

Implementation of a Ternary Lattice Boltzmann Model in LAMMPS

A. K. Gokul Raman^{b,d}, James P. Andrews^c, Ulf D. Schiller^{a,b,c,*}

^a*Department of Computer & Information Sciences, University of Delaware, Newark, DE 19716, USA*

^b*Department of Materials Science and Engineering, University of Delaware, Newark, DE 19716, USA*

^c*Department of Materials Science and Engineering, Clemson University, 161 Surrine Hall, Clemson, SC 29634, USA*

^d*Department of Chemical Engineering, Indian Institute of Technology, Gandhinagar 382055, India*

Abstract

The properties of multicomponent fluids are governed by the interplay of phase behavior, fluid dynamics, and interfacial thermodynamics. A mixture formulation that leverages this interplay is an important aspect in many fabrication processes based on emulsion templating. The lattice Boltzmann method (LBM) has become a popular approach for simulating hydrodynamic effects in complex fluids and soft matter. Here we present an implementation of a ternary lattice Boltzmann model that allows to simulate a mixture of three immiscible fluids. We build on the LATBOLTZ extension of the open-source package LAMMPS and implement a ternary free energy model recently introduced by Semperebon et al. [Phys. Rev. E 93, 033305 (2016)]. We validate the static and dynamic properties by simulating liquid lenses, double emulsions, and ternary mixtures. From the simulations, we obtain the complete morphology diagram of the ternary mixture in composition space. We further discuss an application of the method to phase segregation of ternary films. The implementation of the ternary LBM in LAMMPS opens vast opportunities for mesoscale simulations of interfacial phenomena and non-equilibrium transport processes in multicomponent fluid mixtures.

*Corresponding author.
E-mail address: uschill@udel.edu

Keywords: Lattice Boltzmann, Ternary fluids, LAMMPS, Multicomponent mixtures, Liquid lens, Double emulsion, Liquid film

PROGRAM SUMMARY

Program Title: fix lb/multicomponent

CPC Library link to program files: (to be added by Technical Editor)

Developer's repository link: <https://github.com/uschille/lammps/tree/ternary-lbm>

Licensing provisions(please choose one): GPLv2

Programming language: C++

Supplementary material:

Journal reference of previous version: C. Denniston, N. Afrasiabian, M.G. Cole-André, F.E. Mackay, S.T.T. Ollila, T. Whitehead. LAMMPS lb/fluid fix version 2: Improved hydrodynamics forces implemented into LAMMPS through a lattice-Boltzmann fluid. *Comput. Phys. Commun.* **275**, 108318 (2022).

Does the new version supersede the previous version? No

Reasons for the new version: The lb/multicomponent fix provides new capabilities for simulating ternary fluid mixtures.

Summary of revisions: An implementation of the ternary free-energy lattice Boltzmann model [1] has been added to the LATBOLTZ extension of LAMMPS. The new functionality is available through the script command *fix lb/multicomponent*.

Nature of problem: Multicomponent fluids involve non-local interactions that give rise to fluid-fluid surface tension and a non-ideal pressure tensor. This requires a modification of the standard lattice Boltzmann method.

Solution method: The implementation extends the existing lattice Boltzmann code in LAMMPS [2] and introduces data structures and methods for the ternary algorithm introduced in [1]. The code carries out the streaming and collision steps for three fluid components using the appropriate equilibrium distributions. The equilibrium distribution is calculated based on thermodynamic quantities derived from the free energy.

Additional comments including restrictions and unusual features: The current version of the lb/multicomponent fix is restricted to the D3Q19 lattice.

References

- [1] C. Semprebon, T. Krüger, H. Kusumaatmaja. Ternary free-energy lattice Boltzmann model with tunable surface tensions and contact angles. *Phys. Rev. E* **93**, 033305 (2016).
- [2] C. Denniston, N. Afrasiabian, M. G. Cole-André, F. E. Mackay, S. T. T. Ollila, T. Whitehead. LAMMPS lb/fluid fix version 2: Improved hydrodynamic forces implemented into LAMMPS through a lattice-Boltzmann fluid. *Comput. Phys. Commun.* **275**, 108318 (2022).

1. Introduction

Demixing of multicomponent solutions plays an important role in phase formation and microstructure evolution in a range of materials including alloys [1, 2], polymer blends [3–5], and liquid emulsions [6–8]. Emulsion systems are extensively used in applications such as food [9], cosmetic [10], and pharmaceutical products [11]. Multicomponent fluid mixtures involving oil, water, and surfactant or gaseous phases are further relevant for advanced oil recovery [12] and mitigation of environmental pollution [13, 14]. In all these areas, mixture formulation and thermodynamics are key to the development of enhanced products and processing techniques.

In the biophysical domain, it has been noted that phase separation and interface formation play an important role in the organization and function of biological cells [15]. For instance, liquid-liquid phase separation of intrinsically disordered proteins underlies the structure and function of membraneless organelles, i.e., liquid compartments that are associated with regulation of biochemical processes in cells [16]. Accordingly, there is growing interest in leveraging phase separation mechanisms to design living materials and tissues that exhibit stimuli responsiveness or self-healing.

One particular area that has attracted considerable interest is the fabrication of porous membranes by solvent/non-solvent induced phase separation. Among the commercially relevant methods, emulsion templating continues to be an active research area for enhanced fabrication of polymeric porous media [17]. Emulsion templates are formed by phase separation, where a matrix of pores is introduced by segregation of sacrificial components. For example, polymer membranes can be fabricated by nonsolvent induced phase separation (NIPS) [18–20], where a polymer-lean phase forms pores in a polymer-rich film that is then crystallized or vitrified to create a porous microstructure.

Another recent development is the fabrication of hierarchical and asymmetric bijels (bicontinuous interfacially stabilized emulsion gels) based on solvent-transfer-induced phase separation (STRIPS) [21, 22]. In this technique, spinodal phase separation is initiated by solvent extraction from a ternary mixture, leading to bicontinuous oil-water domains. These domains can be stabilized by introducing nanoparticles that lead to interfacial jamming. The resulting bijels have mechanical and transport properties that are determined by the morphology arising during the separation and jamming process.

These processes exploit the fluid morphology that emerges during solvent/non-solvent induced phase separation, and the resulting pore size is linked to the segregation of fluid domains. An understanding of the kinetics of domain growth is thus essential for the controlled and economically viable fabrication of emulsion templates. However, our quantitative understanding of kinetics of phase segregation of ternary mixtures remains limited, and experimental investigation of the phase behavior and mass-transfer kinetics is laborious and time consuming.

Theoretical models for demixing of multicomponent solutions involve the treatment of interfaces and can be divided into sharp and diffuse interface models. In the latter, the interface emerges intrinsically from the thermodynamic formulation such that no explicit interface tracking is necessary. Diffuse interface models can deal with the inhomogeneous distribution of species and are commonly based on a phase-field formulation; the classical case is the Cahn-Hilliard model [23] that describes the mass balance and diffusion of species in a mixture. The original Cahn-Hilliard equation can be coupled to fluid dynamics by introducing advection with fluid flow, where the fluid velocity is governed by the Navier-Stokes equation.

Various computational approaches are available to solve the hydrodynamic and diffusive transport equations for multicomponent mixtures, including coarse-grained molecular dynamics [24], Monte Carlo [25, 26], dissipative particle dynamics [27], finite difference and finite element methods [28, 29], and lattice Boltzmann (LB) methods [30–34]. Several flavors of lattice Boltzmann methods for binary fluid mixtures have been introduced, such as the Shan-Chen model [30, 31], the color gradient model [35, 36], and free-energy models [32, 33]. However, most of these models have focused on the case of binary fluids, even though approaches like fluctuating hydrodynamics can in principle be formulated for an arbitrary number of components. An extension of the Shan-Chen lattice Boltzmann method to ternary mixtures has been introduced by Chen et al., but treated the third component exclusively as a surfactant. A specific multi-fluid model for ternary polymer solution was put forward by Tree et al. [37, 38] and solved numerically using a semi-implicit time stepping scheme. Recently, Semprebon et al. [39] have presented a free-energy LB model based on a general multicomponent formulation. This approach allows to independently tune the surface tensions between species by means of an appropriately parameterized free-energy functional.

Lattice Boltzmann methods are appealing for simulations of systems with

complex boundaries. They also can be coupled to molecular dynamics to simulate the behavior of colloidal and polymeric systems in a multiscale fashion. In addition, the LB algorithm is embarrassingly parallel and thus suitable for high-performance computing. The lattice Boltzmann method has thus found its way into several popular open-source software packages including LAMMPS [40], ESPResSo and waLBerla [41, 42], LB3D [43], PALABOS [44], and OpenLB [45]. Among these packages, ESPResSo and LAMMPS were the first to offer a hybrid coupling of LB and molecular dynamics [46–49]. The coupling in LAMMPS has further been extended to a dynamic coupling that allows to resolve dynamical changes in interfaces [50]. However, the more widely adopted open-source LB implementations remain focused on single-component fluids.

In this work, we present an implementation of the ternary free-energy lattice Boltzmann method in the LAMMPS software package. We adopt the free-energy formulation by Semperebon et al. [39] and extend the current LB implementation by Denniston, Karttunen, and co-workers [51, 52]. We validated the implementation by simulating liquid lenses and double emulsions and verifying that the imposed surface tensions and the measured angles at the three-phase contact line are in agreement with the Neumann triangle relation. We further verified the scaling of domain growth with time for decomposition of a symmetric ternary mixture. We demonstrate the use of the new implementation for simulations of phase separation in ternary films. Our implementation of the multicomponent-fluid model in LAMMPS can be the basis for future extensions including particle couplings with solvation interactions [53] or the incorporation of thermal fluctuations in the multicomponent LB, which will enable investigations of phase behavior and capillarity in a wide range of complex multicomponent mixtures.

2. Theoretical Formulation

2.1. Ternary Fluid Mixture

Following Semperebon et al. [39], we consider a ternary fluid mixture whose composition is specified by the (mass) concentrations of three components C_i , $i = 1, 2, 3$. Since the lattice Boltzmann method recovers the equations of motion of an incompressible fluid, we further assume the constraint $C_1 + C_2 + C_3 = \rho_0 \approx \text{constant}$. In this case, it is convenient to replace

the concentrations C_i by the density ρ and two order parameters ϕ and ψ

$$\rho = C_1 + C_2 + C_3, \quad \phi = C_1 - C_2, \quad \psi = C_3. \quad (1)$$

The fluid motion is then described by the continuity and Navier-Stokes equations

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \vec{u}) &= 0, \\ \partial_t (\rho \vec{u}) + \nabla \cdot (\rho \vec{u} \vec{u}) &= -\nabla p + \nabla \cdot \eta [\nabla \vec{u} + (\nabla \vec{u})^\top] - \rho \nabla \mu_\rho - \phi \nabla \mu_\phi - \psi \nabla \mu_\psi, \end{aligned} \quad (2)$$

where the pressure p and the chemical potential μ_ρ , μ_ϕ , and μ_ψ are thermodynamic variables that will be derived from a free energy discussed in section 2.2. The advection-diffusion of the order parameters is described by the Cahn-Hilliard equations

$$\begin{aligned} \partial_t \phi + \nabla \cdot (\phi \vec{u}) &= D_\phi \nabla^2 \mu_\phi, \\ \partial_t \psi + \nabla \cdot (\psi \vec{u}) &= D_\psi \nabla^2 \mu_\psi, \end{aligned} \quad (3)$$

where D_ϕ and D_ψ are the diffusion coefficients of the order parameters.

It is worth noting that, as a consequence of the transformation to the phase fields ρ , ϕ , and ψ , the formulation is not entirely symmetric in the components C_1 , C_2 , and C_3 of the mixture. Li and Wagner [54] have argued that manifest symmetry in the components is a theoretical necessity for an LB method to retain generality. In practice, however, the lack of symmetry in the present formulation does not noticeably affect the results (cf. section 4).

2.2. Ternary Free Energy

A common choice for immiscible fluids is a double-well free energy with bulk terms of the form $C_i^2(1 - C_i)^2$, which gives rise to two bulk minima $C_i = 0$ and $C_i = 1$. In addition, interfacial effects can be incorporated by gradient terms of the form $(\nabla C_i)^2$. Hence a suitable form of the free energy functional for a ternary fluid mixture is [39, 55]

$$F = \int_V \sum_{i=1}^3 \left[\frac{\lambda_i}{2} C_i^2 (1 - C_i)^2 + \frac{\kappa_i}{2} (\nabla C_i)^2 \right] dV. \quad (4)$$

Note that more general forms of the free energy could, in principle, include cross terms in the derivatives $\nabla C_i \cdot \nabla C_j$. However, Boyer and Lapuerta [55] have pointed out that such cross terms can only enter in the form $|\nabla \sum C_i|^2$,

which vanishes in the incompressible limit. It can be shown that for the free energy (4), the surface tension between two components m and n is given by [39]

$$\gamma_{mn} = \frac{\sqrt{(\lambda_m + \lambda_n)(\kappa_m + \kappa_n)}}{6}. \quad (5)$$

In terms of the density ρ and the order parameters ϕ and ψ , the free energy reads

$$\begin{aligned} F = \int_V & \left[\frac{\lambda_1}{32} (\rho + \phi - \psi)^2 (2 + \psi - \rho - \phi)^2 + \frac{\kappa_1}{8} (\nabla \rho + \nabla \phi - \nabla \psi)^2 \right. \\ & + \frac{\lambda_2}{32} (\rho - \phi - \psi)^2 (2 + \psi - \rho + \phi)^2 + \frac{\kappa_2}{8} (\nabla \rho - \nabla \phi - \nabla \psi)^2 \\ & \left. + \frac{\lambda_3}{2} \psi^2 (1 - \psi)^2 + \frac{\kappa_3}{2} (\nabla \psi)^2 \right] dV. \end{aligned} \quad (6)$$

The chemical potentials can be obtained by variation of the free energy functional

$$\begin{aligned} \mu_\rho &= \frac{\delta F}{\delta \rho} = \sum_{i=1}^3 \frac{\delta F}{\delta C_i} \frac{\delta C_i}{\delta \rho} \\ &= \frac{\lambda_1}{8} (\rho + \phi - \psi)(\rho + \phi - \psi - 1)(\rho + \phi - \psi - 2) \\ &\quad + \frac{\lambda_2}{8} (\rho - \phi - \psi)(\rho - \phi - \psi - 1)(\rho - \phi - \psi - 2) \\ &\quad - \frac{\kappa_1 + \kappa_2}{4} (\nabla^2 \rho - \nabla^2 \psi) - \frac{\kappa_1 - \kappa_2}{4} \nabla^2 \phi \end{aligned} \quad (7)$$

$$\begin{aligned} \mu_\phi &= \frac{\delta F}{\delta \phi} = \sum_{i=1}^3 \frac{\delta F}{\delta C_i} \frac{\delta C_i}{\delta \phi} \\ &= \frac{\lambda_1}{8} (\rho + \phi - \psi)(\rho + \phi - \psi - 1)(\rho + \phi - \psi - 2) \\ &\quad - \frac{\lambda_2}{8} (\rho - \phi - \psi)(\rho - \phi - \psi - 1)(\rho - \phi - \psi - 2) \\ &\quad - \frac{\kappa_1 - \kappa_2}{4} (\nabla^2 \rho - \nabla^2 \psi) - \frac{\kappa_1 + \kappa_2}{4} \nabla^2 \phi \end{aligned} \quad (8)$$

$$\begin{aligned}
\mu_\psi &= \frac{\delta F}{\delta \rho} = \sum_{i=1}^3 \frac{\delta F}{\delta C_i} \frac{\delta C_i}{\delta \psi} \\
&= -\frac{\lambda_1}{8}(\rho + \phi - \psi)(\rho + \phi - \psi - 1)(\rho + \phi - \psi - 2) \\
&\quad - \frac{\lambda_2}{8}(\rho - \phi - \psi)(\rho - \phi - \psi - 1)(\rho - \phi - \psi - 2) \\
&\quad + \lambda_3 \psi(1 - \psi)(1 - 2\psi) \\
&\quad + \frac{\kappa_1 + \kappa_2}{4}(\nabla^2 \rho - \nabla^2 \psi) + \frac{\kappa_1 - \kappa_2}{4} \nabla^2 \phi - \kappa_3 \nabla^2 \psi
\end{aligned} \tag{9}$$

Finally, the pressure tensor can be obtained from the relation

$$\nabla \cdot P = \nabla p + \sum_{i=1}^3 C_i \nabla \mu_i = \nabla p + \rho \nabla \mu_\rho + \phi \nabla \mu_\phi + \psi \nabla \mu_\psi, \tag{10}$$

where the second equality follows from the linearity of the variable transformation (1). The bulk pressure p is given by

$$\begin{aligned}
p &= \rho c_s^2 + (\lambda_1 + \lambda_2) \left[\frac{3}{32} \rho^2 + \frac{3}{32} \phi^4 + \frac{9}{16} \rho^2 \phi^2 + \frac{9}{16} \rho^2 \psi^2 + \frac{9}{16} \phi^2 \psi^2 \right. \\
&\quad - \frac{3}{8} \rho^3 \psi - \frac{3}{8} \rho \psi^3 - \frac{3}{4} \rho \phi^2 - \frac{3}{4} \rho \psi^2 + \frac{3}{4} \rho^2 \psi + \frac{3}{4} \phi^2 \psi \\
&\quad \left. - \frac{1}{4} \rho^3 + \frac{1}{8} \rho^2 + \frac{1}{8} \phi^2 - \frac{1}{4} \rho \psi - \frac{9}{8} \rho \phi^2 \psi \right] \\
&\quad + (\lambda_1 - \lambda_2) \left[\frac{3}{8} \rho^3 \phi + \frac{3}{8} \rho^3 \psi - \frac{3}{8} \phi^3 \psi - \frac{3}{8} \phi \psi^3 - \frac{3}{4} \rho^2 \phi - \frac{3}{4} \phi \psi^2 \right. \\
&\quad \left. + \frac{1}{4} \rho \phi - \frac{1}{4} \phi \psi - \frac{1}{4} \phi^3 + \frac{9}{8} \rho \phi \psi^2 - \frac{9}{8} \rho^2 \phi \psi + \frac{3}{2} \rho \phi \psi \right] \\
&\quad + \frac{1}{4} (\lambda_1 + \lambda_2 - 8\lambda_3) \psi^3 + (\lambda_1 + \lambda_2 + 16\lambda_3) \left[\frac{3}{32} \psi^4 + \frac{1}{32} \psi^2 \right].
\end{aligned} \tag{11}$$

With the fields ρ , ϕ , ψ , hydrodynamic velocity $\vec{u}$, bulk pressure p , and the chemical potentials μ_ρ , μ_ϕ , and μ_ψ , all necessary thermodynamic quantities are available.

2.3. Ternary Lattice Boltzmann Model

The lattice Boltzmann method (LBM) is a discrete kinetic scheme for simulating diffusive and hydrodynamic transport phenomena. The variables

of the LBM are distribution functions, sometimes termed populations, that represent the mass density of fluid on a discrete lattice. Each population is associated with a discrete velocity vector $\vec{c}_i$ connecting lattice sites. In the multicomponent LBM, multiple sets of populations are used to represent the different fluid species or, alternatively, the corresponding order parameters. For the case of a ternary fluid mixture, we use the following relation between distribution functions and macroscopic variables

$$\rho(\vec{r}, t) = \sum_i f_i(\vec{r}, t), \quad \phi(\vec{r}, t) = \sum_i g_i(\vec{r}, t), \quad \psi(\vec{r}, t) = \sum_i h_i(\vec{r}, t). \quad (12)$$

The macroscopic fluid velocity is obtained as the first moment of the distribution functions f_i associated with the density ρ

$$\rho(\vec{r}, t) \vec{u}(\vec{r}, t) = \sum_i f_i(\vec{r}, t) \vec{c}_i. \quad (13)$$

The time-evolution of the distribution functions is described by the standard Bhatnagar–Gross–Krook (BGK) lattice Boltzmann equation

$$f_i(\vec{r} + \Delta t \vec{c}_i, t + \Delta t) = f_i(\vec{r}, t) - \frac{\Delta t}{\tau} [f_i(\vec{r}, t) - f_i^{\text{eq}}(\vec{r}, t)], \quad (14)$$

$$g_i(\vec{r} + \Delta t \vec{c}_i, t + \Delta t) = g_i(\vec{r}, t) - \frac{\Delta t}{\tau_\phi} [g_i(\vec{r}, t) - g_i^{\text{eq}}(\vec{r}, t)], \quad (15)$$

$$h_i(\vec{r} + \Delta t \vec{c}_i, t + \Delta t) = h_i(\vec{r}, t) - \frac{\Delta t}{\tau_\psi} [h_i(\vec{r}, t) - h_i^{\text{eq}}(\vec{r}, t)]. \quad (16)$$

The BGK approximation can be generalized to multiple relaxation times using a collision matrix instead of the relaxation time τ [56]. To complete the specification of the ternary LBM, the local equilibrium distribution functions need to be chosen. The form of the equilibrium distribution functions depends on the underlying lattice. The equilibrium functions for the D3Q19 lattice are given in appendix A.

It can be shown via a Chapman-Enskog expansion that in the continuum limit, the above LBM recovers the continuity, Navier-Stokes, and Cahn-Hilliard equations. The viscosity of the fluid and the diffusion coefficients of the order parameters are related to the relaxation times:

$$\eta = \rho c_s^2 \left(\tau - \frac{\Delta t}{2} \right), \quad D_\phi = \Gamma_\phi \left(\tau_\phi - \frac{\Delta t}{2} \right), \quad D_\psi = \Gamma_\psi \left(\tau_\psi - \frac{\Delta t}{2} \right). \quad (17)$$

3. Implementation

We have chosen to extend the single-phase lattice Boltzmann implementation in LAMMPS [51, 52]. LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [40] is a widely used open-source code, originally developed for Molecular Dynamics (MD), that offers the flexibility to extend the time integrator via calls to user defined functions. While at present, the ternary LBM model does not include a particle-fluid coupling, contributing our implementation to LAMMPS paves the way to extending the hybrid Molecular Dynamics-Lattice Boltzmann methods that are available for single-phase fluids [46, 49, 50, 57].

3.1. Data structures

The data structures are closely aligned with the ideal gas LB implementation provided in `FixLbFluid` [51, 52]. The LB populations for the three components are stored in 4D arrays that are allocated using `memory->create()`. The hydrodynamic fields ρ , ϕ , ψ and $\vec{u}$, and the thermodynamic quantities p_{bulk} , μ_ψ , and μ_ϕ are stored in similar arrays. In addition, arrays are allocated for the gradients and Laplacians of ρ , ϕ and ψ . The LB populations are thus stored in an array-of-structures. This may not be the most efficient memory layout, however, it maintains compatibility with related LAMMPS extensions [50–52] and facilitates the adoption of particle-fluid coupling. To avoid overwriting populations during the streaming operations, the updated populations are written to a second array for each component, and the pointers to the primary and secondary array are swapped at the end of the `lb_update()` method. More sophisticated streaming algorithms [58] and vectorization are left as possible future improvements.

3.2. Time stepping structure of the implementation

The implementation extends the previous `lb/fluid fix` in LAMMPS by inheriting from the class `FixLbFluid`. The initialization of the lattice data structures and the multicomponent fluid variables takes place during instantiation of the `FixLbFluid` class. The `fix` command integrates the lattice Boltzmann update in the LAMMPS via callback methods that are invoked in specific stages of a simulation. We have assumed that the timestepping is structured in terms of the velocity-Verlet method. A flowchart of the integration of the lattice Boltzmann update in LAMMPS is shown in Fig. 1. The update of the lattice populations is performed at the beginning of each time

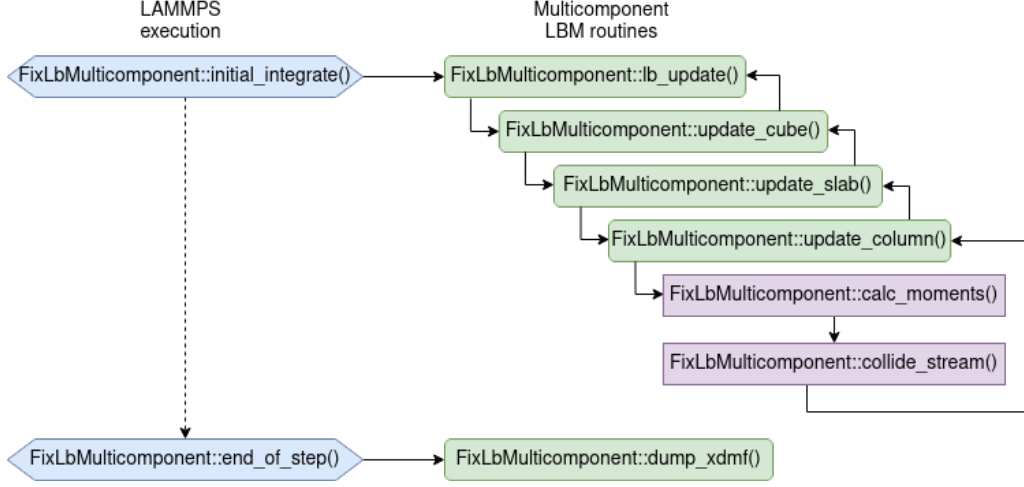


Figure 1: Flowchart illustrating the integration of *fix lb/multicomponent* into the LAMMPS time stepping. The lattice Boltzmann update is invoked during *initial_integrate()*. File I/O is performed during *end_of_step()*.

step when *initial_integrate()* is invoked (the details of the *lb_update()* method are described in more detail in the following section). This ordering allows to handle future implementations of fluid-particle couplings in the *post_integrate()*, *pre_force()*, and *post_force()* methods. Note that the hydrodynamic variables (density, velocity, pressure) are not calculated at $t + \Delta t$ in *lb_update()*, as they are only required when writing to file. The output method *dump_xdmf()* is invoked during the *end_of_step()* method. At configurable intervals, the hydrodynamic variables are calculated and written to disk in the XDMF file format.

3.3. Pseudo-code of implementation

The update of the lattice Boltzmann variables is implemented as a push scheme [59] that performs the collisions for a lattice site and then streams the post-collisional distributions to neighboring lattice sites according to the discrete velocity set (the current implementation supports the D3Q19 lattice). In contrast to ideal gas LB models, however, the interactions between different fluid components require the evaluation of the gradients and Laplacians of the fields ϕ and ψ and thus render the collision step non-local. Therefore, data dependencies need to be considered in both the collision step and the streaming step. The pseudo-code of the implementation is shown in Algorithm 1. To avoid having to loop over the full lattice more than once, the

Algorithm 1 Ternary Lattice Boltzmann Algorithm

```

 $(x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max}) \leftarrow (0, L_x + 2h_x, 0, L_y + 2h_y, 0, L_z + 2h_z)$ 

for  $n = 1$  to  $T$  do                                      $\triangleright$  loop over  $T$  timesteps
    HALO_COMM                                            $\triangleright$  halo communication between processors
    READ_SLAB( $x_{\min}, y_{\min}, y_{\max}, z_{\min}, z_{\max}$ )
    READ_SLAB( $x_{\min} + 1, y_{\min}, y_{\max}, z_{\min}, z_{\max}$ )

    for  $x = x_{\min} + 2$  to  $x_{\max}$  do                        $\triangleright$  loop over 2D slabs
        READ_COLUMN( $x, y_{\min}, z_{\min}, z_{\max}$ )
        READ_COLUMN( $x, y_{\min} + 1, z_{\min}, z_{\max}$ )

        for  $y = y_{\min} + 2$  to  $y_{\max}$  do                    $\triangleright$  loop over 1D columns
            READ_SITE( $x, y, z_{\min}$ )
            READ_SITE( $x, y, z_{\min} + 1$ )

            for  $z = z_{\min} + 2$  to  $z_{\max}$  do                $\triangleright$  loop over sites
                 $\rho, \psi, \phi, \vec{u}, p \leftarrow \text{CALC\_MOMENTS}(x, y, z)$ 
                 $f^{\text{eq}}, g^{\text{eq}}, h^{\text{eq}} \leftarrow \text{CALC\_EQUILIBRIUM}(x - 1, y - 1, z - 1)$ 

                for  $i = 1$  to  $Q$  do                            $\triangleright$  loop over lattice velocities  $\vec{c}_i$ 
                     $f_i(x - 1 + c_{ix}, y - 1 + c_{iy}, z - 1 + c_{iz}) \leftarrow$ 
                         $f_i(x - 1, y - 1, z - 1) - \frac{1}{\tau_r} [f_i(x - 1, y - 1, z - 1) - f_i^{\text{eq}}]$ 
                     $g_i(x - 1 + c_{ix}, y - 1 + c_{iy}, z - 1 + c_{iz}) \leftarrow$ 
                         $g_i(x - 1, y - 1, z - 1) - \frac{1}{\tau_p} [g_i(x - 1, y - 1, z - 1) - g_i^{\text{eq}}]$ 
                     $h_i(x - 1 + c_{ix}, y - 1 + c_{iy}, z - 1 + c_{iz}) \leftarrow$ 
                         $h_i(x - 1, y - 1, z - 1) - \frac{1}{\tau_s} [h_i(x - 1, y - 1, z - 1) - h_i^{\text{eq}}]$ 
                end for
            end for
        end for
    end for

```



Figure 2: Illustration of the staggered lattice update. When the update loop arrives at site (x, y) , the sites with lower indices (green and purple) have already been read. Once the site (x, y) (yellow) has been read, site $(x - 1, y - 1)$ (purple) can be streamed to its neighbors. The illustration shows the situation in 2D; the generalization to 3D is straightforward.

collide-stream update is implemented in a staggered fashion as illustrated in Fig. 2. Essentially, the fields needed for the gradient calculations are read ahead at a lattice site (x, y, z) , and the collisions are then performed at the previously visited lattice sites $(x - 1, y - 1, z - 1)$. We have split the full 3D loop over the lattice into separate methods for reads and updates of a 3D domain (`update_cube`), a 2D slab (`read_slab/update_slab`), a 1D column (`read_column/update_column`), and a single lattice site (`read_site/write_site`). It is important to note that these methods only stream lattice sites in the interior of the passed domain, leaving one lattice layer in all directions unstreamed and to be handled by boundary conditions. These lattice layers are part of a halo-region around the inner domain that serves both the parallelization and periodic boundary conditions. The halo regions consist of two lattice layers in all directions to facilitate the calculation of gradients and Laplacians. The halo regions are communicated to neighboring MPI rank through `MPI_Isend` and `MPI_Irecv` calls. Periodic boundary conditions are accounted for during the construction of the MPI rank neighbor lists in LAMMPS. The design of the loops and the halo regions facilitates future implementation of other collision models (e.g., multi-relaxation time models) and enhanced streaming kernels with reduced memory requirements [58]. The MPI communication routines are designed to allow overlapping of communication and computation. We have explored the effect of overlapping computation and communication on the performance of the current implementation, however, this did not lead to a significant speedup for the systems investigated in this work. Therefore, the code distributed with this paper performs the communication before the lattice update and, by default, does not overlap communication and computation. A thorough exploration

init <method>	ternary fluid configuration
init droplet	binary droplet composed of C_1 and C_2 ($C_3 = 0$)
init liquid_lens	liquid lens of component C_3 between layers of C_1 and C_2
init double_emulsion	double emulsion droplet of components C_1 and C_2 surrounded by C_3
init film	ternary film of composition $C_{1,f}$, $C_{2,f}$, $C_{3,f}$ surrounded by bulk mixture
init mixed_droplet	ternary droplet of composition $C_{1,d}$, $C_{2,d}$, $C_{3,d}$ surrounded by bulk mixture

Table 1: The *lb/multicomponent* fix provides several initialization methods to set up a ternary fluid with a specific configuration including liquid lenses, double emulsions, and ternary films.

of possible performance improvements of the current implementation is left for future work.

3.4. Installation

The ternary LBM is integrated in LAMMPS as a user package. It can be installed in an existing LAMMPS instance by downloading the files distributed with this paper. The source files for the *lb/multicomponent* fix and the patched *lb/fluid* fix should be placed in the directory *src/LATBOLTZ* within the LAMMPS source tree. A full version of LAMMPS including the ternary LBM can also be obtained from the authors' development branch on Github at <https://github.com/uschille/lammps/tree/ternary-lbm>. The package can be included in the LAMMPS build using either GNU Make (`make yes-latboltz`) or CMake (`cmake -D PKG_MPI=yes PKG_LATBOLTZ=yes`). These commands copy the package files into the main source tree and LAMMPS can then be compiled following the steps described in the LAMMPS documentation.

3.5. Script commands

The ternary lattice Boltzmann method can be invoked using the script command *lb/multicomponent*. The syntax of the command is as follows:

```
fix id group lb/multicomponent nevery viscosity density
D3Q19 dx 1 dumpxdmf <output_interval> <tag>
kappa1 <kappa1> kappa2 <kappa2> kappa3 <kappa3>
tau_r <tau_r> tau_p <tau_p> tau_s <tau_s>
C1 <C1> C2 <C2> C3 <C3>
init <init_method>
```

The first 9 positional arguments of the fix command are parsed in the constructor of *FixLbFluid*. They are required to set the parameters required to create the LB lattice in the parent constructor and should be given in the form

compatible with the `lb/fluid` fix. Parameters beyond the 9th positional argument are optional and are parsed by the new class `FixLbMulticomponent`. They can be used to set the surface tensions (`kappa1`, `kappa2`, `kappa3`), viscosity (`tau_r`) and mobilities of the ternary fluid components (`tau_p`, `tau_s`). In addition, it is possible to specify the bulk composition (`C1`, `C2`, `C3`) of the ternary mixture. By default, the mixture is initialized with the bulk concentrations to which a random perturbation is added to initiate demixing. This can also be explicitly specified with the `init mixture` argument. In addition, the `fix lb/multicomponent` command provides several options for initialization of specific fluid configurations (cf. Table 1).

3.6. Extensibility

The new LBM code was designed such that future enhancements and variations of the multicomponent model can easily be implemented. This includes possible optimizations of the data structures through vectorization and/or loop blocking, and the use of alternative collision models such as multi-relaxation time (MRT) collision kernels or the entropic lattice Boltzmann method [60]. Furthermore, our implementation was designed to mirror the data structures of the `lb/fluid` fix in order to facilitate interoperability with the available particle-fluid interactions [48, 49]. For instance, we anticipate that the `lb/viscous` fix can be extended to the multicomponent LB implementation, although the particles will only interact with the fluid through viscous drag. A possible future extension is the incorporation of solvation forces between particles and the three fluid components [53].

4. Results and Discussion

We validated the implementation by analyzing the static and dynamic properties of ternary fluid mixtures for a few simple test cases. Furthermore, to demonstrate an application of the method, we present results for the separation of ternary films surrounded by a bulk solvent phase.

4.1. Validation

One of the advantages of the ternary free energy model is that the surface tensions can be tuned directly through the coefficients κ_i of the square-gradient terms. As outlined by Semprebon et al. [39], the expression for the surface tension between two phases m and n is

$$\gamma_{mn} = \frac{\sqrt{(\kappa'_m + \kappa'_n)(\kappa_m + \kappa_n)}}{6} = \frac{\alpha}{6}(\kappa_m + \kappa_n). \quad (18)$$

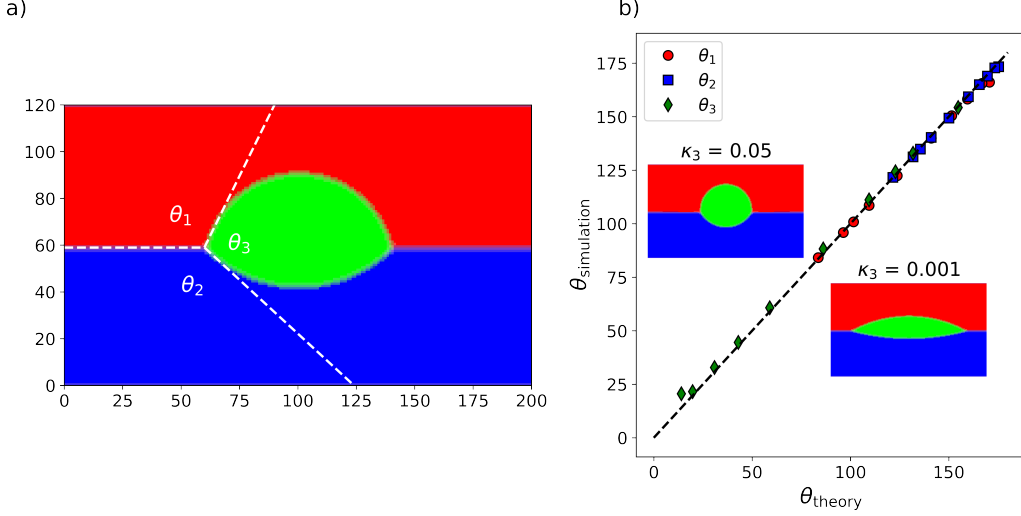


Figure 3: Liquid lens: a) Illustration of the tangent lines and Neumann angles at the three phase contact line. b) Correlation of the measured angles and the analytical solution of the Neumann triangle relation. The insets show the liquid lens configuration at $\kappa_3 = 0.05$ and $\kappa_3 = 0.001$, respectively.

At the three phase contact line, force balance between the three surface tensions can be expressed in terms of the Neumann triangle relation

$$\frac{\gamma_{12}}{\sin \theta_3} = \frac{\gamma_{23}}{\sin \theta_1} = \frac{\gamma_{31}}{\sin \theta_2}. \quad (19)$$

To validate the static properties of the implementation, we have obtained the angles θ_1 , θ_2 , and θ_3 for liquid lenses and double emulsion droplets.

4.1.1. Three-phase contact angles for liquid lenses and double emulsions

We performed simulations of liquid lenses and double emulsions in a quasi-2D simulation box of size $L \times H \times T$ with $H = 120\Delta x$ and $T = 6\Delta x$ where Δx is the lattice spacing (in our implementation, at least six lattice sites are required for periodic boundary conditions). The size L of the simulation box was varied to accommodate the length of the stable liquid lenses ($L = 200\Delta x$ for $\kappa_3 < 0.0002$ and $L = 260$ for $\kappa_3 \leq 0.0002$). The surface tension parameters $\kappa_1 = 0.01$ and $\kappa_2 = 0.02$ were kept fixed, while the third parameter κ_3 was varied to produce different surface tensions and Neumann angles. For liquid lenses, we set up a droplet of pure component C_3 with its center place at the interface between layers of pure components C_1 and

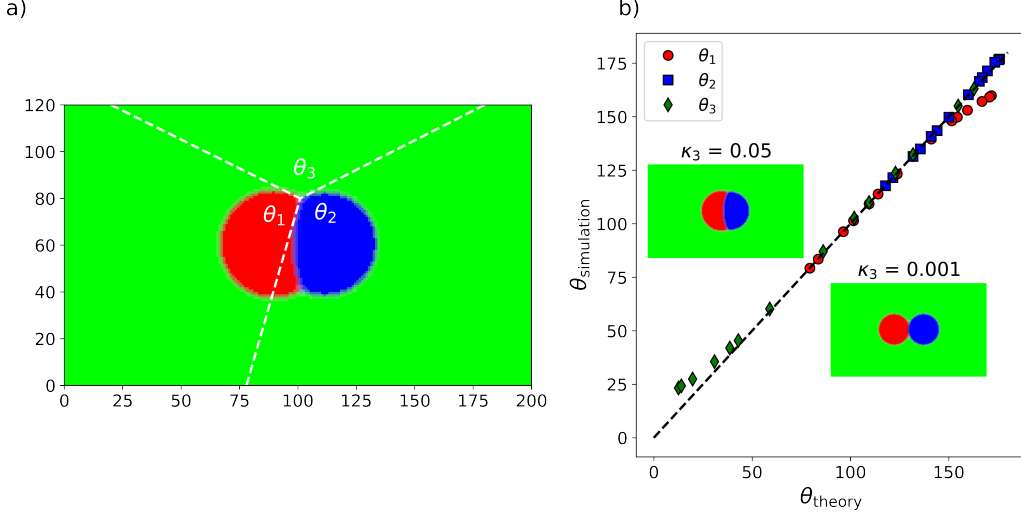


Figure 4: Double emulsion: a) Illustration of the tangent lines and Neumann angles at the three phase contact line. b) Correlation of the measured angles and the analytical solution of the Neumann triangle relation. The insets show the double emulsion configuration at $\kappa_3 = 0.05$ and $\kappa_3 = 0.001$, respectively.

C_2 . For double emulsion droplets, we set up a droplet consisting of one half of pure component C_1 and one half of pure component C_2 in a bulk fluid of pure component C_3 . We ran the simulations until the liquid lens or double emulsion configuration had equilibrated.

To measure the Neumann angles, we determined the isocontours of the fields ϕ and ψ and fitted circles to each segment corresponding to interfaces between two components. We then calculated the tangent slopes at the intersection points to calculate the angles as illustrated in Fig. 3a and Fig. 4a.

Fig. 3b and Fig. 4b show the correlation of the measured and predicted Neumann angles for the liquid lens and double emulsion, respectively. The results show excellent agreement of the Neumann angles over the range of simulated κ_3 values. Only for values that yield a Neumann angle $\theta_3 \lesssim 10^\circ$, deviations between the measured and predicted become noticeable. This is due to the diffuse nature of the interface in the LBM [39], which prevents an accurate measurement of the angle in these configurations. These results indicate that the implementation produces the correct equilibrium thermodynamics of the ternary free energy (4).

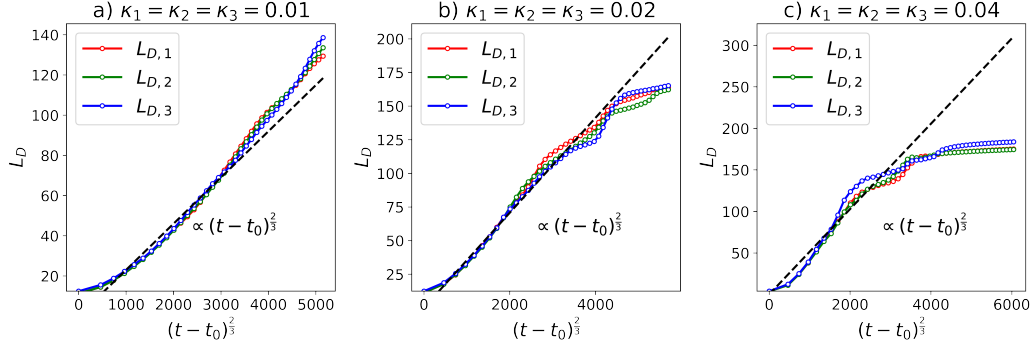


Figure 5: Domain growth during ternary phase separation. The plots show the domain size as a function of scaled time $(t - t_0)^{2/3}$, where the offset t_0 is obtained from the fitting procedure. The domain growth follows the expected hydrodynamic scaling until a domain network forms that slows down coarsening.

4.1.2. Domain growth

To validate the dynamic behavior of the multicomponent mixture, we simulated spinodal decomposition of ternary mixtures with equal concentrations of the three components $C_1 = C_2 = C_3 = 1/3$ and equal surface tension parameters $\kappa_1 = \kappa_2 = \kappa_3 = 0.04$. Since the system is governed by hydrodynamics, we expect to observe the inertial regime of domain growth. In this case, the characteristic velocity for the length scale L is given by $u = (\sigma/\rho)^{1/2} L^{-1/2}$ which yields the theoretical scaling of the domain size $L \sim ut \sim (\sigma/\rho)^{1/3} t^{2/3}$ [61, 62]. In contrast, the theoretical scaling of the domain size in the viscous regime is $L \sim (\sigma/\eta)t$.

To calculate the domain size, we use the fact that for well developed interfaces, the quantity $\chi_m = \langle C_m^2 (1 - C_m)^2 \rangle 6V/\alpha$ represents the area of the diffuse interfaces that separate component m from the other two components in the system volume V . We can thus estimate the characteristic domain size as $L = V/\chi_m$.

Fig. 5 shows the evolution of the domain size in the simulations for three different surface tensions. In all cases the theoretical scaling $L \sim (t - t_0)^{2/3}$ is observed from the onset of spinodal separation at t_0 (once the diffuse interfaces have completely formed). After the spinodal stage the coarsening dynamics changes due to the formation of a foam-like ternary network. The domain growth continues in a more step-like manner which suggests that coarsening proceeds by ripening and coalescence of domains. The onset of different coarsening mechanisms depends on the surface tension κ_i .

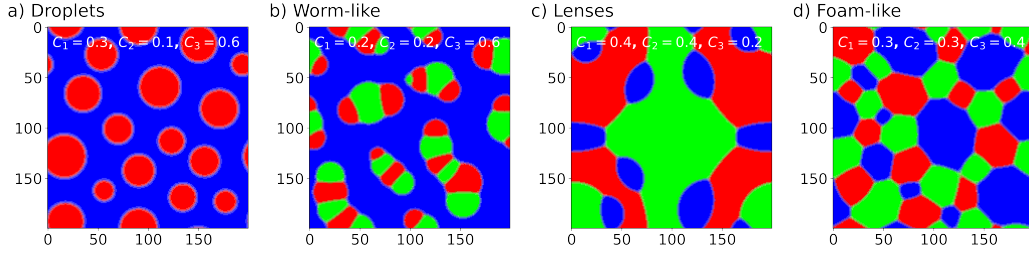


Figure 6: Examples of ternary morphologies that emerge at different mixture compositions. a) Droplets emerge for nearly binary mixtures. b) Worm-like double emulsions form when the minority components have equal concentration. c) Liquid lenses form when one minority component is less abundant. d) A foam-like emulsion emerges for nearly symmetric composition of the three components.

4.2. Ternary Morphology Diagram

To demonstrate the range of morphologies that can emerge during spinodal decomposition of ternary mixtures, we have performed exhaustive simulations of mixtures with different compositions. Fig. 6 shows examples of distinct morphologies that emerge at different compositions.

When only a small amount of the third component is present (i.e., the mixtures is nearly binary), a droplet emulsion is formed as shown in Fig. 6a. When two minority components are present at equal concentration, a peculiar double-emulsion emerges where the minority components form worm-like structures as shown in Fig. 6b. In contrast, when two majority components are present at equal concentration, the third component forms liquid lenses at the interface between the majority components as shown in Fig. 6c. When the concentrations of the three components are nearly equal, we find a foam-like network of separated domains as shown in Fig. 6d. The figures shows that the separated domains are predominantly hexagonal but other shapes are also present, however, a detailed investigation of the foam structure and its properties is beyond the scope of this work.

Fig. 7 shows a ternary morphology diagram where the segregated structures have been labeled with different symbols. The classification of the structures is based on visual inspection of simulation snapshots as discussed in the previous paragraph. Since the properties of the three components are such that the ternary mixture is symmetric, in principle only a third of the points shown in the ternary diagram need to be simulated. Nevertheless, simulations were also performed for the symmetric compositions as a further validation of the implementation. This is a useful verification since the

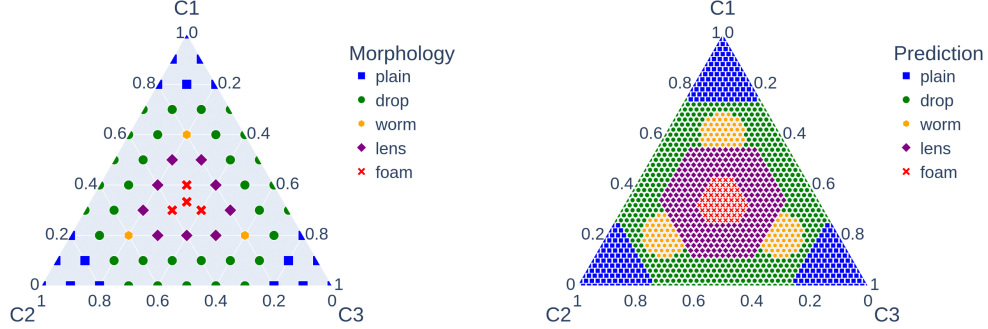


Figure 7: The ternary morphology diagram for the free energy Eq. (4). a) Each composition has been labeled according to the observed morphology (cf. Fig. 6 (single phase: blue squares, droplets: green circles, worm-like: yellow hexagons, lenses: purple diamonds, foam-like: red x-marks). b) Predicted morphologies at additional compositions. The prediction is based on a support vector machine (SVM) that was trained on the data points shown in a).

algorithm does not treat the three components in a completely symmetric fashion. The results in Fig. 8 show that, despite the lack of manifest symmetry in the formulation, the structure and dynamics of ternary mixtures remain invariant under permutations of the components.

The ternary diagram in Fig. 7 shows an approximation of the boundaries between different morphologies. The data was obtained by training a support vector machine (SVM) on the data points of Fig. 7a, using the assigned labels for minimization of the classification error. We then used the trained SVM to determine the labels for a much larger number of compositions as plotted in Fig. 7b. It is worth noting that this is only a first approximation of the full ternary diagram of the free energy model. Nevertheless, our approach demonstrates how the estimated phase boundaries can provide guidance on what morphologies can be expected for specific mixture formulations.

4.3. Ternary Films

Ternary phase segregation plays a central role in a number of processing techniques for structured materials systems. To demonstrate a possible application of the method, we performed simulations of the separation of ternary films surrounded by a bulk solvent phase. Similar systems are of interest for the study of solvent/non-solvent induced phase separation processes in polymer solutions can be used, e.g., for membrane fabrication [17, 37].

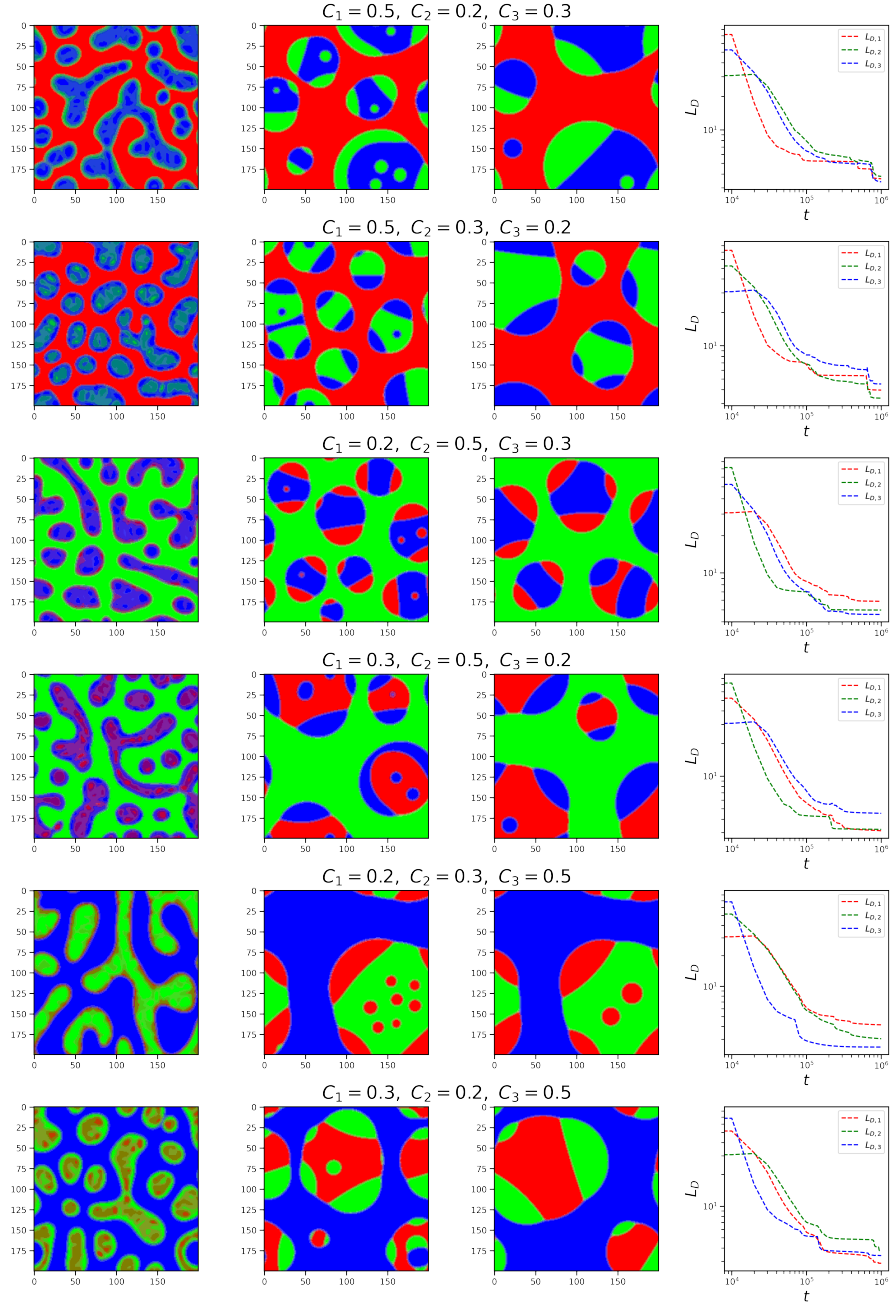


Figure 8: The structure and dynamics of ternary mixtures remains invariant under permutations of the components, although the algorithm does not treat the three components in a completely symmetric fashion. Minor differences between simulations with permuted components can be attributed to the random initial conditions and the periodic boundary conditions.

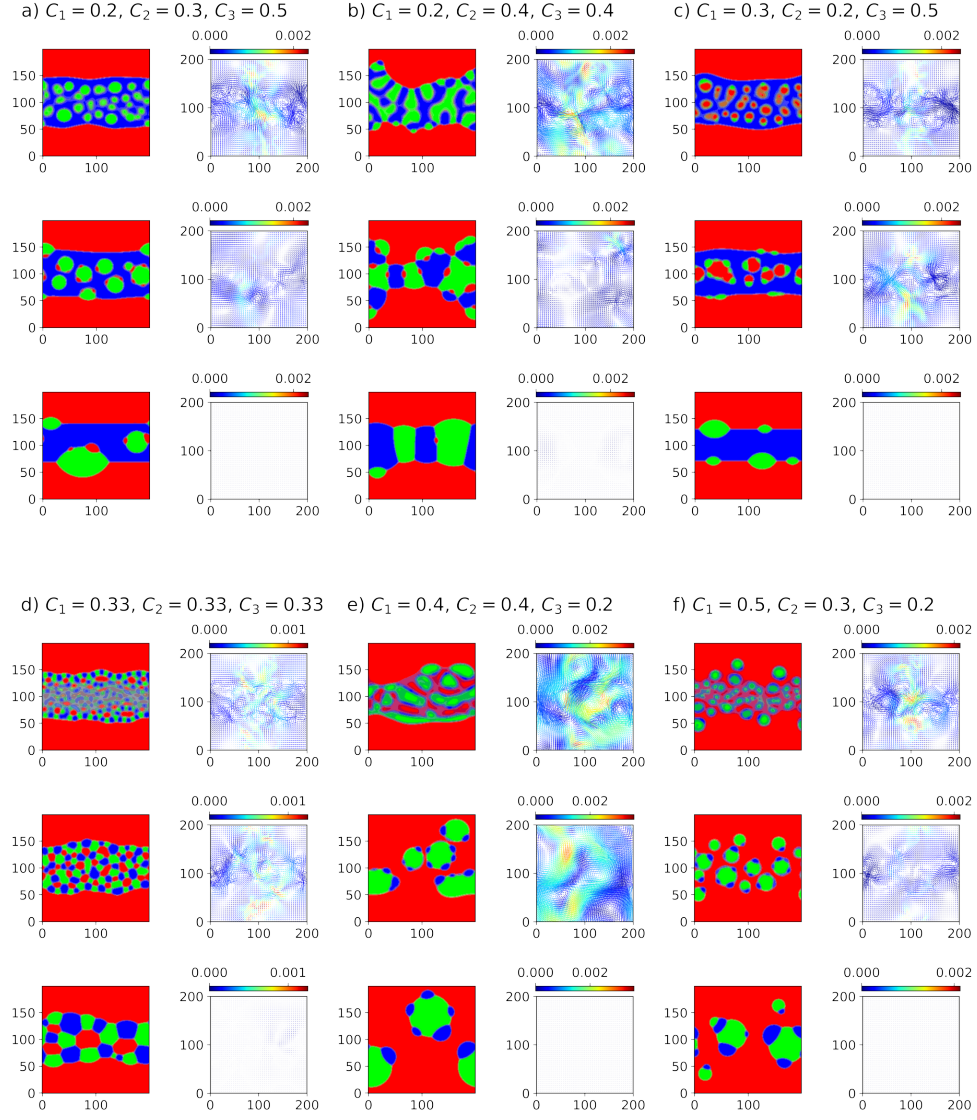


Figure 9: Phase segregation of a ternary film in contact with a bulk solvent. The subplots show the ternary film for different initial composition; the columns show snapshots of the emulsion structure and velocity field at different time steps.

We set up a ternary film of thickness $0.5L$ and variable composition around the center plane of the simulation box that is in contact with a bulk solution in the remainder of the box. The concentration difference between the film and the bulk solvent leads to diffusion while the film is separating. Fig. 9 shows snapshots of the film morphology at different times along with the fluid flow field. For small enough concentration of the bulk solvent C_1 in the initially mixed film, the film remains intact as the solvent diffuses into the bulk phase. Similar to our observation for ternary mixtures, we find that the minority components C_2 and C_3 can form liquid lenses (cf. Fig. 9a,c) and worm-like structures (cf. Fig. 9b). The latter are observed at equal concentration of C_2 and C_3 . At equal concentrations of all three components in the initial film, it segregates into a foam-like structure, where some solvent domains of C_1 remain entrapped in the network formed by C_2 and C_3 . It can be expected that these domains will very slowly diffuse into bulk phase, which is beyond the simulated time. If the concentration of the solvent C_1 in the initial film is large, the film breaks up to form an emulsion consisting of a continuous solvent phase and double emulsion droplets. The double emulsion droplets consist of droplets of the more abundant minority component, while the less abundant minority component forms liquid lenses at the surface of the droplets (cf. Fig. 9e and 9f) The size ratio of the droplets and lenses appears to depend on the concentration ratio of the minority components.

The separation of the film is accompanied by the flow patterns shown in the right columns at different times. The flow fields show curving flow patterns that correlate with the size of the emulsion domains. This is consistent with the formation of spinodal interfaces that move through the liquid. While a detailed analysis of the flow patterns is beyond the scope of this work, it is worth noting that roll cells in the velocity field have been associated with macrovoid formation [38] and we anticipate that the ternary LB model is suited to contribute further insight along these lines.

4.4. Performance and Scalability

Finally, we briefly report on the performance of our implementation. Benchmark simulations were run for phase separation of a symmetric ternary mixture in a box of size $L = 256^3$. The simulations were run on Clemson University's Palmetto cluster using Dell R750 nodes equipped with Intel Xeon 8358 processors (64 cores) and interconnected with Infiniband HDR network. The number of allocated cores was varied from 16 to 2048, and for

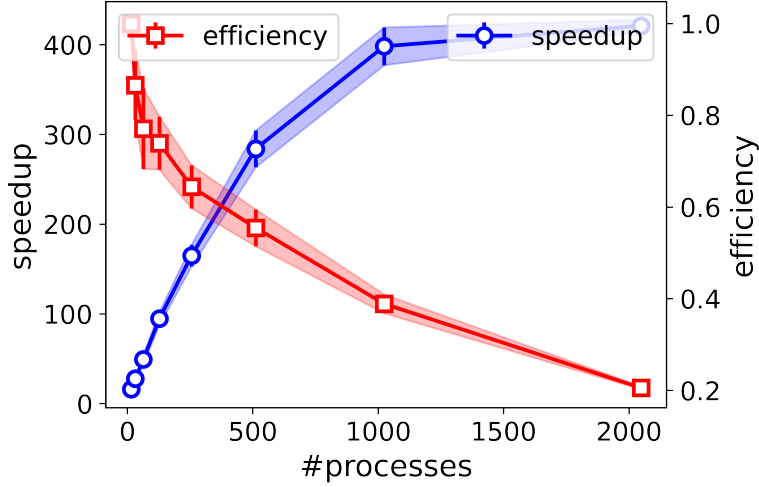


Figure 10: Performance benchmark of the ternary lattice Boltzmann implementation. Speedup and efficiency are shown as a function of the number of MPI processes in the range from 64 to 2048 cores. The error bars indicate the standard deviation over five identical runs.

each selection the simulations were repeated five times to account for potential contention on the cluster. Fig. 10 shows the speedup and efficiency. Speedup was calculated as the ratio of run times $(16 \cdot t_{16})/t_p$ where t_{16} is the average run time on 16 cores and t_p the execution time on p cores. Efficiency was calculated as the ratio $(16 \cdot t_{16})/(p \cdot t_p)$. The results in Fig. 10 show that worthwhile speedups can be achieved with up to 512 cores. Beyond 512 cores, the efficiency drops below 0.5, and virtually no speedup is gained by increasing the number of cores from 1024 to 2048. These results are satisfactory for practical purposes: using 512 cores, for instance, a ternary LB simulation completes 1 million time steps in 2-3 hours. Nevertheless, there is room for improvement of the scalability for larger systems and we have alluded to some opportunities in section 3.

5. Summary

We have introduced an implementation of a multicomponent lattice Boltzmann method based on a free energy formulation for ternary fluid mixtures. The method was implemented in LAMMPS as an extension of the LATBOLTZ package described elsewhere [52]. We outlined the theoretical formulation of

the ternary lattice Boltzmann method for a particular free energy and described the implementation of the algorithm as a user fix for the LAMMPS package. The implementation exhibits satisfactory parallel performance for system sizes that are typically used in soft matter/complex fluids simulations. We have validated the implementation by checking the Neumann relation for the three phase contact line of liquid lenses and double emulsion droplets, and by verifying the correct scaling regime of the domain size with time. We have used the simulation to map out the morphology diagram of ternary mixtures over the full range of compositions. In addition, we demonstrated that the implementation can be used to study the phase morphologies and velocity patterns that emerge during segregation of ternary films.

We anticipate that the ternary LBM provides an avenue for simulations of complex phase behavior in a broad range of emulsion systems, including solvent/non-solvent induced phase separation processes that are relevant for practical applications in chemical engineering and materials science. The implementation also paves the way to future extensions of the method such as particle-fluid coupling for fluid mixtures, thermal fluctuations in multi-component fluids, or reactive flows. This will open up new possibilities for mesoscale simulations of interfacial phenomena and non-equilibrium transport processes in multiphase and multicomponent fluid mixtures.

Declaration of competing interest

The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Colin Denniston and Mikko Karttunen for valuable discussions on their implementation of the single-phase LBM in LAMMPS. The authors acknowledge an attempt by Fang Wang to implement the ternary LBM in LAMMPS. Clemson University is acknowledged for generous allotment of compute time on the Palmetto cluster. This work was supported in part by the National Science Foundation under NSF Award DMR-1944942 and OIA-2131996. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

CRediT Author Statement

AKGR: Investigation, Data Curation, Visualization, Writing - Review & Editing.

JPA: Visualization, Formal Analysis.

UDS: Conceptualization, Methodology, Software, Investigation, Data Curation, Visualization, Formal Analysis, Supervision, Project Administration, Funding Acquisition, Writing - Original Draft, Writing - Review & Editing.

Appendix A. Equilibrium distributions of the D3Q19 model

The implementation presented in this work uses the D3Q19 model. The local equilibrium distribution functions for the D3Q19 model are given as [39]

$$\begin{aligned}
f_i^{\text{eq}} = & w_i \left(\frac{p_b}{c_s^2} + \frac{\rho \vec{u} \cdot \vec{c}_i}{c_s^2} + \frac{\rho \vec{u} \vec{u} : (\vec{c}_i \vec{c}_i - c_s^2 \mathbf{1})}{2c_s^4} \right) \\
& - \frac{w_i}{c_s^2} (\kappa_{\rho\rho} \rho \nabla^2 \rho + \kappa_{\phi\phi} \phi \nabla^2 \phi + \kappa_{\psi\psi} \psi \nabla^2 \psi) \\
& - \frac{w_i}{c_s^2} (\kappa_{\rho\phi} \rho \nabla^2 \phi + \kappa_{\rho\psi} \rho \nabla^2 \psi + \kappa_{\phi\psi} \phi \nabla^2 \psi) \\
& + \frac{\kappa_{\rho\rho}}{c_s^2} (w_i^{xx} \partial_x \rho \partial_x \rho + w_i^{yy} \partial_y \rho \partial_y \rho + w_i^{zz} \partial_z \rho \partial_z \rho \\
& + w_i^{xy} \partial_x \rho \partial_y \rho + w_i^{yz} \partial_y \rho \partial_z \rho + w_i^{zx} \partial_z \rho \partial_x \rho) \\
& + \frac{\kappa_{\phi\phi}}{c_s^2} (w_i^{xx} \partial_x \phi \partial_x \phi + w_i^{yy} \partial_y \phi \partial_y \phi + w_i^{zz} \partial_z \phi \partial_z \phi \\
& + w_i^{xy} \partial_x \phi \partial_y \phi + w_i^{yz} \partial_y \phi \partial_z \phi + w_i^{zx} \partial_z \phi \partial_x \phi) \\
& + \frac{\kappa_{\psi\psi}}{c_s^2} (w_i^{xx} \partial_x \psi \partial_x \psi + w_i^{yy} \partial_y \psi \partial_y \psi + w_i^{zz} \partial_z \psi \partial_z \psi \\
& + w_i^{xy} \partial_x \psi \partial_y \psi + w_i^{yz} \partial_y \psi \partial_z \psi + w_i^{zx} \partial_z \psi \partial_x \psi) \\
& + \frac{2\kappa_{\rho\phi}}{c_s^2} (w_i^{xx} \partial_x \rho \partial_x \phi + w_i^{yy} \partial_y \rho \partial_y \phi + w_i^{zz} \partial_z \rho \partial_z \phi) \\
& + \frac{\kappa_{\rho\phi}}{c_s^2} (w_i^{xy} \partial_x \rho \partial_y \phi + w_i^{xy} \partial_x \phi \partial_y \rho + w_i^{yz} \partial_y \rho \partial_z \phi \\
& + w_i^{yz} \partial_y \phi \partial_z \rho + w_i^{zx} \partial_z \rho \partial_x \phi + w_i^{zx} \partial_z \phi \partial_x \rho) \\
& + \frac{2\kappa_{\rho\psi}}{c_s^2} (w_i^{xx} \partial_x \rho \partial_x \psi + w_i^{yy} \partial_y \rho \partial_y \psi + w_i^{zz} \partial_z \rho \partial_z \psi) \\
& + \frac{\kappa_{\rho\psi}}{c_s^2} (w_i^{xy} \partial_x \rho \partial_y \psi + w_i^{xy} \partial_x \psi \partial_y \rho + w_i^{yz} \partial_y \rho \partial_z \psi \\
& + w_i^{yz} \partial_y \psi \partial_z \rho + w_i^{zx} \partial_z \rho \partial_x \psi + w_i^{zx} \partial_z \psi \partial_x \rho) \\
& + \frac{2\kappa_{\phi\psi}}{c_s^2} (w_i^{xx} \partial_x \phi \partial_x \psi + w_i^{yy} \partial_y \phi \partial_y \psi + w_i^{zz} \partial_z \phi \partial_z \psi) \\
& + \frac{\kappa_{\phi\psi}}{c_s^2} (w_i^{xy} \partial_x \phi \partial_y \psi + w_i^{xy} \partial_x \psi \partial_y \phi + w_i^{yz} \partial_y \phi \partial_z \psi \\
& + w_i^{yz} \partial_y \psi \partial_z \phi + w_i^{zx} \partial_z \phi \partial_x \psi + w_i^{zx} \partial_z \psi \partial_x \phi),
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
g_i^{\text{eq}} &= w_i \left(\frac{\Gamma_\phi \mu_\phi}{c_s^2} + \frac{\phi \vec{u} \cdot \vec{c}_i}{c_s^2} + \frac{\phi \vec{u} \vec{u} : (\vec{c}_i \vec{c}_i - c_s^2 \mathbf{1})}{2c_s^4} \right), \\
h_i^{\text{eq}} &= w_i \left(\frac{\Gamma_\psi \mu_\psi}{c_s^2} + \frac{\psi \vec{u} \cdot \vec{c}_i}{c_s^2} + \frac{\psi \vec{u} \vec{u} : (\vec{c}_i \vec{c}_i - c_s^2 \mathbf{1})}{2c_s^4} \right).
\end{aligned} \tag{A.2}$$

The above expressions apply for $i > 0$, and the equilibrium distributions for $i = 0$ are computed such that the relations (12) are satisfied.

The weight coefficients w_i and $w_i^{\alpha\beta}$ ($\alpha, \beta \in \{x, y, z\}$) also depend on the chosen lattice model. For the specific ordering of the lattice vectors in the LAMMPS code, they are given as

$$\begin{aligned}
w_{1-6} &= \frac{1}{18} \\
w_{7-18} &= \frac{1}{36} \\
w_{1,3}^{xx} &= w_{2,4}^{yy} = w_{5,6}^{zz} = \frac{5}{36} \\
w_{2,4,5,6}^{xx} &= w_{1,3,5,6}^{yy} = w_{1-4}^{zz} = -\frac{1}{9} \\
w_{7-14}^{xx} &= w_{7-10,15-18}^{yy} = w_{11-18}^{zz} = -\frac{1}{72} \\
w_{7-10}^{xx} &= w_{11-14}^{yy} = w_{15-18}^{zz} = \frac{1}{36} \\
w_{1-6}^{xy} &= w_{1-6}^{yz} = w_{1-6}^{zx} = 0 \\
w_{7,10}^{xy} &= w_{15,18}^{yz} = w_{11,14}^{zx} = \frac{1}{12} \\
w_{8,9}^{xy} &= w_{16,17}^{yz} = w_{12,13}^{zx} = -\frac{1}{12} \\
w_{11-18}^{xy} &= w_{7-14}^{yz} = w_{7-10,15-18}^{zx} = 0
\end{aligned} \tag{A.3}$$

The pressure p and the chemical potentials μ_ρ , μ_ϕ , and μ_ψ , which enter the LBM through the equilibrium functions, follow from the thermodynamic description of the ternary mixture given in section 2.2.

It can be shown that the above choice of equilibrium distributions recovers the continuity, Navier-Stokes, and Cahn-Hilliard equations. The full pressure

tensor of the model is given by

$$\begin{aligned}
P_{\alpha\beta} = & p\delta_{\alpha\beta} \\
& + \kappa_{\rho\rho} \left[(\partial_\alpha\rho)(\partial_\beta\rho) - \frac{1}{2}(\partial_\gamma\rho)^2\delta_{\alpha\beta} - \rho(\partial_\gamma\partial_\gamma\rho)\delta_{\alpha\beta} \right] \\
& + \kappa_{\phi\phi} \left[(\partial_\alpha\phi)(\partial_\beta\phi) - \frac{1}{2}(\partial_\gamma\phi)^2\delta_{\alpha\beta} - \phi(\partial_\gamma\partial_\gamma\phi)\delta_{\alpha\beta} \right] \\
& + \kappa_{\psi\psi} \left[(\partial_\alpha\psi)(\partial_\beta\psi) - \frac{1}{2}(\partial_\gamma\psi)^2\delta_{\alpha\beta} - \psi(\partial_\gamma\partial_\gamma\psi)\delta_{\alpha\beta} \right] \\
& + \kappa_{\rho\phi} [(\partial_\alpha\rho)(\partial_\beta\phi) + (\partial_\alpha\phi)(\partial_\beta\rho) \\
& \quad - (\partial_\gamma\rho)(\partial_\gamma\phi)\delta_{\alpha\beta} - \rho(\partial_\gamma\partial_\gamma\phi)\delta_{\alpha\beta} - \phi(\partial_\gamma\partial_\gamma\rho)\delta_{\alpha\beta}] \\
& + \kappa_{\rho\psi} [(\partial_\alpha\rho)(\partial_\beta\psi) + (\partial_\alpha\psi)(\partial_\beta\rho) \\
& \quad - (\partial_\gamma\rho)(\partial_\gamma\psi)\delta_{\alpha\beta} - \rho(\partial_\gamma\partial_\gamma\psi)\delta_{\alpha\beta} - \psi(\partial_\gamma\partial_\gamma\rho)\delta_{\alpha\beta}] \\
& + \kappa_{\phi\psi} [(\partial_\alpha\phi)(\partial_\beta\psi) + (\partial_\alpha\psi)(\partial_\beta\phi) \\
& \quad - (\partial_\gamma\phi)(\partial_\gamma\psi)\delta_{\alpha\beta} - \phi(\partial_\gamma\partial_\gamma\psi)\delta_{\alpha\beta} - \psi(\partial_\gamma\partial_\gamma\phi)\delta_{\alpha\beta}],
\end{aligned} \tag{A.4}$$

where the mixed coefficients are defined as follows

$$\begin{aligned}
\kappa_{\rho\rho} = \kappa_{\phi\phi} = -\kappa_{\rho\psi} &= \frac{\kappa_1 + \kappa_2}{4}, \\
\kappa_{\rho\phi} = -\kappa_{\phi\psi} &= \frac{\kappa_1 - \kappa_2}{4}, \\
\kappa_{\psi\psi} &= \frac{\kappa_1 + \kappa_2 + 4\kappa_3}{4}.
\end{aligned} \tag{A.5}$$

References

- [1] V. M. Lopez-Hirata, C. R. Escamilla-Illescas, R. Calva-Luna, M. L. Saucedo-Muñoz, E. O. Avila-Davila, J. D. Villegas-Cardenas, Thermodynamic and Kinetic Characteristics of Spinodal Decomposition in Ternary Alloys, *Frontiers in Materials* 9 (2022).
URL <https://www.frontiersin.org/articles/10.3389/fmats.2022.901421>
- [2] N. Derimow, R. Abbaschian, Liquid Phase Separation in High-Entropy Alloys—A Review, *Entropy* 20 (11) (2018) 890. doi:10.3390/e20110890.
URL <https://www.mdpi.com/1099-4300/20/11/890>

- [3] L. Leibler, Theory of Microphase Separation in Block Copolymers, *Macromolecules* 13 (6) (1980) 1602–1617. doi:10.1021/ma60078a047. URL <https://doi.org/10.1021/ma60078a047>
- [4] H. Tanaka, T. Nishi, New Types of Phase Separation Behavior during the Crystallization Process in Polymer Blends with Phase Diagram, *Physical Review Letters* 55 (10) (1985) 1102–1105. doi:10.1103/PhysRevLett.55.1102. URL <https://link.aps.org/doi/10.1103/PhysRevLett.55.1102>
- [5] P. K. Janert, M. Schick, Phase behavior of binary homopolymer/diblock blends: Temperature and chain length dependence, *Macromolecules* 31 (4) (1998) 1109–1113. doi:10.1021/ma971093g. URL <https://doi.org/10.1021/ma971093g>
- [6] Y.-T. Hu, Y. Ting, J.-Y. Hu, S.-C. Hsieh, Techniques and methods to study functional characteristics of emulsion systems, *Journal of Food and Drug Analysis* 25 (1) (2017) 16–26. doi:10.1016/j.jfda.2016.10.021. URL <http://www.sciencedirect.com/science/article/pii/S1021949816301831>
- [7] J. Church, J. G. Lundin, D. Diaz, D. Mercado, M. R. Willner, W. H. Lee, D. M. Paynter, Identification and characterization of bilgewater emulsions, *Science of The Total Environment* 691 (2019) 981–995. doi:10.1016/j.scitotenv.2019.06.510. URL <http://www.sciencedirect.com/science/article/pii/S0048969719330852>
- [8] A. J. Bray, Theory of phase-ordering kinetics, *Advances in Physics* 51 (2) (2002) 481–587. doi:10.1080/00018730110117433. URL <https://doi.org/10.1080/00018730110117433>
- [9] E. Dickinson, Food emulsions and foams: Stabilization by particles, *Current Opinion in Colloid & Interface Science* 15 (1) (2010) 40–49. doi:10.1016/j.cocis.2009.11.001. URL <http://www.sciencedirect.com/science/article/pii/S1359029409001010>

- [10] D. Venkataramani, A. Tsulaia, S. Amin, Fundamentals and applications of particle stabilized emulsions in cosmetic formulations, *Advances in Colloid and Interface Science* 283 (2020) 102234. doi:10.1016/j.cis.2020.102234.
URL <https://www.sciencedirect.com/science/article/pii/S0001868620301585>
- [11] T. F. Tadros, 2. Pharmaceutical emulsions, in: 2. Pharmaceutical emulsions, De Gruyter, 2018, pp. 19–44.
URL <https://www.degruyter.com/document/doi/10.1515/9783110587982-003/html>
- [12] M. Leverett, W. Lewis, Steady Flow of Gas-oil-water Mixtures through Unconsolidated Sands, *Transactions of the AIME* 142 (01) (1941) 107–116. doi:10.2118/941107-G.
URL <https://doi.org/10.2118/941107-G>
- [13] C. Simon, H. Dörksen, H. Dornbusch, Pollution Reduction By Diesel-Water Emulsions, *MTZ worldwide* 74 (1) (2013) 48–53. doi:10.1007/s38313-013-0009-4.
URL <https://doi.org/10.1007/s38313-013-0009-4>
- [14] E. Kan, M. A. Deshusses, Development of foamed emulsion bioreactor for air pollution control, *Biotechnology and Bioengineering* 84 (2) (2003) 240–244. doi:10.1002/bit.10767.
- [15] J. Berry, C. P. Brangwynne, M. Haataja, Physical principles of intracellular organization via active and passive phase transitions, *Reports on Progress in Physics* 81 (4) (2018) 046601. doi:10.1088/1361-6633/aaa61e.
URL <https://doi.org/10.1088/1361-6633/aaa61e>
- [16] S. Alberti, A. Gladfelter, T. Mittag, Considerations and Challenges in Studying Liquid-Liquid Phase Separation and Biomolecular Condensates, *Cell* 176 (3) (2019) 419–434. doi:10.1016/j.cell.2018.12.035.
URL <https://www.sciencedirect.com/science/article/pii/S0092867418316490>
- [17] F. Wang, P. Altschuh, L. Ratke, H. Zhang, M. Selzer, B. Nestler, Progress Report on Phase Separation in Polymer Solutions, *Advanced*

- Materials 31 (26) (2019) 1806733. doi:10.1002/adma.201806733.
 URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201806733>
- [18] C. A. Smolders, A. J. Reuvers, R. M. Boom, I. M. Wienk, Microstructures in phase-inversion membranes. Part 1. Formation of macrovoids, *Journal of Membrane Science* 73 (2) (1992) 259–275. doi:10.1016/0376-7388(92)80134-6.
 URL <https://www.sciencedirect.com/science/article/pii/0376738892801346>
- [19] P. van de Witte, P. J. Dijkstra, J. W. A. van den Berg, J. Feijen, Phase separation processes in polymer solutions in relation to membrane formation, *Journal of Membrane Science* 117 (1) (1996) 1–31. doi:10.1016/0376-7388(96)00088-9.
 URL <https://www.sciencedirect.com/science/article/pii/0376738896000889>
- [20] T. Watanabe, C. G. Lopez, J. F. Douglas, T. Ono, J. T. Cabral, Microfluidic Approach to the Formation of Internally Porous Polymer Particles by Solvent Extraction, *Langmuir* 30 (9) (2014) 2470–2479. doi:10.1021/la404506b.
 URL <https://doi.org/10.1021/la404506b>
- [21] M. F. Haase, K. J. Stebe, D. Lee, Continuous Fabrication of Hierarchical and Asymmetric Bijel Microparticles, Fibers, and Membranes by Solvent Transfer-Induced Phase Separation (STRIPS), *Advanced Materials* 27 (44) (2015) 7065–7071. doi:10.1002/adma.201503509.
 URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201503509>
- [22] M. F. Haase, H. Jeon, N. Hough, J. H. Kim, K. J. Stebe, D. Lee, Multifunctional nanocomposite hollow fiber membranes by solvent transfer induced phase separation, *Nature Communications* 8 (1) (2017) 1234. doi:10.1038/s41467-017-01409-3.
 URL <https://www.nature.com/articles/s41467-017-01409-3>
- [23] J. W. Cahn, J. E. Hilliard, Free Energy of a Nonuniform System. I. Interfacial Free Energy, *The Journal of Chemical Physics* 28 (2) (1958) 258–267. doi:10.1063/1.1744102.

- URL <http://scitation.aip.org/content/aip/journal/jcp/28/2/10.1063/1.1744102>
- [24] M. Laradji, S. Toxvaerd, O. G. Mouritsen, Molecular Dynamics Simulation of Spinodal Decomposition in Three-Dimensional Binary Fluids, *Physical Review Letters* 77 (11) (1996) 2253–2256. doi:10.1103/PhysRevLett.77.2253.
URL <https://link.aps.org/doi/10.1103/PhysRevLett.77.2253>
 - [25] E. Sander, T. Wanner, Monte Carlo Simulations for Spinodal Decomposition, *Journal of Statistical Physics* 95 (5) (1999) 925–948. doi:10.1023/A:1004550416829.
URL <https://doi.org/10.1023/A:1004550416829>
 - [26] A. Milchev, D. W. Heermann, K. Binder, Monte-Carlo simulation of the Cahn-Hilliard model of spinodal decomposition, *Acta Metallurgica* 36 (2) (1988) 377–383. doi:10.1016/0001-6160(88)90013-2.
URL <https://www.sciencedirect.com/science/article/pii/0001616088900132>
 - [27] T. Soddemann, B. Dünweg, K. Kremer, Dissipative particle dynamics: A useful thermostat for equilibrium and nonequilibrium molecular dynamics simulations, *Physical Review E* 68 (4) (2003) 046702. doi:10.1103/PhysRevE.68.046702.
URL <https://link.aps.org/doi/10.1103/PhysRevE.68.046702>
 - [28] A. Nonaka, Y. Sun, J. Bell, A. Donev, Low Mach number fluctuating hydrodynamics of binary liquid mixtures, *Communications in Applied Mathematics and Computational Science* 10 (2) (2015) 163–204. doi:10.2140/camcos.2015.10.163.
URL <https://msp.org/camcos/2015/10-2/p03.xhtml>
 - [29] A. Donev, A. Nonaka, A. K. Bhattacharjee, A. L. Garcia, J. B. Bell, Low Mach number fluctuating hydrodynamics of multispecies liquid mixtures, *Physics of Fluids* 27 (3) (2015) 037103. doi:10.1063/1.4913571.
URL <https://aip.scitation.org/doi/full/10.1063/1.4913571>
 - [30] X. Shan, H. Chen, Lattice Boltzmann model for simulating flows with multiple phases and components, *Phys. Rev. E* 47 (3) (1993) 1815–1819.
URL <http://link.aps.org/doi/10.1103/PhysRevE.47.1815>

- [31] X. W. Shan, H. D. Chen, Simulation Of Nonideal Gases And Liquid-Gas Phase-Transitions By The Lattice Boltzmann-Equation, *Phys. Rev. E* 49 (4) (1994) 2941–2948.
- [32] M. R. Swift, W. R. Osborn, J. M. Yeomans, Lattice Boltzmann Simulation Of Nonideal Fluids, *Phys. Rev. Lett.* 75 (5) (1995) 830–833.
- [33] M. R. Swift, E. Orlandini, W. R. Osborn, J. M. Yeomans, Lattice Boltzmann simulations of liquid-gas and binary fluid systems, *Phys. Rev. E* 54 (5) (1996) 5041–5052.
URL <http://link.aps.org/doi/10.1103/PhysRevE.54.5041>
- [34] A. J. Wagner, J. M. Yeomans, Phase separation under shear in two-dimensional binary fluids, *Physical Review E* 59 (4) (1999) 4366–4373.
doi:10.1103/PhysRevE.59.4366.
URL <https://link.aps.org/doi/10.1103/PhysRevE.59.4366>
- [35] D. H. Rothman, J. M. Keller, Immiscible cellular-automaton fluids, *Journal of Statistical Physics* 52 (3) (1988) 1119–1127. doi:10.1007/BF01019743.
URL <https://doi.org/10.1007/BF01019743>
- [36] A. K. Gunstensen, D. H. Rothman, S. Zaleski, G. Zanetti, Lattice Boltzmann model of immiscible fluids, *Physical Review A* 43 (8) (1991) 4320–4327. doi:10.1103/PhysRevA.43.4320.
URL <https://link.aps.org/doi/10.1103/PhysRevA.43.4320>
- [37] D. R. Tree, K. T. Delaney, H. D. Cenicerros, T. Iwama, G. H. Fredrickson, A multi-fluid model for microstructure formation in polymer membranes, *Soft Matter* 13 (16) (2017) 3013–3030. doi:10.1039/C6SM02839J.
URL <https://pubs.rsc.org/en/content/articlelanding/2017/sm/c6sm02839j>
- [38] D. R. Tree, T. Iwama, K. T. Delaney, J. Lee, G. H. Fredrickson, Marangoni Flows during Nonsolvent Induced Phase Separation, *ACS Macro Letters* 7 (5) (2018) 582–586. doi:10.1021/acsmacrolett.8b00012.
URL <https://doi.org/10.1021/acsmacrolett.8b00012>

- [39] C. Semperebon, T. Krüger, H. Kusumaatmaja, Ternary free-energy lattice Boltzmann model with tunable surface tensions and contact angles, *Physical Review E* 93 (3) (2016) 033305. doi:10.1103/PhysRevE.93.033305.
URL <https://link.aps.org/doi/10.1103/PhysRevE.93.033305>
- [40] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. Michael Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. Stevens, J. Tranchida, C. Trott, S. J. Plimpton, LAMMPS - A flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales, *Computer Physics Communications* (2021) 108171doi:10.1016/j.cpc.2021.108171.
URL <https://www.sciencedirect.com/science/article/pii/S0010465521002836>
- [41] A. Arnold, O. Lenz, S. Kesselheim, R. Weeber, F. Fahrenberger, D. Roehm, P. Košovan, C. Holm, ESPResSo 3.1: Molecular Dynamics Software for Coarse-Grained Models, in: M. Griebel, M. A. Schweitzer (Eds.), *Meshfree Methods for Partial Differential Equations VI*, no. 89 in *Lecture Notes in Computational Science and Engineering*, Springer Berlin Heidelberg, 2013, pp. 1–23.
URL http://link.springer.com/chapter/10.1007/978-3-642-32979-1_1
- [42] C. Feichtinger, S. Donath, H. Köstler, J. Götz, U. Rüde, Walberla: Hpc software design for computational engineering simulations, *Journal of Computational Science* 2 (2) (2011) 105 – 112. doi:https://doi.org/10.1016/j.jocs.2011.01.004.
URL <http://www.sciencedirect.com/science/article/pii/S1877750311000111>
- [43] S. Schmieschek, L. Shamardin, S. Frijters, T. Krüger, U. D. Schiller, J. Harting, P. V. Coveney, LB3D: A parallel implementation of the Lattice-Boltzmann method for simulation of interacting amphiphilic fluids, *Comput. Phys. Commun.* 217 (2017) 149–161. doi:10.1016/j.cpc.2017.03.013.
URL <http://www.sciencedirect.com/science/article/pii/S0010465517301017>

- [44] M. Tian, W. Gu, J. Pan, M. Guo, Performance analysis and optimization of palabos on petascale sunway bluelight mpp supercomputer, in: K. Li, Z. Xiao, Y. Wang, J. Du, K. Li (Eds.), *Parallel Computational Fluid Dynamics*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 311–320.
- [45] M. J. Krause, A. Kummerländer, S. J. Avis, H. Kusumaatmaja, D. Dapelo, F. Klemens, M. Gaedtke, N. Hafen, A. Mink, R. Trunk, J. E. Marquardt, M.-L. Maier, M. Haussmann, S. Simonis, OpenLB—Open source lattice Boltzmann code, *Computers & Mathematics with Applications* 81 (2021) 258–288. doi:10.1016/j.camwa.2020.04.033.
URL <https://www.sciencedirect.com/science/article/pii/S0898122120301875>
- [46] T. T. Pham, U. D. Schiller, J. R. Prakash, B. Dünweg, Implicit and explicit solvent models for the simulation of a single polymer chain in solution: Lattice Boltzmann vs Brownian dynamics, *J. Chem. Phys.* 131 (2009) 164114. doi:10.1063/1.3251771.
URL http://jcp.aip.org/resource/1/jcpsa6/v131/i16/p164114_s1
- [47] J. Smiatek, M. Sega, C. Holm, U. D. Schiller, F. Schmid, Mesoscopic simulations of the counterion-induced electro-osmotic flow: A comparative study, *J. Chem. Phys.* 130 (24) (2009) 244702. doi:10.1063/1.3152844.
URL <http://link.aip.org/link/?JCP/130/244702/1>
- [48] S. T. T. Ollila, C. Denniston, M. Karttunen, T. Ala-Nissila, Fluctuating lattice-Boltzmann model for complex fluids, *J. Chem. Phys.* 134 (6) (2011) 064902–19.
URL <http://link.aip.org/link/?JCP/134/064902/1>
- [49] F. E. Mackay, K. Pastor, M. Karttunen, C. Denniston, Modeling the behavior of confined colloidal particles under shear flow, *Soft Matter* 10 (43) (2014) 8724–8730. doi:10.1039/C4SM01812E.
URL <http://pubs.rsc.org/en/content/articlelanding/2014/sm/c4sm01812e>
- [50] X. Yu, M. Dutt, Implementation of dynamic coupling in hybrid Molecular Dynamics–Lattice Boltzmann approach: Modeling aggregation of

- amphiphiles, *Computer Physics Communications* 257 (2020) 107287.
doi:10.1016/j.cpc.2020.107287.
URL <https://www.sciencedirect.com/science/article/pii/S0010465520300977>
- [51] F. E. Mackay, S. T. T. Ollila, C. Denniston, Hydrodynamic forces implemented into LAMMPS through a lattice-Boltzmann fluid, *Computer Physics Communications* 184 (8) (2013) 2021–2031.
doi:10.1016/j.cpc.2013.03.024.
URL <https://www.sciencedirect.com/science/article/pii/S001046551300132X>
- [52] C. Denniston, N. Afrasiabian, M. G. Cole-André, F. E. Mackay, S. T. T. Ollila, T. Whitehead, LAMMPS lb/fluid fix version 2: Improved hydrodynamic forces implemented into LAMMPS through a lattice-Boltzmann fluid, *Computer Physics Communications* 275 (2022) 108318. doi:10.1016/j.cpc.2022.108318.
URL <https://www.sciencedirect.com/science/article/pii/S0010465522000364>
- [53] M. Sega, M. Sbragaglia, S. S. Kantorovich, A. O. Ivanov, Mesoscale structures at complex fluid-fluid interfaces: a novel lattice Boltzmann/molecular dynamics coupling, *Soft Matter* 9 (42) (2013) 10092–10107. doi:10.1039/C3SM51556G.
URL <http://dx.doi.org/10.1039/C3SM51556G>
- [54] Q. Li, A. J. Wagner, Symmetric free-energy-based multicomponent lattice Boltzmann method, *Physical Review E* 76 (3) (2007) 036701. doi:10.1103/PhysRevE.76.036701.
URL <https://link.aps.org/doi/10.1103/PhysRevE.76.036701>
- [55] F. Boyer, C. Lapuerta, Study of a three component Cahn-Hilliard flow model, *ESAIM: Mathematical Modelling and Numerical Analysis* 40 (4) (2006) 653–687. doi:10.1051/m2an:2006028.
URL <https://www.cambridge.org/core/journals/esaim-mathematical-modelling-and-numerical-analysis/article/study-of-a-three-component-cahnhilliard-flow-model/7C5A107014E1CDA4317FBED029691613>

- [56] C. M. Pooley, K. Furtado, Eliminating spurious velocities in the free-energy lattice Boltzmann method, *Physical Review E* 77 (4) (2008) 046702. doi:10.1103/PhysRevE.77.046702.
URL <http://link.aps.org/doi/10.1103/PhysRevE.77.046702>
- [57] B. Dünweg, A. J. C. Ladd, Lattice Boltzmann Simulations of Soft Matter Systems, *Advanced Computer Simulation Approaches for Soft Matter Sciences III* 221 (2009) 89–166. doi:10.1007/978-3-540-87706-6_2.
URL https://doi.org/10.1007/978-3-540-87706-6_2
- [58] M. Wittmann, T. Zeiser, G. Hager, G. Wellein, Comparison of different propagation steps for lattice Boltzmann methods, *Mesoscopic Methods in Engineering and Science* 65 (6) (2013) 924–935. doi:10.1016/j.camwa.2012.05.002.
URL <http://www.sciencedirect.com/science/article/pii/S0898122112003835>
- [59] G. Wellein, T. Zeiser, G. Hager, S. Donath, On the single processor performance of simple lattice Boltzmann kernels, *Computers & Fluids* 35 (8-9) (2006) 910 – 919, proceedings of the First International Conference for Mesoscopic Methods in Engineering and Science. doi:DOI:10.1016/j.compfluid.2005.02.008.
URL <http://www.sciencedirect.com/science/article/B6V26-4HVDYJJ-4/2/21fc67683bc889045d128a80434bc31d>
- [60] M. Wöhrwag, C. Semprebon, A. Mazloomi Moqaddam, I. Karlin, H. Kusumaatmaja, Ternary Free-Energy Entropic Lattice Boltzmann Model with a High Density Ratio, *Physical Review Letters* 120 (23) (2018) 234501. doi:10.1103/PhysRevLett.120.234501.
URL <https://link.aps.org/doi/10.1103/PhysRevLett.120.234501>
- [61] H. Furukawa, Role of inertia in the late stage of the phase separation of a fluid, *Physica A: Statistical Mechanics and its Applications* 204 (1) (1994) 237–245. doi:10.1016/0378-4371(94)90428-6.
URL <https://www.sciencedirect.com/science/article/pii/0378437194904286>
- [62] V. M. Kendon, J.-C. Desplat, P. Bladon, M. E. Cates, 3D Spinodal Decomposition in the Inertial Regime, *Physical Review Letters* 83 (3)

(1999) 576–579. doi:10.1103/PhysRevLett.83.576.
URL <https://link.aps.org/doi/10.1103/PhysRevLett.83.576>