Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2022-0053

Vol. **33**, No. 1, pp. 330-348
January 2023

Numerical Stability Analysis for a Stationary and Translating Droplet at Extremely Low Viscosity Values Using the Lattice Boltzmann Method Color-Gradient Multi-Component Model with Central Moments Formulation

Karun P. N. Datadien[1,*], Gianluca Di Staso[1,2] and Federico Toschi[1,3]

[1] *Department of Applied Physics and Science Education, Eindhoven University of Technology, 5600 MB Eindhoven, Netherlands.*
[2] *FLOW Matters Consultancy B.V., 5612AE, Eindhoven, The Netherlands.*
[3] *Istituto per le Applicazioni del Calcolo, Consiglio Nazionale delle Ricerche, 00185 Rome, Italy.*

Received 19 February 2022; Accepted (in revised version) 25 October 2022

Abstract. Multicomponent models based on the Lattice Boltzmann Method (LBM) have clear advantages with respect to other approaches, such as good parallel performances and scalability and the automatic resolution of breakup and coalescence events. Multicomponent flow simulations are useful for a wide range of applications, yet many multicomponent models for LBM are limited in their numerical stability and therefore do not allow exploration of physically relevant low viscosity regimes. Here we perform a quantitative study and validations, varying parameters such as viscosity, droplet radius, domain size and acceleration for stationary and translating droplet simulations for the color-gradient method with central moments (CG-CM) formulation, as this method promises increased numerical stability with respect to the non-CM formulation. We focus on numerical stability and on the effect of decreasing grid-spacing, i.e. increasing resolution, in the extremely low viscosity regime for stationary droplet simulations. The effects of small- and large-scale anisotropy, due to grid-spacing and domain-size, respectively, are investigated for a stationary droplet. The effects on numerical stability of applying a uniform acceleration in one direction on the domain, i.e. on both the droplet and the ambient, is explored into the low viscosity regime, to probe the numerical stability of the method under dynamical conditions.

AMS subject classifications: 76-XX

Key words: Lattice Boltzmann method, multicomponent flow, numerical stability, low viscosity.

*Corresponding author. *Email addresses:* k.p.n.datadien@tue.nl (K. P. N. Datadien), g.di.staso@tue.nl (G. Di Staso), f.toschi@tue.nl (F. Toschi)

1 Introduction

The numerical modeling of multiphase/multicomponent fluids is still a challenge and the Lattice Boltzmann Method (LBM) has shown great potential in this field [1]. Several models for simulating multiphase/multicomponent flows using the LBM have been proposed over the last three decades, including the color gradient (CG) model [2], the pseudopotential model [3], the free-energy model [4] and the mean-field model [5]. In the current work we focus on exploring the capabilities of a recent formulation of the CG model, i.e. the central moments formulation as proposed by [6], which promises increased numerical stability while retaining physical accuracy of the standard CG model formulation [7]. To explore the limits of this formulation in terms of accessible parameter range and the strength of the spurious currents produced around a droplet-ambient interface, we simulate firstly a quiescent droplet in a cubic domain with periodic boundary conditions. Although the system is stationary, spurious currents will arise due to discretization errors, concentrated mainly near to the droplet-ambient interface [8,9]. Therefore the simulated system is not perfectly static, as it should, despite the droplet being stationary. The kinetic energy, E_{kin}, is non-zero due to the presence of spurious currents, which we measure for a variety of parameters, such as material properties, droplet radius and simulation domain size.

Firstly, the viscosities of the droplet and surrounding ambient are pushed down to the point where physical accuracy is severely compromised, in order to find the limits to which the viscosity parameters can be pushed. We also consider the effect of small-scale anisotropy on the total kinetic energy E_{kin}, caused by the finite sized spacing between grid-points, Δx, adopted for simulations in LBM. By running several simulations with increasing droplet radius, $R = 8$, 12, 16, we can quantify the effect of this small-scale anisotropy, which is essentially a resolution effect. Furthermore we investigate the effect of large-scale anisotropy, caused by using periodic boundary conditions. In this setup the droplet is influenced by itself, but not equally in every direction, e.g. across the diagonal of the simulation domain, the distance of the droplet to itself is greater than across the horizontal. To quantify this effect we increase domain sidelength L, while keeping R constant, thereby increasing the distance from the droplet to the surrounding boundaries to investigate the influence of L on E_{kin}.

Finally we consider the case of a translating droplet put in motion by accelerating the entire domain, i.e. droplet and ambient, in one direction. This is done for varying densities, viscosities and acceleration values. Through this procedure we quantify a stable and unstable regime for a droplet in a moving frame of reference, thereby testing Galilean invariance of the system.

2 Numerical method: Color-gradient with central moments formulation

The color-gradient (CG) lattice Boltzmann method (LBM) multicomponent model with enhanced equilibria and a central moments (CM) formulation, as in [6], is used for all simulations. The CM formulation offers additional stability and a wider accessible parameter range [6, 7] when compared to non-CM implementations. We use a D3Q27 stencil for all simulations, for further increased stability and accuracy. Within the CG framework, to simulate multiphase or immiscible multicomponent flows, a particle distribution function is evolved according to the lattice Boltzmann equation for each of the components:

$$f_i(\vec{x}+\vec{c}\Delta t,t+\Delta t)=f_i(\vec{x},t)+\Omega_i(f_i(\vec{x},t))\Delta t \tag{2.1}$$

with $f_i(\vec{x},t)$ being the discrete particle distribution functions at lattice site $\vec{x}$ at time t. From now on we will use the standard convention that $\Delta t=1$. As usual, the density of the fluid is recovered by summing the particle distributions as

$$\rho=\sum_i f_i. \tag{2.2}$$

The velocity $\vec{u}=(u_x,u_y,u_z)$ is evaluated as the first order moment of the distribution according to the relation

$$\rho\vec{u}=\sum_i f_i\vec{c}_i. \tag{2.3}$$

The following discrete velocities $\vec{c}_i=(|c_{ix}\rangle,|c_{iy}\rangle,|c_{iz}\rangle)$ are used:

$$\begin{aligned}
|c_{ix}\rangle&=[0,1,-1,0,0,0,0,1,1,-1,-1,1,-1,1,-1,0,0,0,0,1,1,1,-1,1,-1,-1,-1]^\intercal,\\
|c_{iy}\rangle&=[0,0,0,1,-1,0,0,1,-1,1,-1,0,0,0,0,1,1,-1,-1,1,1,-1,1,-1,-1,1,-1]^\intercal,\\
|c_{iz}\rangle&=[0,0,0,0,0,1,-1,0,0,0,0,1,1,-1,-1,1,-1,1,-1,1,-1,1,1,-1,1,-1,-1]^\intercal,
\end{aligned} \tag{2.4}$$

with the notation $|\bullet\rangle$ indicating a column vector. The superscript $\intercal$ denotes the transpose operator. The Bhatnagar-Gross-Krook (BGK) collision operator implements relaxation towards the equilibrium and has the form

$$\Omega_i(f_i(\vec{x},t))=-\omega[f_i(\vec{x},t)-f_i^{eq}(\rho(\vec{x},t),\vec{u}(\vec{x},t))] \tag{2.5}$$

with ω being the rate of relaxation towards the discrete local equilibrium, f_i^{eq}, and defined as $\omega=(\nu/c_s^2+1/2)^{-1}$, with kinematic viscosity ν and lattice speed of sound c_s. The local equilibrium f_i^{eq} is calculated as:

$$f_i^{eq}(\rho(\vec{x},t),\vec{u}(\vec{x},t))=\rho w_i\left(1+\frac{3}{c^2}\vec{c}_i\cdot\vec{u}+\frac{9}{2c^4}(\vec{c}_i\cdot\vec{u})^2-\frac{3}{2c^2}(\vec{u})^2\right). \tag{2.6}$$

The weights w_i used in our implementation are $w_0=8/27$, $w_{1,\cdots,6}=2/27$, $w_{7,\cdots,18}=1/54$, $w_{19,\cdots,26}=1/216$. In addition to the collision step, the algorithm includes the streaming step:

$$f_i(\vec{x}+\vec{c}\Delta t,t+\Delta t)=f_i^*(\vec{x},t). \tag{2.7}$$

Note that the above equations are used for single-phase simulations, but for the multiphase/multicomponent simulations presented in this work we add the CG model. The central moments formulation requires a more complex calculation for the equilibrium population f_i^{eq} for each discrete velocity i [7]. Details on these calculations, derivations and a more in-depth treatment of the implementation can be found in [6] where the authors provide a Matlab script to assist in calculating all required terms for implementing the model. The core concept of the CM formulation is to shift lattice directions by the local fluid velocity, whereby we can define $\overline{\mathbf{c}}_i = [|\overline{c}_{ix}\rangle, |\overline{c}_{iy}\rangle, |\overline{c}_{iz}\rangle]$ as

$$|\overline{c}_{ix}\rangle = |c_{ix} - u_x\rangle, \quad |\overline{c}_{iy}\rangle = |c_{iy} - u_y\rangle, \quad |\overline{c}_{iz}\rangle = |c_{iz} - u_z\rangle. \tag{2.8}$$

For completeness we list here the steps required in the implementation of the single-phase CM formulation.

1. Computation of macroscopic variables by Eq. (2.3).

2. Evaluation of postcollision central moments k_i^* as given by Eq. (A.1).

3. Computation of postcollision raw moments, $|r_i^*\rangle = [r_0^*, \cdots, r_i^*, \cdots, r_{26}^*]^\intercal$, from the postcollision central moments according to, $|r_i^*\rangle = \mathbf{N}^{-1}|k_i^*\rangle$, where $\mathbf{N} = \mathbf{T}\mathbf{M}^{-1}$, with the transformation matrix $\mathbf{T}$ of the form given by Eq. (A.3) and $\mathbf{M}$ given by Eq. (A.4).

4. Reconstruction of postcollision populations as $|f_i^*\rangle = \mathbf{M}^{-1}|r_i^*\rangle$, resulting in Eq. (A.5).

5. Perform the streaming step according to Eq. (2.1).

Additional variables should be calculated when implementing the multi-component color-gradient model, such as the color-field ρ^N, speed of sound in the red (more dense) and blue (less dense) fluids, α_R and α_B and coefficients A_r and A_b used for the calculation of surface tension. In effect, the CG model can be included in the standard way and for more specifics on this, the reader is referred to Appendix D in [7] which outlines the required steps explicitly. For the inclusion of enhanced equilibrium terms the reader is further referred to [10] and [11]. The Matlab script provided in the supplementary material of [6] can then be modified to include those enhanced equilibrium terms, resulting in altered versions of the post collision central moments, k_i^*, and post collision raw moments, r_i^*, from those shown above. The altered versions are omitted here due to their cumbersome and highly complex form. Note that the discrete velocities used in this work differ from the ones reported in [6] and [7] resulting in a different set of equations for the postcollision populations in this work, i.e. Eq. (A.5), compared to those reported in [6].

In the color-gradient nomenclature the denser fluid is referred to as the red fluid and the less dense fluid as the blue fluid. The densities are therefore denoted as ρ_R and ρ_B, respectively. These densities may fluctuate depending on pressure and to that end we further define the constant initial densities (used at the start of a simulation) as ρ_R^0 and ρ_B^0.

To locate the interface in multicomponent simulations we first define the order parameter, or color-field, as

$$\rho^N = \left(\frac{\rho_R}{\rho_R^0} - \frac{\rho_B}{\rho_B^0}\right) \Big/ \left(\frac{\rho_R}{\rho_R^0} + \frac{\rho_B}{\rho_B^0}\right). \tag{2.9}$$

When, e.g. placing a droplet of density ρ_R in an ambient fluid of density ρ_B, we consider its interface position to be at $\rho^N = 0$. As an indication of computational performance we performed a side-by-side static droplet simulation using the standard CG implementation (without central moments formulation) and the CG-CM implementation. The total runtime is 114.29 s for CG and 125.71 s for CG-CM. The CG-CM model is therefore 10% slower. It should be noted that optimizations to the implementation of CG-CM are possible (code-wise), which would close the performance gap.

3 Simulations

In the following two sections, we present the static droplet simulations mentioned in the introduction for: (1) a unity density ratio $\rho_R/\rho_B = 1$ and for (2) a large density ratio $\rho_R/\rho_B = 1000$. These are followed by two sections on the translating droplet simulation cases, again for a unity density ratio $\rho_R/\rho_B = 1$ and a large density ratio $\rho_R/\rho_B = 1000$. One of the intended applications, explored in future work, is to use the CG-CM model for inkjetting simulations, using approximately the same parameters as given in Table 1, Set 5. To this end the viscosity ratio is kept constant at $\nu_R/\nu_B = 1/2$, which is realistic for the viscosity of ink, ν_R, and air, ν_B.

3.1 Static droplet

A stationary droplet with radius $R = 8, 12, 16$ is initialized in the center of a cubic domain with sides of length $L = 64, 96, 128$. Periodic boundary conditions are applied on the domain surfaces. The droplet has a density of $\rho_R = 1, 1000$ (two cases are considered) and the surrounding ambient fluid has density $\rho_B = 1$. Kinematic viscosities ν_R and ν_B of respectively the droplet and ambient, are determined by their respective relaxation times which we set to the values $\tau_R = 0.51, 0.501, 0.5001, 0.50001$ and $\tau_B = 0.52, 0.502, 0.5002, 0.50002$. The associated viscosity is calculated as $\nu = c_s^2(\tau - 0.5)$, with the lattice speed of sound $c_s = 1/\sqrt{3}$. For the parameter scans we perform a total of 72 runs for the static droplet simulations, using the parameter sets shown in Table 1 for each combination of R and L.

3.1.1 Unity density ratio $\rho_R/\rho_B = 1$

We measure the total kinetic energy of the system, E_{kin}, which in this stationary system is due to the presence of spurious currents. These currents mostly concentrate around the droplet-ambient interface, but are expected to reduce in severity with lower curvature, i.e. larger drop radius R. To be able to compare results we therefore normalize E_{kin} by the droplet surface area $A = 4\pi R^2$. Starting with a unity density ratio, $\rho_R/\rho_B = 1$, we find

Table 1: Input parameters for the static droplet simulations parameter scans: red and blue fluid density ρ_R and ρ_B, relaxation times τ_R and τ_B, kinematic viscosities ν_R and ν_B, surface tension σ, radius R and cubic domain side-length L in lattice units.

Set	ρ_R	ρ_B	τ_R	τ_B	ν_R	ν_B	σ	R	L
1	1.0	1.0	0.51	0.52	$1/3\cdot10^{-3}$	$2/3\cdot10^{-3}$	$1\cdot10^{-4}$	8, 12, 16	64, 96, 128
2	1.0	1.0	0.501	0.502	$1/3\cdot10^{-4}$	$2/3\cdot10^{-4}$	$1\cdot10^{-4}$	8, 12, 16	64, 96, 128
3	1.0	1.0	0.5001	0.5002	$1/3\cdot10^{-5}$	$2/3\cdot10^{-5}$	$1\cdot10^{-4}$	8, 12, 16	64, 96, 128
4	1.0	1.0	0.50001	0.50002	$1/3\cdot10^{-6}$	$2/3\cdot10^{-6}$	$1\cdot10^{-4}$	8, 12, 16	64, 96, 128
5	1000	1.0	0.51	0.52	$1/3\cdot10^{-3}$	$2/3\cdot10^{-3}$	$1\cdot10^{-4}$	8, 12, 16	64, 96, 128
6	1000	1.0	0.501	0.502	$1/3\cdot10^{-4}$	$2/3\cdot10^{-4}$	$1\cdot10^{-4}$	8, 12, 16	64, 96, 128
7	1000	1.0	0.5001	0.5002	$1/3\cdot10^{-5}$	$2/3\cdot10^{-5}$	$1\cdot10^{-4}$	8, 12, 16	64, 96, 128
8	1000	1.0	0.50001	0.50002	$1/3\cdot10^{-6}$	$2/3\cdot10^{-6}$	$1\cdot10^{-4}$	8, 12, 16	64, 96, 128

that in every case considered the normalized kinetic energy inside the domain, E_{kin}/A, is monotonically increasing, see Fig. 1. It is clear from these results that increasing viscosity results in significantly reduced spurious currents. There is also a slight dependence of E_{kin}/A on R, where the smallest drop radius $R=8$ results in the lowest total kinetic energy and $R=16$ the highest, with the exception of the lowest viscosity case, where $R=12$ results in the most kinetic energy content.

A difference in equilibration time is also noticeable, as the higher viscosity simulations attain a steady state sooner than the lower viscosity simulations. To further investigate this and quantify the dependence of equilibration time on viscosity, we fit the data presented in Fig. 1 according to the function $f(t)=a\cdot(1-\exp(-t/\tau_s))$. The result of several fits are shown in Fig. 2. We consider τ_s to be the characteristic equilibration timescale for this system. From the results shown in Fig. 3 we find that the characteristic kinetic energy equilibration timescale τ_s is strongly affected by viscosity values, marginally affected by R and the domain size effect is negligible. For the highest viscosity case, τ_s is two orders of magnitude larger than for the lowest viscosity case.

We understand that the buildup process of the spurious currents is associated with a timescale analogous to the one present in dissipative phenomena, where the timescale is proportional to $length^2/viscosity$. This corresponds for our setup to $\tau_s \propto R^2/\nu_R$ and such dependency is shown in the inset of Fig. 3 by plotting τ_s as a function of R for the highest viscosity value considered, $\nu_R=1/3\cdot10^{-3}$.

3.1.2 Large density ratio $\rho_R/\rho_B=1000$

We now consider a similar, but more challenging setup with density ratio $\rho_R/\rho_B=1000$, which is relevant for e.g. simulating water and air or other liquid and gas systems. Once again we measure E_{kin}/A for the different sets of simulations and the results are reported in Fig. 4. Two different regimes are clearly visible. For the two sets of simulations with lower viscosities, E_{kin}/A is increasing with time. The opposite is the case for the two sets

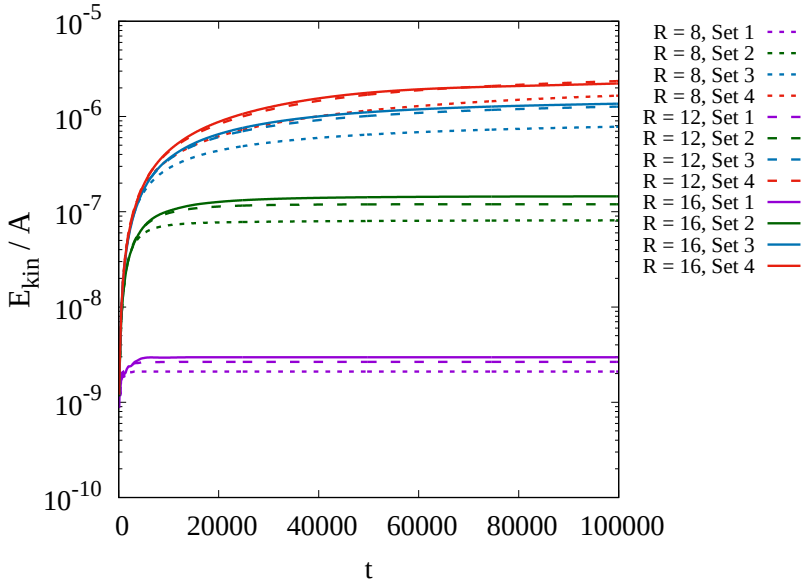

Figure 1: The kinetic energy integrated over the domain and normalized by the droplet surface area E_{kin}/A is measured as a function of time t. Here droplet density $\rho_R=1$ and domain side-length $L=64$. Simulation parameters corresponding to Set 1 through Set 4 are reported in Table 1. A clear trend is visible as lowering the viscosities leads to higher E_{kin}, which is a measure of spurious currents. Furthermore, the equilibration time before a steady state is attained is increased as viscosities are lowered.

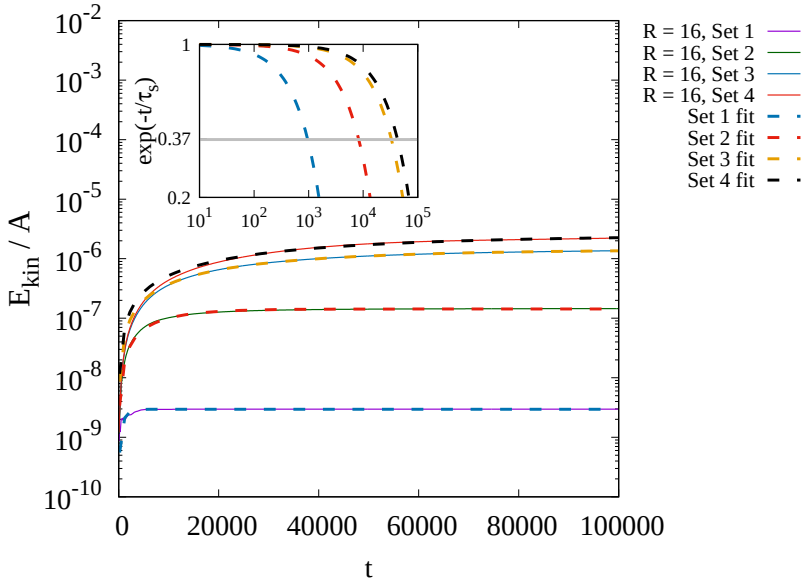

Figure 2: Fit function $f(t)=a\cdot(1-\exp(-t/\tau_s))$ applied to data from simulations with $R=16$. The fit function is used to obtain a characteristic equilibration timescale, τ_s, of E_{kin}/A. The inset shows $\exp(-t/\tau_s)$ as a function of time t, where the crossing with the grey line indicates the value of τ_s at which the kinetic energy has decreased by a factor $1/e$. Values of τ_s for all simulations are reported in Fig. 3.

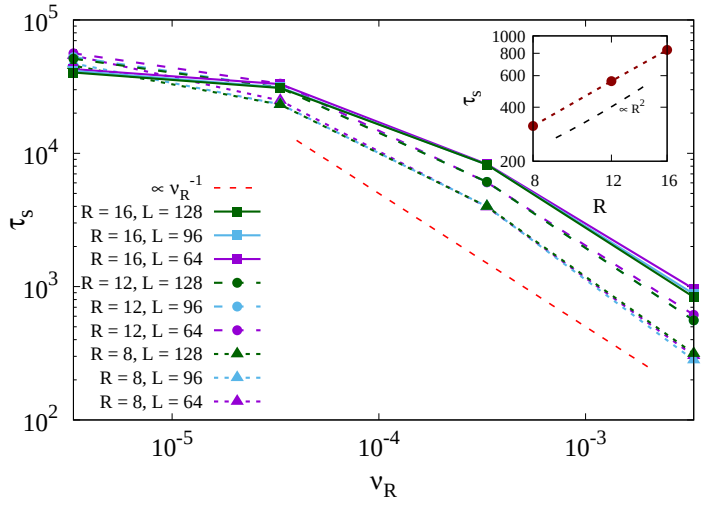

Figure 3: Characteristic kinetic energy equilibration timescale τ_s as a function of kinematic viscosity $\nu_R = c_s^2(\tau_R - 0.5)$ obtained using the fit function $f(t) = a \cdot (1 - \exp(-t/\tau_s))$ applied on the data shown in Fig. 1. A clear influence of ν_R on τ_s is shown, where a higher ν_R leads to significantly faster equilibration times, as the equilibration timescale decreases as approximately $\tau_s \propto \nu_R^{-1}$, indicated by the red dashed line, for the lower viscosity values in particular. A smaller droplet radius R leads to slightly faster equilibration times as indicated by the decreased τ_s value. The inset shows τ_s as a function of R for the cases with parameter Set 4 (and $L = 128$), i.e. with viscosity $\nu_R = 1/3 \cdot 10^{-3}$, its slope indicating $\tau_s \propto R^2/\nu_R$.

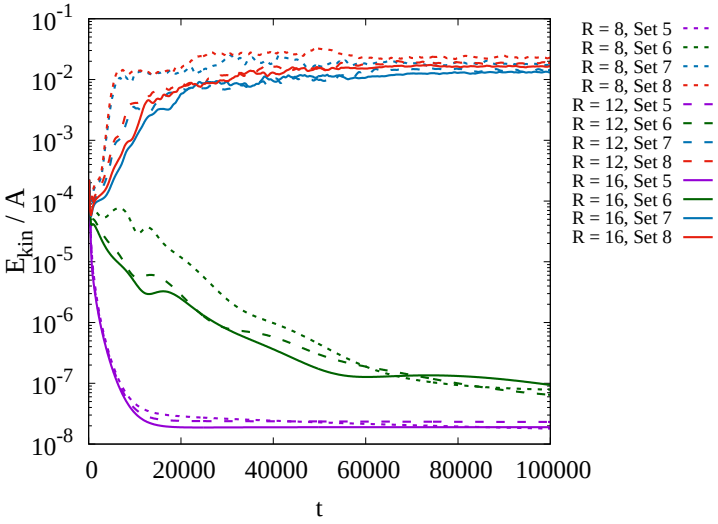

Figure 4: The kinetic energy integrated over the domain and normalized by the droplet surface area E_{kin}/A is measured as a function of time t. Here droplet density $\rho_R = 1000$ and domain side-length $L = 64$. Simulation parameters corresponding to Set 4 through Set 8 are reported in Table 1. A clear trend is visible where lowering the viscosities leads to higher E_{kin}, which is a measure of spurious currents. Furthermore, for the lower viscosity cases we see an increase in E_{kin} after initialization (these correspond to the unstable droplets pictured in Fig. 5), whereas we see a decrease in the higher viscosity (stable) cases.

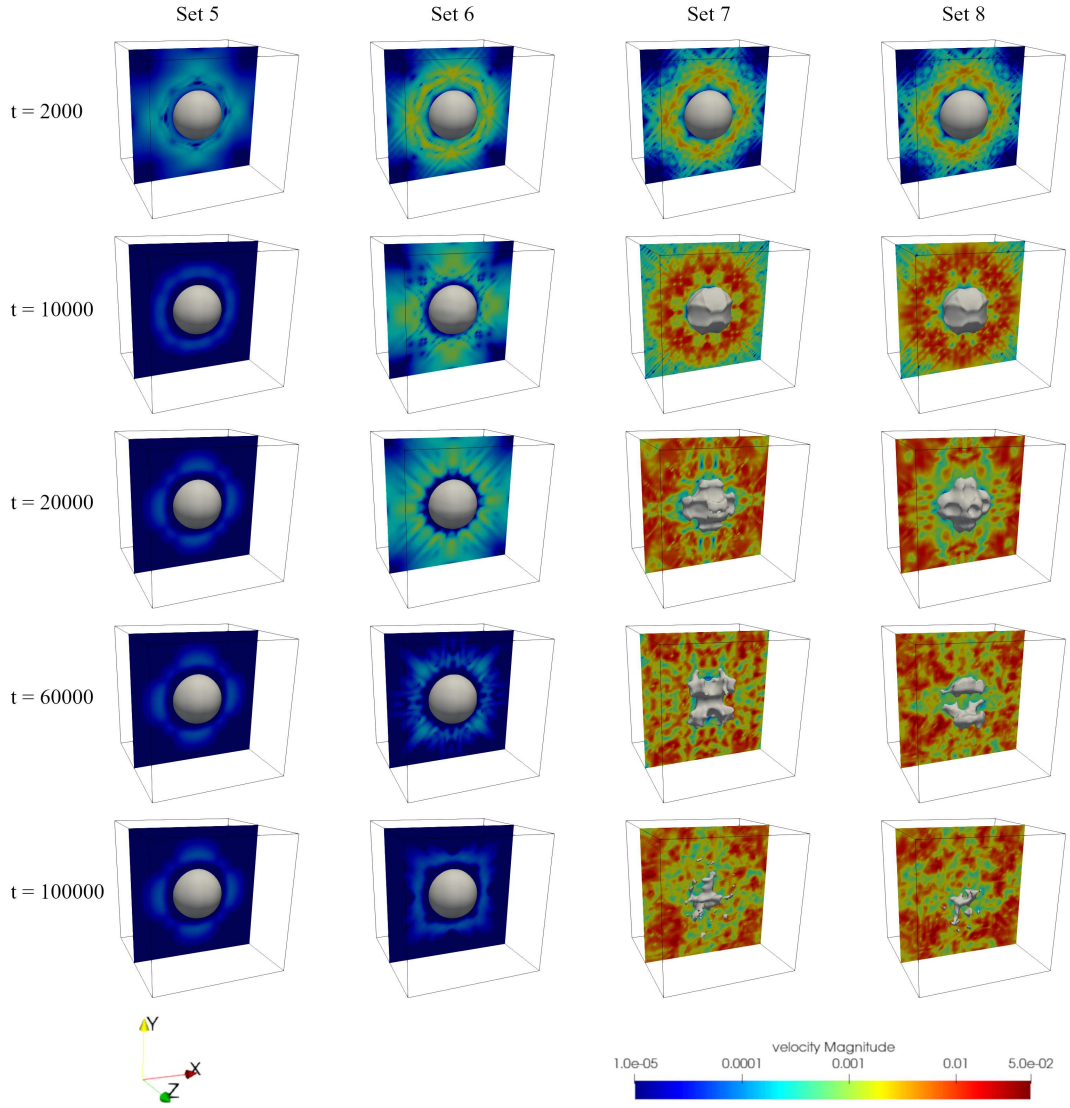

Figure 5: Visualizations showing the isosurface of the droplet-ambient interface and a cross-sectional (colored) area of the velocity field. Shown here are simulations with domain side-length $L=64$, droplet radius $R=12$ and droplet density $\rho_R=1000$ at times $t=2000$, 10000, 20000, 60000, 100000. Viscosity values differ per column and can be found in Table 1, where the leftmost column has the highest viscosity values and is most stable, and the rightmost column has the lowest viscosity values and is least stable.

with higher viscosities, which show decreasing E_{kin}/A over time. Once again the lower the viscosity, the lower E_{kin}/A values we find. Visualizations of these simulations show that in the cases of increasing kinetic energy the droplets are in fact unstable, see Fig. 5. Note however that none of the simulations were *numerically* unstable.

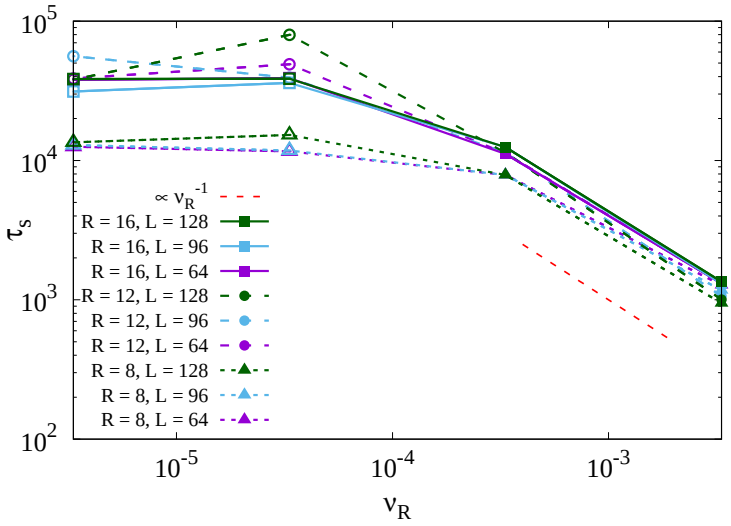

Figure 6: Characteristic kinetic energy equilibration timescale τ_s as a function of kinematic viscosity $\nu_R = c_s^2(\tau_R - 0.5)$ obtained using fit functions $f(t) = a\cdot(1-\exp(-t/\tau_s))$ for parameter Set 7 and 8, and $g(t) = a\cdot(1-\exp(-t/\tau_s))+c$ for Set 5 and 6, on the data shown in Fig. 4. A clear influence of ν_R on τ_s is seen for the highest two viscosity values (Set 5 and 6, filled symbols), where a higher ν_R leads to significantly faster equilibration times, as the equilibration timescale decreases according to $\tau_s \propto \nu_R^{-1}$. A smaller droplet radius R leads to slightly faster equilibration times as indicated by the decreased τ_s value. For the lower two viscosity simulations (Set 7 and 8, hollow symbols) there is no clear trend or difference between the two. This can be explained by the chaotic nature of these systems, as illustrated in Fig. 5.

Also for this case with density ratio $\rho_R/\rho_B = 1000$ we investigate the equilibration time, in a similar manner as before using the fit function $f(t) = a\cdot(1-\exp(-t/\tau_s))$ for parameter Set 7 and 8. For parameter Set 5 and 6 we use a slightly modified fit function, since E_{kin}/A decreases over time for these parameters, as opposed to increasing. The modified function $g(t) = a\cdot(1-\exp(-t/\tau_s))+c$ gives a closer fit for the data sets corresponding to parameter Set 5 and 6.

The characteristic kinetic energy equilibration timescale values are obtained through the fitting procedure and are reported in Fig. 6. We find that equilibration time is strongly affected by viscosity values and by R. The effect of domain size is more pronounced for the two lowest viscosity cases, when compared to the unity density simulations. A higher ν_R leads to significantly faster equilibration times, as the equilibration timescale decreases according to $\tau_s \propto \nu_R^{-1}$. For Set 7 and 8, i.e. the two lowest viscosity simulations, there is no clear trend or difference between the two in terms of τ_s. This can be explained by the unstable chaotic nature of these systems, as illustrated in Fig. 5.

3.2 Translating droplet

When investigating physical phenomena it is rare that droplets stay at rest completely. Therefore we now consider the stability of the numerical scheme for a droplet moving

along with the ambient, being accelerated at the same rate. We once again look at the cases where $\rho_R/\rho_B=1$, 1000 and vary the acceleration $a=10^{-5}$, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9} and viscosity values $\nu_R=1/3\cdot10^{-3}$, $1/3\cdot10^{-4}$ and $\nu_B=2/3\cdot10^{-3}$, $2/3\cdot10^{-4}$ to get a better understanding of the limits to the numerical stability of this scheme when the droplet along with the ambient is accelerated. For the following simulations L and R are kept constant at $L=64$ and $R=12$.

3.2.1 Unity density ratio $\rho_R/\rho_B=1$

We first consider the unity density ratio, $\rho_R/\rho_B=1$, case. As was done for the stationary cases, we integrate E_{kin} over the entire domain. We then subtract the kinetic energy due to the applied acceleration integrated over the whole domain $E_{kin,a}=\int_D 1/2mv^2$. We hereby acquire the kinetic energy only due to spurious currents, $E_{kin}-E_{kin,a}$. The results shown in Fig. 7 for the case with $\nu_R=1/3\cdot10^{-3}$ show that the lower the acceleration, the less spurious currents one can expect. We also find that the limit of numerical stability is reached early for the case where $a=10^{-5}$, which becomes numerically unstable around time t=40000, at which point the velocity v=$at\approx$0.4. All other simulations are stable up to the final simulation time of $t=100000$. We see that kinetic energy related to the spurious currents increases monotonically with a t^2 dependency for all cases since $E_{kin}-E_{kin,a}\propto a^2t^2$, as illustrated by the trendline plotted in Fig. 7 and Fig. 8.

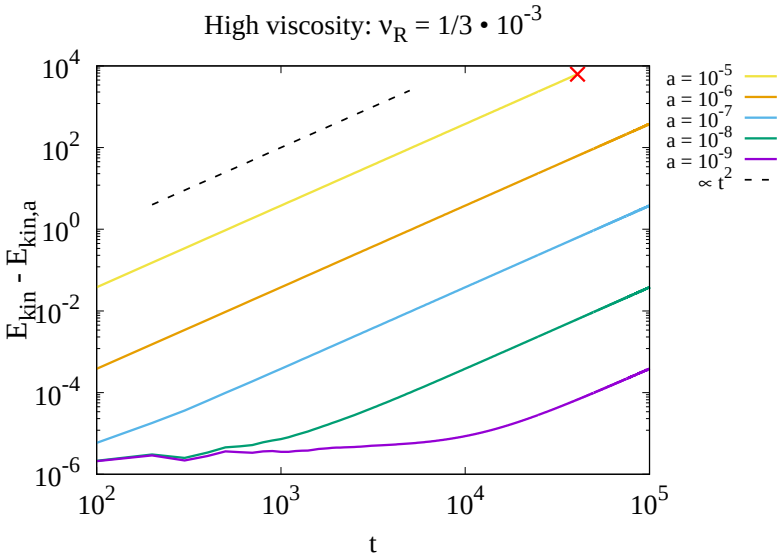

Figure 7: Total excess kinetic energy $E_{kin}-E_{kin,a}$, i.e. the total kinetic energy E_{kin} minus the kinetic energy due to domain acceleration $E_{kin,a}$, as a function of time t. Parameters used are $L=64$, $R=12$, $\nu_R=1/3\cdot10^{-3}$ and $\nu_B=2/3\cdot10^{-3}$. Numerical instability occurs for acceleration value $a=10^{-5}$ around time $t=40000$. In all cases $E_{kin}-E_{kin,a}\propto t^2$, apart from an initial slower increase for the lower acceleration cases. Lower acceleration leads to significantly lower spurious currents as measured by the reduction in $E_{kin}-E_{kin,a}$.

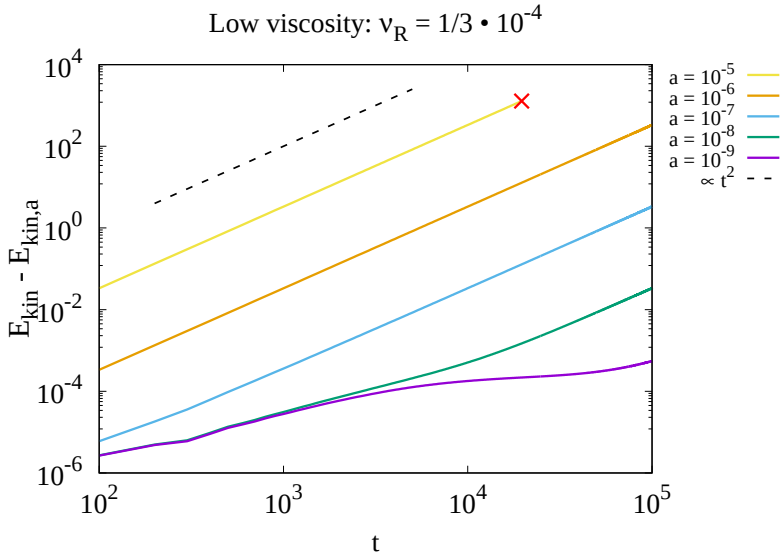

Figure 8: Total excess kinetic energy $E_{kin}-E_{kin,a}$, i.e. the total kinetic energy E_{kin} minus the kinetic energy due to domain acceleration $E_{kin,a}$, as a function of time t. Parameters used are $L=64$, $R=12$, $\rho_R=1$, $\rho_B=1$, $\nu_R=1/3\cdot10^{-4}$ and $\nu_B=2/3\cdot10^{-4}$. Numerical instability occurs for acceleration value $a=10^{-5}$ around time $t=20000$. For most cases $E_{kin}-E_{kin,a}\propto t^2$, apart from an slower increase for the lower acceleration cases. Lower acceleration leads to significantly lower spurious currents as measured by the reduction in $E_{kin}-E_{kin,a}$.

Keeping everything equal, but reducing the viscosity by setting $\nu_R=1/3\cdot10^{-4}$ we retrieve the results shown in Fig. 8. We note that nearly identical $E_{kin}-E_{kin,a}$ is reached at the end of the simulation for nearly all cases. Differences are that the case with $a=10^{-5}$ shows numerical instability earlier at time $t=20000$ and the case with $a=10^{-9}$ has a faster initial rise in E_{kin}.

3.2.2 Large density ratio $\rho_R/\rho_B=1000$

We now consider the high density ratio case, with $\rho_R/\rho_B=1000$. Results are shown in Fig. 9 and Fig. 10 for $\nu_R=1/3\cdot10^{-3}$ and $\nu_R=1/3\cdot10^{-4}$ respectively. In general, we see significantly reduced numerical stability compared to the unity density case, as all simulations with $a\leq10^{-7}$ show numerical instability before $t=100000$ for the high viscosity case, see Fig. 9. Specifically, at $\nu_R=1/3\cdot10^{-3}$ with $a=10^{-5}$, 10^{-6}, 10^{-7} numerical instability occurs at times $t=3300$, 16000, 74900, which correspond to velocities $v\approx0.033$, 0.016, 0.007 respectively. Similar behavior is seen for the lower viscosity case, Fig. 10, however, numerical instability does occur faster for the lower viscosity case, i.e. the simulations crash earlier. Specifically, at $\nu_R=1/3\cdot10^{-4}$ with $a=10^{-5}$, 10^{-6}, 10^{-7} numerical instability occurs at times $t=1800$, 11300, 30000, which correspond to velocities $v\approx0.018$, 0.011, 0.003 respectively. Furthermore the kinetic energy does not increase monotonically as was the case for a unity density ratio. We see an initial decrease, followed by a gradual increase

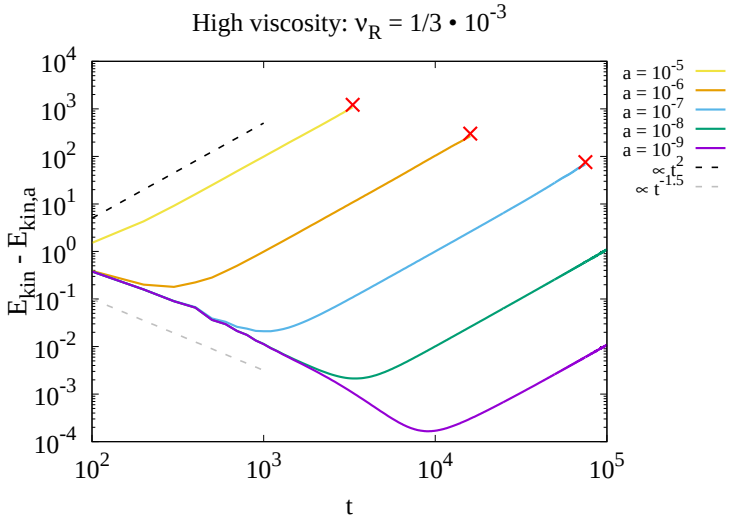

Figure 9: Total excess kinetic energy $E_{kin}-E_{kin,a}$, i.e. the total kinetic energy E_{kin} minus the kinetic energy due to domain acceleration $E_{kin,a}$, as a function of time t. Parameters used are $L=64$, $R=12$, $\rho_R=1000$, $\rho_B=1$, $\nu_R=1/3\cdot10^{-3}$ and $\nu_B=2/3\cdot10^{-3}$. Numerical instability occurs for acceleration value $a\leq10^{-7}$. In all cases $E_{kin}-E_{kin,a}\propto t^2$, after an initial decrease of approximately $E_{kin}-E_{kin,a}\propto t^{-1.5}$. The decrease in $E_{kin}-E_{kin,a}$ continues for a longer period of time as a is decreased. Lower acceleration leads to significantly lower spurious currents as measured by the reduction in $E_{kin}-E_{kin,a}$.

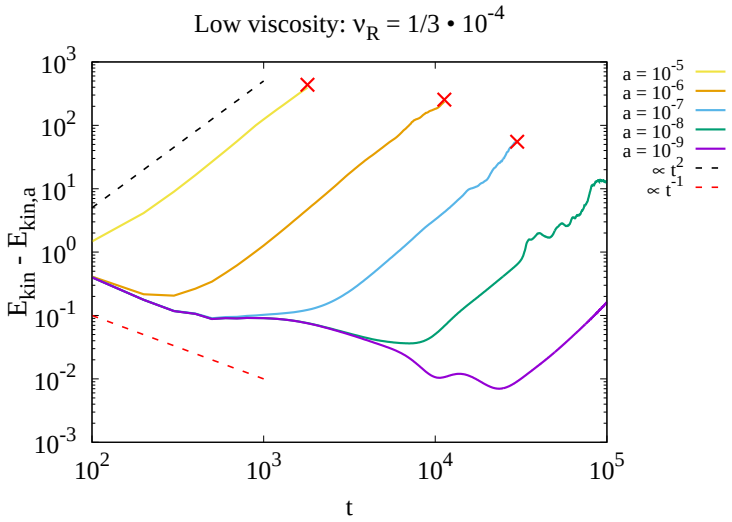

Figure 10: Total excess kinetic energy $E_{kin}-E_{kin,a}$, i.e. the total kinetic energy E_{kin} minus the kinetic energy due to domain acceleration $E_{kin,a}$, as a function of time t. Parameters used are $L=64$, $R=12$, $\rho_R=1000$, $\rho_B=1$, $\nu_R=1/3\cdot10^{-4}$ and $\nu_B=2/3\cdot10^{-4}$. Numerical instability occurs for acceleration value $a\leq10^{-7}$ and occurs earlier than for the higher viscosity case. In all cases $E_{kin}-E_{kin,a}\propto t^2$, after an initial decrease of approximately $E_{kin}-E_{kin,a}\propto t^{-1}$. The decrease in $E_{kin}-E_{kin,a}$ continues for a longer period of time as a is decreased. Lower acceleration leads to significantly lower spurious currents as measured by the reduction in $E_{kin}-E_{kin,a}$.

of $E_{kin}-E_{kin,a}$. Finally we note that $E_{kin}-E_{kin,a}$ is higher for all numerically stable simulations compared to the equivalent $\rho_R/\rho_B=1$ cases.

Comparing the two viscosity value cases, $\nu_R=1/3\cdot10^{-3}$ and $\nu_R=1/3\cdot10^{-4}$, we note that nearly identical $E_{kin}-E_{kin,a}$ is reached at the end of the simulation for each case with the same value of a. Differences are that the cases with $a\leq10^{-7}$ show numerical instability slightly earlier for the lower viscosity cases.

4 Conclusions

Numerically stable multicomponent simulations at extremely low viscosity values are possible using the color-gradient method central moments (CG-CM) model, as shown by the cases presented in this work. All stationary droplet simulations presented were numerically stable, but in the extreme cases of high density ratio, $\rho_R/\rho_B=1000$ combined with low viscosities, $\nu_R=1/3\cdot10^{-5}$, $1/3\cdot10^{-6}$, the droplet interface becomes completely deformed due to the presence of strong spurious currents leading to unphysical results. In general higher viscosity values lead to faster equilibration times and lower kinetic energy E_{kin} due to spurious currents. Increased droplet radius R increases total E_{kin} due to increased surface area around which spurious currents occur. Normalizing E_{kin} by the surface area of the droplet A shows that even after normalization the lowest E_{kin}/A occurs with the smallest R, all other parameters kept equal. Increasing domain size length L has a negligible effect on equilibration time.

Adding movement to the ambient and droplet by applying an acceleration to the entire domain did show cases of numerical instability, which was not seen for the stationary droplet simulations. For the case with $\rho_R/\rho_B=1$, $\nu_R=1/3\cdot10^{-3}$, $1/3\cdot10^{-4}$ and $a=10^{-5}$ the simulation was unstable. Increasing the density ratio to $\rho_R/\rho_B=1000$ made the system more sensitive to numerical instability, evidenced by the fact that several simulations using $a\leq10^{-7}$ were unstable. We can conclude that dynamic systems are more susceptible to numerical instability than stationary systems and higher density ratios and lower viscosities negatively impact stability. The work presented gives a clear indication of the numerical stability limits using CG-CM in the low viscosity regime. This is especially relevant for simulating e.g. water-air type systems, which require low kinematic viscosity parameters and high density ratio $\rho_R/\rho_B\approx1000$.

In future work we will show that the CG-CM model is suitable for inkjetting simulations on a micrometer scale, which is an example of an industrial application involving a high density ratio $\rho_R/\rho_B=1000$. It is possible to perform accurate inkjetting simulations using the model with parameters approximating those of Set 5 from Table 1. Specifically, for the jetting simulations we use the following parameters in SI units: density of ink $\rho_R=1080\text{kg/m}^3$, density of air $\rho_B=1.08\text{kg/m}^3$, viscosity of ink $\nu_R=0.83\cdot10^{-}5m^2/s$, viscosity of air $\nu_B=1.67\cdot10^{-}5m^2/s$, surface tension $\sigma=2.8\cdot10^{-2}\text{N/m}$ and approximate generated droplet radius $R=15\mu\text{m}$. As predicted by the current work, the simulation is numerically stable and accurate for these parameters.

Acknowledgments

This work is part of the Netherlands Organization for Scientific Research (NWO) research project High Tech Systems and Materials (HTSM), with project number 13912. The authors thank the NWO domain Science for the use of supercomputer facilities and co-financers Canon Production Printing Holding B.V., University of Twente and Eindhoven University of Technology for financial support.

A Appendix

Here we provide additional equations used for the implementation of the CM formulation as described in section 2, where the five implementation steps are listed, which we repeat here for convenience:

1. Computation of macroscopic variables by Eq. (2.3).

2. Evaluation of postcollision central moments k_i^* as given by Eq. (A.1).

3. Computation of postcollision raw moments, $|r_i^*\rangle = [r_0^*, \cdots, r_i^*, \cdots, r_{26}^*]^\intercal$, from the postcollision central moments according to, $|r_i^*\rangle = \mathbf{N}^{-1}|k_i^*\rangle$, where $\mathbf{N} = \mathbf{T}\mathbf{M}^{-1}$, with the transformation matrix $\mathbf{T}$ of the form given by Eq. (A.3) and $\mathbf{M}$ given by Eq. (A.4).

4. Reconstruction of postcollision populations as $|f_i^*\rangle = \mathbf{M}^{-1}|r_i^*\rangle$, resulting in Eq. (A.5).

5. Perform the streaming step according to Eq. (2.1).

For the second implementation step the following postcollision central moments k_i^* are used:

$$\begin{aligned}
k_0^* &= \rho, \\
k_4^* &= (1-\omega)k_4, \\
k_5^* &= (1-\omega)k_5, \\
k_6^* &= (1-\omega)k_6, \\
k_7^* &= (1-\omega)k_7, \\
k_8^* &= (1-\omega)k_8, \\
k_9^* &= 3\rho c_s^2, \\
k_{17}^* &= \rho c_s^2, \\
k_{18}^* &= \rho c_s^4, \\
k_{26}^* &= \rho c_s^6,
\end{aligned} \tag{A.1}$$

where $k^*_{1,\cdots,3}=k^*_{10,\cdots,16}=k^*_{19,\cdots,25}=0$. The five precollision central moments are calculated as

$$\begin{aligned} k_4&=\sum_i f_i\overline{c}_{ix}\overline{c}_{iy},\quad k_5=\sum_i f_i\overline{c}_{ix}\overline{c}_{iz},\quad k_6=\sum_i f_i\overline{c}_{iy}\overline{c}_{iz},\\ k_7&=\sum_i f_i(\overline{c}^2_{ix}-\overline{c}^2_{iy}),\quad k_8=\sum_i f_i(\overline{c}^2_{ix}-\overline{c}^2_{iz}). \end{aligned} \tag{A.2}$$

For the third implementation step the postcollision central moments are calculated as to $|r^*_i\rangle=\mathbf{N}^{-1}|k^*_i\rangle$, where $\mathbf{N}=\mathbf{T}\mathbf{M}^{-1}$, with the matrices $\mathbf{T}$ and $\mathbf{M}$ being

$$\mathbf{T}=\mathbf{N}\mathbf{M}=\begin{bmatrix} \langle|\mathbf{c}_i|^0| \\ \langle\overline{c}_{ix}| \\ \langle\overline{c}_{iy}| \\ \langle\overline{c}_{iz}| \\ \langle\overline{c}_{ix}\overline{c}_{iy}| \\ \langle\overline{c}_{ix}\overline{c}_{iz}| \\ \langle\overline{c}_{iy}\overline{c}_{iz}| \\ \langle\overline{c}^2_{ix}-\overline{c}^2_{iy}| \\ \langle\overline{c}^2_{ix}-\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}+\overline{c}^2_{iy}+\overline{c}^2_{iz}| \\ \langle\overline{c}_{ix}\overline{c}^2_{iy}+\overline{c}_{ix}\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}_{iy}+\overline{c}_{iy}\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}_{iz}+\overline{c}^2_{iy}\overline{c}_{iz}| \\ \langle\overline{c}_{ix}\overline{c}^2_{iy}-\overline{c}_{ix}\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}_{iy}-\overline{c}_{iy}\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}_{iz}-\overline{c}^2_{iy}\overline{c}_{iz}| \\ \langle\overline{c}_{ix}\overline{c}_{iy}\overline{c}_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}^2_{iy}+\overline{c}^2_{ix}\overline{c}^2_{iz}+\overline{c}^2_{iy}\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}^2_{iy}+\overline{c}^2_{ix}\overline{c}^2_{iz}-\overline{c}^2_{iy}\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}^2_{iy}-\overline{c}^2_{ix}\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}_{iy}\overline{c}_{iz}| \\ \langle\overline{c}_{ix}\overline{c}^2_{iy}\overline{c}_{iz}| \\ \langle\overline{c}_{ix}\overline{c}_{iy}\overline{c}^2_{iz}| \\ \langle\overline{c}_{ix}\overline{c}^2_{iy}\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}_{iy}\overline{c}^2_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}^2_{iy}\overline{c}_{iz}| \\ \langle\overline{c}^2_{ix}\overline{c}^2_{iy}\overline{c}^2_{iz}| \end{bmatrix}, \tag{A.3}$$

$$\mathbf{M}=\begin{bmatrix}
\langle|\mathbf{c}_i|^0| \\
\langle c_{ix}| \\
\langle c_{iy}| \\
\langle c_{iz}| \\
\langle c_{ix}c_{iy}| \\
\langle c_{ix}c_{iz}| \\
\langle c_{iy}c_{iz}| \\
\langle c_{ix}^2-c_{iy}^2| \\
\langle c_{ix}^2-c_{iz}^2| \\
\langle c_{ix}^2+c_{iy}^2+c_{iz}^2| \\
\langle c_{ix}c_{iy}^2+c_{ix}c_{iz}^2| \\
\langle c_{ix}^2c_{iy}+c_{iy}c_{iz}^2| \\
\langle c_{ix}^2c_{iz}+c_{iy}^2c_{iz}| \\
\langle c_{ix}c_{iy}^2-c_{ix}c_{iz}^2| \\
\langle c_{ix}^2c_{iy}-c_{iy}c_{iz}^2| \\
\langle c_{ix}^2c_{iz}-c_{iy}^2c_{iz}| \\
\langle c_{ix}c_{iy}c_{iz}| \\
\langle c_{ix}^2c_{iy}^2+c_{ix}^2c_{iz}^2+c_{iy}^2c_{iz}^2| \\
\langle c_{ix}^2c_{iy}^2+c_{ix}^2c_{iz}^2-c_{iy}^2c_{iz}^2| \\
\langle c_{ix}^2c_{iy}^2-c_{ix}^2c_{iz}^2| \\
\langle c_{ix}^2c_{iy}c_{iz}| \\
\langle c_{ix}c_{iy}^2c_{iz}| \\
\langle c_{ix}c_{iy}c_{iz}^2| \\
\langle c_{ix}c_{iy}^2c_{iz}^2| \\
\langle c_{ix}^2c_{iy}c_{iz}^2| \\
\langle c_{ix}^2c_{iy}^2c_{iz}| \\
\langle c_{ix}^2c_{iy}^2c_{iz}^2|
\end{bmatrix}. \tag{A.4}$$

For the fourth implementation step the postcollision populations are calculated as $|f_i^*\rangle = \mathbf{M}^{-1}|r_i^*\rangle$, resulting in:

$$\begin{aligned}
f_0^* &= r_0^*-r_9^*+r_{17}^*-r_{26}^*,\\
f_1^* &= (r_7^*+r_8^*+r_9^*)/6-(r_{17}^*+r_{18}^*)/4+(r_1^*+r_{23}^*+r_{26}^*-r_{10}^*)/2,\\
f_2^* &= (r_7^*+r_8^*+r_9^*)/6-(r_{17}^*+r_{18}^*)/4+(-r_1^*-r_{23}^*+r_{26}^*+r_{10}^*)/2,\\
f_3^* &= (r_{18}^*-3r_{17}^*)/8-r_7^*/3-r_{19}^*/4+(r_8^*+r_9^*)/6+(r_2^*+r_{24}^*+r_{26}^*-r_{11}^*)/2,\\
f_4^* &= (r_{18}^*-3r_{17}^*)/8-r_7^*/3-r_{19}^*/4+(r_8^*+r_9^*)/6+(r_{11}^*+r_{26}^*-r_2^*-r_{24}^*)/2,\\
f_5^* &= (r_{18}^*-3r_{17}^*)/8-r_8^*/3+r_{19}^*/4+(r_7^*+r_9^*)/6+(r_3^*+r_{25}^*+r_{26}^*-r_{12}^*)/2,
\end{aligned}$$

$$
\begin{aligned}
f_6^* &= (r_{18}^* - 3r_{17}^*)/8 - r_8^*/3 + r_{19}^*/4 + (r_7^* + r_9^*)/6 - (r_3^* + r_{25}^* - r_{12}^* - r_{26}^*)/2,\\
f_7^* &= (r_{17}^* + r_{18}^*)/16 - (r_{22}^* + r_{23}^* + r_{24}^* + r_{26}^* - r_4^*)/4 + (r_{10}^* + r_{11}^* + r_{13}^* + r_{14}^* + r_{19}^*)/8,\\
f_8^* &= (r_{17}^* + r_{18}^*)/16 + (r_{10}^* + r_{13}^* + r_{19}^* - r_{11}^* - r_{14}^*)/8 - (r_4^* + r_{23}^* + r_{26}^* - r_{22}^* - r_{24}^*)/4,\\
f_9^* &= (r_{17}^* + r_{18}^*)/16 + (r_{11}^* + r_{14}^* + r_{19}^* - r_{10}^* - r_{13}^*)/8 - (r_4^* + r_{24}^* + r_{26}^* - r_{22}^* - r_{23}^*)/4,\\
f_{10}^* &= (r_{17}^* + r_{18}^*)/16 + (r_4^* + r_{23}^* + r_{24}^* - r_{22}^* - r_{26}^*)/4 - (r_{10}^* + r_{11}^* + r_{13}^* + r_{14}^* - r_{19}^*)/8,\\
f_{11}^* &= (r_{17}^* + r_{18}^*)/16 + (r_{10}^* + r_{12}^* + r_{15}^* - r_{13}^* - r_{19}^*)/8 - (r_{21}^* + r_{23}^* + r_{25}^* + r_{26}^* - r_5^*)/4,\\
f_{12}^* &= (r_{17}^* + r_{18}^*)/16 + (r_{12}^* + r_{13}^* + r_{15}^* - r_{10}^* - r_{19}^*)/8 - (r_5^* + r_{25}^* + r_{26}^* - r_{21}^* - r_{23}^*)/4,\\
f_{13}^* &= (r_{17}^* + r_{18}^*)/16 - (r_5^* + r_{23}^* + r_{26}^* - r_{21}^* - r_{25}^*)/4 - (r_{12}^* + r_{13}^* + r_{15}^* + r_{19}^* - r_{10}^*)/8, \qquad \text{(A.5)}\\
f_{14}^* &= (r_{17}^* + r_{18}^*)/16 + (r_5^* + r_{23}^* + r_{25}^* - r_{21}^* - r_{26}^*)/4 - (r_{10}^* + r_{12}^* + r_{15}^* + r_{19}^* - r_{13}^*)/8,\\
f_{15}^* &= (r_{11}^* + r_{12}^* + r_{17}^* - r_{14}^* - r_{15}^* - r_{18}^*)/8 - (r_{20}^* + r_{24}^* + r_{25}^* + r_{26}^* - r_6^*)/4,\\
f_{16}^* &= (r_{11}^* + r_{15}^* + r_{17}^* - r_{12}^* - r_{14}^* - r_{18}^*)/8 - (r_6^* + r_{24}^* + r_{26}^* - r_{20}^* - r_{25}^*)/4,\\
f_{17}^* &= (r_{12}^* + r_{14}^* - r_{17}^* - r_{11}^* - r_{15}^* - r_{18}^*)/8 - (r_6^* + r_{25}^* + r_{26}^* - r_{20}^* - r_{24}^*)/4,\\
f_{18}^* &= (r_{14}^* + r_{15}^* + r_{17}^* - r_{11}^* - r_{12}^* - r_{18}^*)/8 + (r_6^* + r_{24}^* + r_{25}^* - r_{20}^* - r_{26}^*)/4,\\
f_{19}^* &= (r_{16}^* + r_{20}^* + r_{21}^* + r_{22}^* + r_{23}^* + r_{24}^* + r_{25}^* + r_{26}^*)/8,\\
f_{20}^* &= (r_{22}^* + r_{23}^* + r_{24}^* + r_{26}^* - r_{16}^* - r_{20}^* - r_{21}^* - r_{25}^*)/8,\\
f_{21}^* &= (r_{21}^* + r_{23}^* + r_{25}^* + r_{26}^* - r_{16}^* - r_{20}^* - r_{22}^* - r_{24}^*)/8,\\
f_{22}^* &= (r_{20}^* + r_{24}^* + r_{25}^* + r_{26}^* - r_{16}^* - r_{21}^* - r_{22}^* - r_{23}^*)/8,\\
f_{23}^* &= (r_{16}^* + r_{20}^* + r_{23}^* + r_{26}^* - r_{21}^* - r_{22}^* - r_{24}^* - r_{25}^*)/8,\\
f_{24}^* &= (r_{16}^* + r_{22}^* + r_{25}^* + r_{26}^* - r_{20}^* - r_{21}^* - r_{23}^* - r_{24}^*)/8,\\
f_{25}^* &= (r_{16}^* + r_{21}^* + r_{24}^* + r_{26}^* - r_{20}^* - r_{22}^* - r_{23}^* - r_{25}^*)/8,\\
f_{26}^* &= (r_{20}^* + r_{21}^* + r_{22}^* + r_{26}^* - r_{16}^* - r_{23}^* - r_{24}^* - r_{25}^*)/8.
\end{aligned}
$$

References

[1] S. Succi. *The lattice Boltzmann equation: For complex states of flowing matter*. 01 2018. doi: 10.1093/oso/9780199592357.001.0001.

[2] Andrew K. Gunstensen, Daniel H. Rothman, Stéphane Zaleski, and Gianluigi Zanetti. Lattice Boltzmann model of immiscible fluids. *Phys. Rev. A*, 43:4320–4327, Apr 1991. doi: 10.1103/PhysRevA.43.4320. URL `https://link.aps.org/doi/10.1103/PhysRevA.43.4320`.

[3] Xiaowen Shan and Hudong Chen. Lattice Boltzmann model for simulating flows with multiple phases and components. *Phys. Rev. E*, 47:1815–1819, Mar 1993. doi: 10.1103/PhysRevE.47.1815. URL `https://link.aps.org/doi/10.1103/PhysRevE.47.1815`.

[4] Michael R. Swift, E. Orlandini, W. R. Osborn, and J. M. Yeomans. Lattice Boltzmann simulations of liquid-gas and binary fluid systems. *Phys. Rev. E*, 54:5041–5052, Nov 1996. doi: 10.1103/PhysRevE.54.5041. URL `https://link.aps.org/doi/10.1103/PhysRevE.54.5041`.

[5] Xiaoyi He, Shiyi Chen, and Raoyang Zhang. A lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability. *Journal of Computational Physics*, 152(2):642–663,

1999. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.1999.6257. URL https://www.sciencedirect.com/science/article/pii/S0021999199962575.

[6] Alessandro De Rosis and Christophe Coreixas. Multiphysics flow simulations using D3Q19 lattice Boltzmann methods based on central moments. *Physics of Fluids*, 32(11):117101, 2020. doi: 10.1063/5.0026316. URL https://doi.org/10.1063/5.0026316.

[7] Alessandro De Rosis, Rongzong Huang, and Christophe Coreixas. Universal formulation of central-moments-based lattice Boltzmann method with external forcing for the simulation of multiphysics phenomena. *Physics of Fluids*, 31(11):117102, Nov 2019. ISSN 1089-7666. doi: 10.1063/1.5124719. URL http://dx.doi.org/10.1063/1.5124719.

[8] Kevin Connington and Taehun Lee. A review of spurious currents in the lattice Boltzmann method for multiphase flows. *Journal of Mechanical Science and Technology*, 26, 12 2012. doi: 10.1007/s12206-012-1011-5.

[9] Karun P. N. Datadien, Gianluca Di Staso, Herman M. A. Wijshoff, and Federico Toschi. A quantitative comparison of physical accuracy and numerical stability of lattice Boltzmann color gradient and pseudopotential multicomponent models for microfluidic applications. *arXiv*, physics.flu-dyn, 2021.

[10] S. Leclaire, N. Pellerin, M. Reggio, and J. Trépanier. Enhanced equilibrium distribution functions for simulating immiscible multiphase flows with variable density ratios in a class of lattice Boltzmann models. *International Journal of Multiphase Flow*, 57:159–168, 2013.

[11] Sébastien Leclaire, Andrea Parmigiani, Orestis Malaspinas, Bastien Chopard, and Jonas Latt. Generalized three-dimensional lattice Boltzmann color-gradient method for immiscible two-phase pore-scale imbibition and drainage in porous media. *Phys. Rev. E*, 95:033306, Mar 2017. doi: 10.1103/PhysRevE.95.033306. URL https://link.aps.org/doi/10.1103/PhysRevE.95.033306.