2. Governing equations and numerical implementation

2.1. Lattice Boltzmann methods

The numerical simulation of multiphase flow problems typically involves solving the macroscopic governing Navier–Stokes (NS) equations within each fluid region, which directly provides hydrodynamic parameters such as velocity and pressure. Here, as discussed in the introduction, we instead use the lattice Boltzmann method (LBM) to solve a mesoscopic problem, whose solution converges to that of NS equations. Additionally, our proposed LBM approach is able to solve two-fluid flows whose density ratio is nearly 1000, typically air and water.

For multiphase flows, a separate model must also be implemented for tracking the motion of interfaces between fluids, which can be represented as sharp or diffused. Here, the latter is assumed and interfaces are tracked by the Cahn–Hilliard advection–diffusion (AD) equation. In the latter, a phase order function is used to smoothly vary fluid properties, such as density and viscosity, across the interface (denoted ϕ and varying between arbitrary values of ϕ_1 and ϕ_2 set for fluid 1 and 2, respectively, with here $\phi_1 > \phi_2$; see details in Section 2.3).

As in the two-dimensional (2D) model of Banari et al. [5], in our three-dimensional (3D) model, both the NS and AD equations will be solved by two distinct LBM schemes, with separate sets of distributions functions and collision operators. In both cases, the *D*3*Q* 19 lattice discretization scheme will be used, which introduces 19 discrete particle velocities in directions $\boldsymbol{e}_i$ defined as,

$$\boldsymbol{e}_i = c.\left\{\begin{array}{c|cccccc|cccccccccccc} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{array}\right\} \tag{2.1}$$

where $c = \Delta x/\Delta t$ denotes the particle propagation speed on the lattice [17], and Δx and Δt are the lattice constant mesh size and time step, respectively.

Details of the LBM equations and schemes are given in the following sections.

2.2. Lattice Boltzmann solution of NS equations

The macroscopic continuity and momentum (i.e., NS) equations for incompressible Newtonian fluids read (using the index summation convention),

$$\partial_\alpha u_\alpha = 0 \tag{2.2}$$

$$\partial_t u_\alpha + \partial_\beta\left(u_\alpha u_\beta\right) = -\frac{1}{\rho}\partial_\beta(p) + \partial_\beta\left\{\mu\left(\partial_\alpha u_\beta + \partial_\beta u_\alpha\right)\right\} + \frac{k}{\rho}\partial_\beta\left\{\frac{\partial\phi}{\partial x_\gamma}\frac{\partial\phi}{\partial x_\gamma}\delta_{\alpha\beta} - \frac{\partial\phi}{\partial x_\alpha}\frac{\partial\phi}{\partial x_\beta}\right\} \tag{2.3}$$

where the last term in Eq. (2.3) is non-standard and represents surface tension effects at the diffuse interface between both fluids [5] (k is related to the ϕ_1 and ϕ_2 and the surface tension coefficient σ_{12}, for fluids 1 and 2, and will be defined in Section 2.3). These equations are solved in a LBM framework using the particle distribution functions g_i, whose time evolution is governed by,

$$g_i(\boldsymbol{x} + \boldsymbol{e}_i\Delta t, t + \Delta t) = g_i(\boldsymbol{x}, t) - \frac{\Delta t}{\tau_g}(g_i(\boldsymbol{x}, t) - g_i^{(\mathrm{eq})}(\boldsymbol{x}, t)) + \Delta t\,(F_i + G_i) \tag{2.4}$$

where F_i and G_i represent effects of the viscous and gravitational volume forces, respectively, τ_g is a relaxation time, and g_i^{eq} are the equilibrium distribution functions defined as,

$$\begin{aligned} g_i^{(\mathrm{eq})} &= w_i\left\{\frac{e_{i\alpha}u_\alpha^*}{c_s^2} + \frac{(e_{i\alpha}u_\alpha^*)^2}{2c_s^4} - \frac{|\boldsymbol{u}^*|^2}{2c_s^2} + \frac{1}{2c_s^2}\left(\tau_g - \frac{1}{2}\Delta t\right)\left(\frac{\partial u_\beta}{\partial x_\alpha} + \frac{\partial u_\alpha}{\partial x_\beta}\right)e_{i\alpha}e_{i\beta}\right\} \\ &\quad + w_i\frac{k}{\rho}T_{\alpha\beta}e_{i\alpha}e_{i\beta} - v_i\frac{k}{2\rho}|\nabla\phi|^2. \end{aligned} \tag{2.5}$$

where $c_s = c/\sqrt{3}$, is the speed of sound and,

$$w_0 = \frac{1}{3}; \qquad w_{1,\ldots,6} = \frac{1}{18}; \qquad w_{7,\ldots,18} = \frac{1}{36}, \tag{2.6}$$

$$v_0 = -\frac{2}{c^2}; \qquad v_i = \frac{3}{c^2}w_i \quad (i = 1, 2, \ldots, 18), \tag{2.7}$$

are LBM weights used for the *D*3*Q* 19 scheme, and,

$$T_{\alpha\beta}(\phi) = \frac{9}{2c^4}\frac{\partial\phi}{\partial x_\alpha}\frac{\partial\phi}{\partial x_\beta} - \frac{3}{2c^4}\frac{\partial\phi}{\partial x_\gamma}\frac{\partial\phi}{\partial x_\gamma}\delta_{\alpha\beta}. \tag{2.8}$$

expresses surface tension effects at the two-fluid interface, together with the last term in Eq. (2.5). The last term in the curly bracket in Eq. (2.5) removes the dependency of τ_g on viscosity, which makes the proposed method extremely stable for the simulation of high Reynolds number flows. For this reason, viscous effects are added back as a volume force in the evolution Eq. (2.4), together with the effect of the gravity force. The volume forces expressing the contributions of viscous effects and gravity force in the LBM governing equation read,

$$F_i = \left(1 - \frac{\Delta t}{2\tau_g}\right) w_i \left[\frac{\boldsymbol{e}_i - \boldsymbol{u}}{c_s^2} + \frac{\boldsymbol{e}_i.\boldsymbol{u}}{c_s^4}\boldsymbol{e}_i\right].\boldsymbol{F} \tag{2.9}$$

$$G_i = \left(1 - \frac{\Delta t}{2\tau_g}\right) w_i \left[\frac{\boldsymbol{e}_i - \boldsymbol{u}}{c_s^2} + \frac{\boldsymbol{e}_i.\boldsymbol{u}}{c_s^4}\boldsymbol{e}_i\right].\boldsymbol{G} \tag{2.10}$$

with $\boldsymbol{G}$ the gravitational acceleration vector, and $\boldsymbol{F} = \frac{1}{\rho}\partial_\beta\left(\mu(\frac{\partial u_\beta}{\partial x_\alpha} + \frac{\partial u_\alpha}{\partial x_\beta})\right)$ the viscous stress per unit mass. Note that the extended body force formulation of Guo et al. [18] has been used in these equations, since body forces vary in time and space.

The presence of density in the standard equilibrium distribution functions used for solving NS equations is the main source of instability identified in various earlier LBM solutions of multiphase flows with high density ratios [10,19]. Here, to eliminate these instabilities, following Banari et al. [5], density was removed from the first part of the equilibrium functions in Eq. (2.5). This will require adding a correction to the velocity field to satisfy mass conservation, as discussed next. It can be shown by applying a Chapman–Enskog expansion that Eq. (2.4) with the equilibrium distribution functions defined in Eq. (2.5) converges to the NS equations (Eqs. (2.3), (2.2)), without the pressure gradient term (referred to as "pressureless" NS equations). The details of the Chapman–Enskog expansion are shown in Appendix A. The fluid velocity corresponding to these pressureless NS equations is found from Eq. (2.5) as,

$$u_\alpha^* = \sum_{i=0}^{b} g_i e_{i\alpha} + \frac{\Delta t}{2}(\boldsymbol{F} + \boldsymbol{G}). \tag{2.11}$$

However, this velocity is not divergence free and hence needs to be corrected. Thus, similar to [9,5], the corrected velocity $\boldsymbol{u}$ is obtained as,

$$\boldsymbol{u} = \boldsymbol{u}^* + \Delta\boldsymbol{u} \tag{2.12}$$

with the correction $\Delta\boldsymbol{u}$ defined as,

$$\Delta\boldsymbol{u} \simeq -\Delta t\,\frac{\nabla p}{\rho} \tag{2.13}$$

as a function of the gradient of the pressure p, which is obtained by solving the following Poisson equation,

$$\nabla.\left(\frac{\Delta t\,\nabla p}{\rho}\right) = \nabla.\boldsymbol{u}^*. \tag{2.14}$$

Similar to [5], this equation is solved iteratively in a LBM framework. For this purpose, an additional set of particle distribution functions h_i ($i = 0, \ldots, 18$) is introduced, whose time evolution is governed by,

$$h_i^n(\boldsymbol{x} + \boldsymbol{e}_i\Delta t, t + \Delta t) = h_i^n(\boldsymbol{x}, t) - \frac{\Delta t}{\tau_h}(h_i^n - h_i^{(\mathrm{eq},n)}) - \frac{1}{3}w_i(\nabla.\boldsymbol{u}^*(t)) \tag{2.15}$$

where n denotes the n-th iteration in the Poisson equation solution (see [9] for details). In this second LBM scheme, the equilibrium distribution functions are defined as,

$$h_i^{(\mathrm{eq},n)} = \frac{w_i p^n(\boldsymbol{x}, t)}{\rho_0 c^2} \tag{2.16}$$

with ρ_0 denoting a reference density and the relaxation time τ_h being related to the density by,

$$\tau_h = \Delta t\left(\frac{\rho_0 c^2}{\rho c_s^2} + \frac{1}{2}\right). \tag{2.17}$$

Once the Eqs. (2.15)–(2.17) are solved for a given iteration n, the pressure is obtained as the zeroth order moment of the particle distribution functions as,

$$p^{n+1} = \rho_0 c^2 \sum_{i=0}^{b} h_i^n. \tag{2.18}$$

This scheme is solved iteratively until convergence is achieved on the pressure for a given time step. Then, the velocity correction is calculated using Eq. (2.13).

As shown in the Chapman–Enskog expansion (Appendix A), the complete set of LBM equations (2.4)–(2.18) solves the complete set of NS equations (2.3). Unlike the classical LBM, however, the relaxation time τ_g can be chosen arbitrarily, independently from the Reynolds number, and the scheme is stable for high density ratios between fluids. It should be pointed out that, incorporating viscous forces as volume forces creates additional spurious terms in the macroscopic NS equation. These terms, however, scale with viscosity and hence are negligible for high Reynolds number flows [5].

2.3. Lattice Boltzmann scheme for interface capturing

In this 3D model, similar to the 2D model of Banari et al. [5], the position of the interface between the 2 fluids is tracked using the Cahn–Hilliard advection–diffusion equation,

$$\partial_t(\phi) + \partial_\alpha(\phi u_\alpha) = M\nabla^2\mu_\phi \tag{2.19}$$

where M is the mobility parameter between two components, and μ_ϕ is the chemical potential (see details below). Diffuse interface schemes offer some advantages as compared to sharp interface schemes, for which fluid parameters such as density and viscosity vary discontinuously across the interface. Such discontinuities often cause numerical noise, which can potentially trigger instabilities, especially for high density and viscosity ratios. By contrast, in diffuse interface schemes, fluid properties continuously and smoothly vary over a short distance across the interface. Additionally, for sharp interface schemes, the accurate computation of the interface curvature and related surface tension forces is very challenging, whereas surface tension effects are intrinsically included in diffuse interface models. Here, similar to [5], the motion of the diffusive interface is modeled by the Cahn–Hilliard equation,

In the model, the chemical potential is defined as,

$$\mu_\phi = \frac{d\Psi}{d\phi} - k\nabla^2\phi \quad \text{with } \Psi(\phi) = (\phi-\phi_2)^2(\phi-\phi_1)^2 \tag{2.20}$$

the bulk free-energy density. Coefficients k and β are related to the surface tension coefficient σ_{12} and interface thickness W by,

$$W = \frac{4}{\phi_1-\phi_2}\sqrt{\frac{k}{2\beta}} \quad \text{and} \quad \sigma_{12} = \frac{(\phi_1-\phi_2)^3}{6}\sqrt{2k\beta}. \tag{2.21}$$

A more detailed derivation and discussion of this model can be found in [5].

Eqs. (2.19)–(2.21), are solved with a LBM using a third set of probability distribution functions, $f_i(x,t)$, whose time evolution is governed by,

$$f_i(\boldsymbol{x}+\boldsymbol{e}_i\Delta t, t+\Delta t) = f_i(\boldsymbol{x},t) - \frac{\Delta t}{\tau_f}(f_i(\boldsymbol{x},t) - f_i^{(\mathrm{eq})}(\boldsymbol{x},t)). \tag{2.22}$$

The corresponding equilibrium distribution functions $f_i^{(\mathrm{eq})}$ read,

$$f_i^{(\mathrm{eq})} = H_i\phi + v_i\frac{M}{\tau_f - \frac{1}{2}\Delta t}\mu_\phi + \phi\, w_i\left\{\frac{e_{i\alpha}u_\alpha}{c_s^2} + \frac{(e_{i\alpha}u_\alpha)^2}{2c_s^4} - \frac{|\boldsymbol{u}|^2}{2c_s^2}\right\}, \tag{2.23}$$

with,

$$H_0 = 1 \quad \text{and} \quad H_{1,2,\ldots,8} = 0 \tag{2.24}$$

and w_i and v_i defined as in Eqs. (2.6) and (2.7). The order parameter ϕ is computed from the first moment of the distribution functions as,

$$\phi = \sum_{i=0}^{b} f_i. \tag{2.25}$$

Once the order parameter computed, the spatial distribution of fluid properties (density ρ and dynamic viscosity μ) in the entire domain, and in particular across the interface can be expressed as,

$$\rho(\phi) = \begin{cases} \rho_2 & \phi \le \phi_2 \\ \dfrac{\phi-\phi_2}{\phi_1-\phi_2}(\rho_1-\rho_2)+\rho_2 & \phi_2 < \phi < \phi_1 \\ \rho_1 & \phi \ge \phi_1. \end{cases} \tag{2.26}$$

Rather than directly expressing the kinematic viscosity as a function of ϕ, it is defined as a function of density as,

$$\mu(\rho) = \frac{\rho-\rho_2}{\rho_1-\rho_2}(\mu_1-\mu_2)+\mu_2. \tag{2.27}$$

This was fund by Banari et al. [5] to yield more accurate results.

2.4. Numerical implementation and performance

The numerical algorithm for this multi-component LBM model is given in Algorithm 1. The entire model is implemented for a parallel solution on a GPGPU, using the CUDA language in the NVIDIA environment. In the algorithm, after initializing all the fluid properties and LBM variables, these are adjusted through initial iterations performed on the interface shape or the field variables to fulfill the governing equations. Specifically, the initially sharp interface between the two fluids specified at time step zero is smoothed out by solving the advection–diffusion equation, for the interface capturing, for a fixed number of initial iterations (e.g., 5000), while assuming a zero velocity field and adjusting interface thickness based on Eq. (2.21). Once physically relevant initial conditions are achieved, the main computational time loop is started. Note that, in the latter, to increase the efficiency of the computations, the convergence of the iterative solution of the Poisson equation is only checked every 20 iterations.

Algorithm 1 Algorithm for the LBM computation of two-fluid flow fields and phase field interface update

Set initial conditions, $\phi = \phi_i$, $\rho = \rho_i$, $\mu = \mu_i$ and $p = 0$

Initialize particle distribution functions, i.e. set
$f_i(\boldsymbol{x}, t_0) = f_i^{eq}(\boldsymbol{x}, t_0)$, $g_i(\boldsymbol{x}, t_0) = g_i^{eq}(\boldsymbol{x}, t_0)$, $h_i(\boldsymbol{x}, t_0) = h_i^{eq}(\boldsymbol{x}, t_0)$

Run initial iterations to improve the initial conditions for the phase field

for $t < t_{end}$ **do**
  Compute $f_i(\boldsymbol{x}, t + \Delta t)$ using Eq. (2.22)
  Compute $g_i(\boldsymbol{x}, t + \Delta t)$ using Eq. (2.4),
  Compute $\phi(\boldsymbol{x}, t + \Delta t)$ and $\boldsymbol{u}^*(\boldsymbol{x}, t + \Delta t)$ with Eqs. (2.25) and (2.11);
  $\rho(\boldsymbol{x}, t + \Delta t)$ and $\mu(\boldsymbol{x}, t + \Delta t)$ are calculated using Eqs. (2.26) and (2.27).
  For every 20 Poisson iterations:
  if $|\frac{p^{n+1} - p^n}{p^n}| > \varepsilon$ **then**
    Compute $p^{n+1}(\boldsymbol{x}, t + \Delta t)$ using Eq. (2.15)–(2.18).
  end if
  Compute $\boldsymbol{u}(\boldsymbol{x}, t + \Delta t)$ using Eq. (2.13).
end for

The performance of our new 3D LBM multiphase model highly depends on the number of Poisson iterations performed at each time step, which depends upon the problem physics that is tackled. For a given relative error threshold ε, we find that the number of iterations required to solve the Poisson equation is initially larger, in the first few time steps of calculations, while the field variables are still dynamically adjusting to the governing equations, but then decreases later on. After this early phase of computations we can evaluate the model efficiency in terms of the usual metric used in LBM models, the "Million of Node Updates Per Second" (*MNUPS*). On a NVIDIA TESLA 2070 GPGPU (448 CUDA cores), the performance of the current scheme is found to be around 20 *MNUPS*, using the largest possible grid that can fit in he 6 Gb RAM, with about 40% of the computational time spent for solving the Poisson equation. While this is less than for standard LBM schemes applied to a single fluid, which may achieve over 100 *MNUPS*, this is still a very respectable performance considering the higher complexity of solving large Reynolds number flows, for two fluids with a high density ratio, while tracking their 3D interface, which may have a very complex geometry in some applications (e.g., ocean breaking waves).

3. Validation

A few increasingly complex validation/benchmarking applications of our LBM scheme are presented in the following, to assess the accuracy and convergence of the presented multiphase model and confirm its relevance from investigating new physics in multi-fluid lows, particularly in he context of air–sea interactions, which is the motivation for this research. All simulations are carried out in a dimensionless framework, using non-dimensional lattice variables (here denoted by a prime) scaled on the basis of a length scale λ, time scale τ and mass scale ϖ. Thus, for the mesh parameters, $\Delta x' = \Delta x/\lambda$, $\Delta t' = \Delta t/\tau$ and $c' = c\tau/\lambda$. It is also customary in LBM to assume that $c' = 1$, which is akin to having the mesh Courant number be unity. If the length scale is further defined as $\lambda = \Delta x$, we then have $\Delta x' = 1$ and this requires $\tau = \Delta t$ and $\Delta t' = 1$ as well. Hence, with these definitions, in lattice variables, we always have $c' = \Delta x' = \Delta t' = 1$ and $c_s' = c'/\sqrt{3} = 1/\sqrt{3}$ [17]. Accordingly, the dimensionless relaxation times are defined as,

$$\tau_g' = \frac{\tau_g}{\Delta t} \quad \text{and} \quad \tau_h' = \frac{1}{2} + \frac{\rho_0'}{\rho'} \tag{3.1}$$