $\rho^0 = \alpha_g \rho_g^0 + \alpha_l \rho_l^0$. Combining these definitions yields

$$K_I = \kappa_I \big(\alpha_g \, \alpha_l\big) \big(\alpha_g + \alpha_l \, R\big). \tag{135}$$

The model by Clift, Grace, and Weber (CGW) introduces a further dependence on the volume fraction into the effective drag coefficient. This dependence can be analyzed by setting $K_I = \kappa_I \, \Lambda$, where

$$\Lambda(\alpha_g) \equiv \big(\alpha_g \, \alpha_l\big) \big(\alpha_g + \alpha_l \, R\big). \tag{136}$$

The function $\Lambda(\alpha_g)$ goes to zero for both $\alpha_g = 0$ and $\alpha_g = 1$, with a positive maximum in between. In the limiting case that the gas is the dispersed phase (namely, that α_g is small enough), the function $\Lambda(\alpha_g)$ can be approximated by its tangent at $\alpha_g = 0$, namely, $\alpha^2 \approx \alpha_g$ and $\rho^0 \approx \rho_l^0$, which imply $\Lambda \approx \Lambda_0 = \alpha_g \, R$ [11,22]. In the case of large-density ratios, the tangent $\Lambda_0 = \alpha_g \, R$ becomes quite steep and this makes the CGW model stiff because it can significantly change the drag force for moderate changes of the void fraction. For the purpose of containing the numerical instability due to this stiffness, let us update the function $\Lambda(\alpha_g)$ only once in a while to let the equations gradually accommodate the changes

$$\Lambda'\big(\hat{x}, \hat{t}\big) = \Lambda\big(\hat{x}, \hat{t}_\gamma\big), \tag{137}$$

where again $\hat{t}_\gamma$ is the highest multiple of n_γ but still smaller than $\hat{t}$, namely, $\hat{t}_\gamma = r_\gamma \, n_\gamma$ where r_γ is a natural number and $r_\gamma \, n_\gamma \leq \hat{t} < (r_\gamma + 1) \, n_\gamma$. Again $n_\gamma \ll N_t$, where N_t is the total number of time steps. It is possible to prove (see previous section) that this approximation also does not spoil the order of convergence of the proposed methodology.

Another important ingredient for realistic simulations by the LBM is the concept of fluid-dynamic similarity. Because of its kinetic origin, the LBM can not deal directly with macroscopic geometric dimensions. Fortunately, the (kinematic) similarity allows one to apply the LBM to a geometrically similar setup (same shape but different sizes) with the same boundary conditions (e.g., no-slip, inlet velocity) and the same relevant dimensionless numbers. In other words, the velocity at any point in the LBM model flow is proportional by a constant scaling factor to the velocity at the same point in the real flow, while maintaining the same flow streamlines. It is important to identify this scaling factor, i.e., a proper way to convert flow quantities in physical units into lattice quantities coded by the Boltzmann scaling. Let us start with the main driving force, namely, the buoyancy force, driven by the gravitational acceleration g, and its LBM counterpart $\hat{g}$ in Boltzmann scaling, which must be selected to ensure numerical stability. The ratio between these two quantities defines the first constraint in terms of c/τ, namely,

$$c/\tau = \frac{g}{\hat{g}}. \tag{138}$$

A second constraint is given by a similar ratio between the drag factor in the CGW model, i.e., κ_I, and its LBM counterpart, i.e., $\hat{\kappa}_I$, which must ensure stable simulations. It is possible to express the last ratio in terms of $c\,\tau$, namely,

$$c\,\tau = \frac{\hat{\kappa}_I}{\kappa_I}. \tag{139}$$

The previous constraints allow us to compute both c and τ. In particular, the so-called lattice speed, which is the average fictitious particle velocity, can be computed as

$$c = \sqrt{(c/\tau)\,(c\tau)} = \sqrt{\frac{g}{\hat{g}}\,\frac{\hat{\kappa}_I}{\kappa_I}}. \tag{140}$$

This poses an upper limit to the maximum magnitude of the velocity field $u_\varphi(x,t)$ which can be simulated by stable simulations, because the latter relation requires that $\hat{u}_\varphi = u_\varphi/c$ is small enough, coherently with the incompressible limit. If these quantities are chosen consistently with regard to realistic setups, this is usually not problematic.

In particular, it is worth comparing the two most important driving forces of the momentum equation for the gas phase given by Equation (113), namely, the interphase momentum exchange force and the buoyancy force. For the purpose of doing so, the relevant fields, e.g., the velocity field $u_\varphi(x,t)$, can be characterized by a reference value identified by the dagger superscript, e.g., $u^\dagger_\varphi$, which is typically the inlet (known) value. Using these reference values, the ratio between the magnitudes of the exchange force and the buoyancy force for the gas phase can be expressed as M_{ex}, and according to the CGW model, it can be computed as

$$M_{ex} = \frac{\kappa_I\,\Lambda^\dagger \left|u^\dagger_g - u^\dagger_l\right|^2/\alpha^\dagger_g}{(R-1)\,g}. \tag{141}$$

where $\Lambda^\dagger = \Lambda(\alpha^\dagger_g) = \alpha^\dagger_g\,\alpha^\dagger_l(\alpha^\dagger_g + \alpha^\dagger_l\,R)$. Computing the same dimensionless number by LBM quantities in Boltzmann scaling yields

$$\hat{M}_{ex} = \frac{\hat{\kappa}_I\,\Lambda^\dagger \left|\hat{u}^\dagger_g - \hat{u}^\dagger_l\right|^2/\alpha^\dagger_g}{(R-1)\,\hat{g}} = (c/\tau)\,\frac{c\,\tau}{c^2}\,M_{ex} = M_{ex}, \tag{142}$$

which clearly proves that $\hat{M}_{ex} = M_{ex}$ and hence the fluid-dynamic similarity ensures the same ratio between the two main driving forces.

3.4. Finite Difference Solver for the Reference Solution

In this section, we develop an algorithm based on the finite difference (FD) method for solving the system of equations given by Equations (112), (113), (115) and (120) in order to obtain numerically a reference solution. The reference solution will be used to validate the LBM results. Hence, in order to simplify the comparisons between the results of the two methods, let us formulate both numerical engines in terms of $(\hat{x},\hat{t})$, where $\hat{x}$ is the space coordinate divided by the distance λ between two neighboring lattice nodes (mean free path) and $\hat{t}$ is the physical time divided by the time τ between two consecutive lattice collisions (mean collision time). Let us define a computational domain which is a multiple N_x of λ and let us search for a solution of the previous system of equations in the nodal points $\hat{x}_i \in \{1,2,\ldots,N_x\}$ where $1 \le i \le N_x$. Let us consider the generic quantity $\hat{u}(\hat{x})$ and let us call the nodal values as $\hat{u}(\hat{x}_i) = \hat{u}_i$. Let us store the nodal values in a vector $\{\hat{u}\} = \{\hat{u}_1,\hat{u}_2,\ldots,\hat{u}_{N_x}\}$ with elements $\{\hat{u}\}(i) = \hat{u}_i$ for $1 \le i \le N_x$. It is possible to include also boundary conditions (BCs) into this nomenclature. From the nodal vector of the generic quantity $\{\hat{u}\}$, by wrapping it with proper BCs, let us construct a larger nodal vector $\langle\hat{u}\rangle = \{\hat{u}_0,\hat{u}_1,\hat{u}_2,\ldots,\hat{u}_{N_x},\hat{u}_{N_x+1}\}$ with elements $\langle\hat{u}\rangle(i) = \hat{u}_i$ for $0 \le i \le (N_x+1)$, where $\hat{u}_0$ and $\hat{u}_{N_x+1}$ are some BCs which must be specified for the generic known (as a

function of the other nodal values). Let us introduce also the following combinatorial rule. Given two known vectors $\langle \hat{u} \rangle$ and $\langle \hat{v} \rangle$, let us assume that $(\langle \hat{u} \rangle \langle \hat{v} \rangle)(i) = \langle \hat{u} \rangle(i) \langle \hat{v} \rangle(i)$. The same combinatorial rule applies also to vectors $\{\cdot\}$, namely, $(\{\hat{u}\}\{\hat{v}\})(i) = \{\hat{u}\}(i)\{\hat{v}\}(i)$.

Moreover, it is possible to introduce the central difference operator $\hat{D}_x : \langle \hat{u} \rangle \to \{\hat{D}_x \hat{u}\}$ (contraction) defined as

$$(\hat{D}_x \langle \hat{u} \rangle)(i) = \{\hat{D}_x \hat{u}\}(i) = [\langle \hat{u} \rangle(i+1) - \langle \hat{u} \rangle(i-1)]/2, \tag{143}$$

where $1 \leq i \leq N_x$. Analogously, by applying differencing formulas in a recursive manner, it is possible to introduce $\hat{D}_x^2 : \langle \hat{u} \rangle \to \{\hat{D}_x^2 \hat{u}\}$ (contraction) defined as

$$(\hat{D}_x^2 \langle \hat{u} \rangle)(i) = \{\hat{D}_x^2 \hat{u}\}(i) = \langle \hat{u} \rangle(i+1) - 2\langle \hat{u} \rangle(i) + \langle \hat{u} \rangle(i-1), \tag{144}$$

where again $1 \leq i \leq N_x$. These are the essential ingredients of the FD engine.

The FD engine searches for a numerical solution of the following vectors: $\{\hat{p}_k\}$, $\{\hat{u}_g\}$, $\{\hat{u}_l\}$, and $\{\hat{\alpha}_g\}$. Multiplying Equation (112) by ε^2 and applying the code scaling yields

$$\frac{d\{\hat{p}_k\}}{d\hat{t}} = -c_s^2 \, \hat{D}_x \langle \hat{u}_g \rangle + c_s^2 \, \hat{D}_x \left[\langle \hat{\alpha}_l \rangle \left(\langle \hat{u}_g \rangle - \langle \hat{u}_l \rangle \right) \right]. \tag{145}$$

Multiplying Equations (113) and (115) by ε^3 and applying the code scaling yields

$$\frac{d\{\hat{u}_g\}}{d\hat{t}} = -\hat{D}_x(\langle \hat{u}_g \rangle \langle \hat{u}_g \rangle) - \hat{D}_x \langle \hat{p}_k \rangle + \nu_g \, \hat{D}_x^2 \langle \hat{u}_g \rangle + \{\hat{S}_g\} \, \{\hat{u}_g\} + \{\hat{\sigma}_g / \hat{\alpha}_g\} \hat{D}_x \langle \hat{\alpha}_g \rangle \ldots + (R-1) \, \hat{g} + \{\hat{K}_I / \hat{\alpha}_g\} \, |\{\hat{u}_l\} - \{\hat{u}_g\}| (\{\hat{u}_l\} - \{\hat{u}_g\}), \tag{146}$$

and

$$\frac{d\{\hat{u}_l\}}{d\hat{t}} = -\hat{D}_x(\langle \hat{u}_l \rangle \langle \hat{u}_l \rangle) - (1/R) \, \hat{D}_x \langle \hat{p}_k \rangle + \nu_l \, \hat{D}_x^2 \langle \hat{u}_l \rangle + \{\hat{S}_l\} \, \{\hat{u}_l\} + \{\hat{\sigma}_l / \hat{\alpha}_l\} \, \hat{D}_x \langle \hat{\alpha}_l \rangle \ldots + (1/R) \, \{\hat{K}_I / \hat{\alpha}_l\} \, |\{\hat{u}_g\} - \{\hat{u}_l\}| (\{\hat{u}_g\} - \{\hat{u}_l\}) + (1/R) \, \{\hat{K}_W / \hat{\alpha}_l\} \, |\{\hat{u}_l\}| \{\hat{u}_l\}, \tag{147}$$

where we assume $\varepsilon \, K_W = \hat{K}_W$ in analogy with the scaling $\hat{K}_I = \varepsilon K_I$ which was adopted in the asymptotic analysis of the LBM schemes. Finally, multiplying Equation (120) by ε^2 and applying the code scaling yields

$$\frac{d\{\hat{\alpha}_g\}}{d\hat{t}} = -\hat{D}_x(\langle \hat{\alpha}_g \rangle \langle \hat{u}_g \rangle). \tag{148}$$

Equations (145)–(148) define a system of ordinary differential equations (ODEs) for $\{\hat{p}_k\}$, $\{\hat{u}_g\}$, $\{\hat{u}_l\}$, and $\{\hat{\alpha}_g\}$, as far as proper boundary conditions are specified for computing the larger vector $\langle \hat{u} \rangle$ from the generic vector $\{\hat{u}\}$. For the kinematic pressure, this mapping is realized by the following wrapping:

$$\langle \hat{p}_k \rangle = \{2 \, \hat{p}_k(1) - \hat{p}_k(2), \hat{p}_k(1), \ldots, \hat{p}_k(N_x), 2 \, c_s^2 - \hat{p}_k(N_x)\}, \tag{149}$$

where $\hat{p}_k(0) = 2 \, \hat{p}_k(1) - \hat{p}_k(2)$ at the inlet is an extrapolation and $\hat{p}_k(N_x + 1) = 2 \, c_s^2 - \hat{p}_k(N_x)$ at the outlet is a Dirichlet condition corresponding to $\hat{p}_k(N_x + 1/2) = c_s^2$ (halfway). Other flow quantities have the following mappings:

$$\langle \hat{u}_g \rangle = \{2 \, \hat{u}_g^{IN} - \hat{u}_g(1), \hat{u}_g(1), \ldots, \hat{u}_g(N_x), 2 \, \hat{u}_g(N_x) - \hat{u}_g(N_x - 1)\}, \tag{150}$$

$$\langle \hat{u}_l \rangle = \{2 \, \hat{u}_l^{IN} - \hat{u}_l(1), \hat{u}_l(1), \ldots, \hat{u}_l(N_x), 2 \, \hat{u}_l(N_x) - \hat{u}_l(N_x - 1)\}, \tag{151}$$

$$\langle \hat{\alpha}_g \rangle = \{2\,\hat{\alpha}_g^{IN} - \hat{\alpha}_g(1), \hat{\alpha}_g(1), \ldots, \hat{\alpha}_g(N_x), 2\,\hat{\alpha}_g(N_x) - \hat{\alpha}_g(N_x - 1)\}, \tag{152}$$

where, for the generic quantity, $\hat{u}(0) = 2\,\hat{u}^{IN} - \hat{u}(1)$ at the inlet is a Dirichlet condition corresponding to $\hat{u}(1/2) = \hat{u}^{IN}$ (halfway) and $\hat{u}(N_x + 1) = 2\,\hat{u}(N_x) - \hat{u}(N_x - 1)$ at the outlet is an extrapolation. In order to be consistent with the incompressible limit, the generic inlet quantity $\hat{y}^{IN} \in \{\hat{\alpha}_g^{IN}, \hat{u}_g^{IN}, \hat{u}_l^{IN}\}$ involved in the previous formulas is progressively increased by the following function:

$$\hat{y}^{IN}(\hat{t}) = (\hat{y}^{max} - \hat{y}^{min}) \tanh(\hat{t}/n_t) + \hat{y}^{min}, \tag{153}$$

where $\hat{y}^{max}$ and $\hat{y}^{min}$ are quantity-specific values which depend on the considered test case. The parameter $n_t < N_t$, where N_t is the total number of time steps.

The above system of ODEs will be solved in Matlab® R2024b by means of *ode45* solver, which is a variable step solver (which means that it automatically chooses the value of the time stepping) and is based on an explicit Runge–Kutta (4,5) formula, the Dormand–Prince pair, namely, a combination of fourth- and fifth-order methods. During the iteration procedure, once every few time steps (e.g., 100 iterations), the solution vectors are smoothed in order to avoid the checkerboard instability by means of a Gaussian-weighted moving average filter (with a window containing 12 points). It is remarkable that LBM schemes do not need such additional smoothing because they are more robust against the checkerboard instability. The reason is due to the third term of the left-hand side of Equation (47) in the LBM asymptotic equations [16].

3.5. LBM Solver

Before discussing practical details of the LBM solver, it is worth realizing that the one-dimensional case is actually degenerate, and this may require some modifications in the formulas involving the transport coefficients. For example, recalling Equation (68) in this case yields

$$\sigma_\varphi = \nu_\varphi^{ef} \frac{\partial u_\varphi}{\partial x} + O(\varepsilon^2), \tag{154}$$

where

$$\nu_\varphi^{ef} = \left(\frac{1}{\omega_\varphi} - \frac{1}{2}\right) - \frac{c_s^2 \psi_\varphi}{\omega_\varphi}. \tag{155}$$

Using Equations (65) and (66) leads to an effective viscosity $\nu_\varphi^{ef} = \nu_\varphi\, d_\varphi$ where d_φ is a corrective factor for one-dimensional degeneracy, namely,

$$d_\varphi = \frac{4}{3} + \frac{\xi_\varphi}{\nu_\varphi}. \tag{156}$$

Hence, two strategies are possible for recovering the given kinematic viscosity are as follows:

- $\xi_\varphi = -(1/3)\,\nu_\varphi$, as reported in the previous section, and consequently $d_\varphi = 1$ (with $\psi_\varphi = 2\,\omega_\varphi \nu_\varphi / c_s^2$), which ensures the maximum consistency with the multidimensional cases but also some small oscillations during the transient dynamics (up to 10% in the tested cases) without impacting on the steady state solution;
- $\xi_\varphi = (5/3)\,\nu_\varphi$ and consequently $d_\varphi = 3$ (with $\psi_\varphi = 0$), which ensures the best performance also during the transient dynamics.

In the following, the second strategy is adopted. This has an impact also on the way one computes the stress in the one-dimensional case. Recalling Equation (61) in this case yields

$$\frac{\partial u_\varphi}{\partial x} = \omega_\varphi(\Pi_\varphi^{eq} - \Pi_\varphi) + \psi_\varphi c_s^2\, S_\varphi + O(\varepsilon^2), \tag{157}$$

and substituting it into Equation (154) yields

$$\sigma_\varphi = \nu_\varphi^{ef}\left[\omega_\varphi(\Pi_\varphi^{eq} - \Pi_\varphi) + \psi_\varphi c_s^2\, S_\varphi\right] + O(\varepsilon^2), \tag{158}$$

where the effective viscosity is used.

For solving the one-dimensional test case, let us adopt the standard implementation for the D1Q3 lattice [1], with the equilibrium distributions reported in Appendix A. Aside from the bulk implementation discussed so far, BCs in LBM are quite critical for stability and efficiency. The proposed BCs are those which showed the best compromise between stability and accuracy. They can be formulated by the two steps described below:

- STEP #1. First of all, the generic flow quantity $\hat{y} \in \{\hat{e}_g, \hat{e}_l, \hat{\alpha}_g, \hat{\alpha}_l, \hat{\beta}_g, \hat{\beta}_l, \hat{u}_g, \hat{u}_l, \hat{c}_g, \hat{c}_l\}$ at the boundary $\hat{x}_B$ (boundary) can be imposed (Dirichlet) or extrapolated. Concerning extrapolation, there are two cases: halfway (hw), where the computed generic quantity is intended at the position $\hat{x}_B$ midway between the last mesh node at $\hat{x}_E$ (edge) and the missing node outside the mesh, namely, $|\hat{x}_B - \hat{x}_E| = 1/2$, and full-way (fw), where the computed quantity is intended in the missing node, namely, $|\hat{x}_B - \hat{x}_E| = 1$. These two cases can be summarized as

$$|\hat{x}_B - \hat{x}_E| = \begin{cases} 1/2, & \text{if half-way (hw),} \\ 1, & \text{if full-way (fw).} \end{cases} \tag{159}$$

  The choice between these two options depends on the LBM boundary condition (see the following step). At the position $\hat{x}_B$, the generic flow quantity can be imposed or extrapolated

$$\hat{y}^{B(in)} = \begin{cases} \hat{y}^{IN}, & \text{if Dirichlet,} \\ (3/2)\,\hat{y}(1) - (1/2)\,\hat{y}(2), & \text{if EX(hw),} \\ 2\,\hat{y}(1) - \hat{y}(2), & \text{if EX(fw),} \end{cases} \tag{160}$$

  and

$$\hat{y}^{B(out)} = \begin{cases} \hat{y}^{OUT}, & \text{if Dirichlet,} \\ (3/2)\,\hat{y}(N_x) - (1/2)\,\hat{y}(N_x - 1), & \text{if EX(hw),} \\ 2\,\hat{y}(N_x) - \hat{y}(N_x - 1), & \text{if EX(fw).} \end{cases} \tag{161}$$

  The proposed BCs for the LBM framework are described in Table 1 for the inlet at $\hat{x} = 1$ and in Table 2 for the outlet at $\hat{x} = N_x$, respectively. For the sake of the following step, the generic flow quantities can be divided in two subgroups: generic densities $\hat{y}_\epsilon \in \{\hat{e}_g, \hat{e}_l, \hat{\alpha}_g, \hat{\alpha}_l, \hat{\beta}_g, \hat{\beta}_l\}$ and generic velocities $\hat{y}_u \in \{\hat{u}_g, \hat{u}_l, \hat{c}_g, \hat{c}_l\}$. All previous options are summarized in Table 1 for the inlet and in Table 2 for the outlet by using the same acronyms (see columns for generic density and for generic velocity).
- STEP #2. Secondly, once the flow quantities are imposed or extrapolated, the usual LBM techniques can be used to transfer the boundary condition at the position $\hat{x}_B$ to the generic distribution function $f_{*\varphi} \in \{f_g, f_l, f_{\alpha g}, f_{\alpha l}, f_{\beta g}, f_{\beta l}\}$ in the mesh node $\hat{x}_E$ during streaming. Let us consider some popular approaches: bounce-back (BB) rule, anti-bounce-back (ABB) rule, and equilibrium (EQ) rule [1]. It is worth noticing also that BB and ABB are formulated in terms of the equilibrium distribution, which must be coherent with the one used in the bulk, namely, incompressible equilibrium (I) $f_I^{eq}(\phi, \hat{e}, \hat{u})$ or standard equilibrium (S) $f_S^{eq}(\hat{\alpha}, \hat{u})$ (see Appendix A). Let us refer to the generic equilibrium distribution function as $f_Y^{eq}(\hat{y}_\epsilon, \hat{y}_u)$, which can coincide with either f_I^{eq} or f_S^{eq} depending on the considered distribution function. The BB, ABB, and

EQ rule for computing the incoming distribution function during the streaming can be expressed as

$$f_{*\varphi}(\hat{x}_E, \hat{t}+1, BB(q^*)) = \begin{cases} f^*_{*\varphi}(\hat{x}_E, \hat{t}, q^*) + \Delta_Y^{BB}(\hat{y}^B_\epsilon, \hat{y}^B_u, q^*), & \text{if BB(Y)}, \\ -f^*_{*\varphi}(\hat{x}_E, \hat{t}, q^*) + \Delta_Y^{ABB}(\hat{y}^B_\epsilon, \hat{y}^B_u, q^*), & \text{if ABB(Y)}, \\ f_Y^{eq}(\hat{y}^B_\epsilon, \hat{y}^B_u, BB(q^*)), & \text{if EQ(Y)}, \end{cases} \tag{162}$$

where $\hat{x}_E$ is the edge node (in the one-dimensional case, it can be either $\hat{x}_E = 1$ or $\hat{x}_E = N_x$), q^* is the identifier of the velocity v_{q^*} leaving the computational domain ($v_{q^*} = -1$ for $\hat{x}_E = 1$ and $v_{q^*} = +1$ for $\hat{x}_E = N_x$), $BB(q^*)$ is the opposite direction (bounce-back), and $f^*_{*\varphi}$ is the generic post-collision distribution function [1]. The operators Δ_Y^{BB} and Δ_Y^{ABB} can be constructed by means of the proper equilibrium for the considered distribution function as

$$\Delta_Y^{BB} = f_Y^{eq}(\hat{y}^B_\epsilon, -\hat{y}^B_u) - f_Y^{eq}(\hat{y}^B_\epsilon, \hat{y}^B_u), \tag{163}$$

$$\Delta_Y^{ABB} = f_Y^{eq}(\hat{y}^B_\epsilon, -\hat{y}^B_u) + f_Y^{eq}(\hat{y}^B_\epsilon, \hat{y}^B_u). \tag{164}$$

The rationale behind the reported choices is the following: The BB rule is typically used to transfer a Dirichlet condition for a generic velocity $\hat{y}_u$, the ABB rule to transfer a Dirichlet condition for a generic density $\hat{y}_\epsilon$ (and hence also for a generic pressure), while the EQ rule to transfer a Dirichlet condition for both a generic velocity and a generic density at the same time. The BB and the ABB rules impose conditions halfway, namely, at position $\hat{x}_B$ where $|\hat{x}_B - \hat{x}_E| = 1/2$, while the EQ rule streams the distribution function from a missing node located at the full-way distance, namely, from the position $\hat{x}_B$ where $|\hat{x}_B - \hat{x}_E| = 1$. See Equations (159)–(161) from which we started.

Table 1. BCs for LBM schemes: INLET ($\hat{x} = 1$). Acronyms stand for bounce back (BB); anti-bounce back (ABB); extrapolation (EX); and equilibrium (EQ). Uppercase letters in parentheses stand for incompressible equilibrium (I) and standard equilibrium (S). Lowercase letters in parentheses stand for halfway (hw) and full-way (fw).

$f_{*\varphi}$	**INLET ($\hat{x} = 1$): Compute $\hat{y}^{B(in)}$**		
	Type	**Generic Density** $\hat{y}^{B(in)}_\epsilon$	**Generic Velocity** $\hat{y}^{B(in)}_u$
f_g	BB(I)	$\hat{\epsilon}_g$: EX(hw)	$\hat{u}_g = \hat{u}_g^{IN}$
f_l	BB(I)	$\hat{\epsilon}_l$: EX(hw)	$\hat{u}_l = \hat{u}_l^{IN}$
$f_{\alpha g}$	EQ(S)	$\hat{\alpha}_g = \hat{\alpha}_g^{IN}$	$\hat{u}_g = \hat{u}_g^{IN}$
$f_{\alpha l}$	EQ(S)	$\hat{\alpha}_l = 1 - \hat{\alpha}_g^{IN}$	$\hat{u}_l = \hat{u}_l^{IN}$
$f_{\beta g}$	EQ(S)	$\hat{\beta}_g$: EX(fw)	$\hat{c}_g$: EX(fw)
$f_{\beta l}$	EQ(S)	$\hat{\beta}_l$: EX(fw)	$\hat{c}_l$: EX(fw)

3.6. Numerical Results and Discussion

In this section, we report the numerical results for some meaningful test cases of the proposed methodology, which are grouped into a preliminary category and an advanced category. The preliminary category (TESTS #1 and #2) is intended to realize the basic essential features of the methodology, while the advanced category (TESTS #3 and #4) discusses challenging features, i.e., very-high-density ratios and realistic phenomenological relations for the interphase momentum exchange. The proposed test cases are also described through their distinctive minimal set of features in Table 3.

Table 2. BCs for LBM schemes: OUTLET ($\hat{x} = N_x$). Acronyms stand for bounce back (BB); anti-bounce back (ABB); extrapolation (EX); and equilibrium (EQ). Uppercase letters in parentheses stand for incompressible equilibrium (I) and standard equilibrium (S). Lowercase letters in parentheses stand for halfway (hw) and full-way (fw).

$f_{*\varphi}$	**OUTLET ($\hat{x} = N_x$): Compute $\hat{y}^{B(out)}$**		
	Type	**Generic Density** $\hat{y}_{\epsilon}^{B(out)}$	**Generic Velocity** $\hat{y}_{u}^{B(out)}$
f_g	ABB(I)	$\hat{\epsilon}_g = 1$	$\hat{u}_g$: EX(hw)
f_l	ABB(I)	$\hat{\epsilon}_l = 1$	$\hat{u}_l$: EX(hw)
$f_{\alpha g}$	ABB(S)	$\hat{\alpha}_g$: EX(hw)	$\hat{u}_g$: EX(hw)
$f_{\alpha l}$	ABB(S)	$\hat{\alpha}_l$: EX(hw)	$\hat{u}_l$: EX(hw)
$f_{\beta g}$	EQ(S)	$\hat{\beta}_g$: EX(fw)	$\hat{c}_g$: EX(fw)
$f_{\beta l}$	EQ(S)	$\hat{\beta}_l$: EX(fw)	$\hat{c}_l$: EX(fw)

Table 3. Numerical test cases of the proposed methodology, organized in a "preliminary" and an "advanced" category. The distinctive parameters are the density ratio R and the possible realistic definition of the momentum exchange coefficient through a drag model.

Category	**Test**	**R**	**Drag Model**
Preliminary	#1	2	None
	#2	5	None
Advanced	#3	833.3	None
	#4	833.3	CGW

First of all, let us consider first the preliminary category with preliminary numerical results. In particular, let us consider the following test case (corresponding to a natural circulation loop driven by gas injection), called TEST #1: $\rho_g^0 = 1.2$, $\rho_l^0 = 2.4$ ($R = 2$), $\nu_g = 1.1667$, $\nu_l = 1.1667$, $\hat{g} = 10^{-6}$, $\hat{K}_I = 10^{-2}$, and $\hat{K}_W = 10^{-2}$. The progressively increased quantities at the inlet (see Equation (153)) for this test case are $\hat{\alpha}_g^{max} = 0.8$ and $\hat{\alpha}_g^{min} = 10^{-2}$ for the dispersed phase volume fraction; $\hat{u}_l^{max} = 10^{-3}$ and $\hat{u}_l^{min} = 0$ for the liquid phase velocity; while, for the dispersed phase velocity, $\hat{u}_g^{max} = 10^{-2}$ and $\hat{u}_g^{min}$ is given by Equation (124), namely,

$$\hat{u}_g^{min} = \frac{1}{\sqrt{\hat{K}_I}} \sqrt{\hat{\alpha}_g^{min} \left(1 - \hat{\alpha}_g^{min}\right) (R-1)\, \hat{g}}. \tag{165}$$

For this test case, we used $N_x = 200$ which is the number of nodal values, $N_t = 6 \times 10^6$ which is number of time steps, and $n_t = 5 \times 10^5 = N_t/12$ which is the characteristic number of time steps which parametrize the transient smooth ramp-up (see Equation (153)). Similarly, for the purpose of testing the capability of the proposed methodology to deal with phases with different average densities, let us consider also another test case, called TEST #2, where $\rho_g^0 = 1.2$, but this time $\rho_l^0 = 6.0$ ($R = 5$), $\hat{g} = 2.5 \times 10^{-7}$, in such a way that $(R-1)\,\hat{g}$ is equal in the two tests discussed so far, and all remaining parameters are the same as in TEST #1.

Next, let us consider the advanced category with advanced numerical results for testing the proposed methodology in challenging setups, i.e., very-high-density ratios and stiff (but realistic) phenomenological relations for the interphase momentum exchange. Let us consider a test case, called TEST #3, where $\rho_g^0 = 1.2$, but this time we take the very challenging value $\rho_l^0 = 1000.0$ ($R = 833.3$), $\hat{g} = 1.2 \times 10^{-9}$, in such a way that $(R-1)\,\hat{g}$ is equal in all cases discussed so far. This third test case is intended to explore how the

solvers deal with a very-large-density ratio—typically very challenging for the LBM—which could trigger some numerical instabilities. For this reason, this test case requires also the stabilization ingredients discussed in Section 3.2: $\gamma = 1$ with the third stabilization strategy (for better consistency) and $n_\gamma = 200 \ll N_t$. All remaining parameters are the same as in other test cases discussed so far.

Moreover, for the purpose of testing also the additional dependency of the drag force on the volume fraction, let us also consider a test case, called TEST #4, where the interphase drag force is modeled by the CGW model. In this case, the simulation parameters must be derived from physical values, which are consistent with a realistic setup (despite the limitations of the one-dimensional configuration). For this aim, let us focus on the experimental setup described in Reference [22] for extracting the most relevant physical quantities: $g = 9.81$, $\kappa_I = 3 \times 0.66/(4 \times 0.002) = 247.5$, $u_g^{max} = 0.3$, and $\alpha_g^{max} = 0.1$, where we assume $U = 1 \ m/s$ as the characteristic flow speed and $L = 1 \ m$ as the characteristic length scale of the flow field. Assuming $\hat{g} = 1.2 \times 10^{-9}$ and $\hat{\kappa}_I = 1.45 \times 10^{-4}$ ($\hat{\kappa}_W = \hat{\kappa}_I$) for stability reasons, applying Equations (138) and (139) yields $c/\tau = 8.18 \times 10^9$ and $c\,\tau = 5.86 \times 10^{-7}$, respectively. Therefore the lattice speed is $c = \sqrt{(c/\tau)\,(c\tau)} = 69.2$. This means that $\hat{u}_g^{max} = u_g^{max}/c = 0.0043$, which ensures a Mach number small enough to be consistent with the incompressible limit. Moreover, let us assume $\hat{u}_l^{max} = \hat{u}_g^{max}/4 = 0.0011$ for having a balanced inlet condition, as we prove below. Adopting $u_g^\dagger = u_g^{max}$, $u_l^\dagger = u_l^{max}$, and $\alpha_g^\dagger = \alpha_g^{max}$ as reference values, it is possible to estimate the ratio between the momentum exchange force and the buoyancy force in the momentum equation of the gas phase by Equation (141), which is equal to $M_{ex} = 1.04$. This means that the drag force of this setup is expected to almost balance the buoyancy force at the inlet. Finally, let us assume $\hat{u}_g^{min} = \hat{u}_g^{max}/10$, $\hat{u}_l^{min} = \hat{u}_l^{max}/10$, and $\alpha_g^{min} = 10^{-2}$. All remaining parameters are the same as in other test cases discussed so far.

Numerical results are reported below for all test cases, comparing those obtained by the FD engine and those by the LBM engine. Figures 1 and 2 present the most relevant features and additional computational details for TEST #1. Similarly, Figures 3 and 4 illustrate the corresponding information for TEST #2. Moreover, Figure 5 shows the main features of TEST #3 about a very-large-density ratio, together with additional computational details for the same test case in Figure 6. Even more remarkably, Figure 7 shows the main features of TEST #4 about a very-large-density ratio with a stiff (but realistic) relation for the drag force. More computational details about this very challenging test case can be found in Figure 8.

Concerning these numerical results, let us comment first the preliminary ones below.

- It is important to highlight that the proposed LBM schemes agree with the FD numerical results in an excellent way. This opens a promising venue for simulating multiphase flows in large HPC facilities. In fact, it is worth highlighting that the proposed methodology and all LBM formulas reported in Section 2 can be automatically applied to any dimension, including all boundary conditions.
- The key point, namely, the artificially compressible continuity equation for each phase, works well, and Figures 2a and 4a prove that the individual phase sources can be computed in LBM schemes without FD corrections.
- LBM schemes without FD corrections can be used effectively to compute all terms in multiphase equations, including gradients of the volume fractions and stress tensors, as clearly reported in Figures 2b–f and 4b–f, where the LBM results show an excellent agreement with those obtained by FD operators.
- It is important to highlight that incompressible multiphase flows has divergence-free mixture velocity, as prescribed by Equation (12), but this condition does not hold for individual phase velocity. The one-dimensional test case is enough to show this important

difference: indeed, $\hat{\alpha}_g \hat{u}_g + \hat{\alpha}_l \hat{u}_l$ is constant (the only way to ensure the divergence-free condition in one dimension), but $\hat{u}_g$ and $\hat{u}_l$ are not. It is clear that Figures 1c and 3c confirm this expectation. Hence the proposed LBM framework ensures, as it should be, the divergence-free condition for the mixture velocity, but it does not force it also for the individual phase velocities. This is one of the most important features of the proposed method in comparison with what is already available in the literature.

- Another consideration is to investigate the relation between the numerical solution at the outlet and the analytical solution derived in the previous section. For the sake of simplicity, let us consider TEST #1. At the outlet, the LBM framework provides $\hat{u}_g(N_x) = 8.4237 \times 10^{-3}$, $\hat{u}_l(N_x) = 3.9750 \times 10^{-3}$, $\hat{\alpha}_g(N_x) = 0.950$, and $\hat{\alpha}_l(N_x) = 0.050$, which corresponds to the following mixture velocity: $\hat{\alpha}_g(N_x)\,\hat{u}_g(N_x) + \hat{\alpha}_l(N_x)\,\hat{u}_l(N_x) = 8.1999 \times 10^{-3}$. In this test case, $r = 0.9496$. Using $\hat{\alpha}_g(N_x)$ instead of $\overline{\alpha}_g$ in Equation (124) leads to $\overline{u}_g^0 = 2.1859 \times 10^{-3}$. Using the latter velocity in Equation (127) and assuming $\overline{u}_g = \hat{u}_g(N_x)$ leads to $\overline{u}_l = 3.9754 \times 10^{-3}$, which is in excellent agreement with $\hat{u}_l(N_x)$ (mismatch 0.01 %). Because in this case r is pretty close to 1, it is also possible to use the approximated formula $\overline{u}_l \approx (\overline{u}_g - \overline{u}_g^0)\,(1 + \overline{u}_g^0/\overline{u}_g)/2 = 3.9282 \times 10^{-3}$ (mismatch 1 %).
- It is also worth comparing the numerical results between TEST #1 and TEST #2. In Figures 1c and 3c the difference is clear in the slope of the kinematic pressure profiles of the two phases for the TEST #1 ($R = 2$) and TEST #2 ($R = 5$), respectively. This proves that the proposed methodology can describe properly the effects due to the density ratio between the phases.

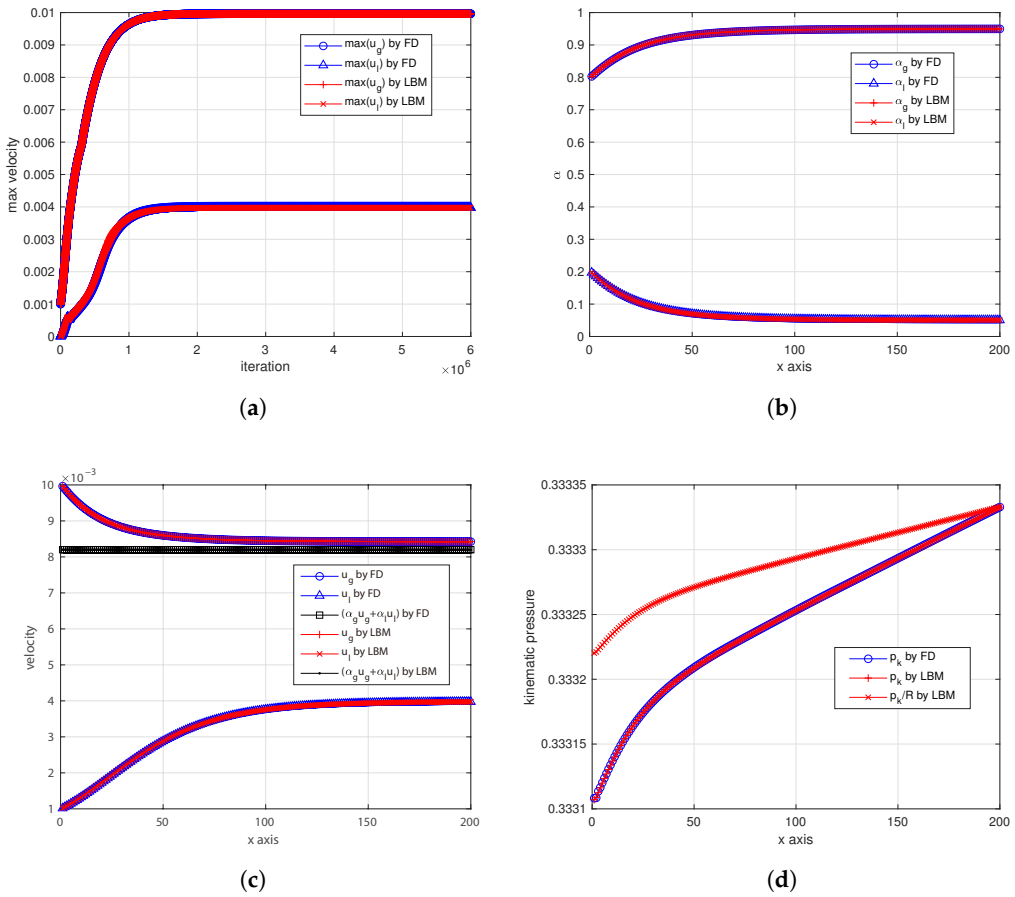
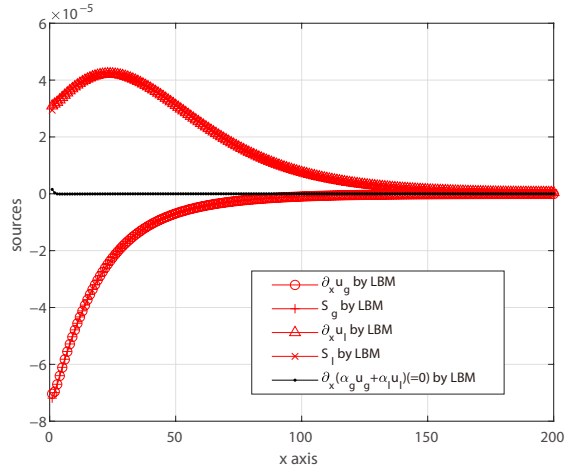
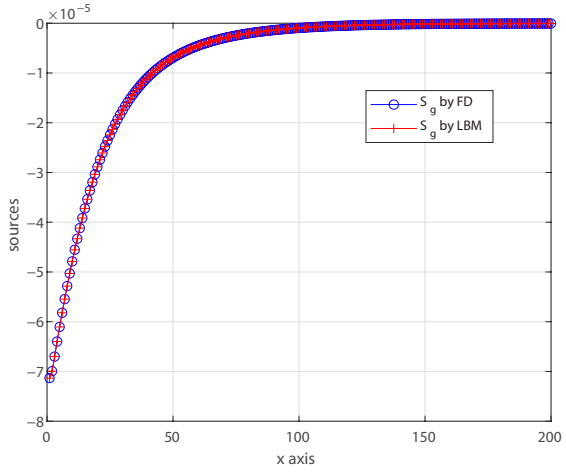


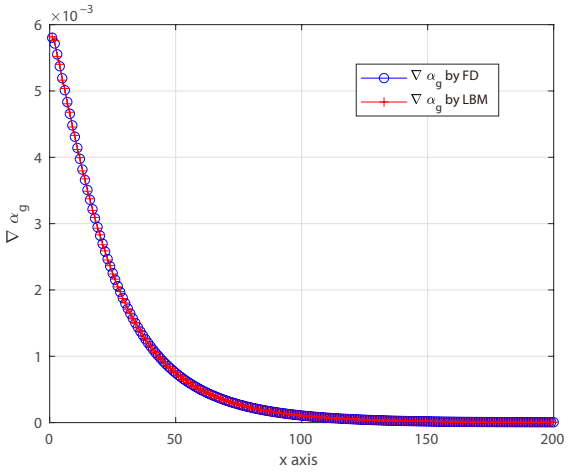
Figure 1. TEST #1: Comparison between numerical results by FD and by LBM ($R = 2$ and $\hat{g} = 10^{-6}$). Multiple panels: (**a**) FD vs. LBM: max velocity; (**b**) FD vs. LBM: volume fractions; (**c**) FD vs. LBM: phase velocities; (**d**) FD vs. LBM: kinematic pressure.



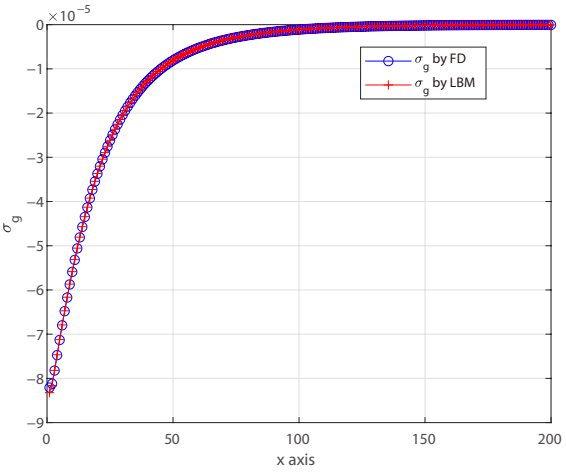
(**a**)



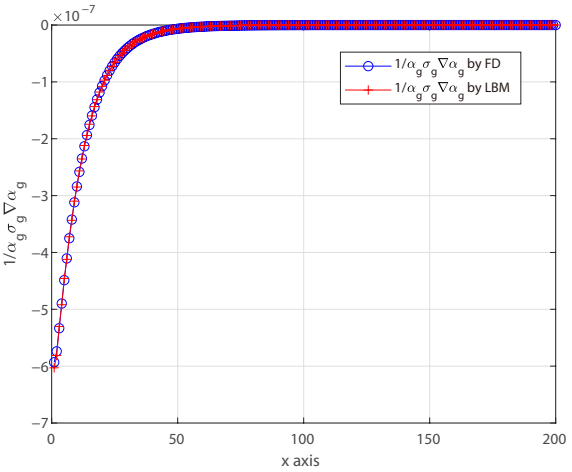
(**b**)



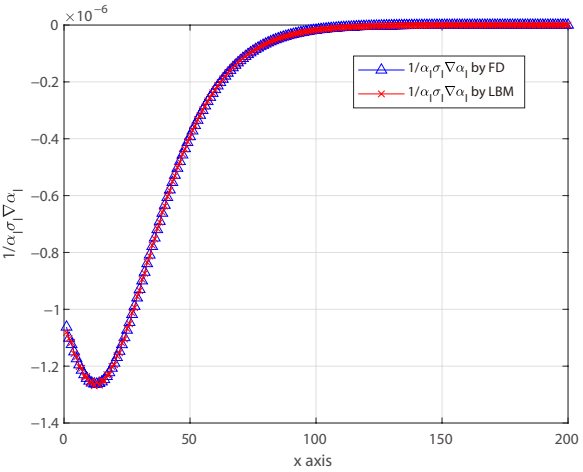
(**c**)



(**d**)

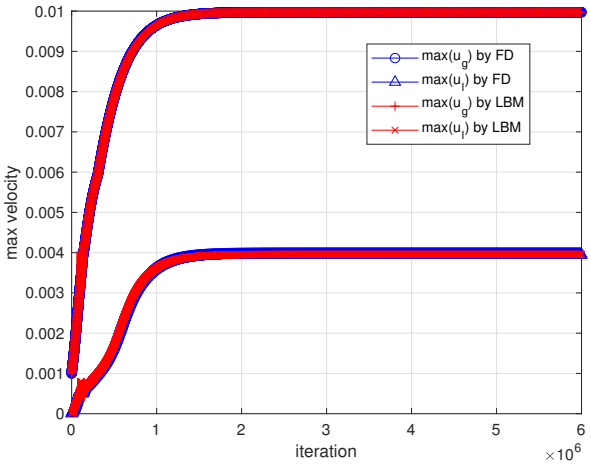


(**e**)

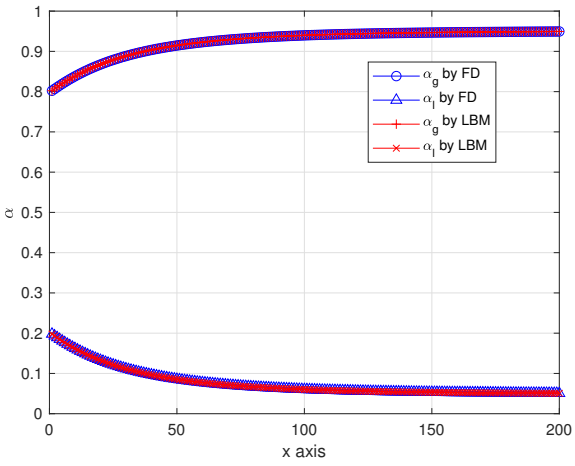


(**f**)

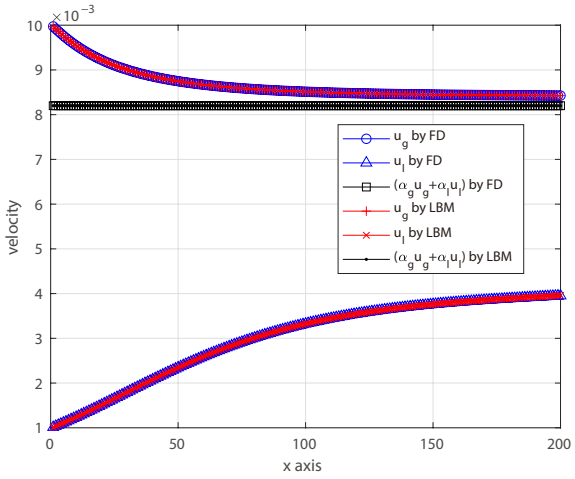
Figure 2. TEST #1: Computational details of the LBM schemes without FD corrections and comparison with numerical results by FD operators ($R = 2$ and $\hat{g} = 10^{-6}$). Multiple panels: (**a**) Phase sources computed by LBM; (**b**) FD vs. LBM: dispersed-phase source; (**c**) FD vs. LBM: dispersed-phase gradient; (**d**) FD vs. LBM: dispersed-phase stress; (**e**) FD vs. LBM: dispersed-phase force; (**f**) FD vs. LBM: liquid-phase force.



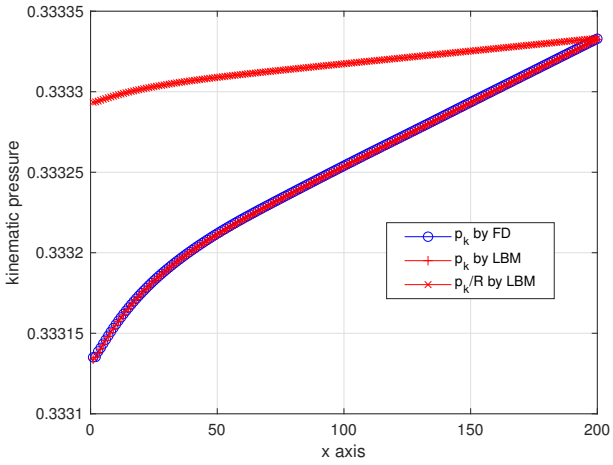
(**a**)



(**b**)

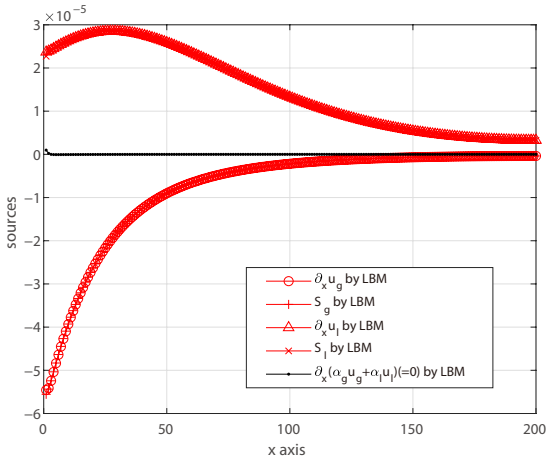


(**c**)

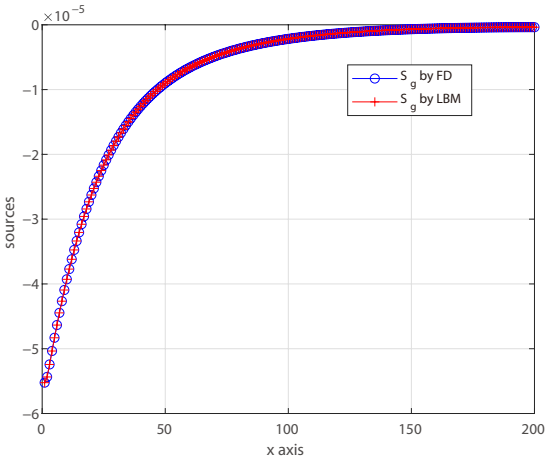


(**d**)

Figure 3. TEST #2: Comparison between numerical results by FD and by LBM ($R = 5$ and $\hat{g} = 2.5 \times 10^{-7}$). Multiple panels: (**a**) FD vs. LBM: max velocity; (**b**) FD vs. LBM: volume fractions; (**c**) FD vs. LBM: phase velocities; (**d**) FD vs. LBM: kinematic pressure.

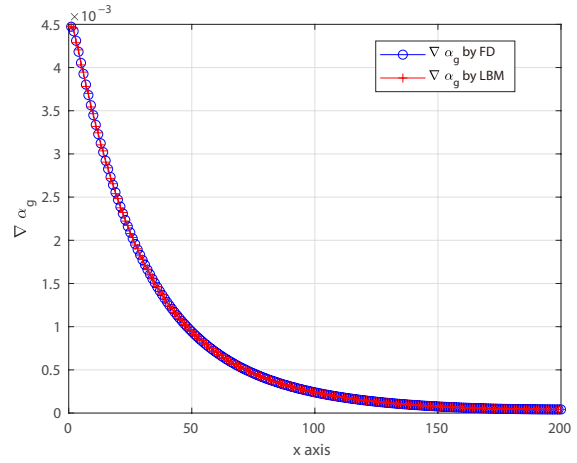


(**a**)

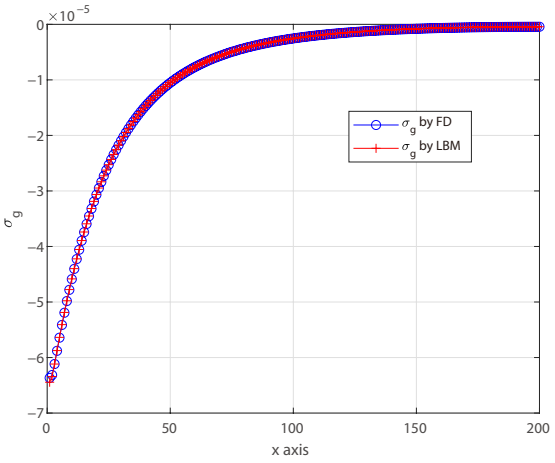


(**b**)

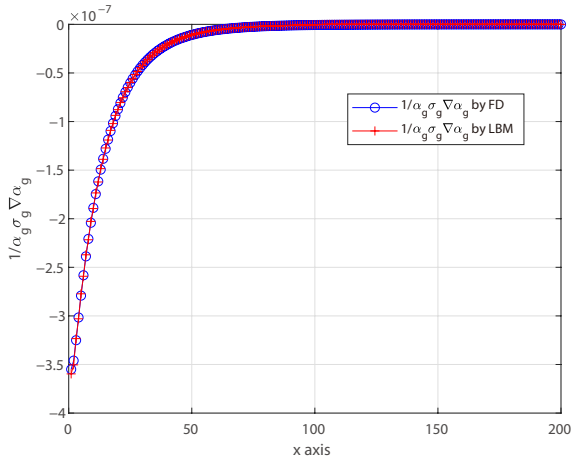
Figure 4. *Cont.*



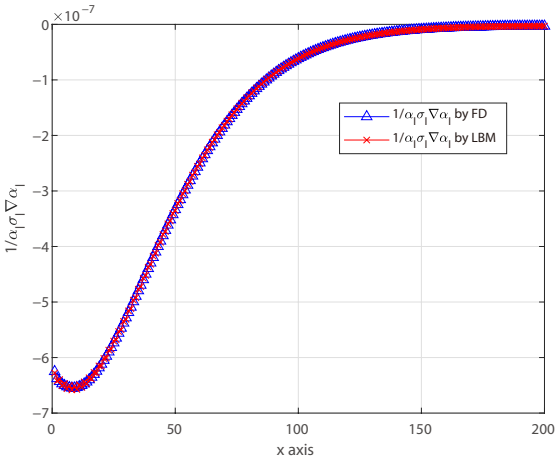
(**c**)



(**d**)

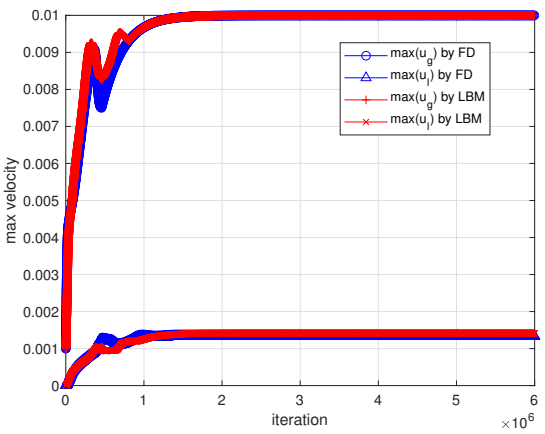


(**e**)

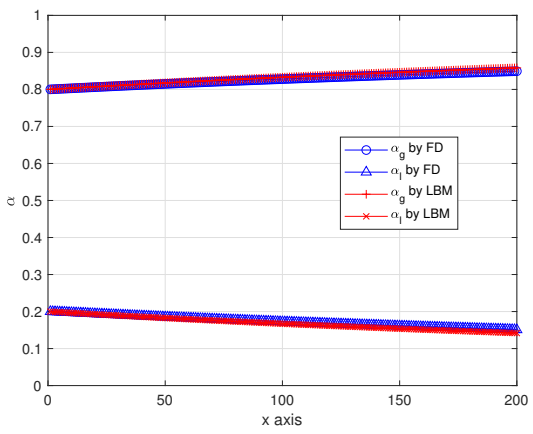


(**f**)

Figure 4. TEST #2: Computational details of the LBM schemes without FD corrections and comparison with numerical results by FD operators ($R = 5$ and $\hat{g} = 2.5 \times 10^{-7}$). Multiple panels: (**a**) Phase sources computed by LBM; (**b**) FD vs. LBM: dispersed-phase source; (**c**) FD vs. LBM: dispersed-phase gradient; (**d**) FD vs. LBM: dispersed-phase stress; (**e**) FD vs. LBM: dispersed-phase force; (**f**) FD vs. LBM: liquid-phase force.

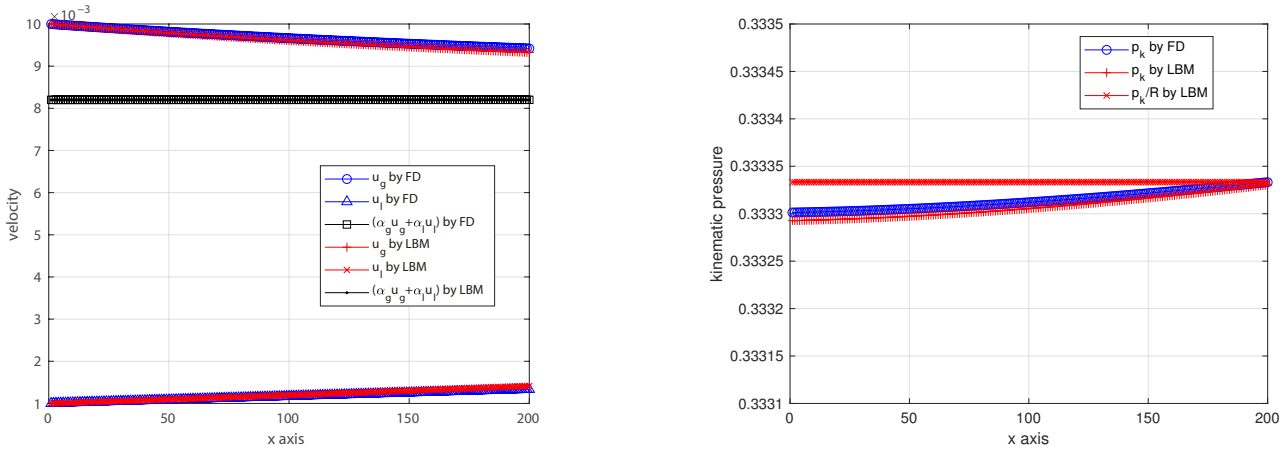


(**a**)



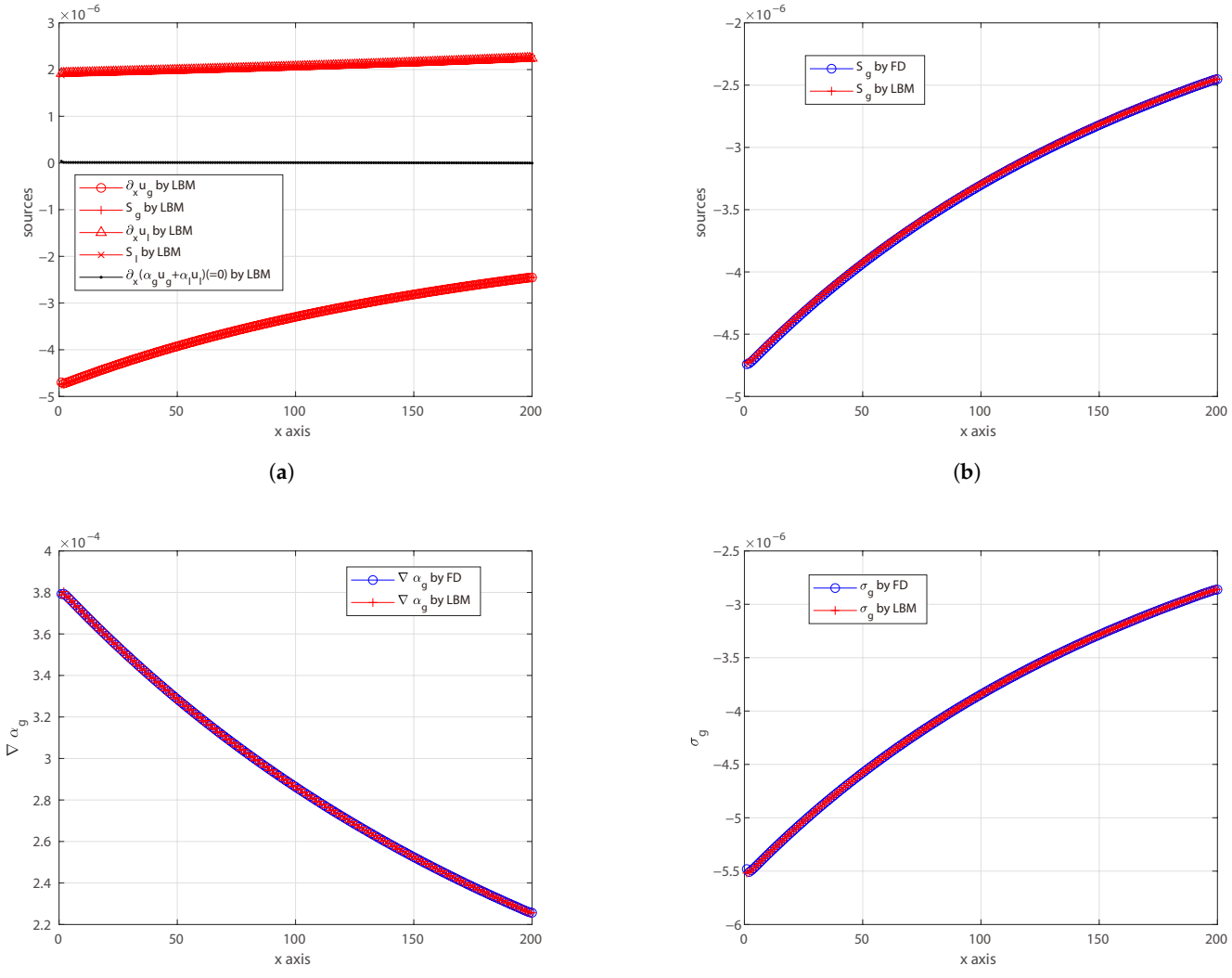
(**b**)

Figure 5. *Cont*.



(**c**) (**d**)

Figure 5. TEST #3: Comparison between numerical results by FD and by LBM ($R = 833.3$ and $\hat{g} = 1.2 \times 10^{-9}$). Multiple panels: (**a**) FD vs. LBM: max velocity; (**b**) FD vs. LBM: volume fractions; (**c**) FD vs. LBM: phase velocities; (**d**) FD vs. LBM: kinematic pressure.



(**a**) (**b**)

(**c**) (**d**)

Figure 6. *Cont.*

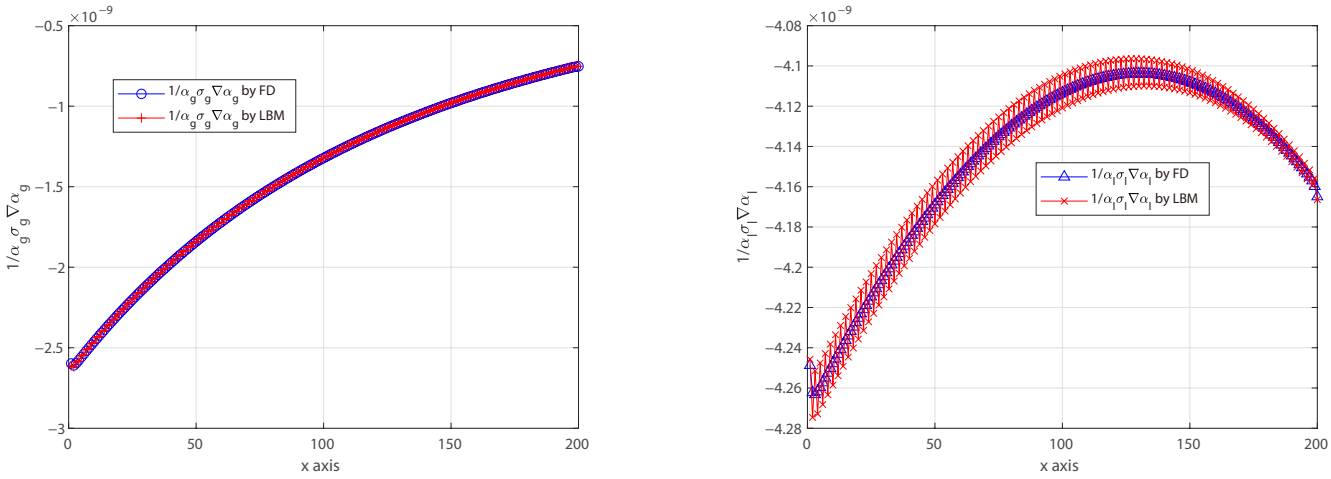


Figure 6. TEST #3: Computational details of the LBM schemes without FD corrections and comparison with numerical results by FD operators ($R = 833.3$ and $\hat{g} = 1.2 \times 10^{-9}$). Multiple panels: (**a**) Phase sources computed by LBM; (**b**) FD vs. LBM: dispersed-phase source; (**c**) FD vs. LBM: dispersed-phase gradient; (**d**) FD vs. LBM: dispersed-phase stress; (**e**) FD vs. LBM: dispersed-phase force; (**f**) FD vs. LBM: liquid-phase force.

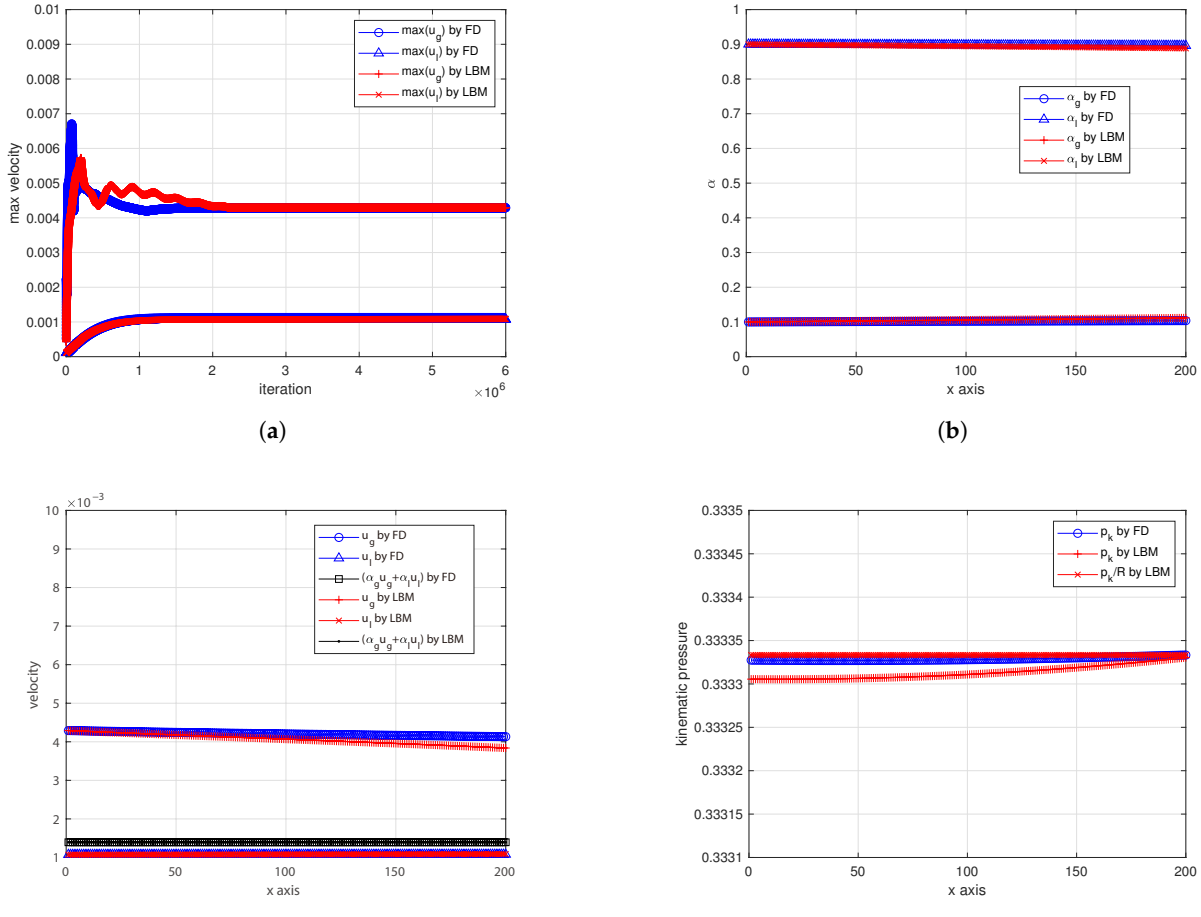


Figure 7. TEST #4: Comparison between numerical results by FD and by LBM ($R = 833.3$, $\hat{g} = 1.2 \times 10^{-9}$, model for drag force by Clift, Grace & Weber with $\hat{\kappa}_I = 1.45 \times 10^{-4}$). Multiple panels: (**a**) FD vs. LBM: max velocity; (**b**) FD vs. LBM: volume fractions; (**c**) FD vs. LBM: phase velocities; (**d**) FD vs. LBM: kinematic pressure.

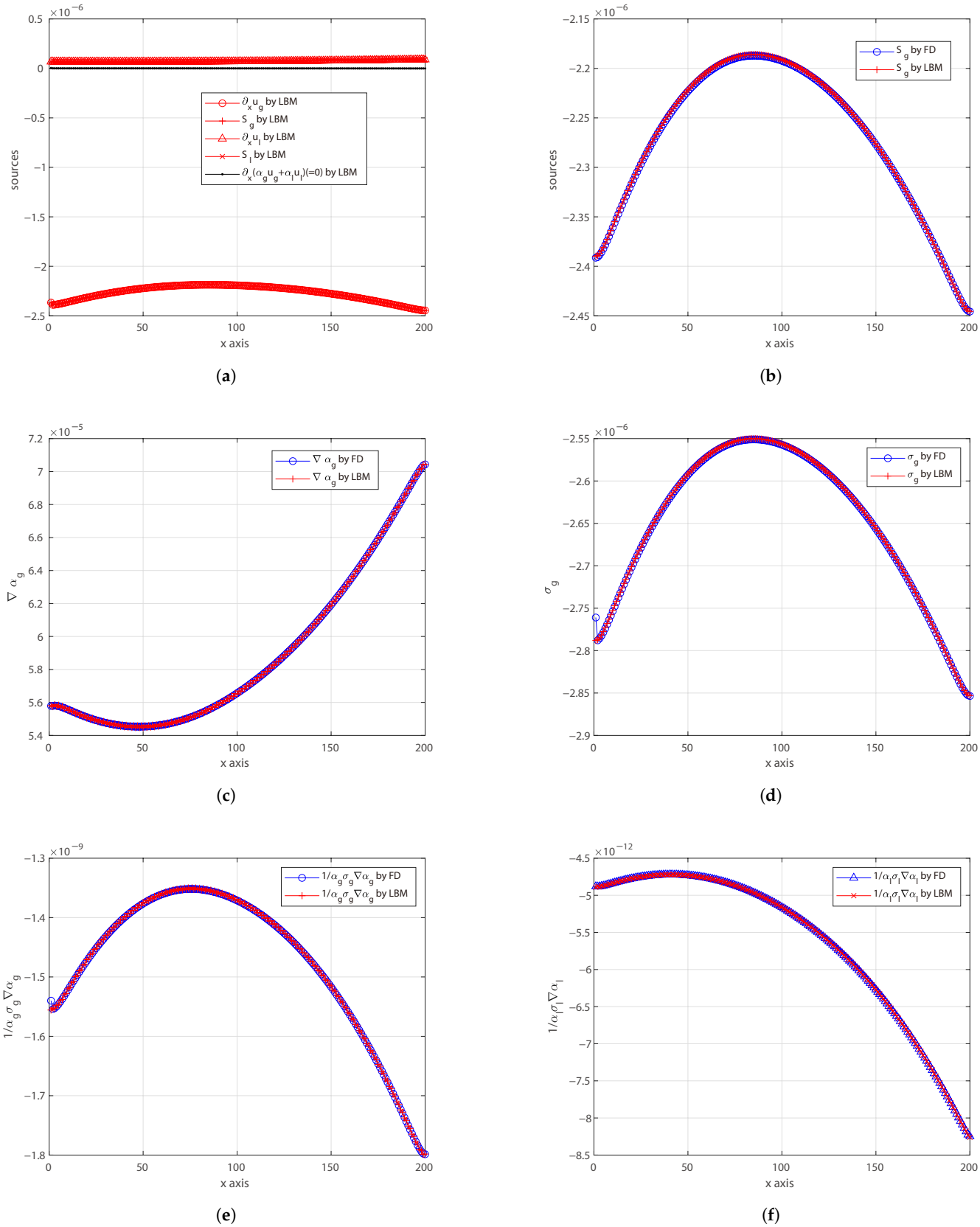


Figure 8. TEST #4: Computational details of the LBM schemes without FD corrections and comparison with numerical results by FD operators ($R = 833.3$, $\hat{g} = 1.2 \times 10^{-9}$, model for drag force by Clift, Grace & Weber with $\hat{\kappa}_I = 1.45 \times 10^{-4}$). Multiple panels: (**a**) Phase sources computed by LBM; (**b**) FD vs. LBM: dispersed-phase source; (**c**) FD vs. LBM: dispersed-phase gradient; (**d**) FD vs. LBM: dispersed-phase stress; (**e**) FD vs. LBM: dispersed-phase force; (**f**) FD vs. LBM: liquid-phase force.

Secondly, advanced results are discussed below:

- The previous preliminary results highlight some challenges, mainly about large-density ratios, which are discussed here in the category about advanced results. In the limit of very-large-density ratios, namely, $R \gg 1$, some terms proportional to $1/R$ in the momentum equation of the liquid phase become very small and hence comparable with the numerical errors. For this reason, TEST #3 ($R = 833.3$) also requires the stabilization ingredients discussed in Section 3.2: $\gamma = 1$ with the third stabilization strategy and $n_\gamma = 200 \ll N_t$. The transient profiles of the LBM results are slightly smoother than those of the FD results, as shown in Figure 5a. The kinematic pressure profiles computed by the two engines in Figure 5d reveal a small discrepancy, but this is acceptable because the pressure gradient is what really matters. The numerical details of the proposed methodology reported in Figure 6 are excellent with the exception of the computed second term of the force G_l (in the one-dimensional case) given by Equation (26) and reported in Figure 6f, which shows some numerical oscillations (but the magnitude of these oscillations is extremely small).
- An important feature to be discussed in the category about the advanced numerical results is the need to include realistic phenomenological relations. In particular, TEST #4 adopts the model by Clift, Grace, and Weber (CGW) [22], which introduces a further dependence on the volume fraction into the effective drag coefficient (stiff coupling). The transient profiles of the LBM results show some discrepancies with regard to those of the FD results, as shown in Figure 7a. In particular, the LBM solution is not necessarily smoother than the reference solution, likely because of the stiffness of the CGW model. At steady state, in spite of the excellent numerical details reported in Figure 8, there is a small mismatch for this mesh in the volume fraction profiles reported in Figure 7b and in the velocity profiles reported in Figure 7c (e.g., looking at the gas phase, there is an 8% discrepancy at the outlet). Moreover, the kinematic pressure profiles computed by the two engines in Figure 7d reveal a negligible discrepancy in absolute terms but which may have an impact on the effective pressure gradient and hence on the momentum equations. It is important to recall that a small discrepancy in α_g at small values, but with very-large-density ratios, may lead to a significant impact on $\Lambda(\alpha_g)$ and hence on the drag force acting in the momentum equation.

4. Conclusions

In spite of its apparent simplicity, the LBM requires a careful design of the overall framework to be used for solving complex systems of partial differential equations, as needed by multiphase flows. By framework, we mean (at least) the following information: minimum set of distribution functions, the definitions of the relevant target quantities as functions of the distribution functions (which are just auxiliaries), the definitions of the equilibrium distribution functions and how to compute them, including the proper equation of states, the proper additional forces, sources, etc. The framework is essential because the LBM is defined in an auxiliary space of variables and also, more importantly, because only a subset of this auxiliary space ensures stable calculations (e.g., zeroth-order moment of the distribution function usually must not be too small in order to avoid singularities, lattice sound speed cannot be too large, etc.).

For the first time to our knowledge, this work proposes a novel LBM framework to solve Eulerian–Eulerian multiphase flow equations without any finite difference correction, including very-large-density ratios and also a realistic model for the drag coefficient. The proposed methodology and all reported LBM formulas can be applied to any dimension. This opens a promising venue for simulating multiphase flows in large HPC facilities and

on novel parallel hardware. This LBM framework consists of six LBM schemes for the two phases, so it naturally comes with a large memory footprint and a certain level of coding complexity. It includes the following: two LBM schemes (f_g and f_l) for artificially compressible continuity equations and momentum equations; two LBM schemes ($f_{\alpha g}$ and $f_{\alpha l}$) for phase volume fractions; and two LBM schemes ($f_{\beta g}$ and $f_{\beta l}$) for phase continuity sources. All of these schemes are run on the same lattice and are coupled with each other, ensuring the best synergy for efficient implementation in large codes with minimum effort. Its memory-demanding nature, however, combined with the intrinsic locality of LBM operations, makes it particularly well-suited for HPC applications. Nonetheless, some limitations were observed. In simulations with very-high-density ratios, small mismatches in the volume fraction profiles—especially when the absolute value of α_g is small—may lead to a significant impact on the drag force computation, particularly when using realistic drag coefficient models. Furthermore, possible numerical instabilities might emerge when extending the model to multidimensional configurations. These aspects suggest directions for future refinement of the method.

Author Contributions: Conceptualization, M.M.P. and P.A.; methodology, M.M.P. and P.A.; software, M.M.P. and P.A.; validation, M.M.P. and P.A.; formal analysis, M.M.P. and P.A.; investigation, M.M.P. and P.A.; resources, P.A.; data curation, M.M.P. and P.A.; writing—original draft preparation, M.M.P. and P.A.; writing—review and editing, M.M.P. and P.A.; visualization, M.M.P. and P.A.; supervision, P.A.; project administration, P.A.; funding acquisition, P.A. All authors have read and agreed to the published version of the manuscript.

Funding: This publication is part of the project PNRR-NGEU, which has received funding from the MUR–DM 352/2022.

Data Availability Statement: Dataset available on request from the authors.

Acknowledgments: This publication is part of the project PNRR-NGEU which has received funding from the MUR–DM 352/2022. The authors would like to acknowledge useful discussions with Keld Lund Nielsen, Fabrizio Podenzani, and Chiara Della Torre. The authors would like to acknowledge the support of Matteo Fasano for project administration.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Computational Details for D1Q3 Lattice

In this appendix, we report the computational formulas of the equilibrium functions used in the proposed LBM framework for the one-dimensional case. Let us consider the D1Q3 lattice [1], where the generic velocity becomes a scalar and it belongs to the lattice $\mathbb{L}$, namely, $v_q \in \mathbb{L}$, where

$$\mathbb{L} = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}. \tag{A1}$$

It is worth highlighting that the D1Q3 lattice is just a projection of the lattices with higher dimensionality on the axis $\boldsymbol{e}_x$ [1], and this makes one confident that it is easy, in the LBM context, to extend the proposed framework to higher dimensionality.

Concerning the equilibrium functions, standard (S) equilibrium given by Equation (72) becomes

$$f_S^{eq}(\hat{\alpha}, \hat{u}) \equiv f^{eq}(\hat{\alpha}, \hat{u}) = \begin{bmatrix} \hat{\alpha}\,(2/3 - \hat{u}^2) \\ \hat{\alpha}\,(1 + 3\,\hat{u} + 3\,\hat{u}^2)/6 \\ \hat{\alpha}\,(1 - 3\,\hat{u} + 3\,\hat{u}^2)/6 \end{bmatrix}. \tag{A2}$$

Incompressible (I) equilibrium given by Equation (37) becomes

$$f_I^{eq}(\phi, \hat{\epsilon}, \hat{u}) = \begin{bmatrix} \hat{\epsilon}\,(3-\phi)/3 - \hat{u}^2 \\ (\hat{\epsilon}\,\phi + 3\,\hat{u} + 3\,\hat{u}^2)/6 \\ (\hat{\epsilon}\,\phi - 3\,\hat{u} + 3\,\hat{u}^2)/6 \end{bmatrix}. \tag{A3}$$

Finally, linearized equilibrium given by Equation (42) becomes

$$f_L^{eq}(\psi, \hat{S}, \hat{G}) = \begin{bmatrix} \hat{S}\,(3-\psi)/3 \\ (\hat{S}\,\psi + 3\,\hat{G})/6 \\ (\hat{S}\,\psi - 3\,\hat{G})/6 \end{bmatrix}. \tag{A4}$$

References

1. Krüger, T.; Kusumaatmaja, H.; Kuzmin, A.; Shardt, O.; Silva, G.; Viggen, E.M. *The Lattice Boltzmann Method: Principles and Practice*; Graduate Texts in Physics; Springer Nature: Cham, Switzerland, 2016.
2. Leclaire, S.; Parmigiani, A.; Malaspinas, O.; Chopard, B.; Latt, J. Generalized three-dimensional lattice Boltzmann color-gradient method for immiscible two-phase pore-scale imbibition and drainage in porous media. *Phys. Rev. E* **2017**, *95*, 033306. [CrossRef] [PubMed]
3. Shan, X.; Chen, H. Lattice Boltzmann model for simulating flows with multiple phases and components. *Phys. Rev. E* **1993**, *47*, 1815–1819. [CrossRef]
4. Coelho, R.C.V.; Moura, C.B.; Telo da Gama, M.M.; Araújo, N.A.M. Wetting boundary conditions for multicomponent pseudopotential lattice Boltzmann. *Int. J. Numer. Methods Fluids* **2021**, *93*, 2570–2580. [CrossRef]
5. Huang, H.; Sukop, M.; Lu, X. *Multiphase Lattice Boltzmann Methods: Theory and Application*; John Wiley & Sons, Incorporated: Hoboken, NJ, USA, 2015.
6. Wang, H.; Yuan, X.; Liang, H.; Chai, Z.; Shi, B. A brief review of the phase-field-based lattice Boltzmann method for multiphase flows. *Capillarity* **2019**, *2*, 33–52. [CrossRef]
7. Reis, T.; Phillips, T.N. Lattice Boltzmann model for simulating immiscible two-phase flows. *J. Phys. A Math. Theor.* **2007**, *40*, 4033. [CrossRef]
8. Manninen, M.; Taivassalo, V.; Kallio, S. *On the Mixture Model for Multiphase Flow*; Number 288 in VTT Publications; VTT Technical Research Centre of Finland: Espoo, Finland, 1996.
9. Garcia-Villalba, M.; Colonius, T.; Desjardins, O.; Lucas, D.; Mani, A.; Marchisio, D.; Matar, O.K.; Picano, F.; Zaleski, S. Numerical methods for multiphase flows. *Int. J. Multiph. Flow* **2025**, *191*, 105285. [CrossRef]
10. Prosperetti, A.; Tryggvason, G. *Computational Methods for Multiphase Flow*; Cambridge University Press: Cambridge, UK, 2007.
11. Maniscalco, F.; Buffo, A.; Marchisio, D.; Vanni, M. Numerical simulation of bubble columns: LES turbulence model and interphase forces blending approach. *Chem. Eng. Res. Des.* **2021**, *173*, 1–4. [CrossRef]
12. Oliveira, P.J.; Issa, R.I. Numerical aspects of an algorithm for the Eulerian simulation of two-phase flows. *Int. J. Numer. Methods Fluids* **2003**, *43*, 1177–1198. [CrossRef]
13. Laney, C.B. *Computational Gasdynamics*; Cambridge University Press: Cambridge, UK 1998.
14. Passalacqua, A.; Fox, R. Implementation of an iterative solution procedure for multi-fluid gas-particle flow models on unstructured grids. *Powder Technol.* **2011**, *213*, 174–187. [CrossRef]
15. Lallemand, P.; Luo, L.S.; Krafczyk, M.; Yong, W.A. The lattice Boltzmann method for nearly incompressible flows. *J. Comput. Phys.* **2021**, *431*, 109713. [CrossRef]
16. Ohwada, T.; Asinari, P. Artificial compressibility method revisited: Asymptotic numerical method for incompressible Navier–Stokes equations. *J. Comput. Phys.* **2010**, *229*, 1698–1723. [CrossRef]
17. Junk, M.; Klar, A.; Luo, L.S. Asymptotic analysis of the lattice Boltzmann equation. *J. Comput. Phys.* **2005**, *210*, 676–704. [CrossRef]
18. Spalding, D.B. *The Numerical Computation of Multi-Phase Flows*; Technical Report; Imperial College of London: London, UK, 1985.
19. Inamuro, T. A lattice kinetic scheme for incompressible viscous flows with heat transfer. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2002**, *360*, 477–484. [CrossRef]
20. Asinari, P.; Ohwada, T.; Chiavazzo, E.; Di Rienzo, A. Link-wise artificial compressibility method. *J. Comput. Phys.* **2012**, *231*, 5109–5143. [CrossRef]
21. Ishii, M.; Hibiki, T. *Thermo-Fluid Dynamics of Two-Phase Flow*; Springer: New York, NY, USA, 2010. [CrossRef]
22. Pfleger, D.; Gomes, S.; Gilbert, N.; Wagner, H.G. Hydrodynamic simulations of laboratory scale bubble columns fundamental studies of the Eulerian–Eulerian modelling approach. *Chem. Eng. Sci.* **1999**, *54*, 5091–5099. [CrossRef]