C. R. Mecanique 343 (2015) 580–588

Contents lists available at ScienceDirect

Comptes Rendus Mecanique

www.sciencedirect.com

Discrete simulation of fluid dynamics

Simulation of liquid–vapour phase separation on GPUs using Lattice Boltzmann models with off-lattice velocity sets

Tonino Biciuşcă [a,b], Adrian Horga [a,c], Victor Sofonea [a,*]

[a] Center for Fundamental and Advanced Technical Research, Romanian Academy, Bd. Mihai Viteazul 24, 300223 Timişoara, Romania
[b] Department of Physics, West University of Timişoara, Bd. Vasile Pârvan 4, 300223 Timişoara, Romania
[c] Department of Computer and Information Science, Linköping University, 581 83 Linköping, Sweden

ARTICLE INFO

Article history:
Received 16 December 2014
Accepted 27 March 2015
Available online 11 August 2015

Keywords:
Lattice Boltzmann
Corner transport upwind
Phase separation

ABSTRACT

We use a two-dimensional Lattice Boltzmann model to investigate the liquid–vapour phase separation in an isothermal van der Waals fluid. The model is based on the expansion of the distribution function up to the third order in terms of Hermite polynomials. In two dimensions, this model is an off-lattice one and has 16 velocities. The Corner Transport Upwind Scheme is used to evolve the corresponding distribution functions on a square lattice. The resulting code allows one to follow the liquid–vapour phase separation on lattices up to 4096×4096 nodes using a Tesla M2090 Graphics Processing Unit.

1. Introduction: two-dimensional Gauss–Hermite Lattice Boltzmann models

The interest in using Lattice Boltzmann (LB) models for the investigation of large-scale multi-phase fluids is widely expanding because of the parallel nature of their basic algorithm, as well as for the easy incorporation of interparticle forces. An important feature of the LB models is the discretization of the velocity space, which is performed using quadrature methods involving a set of orthogonal polynomials, e.g., the Hermite polynomials [1–3].

The general procedure described in [1–3] allows one to construct the family of two-dimensional ($2D$) Gauss–Hermite Lattice Boltzmann models, denoted as $HLB(N; Q_x, Q_y)$, where N is the order of the model and $Q_l \geq N+1$, $l \in \{x, y\}$ are the quadrature orders along each coordinate axis in the velocity space. The number of vectors in the velocity set of the model $HLB(N; Q_x, Q_y)$ is $Q_x \times Q_y$. A $2D$ Lattice Boltzmann (LB) model is said to be of order N if all the moments

$$\mathcal{M}^{(s_x,s_y)}(\boldsymbol{x},t) = \int f^{\mathrm{eq}} \xi_x^{s_x} \xi_y^{s_y} \mathrm{d}\boldsymbol{\xi}, \qquad 0 \leq s_x + s_y \leq N \tag{1}$$

of the equilibrium distribution function $f^{\mathrm{eq}} \equiv f^{\mathrm{eq}}(\boldsymbol{x}, \boldsymbol{\xi}, t)$ are exactly recovered [1–5]. In this paper, we will consider only the case $Q_x = Q_y = Q$, as usual in LB models.

The tensor Hermite polynomials $\boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi}) \equiv \mathcal{H}_{\boldsymbol{\alpha}}^{(n)}(\boldsymbol{\xi})$, $n = 0, 1, 2 \ldots$, $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, $\alpha_i \in \{x, y\}$, $1 \leq i \leq n$ [1,2,6], are orthogonal with respect to the weight function $\omega(\boldsymbol{\xi})$:

$$\int \omega(\boldsymbol{\xi}) \mathcal{H}_{\boldsymbol{\alpha}}^{(n)}(\boldsymbol{\xi}) \mathcal{H}_{\boldsymbol{\beta}}^{(m)}(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \delta_{n,m} \boldsymbol{\delta}_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{(n)} \tag{2}$$

* Corresponding author.
E-mail addresses: biciusca.tonino@gmail.com (T. Biciuşcă), horga_adi@yahoo.com (A. Horga), sofonea@acad-tim.tm.edu.ro (V. Sofonea).

http://dx.doi.org/10.1016/j.crme.2015.07.011
1631-0721/

where

$$\omega(\boldsymbol{\xi}) = \frac{1}{2\pi}\exp[-\boldsymbol{\xi}^2/2] \tag{3}$$

$$\delta_{n,m} = \begin{cases} 1\ , & m = n \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\boldsymbol{\delta}^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \begin{cases} 1\ , & \text{when } \{\beta_1, \beta_2, \ldots, \beta_n\} \text{ is a permutation of } \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The Gauss–Hermite lattice Boltzmann models $HLB(N;\, Q\,,\, Q\,)$ take advantage of the orthogonality relation, Eq. (2). We briefly underline here the main steps used to construct these models [1,2]:

- the particle distribution function $f \equiv f(\boldsymbol{x}, \boldsymbol{\xi}, t)$ is projected on the orthogonal basis in the momentum space, formed up by the tensor Hermite polynomials $\boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi})$:

$$f(\boldsymbol{x}, \boldsymbol{\xi}, t) = \omega(\boldsymbol{\xi}) \sum_{n=0}^{\infty} \frac{1}{n!}\, \boldsymbol{a}^{(n)}(\boldsymbol{x}, t)\, \boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi}) \tag{6}$$

$$\boldsymbol{a}^{(n)}(\boldsymbol{x}, t) = \int f(\boldsymbol{x}, \boldsymbol{\xi}, t)\, \boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi} \tag{7}$$

  and a similar projection is performed for the equilibrium distribution function f^{eq};
- the derivative $\nabla_{\boldsymbol{\xi}} f$ that appears in the force term of the Boltzmann equation is calculated using the derivatives of Hermite polynomials:

$$\nabla_{\boldsymbol{\xi}} f = \sum_{n=0}^{\infty} \frac{1}{n}\, \boldsymbol{a}^{(n)}(\boldsymbol{x}, t)\, \nabla_{\boldsymbol{\xi}} \left[\omega(\boldsymbol{\xi})\, \boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi})\right] = -\,\omega(\boldsymbol{\xi}) \sum_{n=0}^{\infty} \frac{1}{n!}\, \boldsymbol{a}^{(n)}(\boldsymbol{x}, t)\, \boldsymbol{\mathcal{H}}^{(n+1)}(\boldsymbol{\xi}) \tag{8}$$

- the resulting Boltzmann equation containing f, f^{eq} and $\nabla_{\boldsymbol{\xi}} f$ projected on the tensor Hermite polynomials is truncated at order N;
- the momentum space is discretized using the Gauss–Hermite quadrature [6] of order $Q = N + 1$ on each Cartesian axis; in the $2D$ space this procedure gives the HLB model of order N with $K = Q \times Q$ vectors $\boldsymbol{\xi}_k$ and the corresponding weights w_k;
- for each velocity vector $\boldsymbol{\xi}_k$, $1 \le k \le K$, there is a corresponding function

$$f_k \equiv f_k(\boldsymbol{x}, t) = \frac{w_k}{\omega(\boldsymbol{\xi}_k)}\, f^N(\boldsymbol{x}, \boldsymbol{\xi}_k, t) \quad , \quad f^N(\boldsymbol{x}, \boldsymbol{\xi}, t) = \omega(\boldsymbol{\xi}) \sum_{n=0}^{N} \frac{1}{n!}\, \boldsymbol{a}^{(n)}(\boldsymbol{x}, t)\, \boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi}) \tag{9}$$

  as well as a corresponding equilibrium function f_k^{eq};
- when using the BGK collision term [2,7–10], the functions f_k evolve on a $2D$ square lattice according to ($\partial_t = \partial/\partial_t$, $\partial_\gamma = \partial/\partial_{x_\gamma}$)

$$\partial_t f_k + \xi_{k,\gamma}\, \partial_\gamma f_k = -\frac{1}{\tau}\left[f_k - f_k^{\mathrm{eq}}\right] + F_k \qquad , \qquad 1 \le k \le K \tag{10}$$

  where τ is the relaxation time and [2]

$$F_k = \boldsymbol{g} \cdot \nabla_{\boldsymbol{\xi}} f = w_k \sum_{n=0}^{N-1} \frac{1}{n!}\, \boldsymbol{g}\boldsymbol{a}^{(n)}(x, t)\, \boldsymbol{\mathcal{H}}^{(n+1)}(\boldsymbol{\xi}_k) \tag{11}$$

The Navier–Stokes equations for isothermal systems are recovered when $N = 3$ (the widely used value $N = 2$ is sufficient only in the incompressible limit). According to [2], the expressions of f_k^{eq} and F_k up to order $N = 3$ are:

$$f_k^{\mathrm{eq}} = w_k \rho \left\{ 1 + \boldsymbol{\xi}_k \cdot \boldsymbol{u} + \frac{1}{2}\left[(\boldsymbol{\xi}_k \cdot \boldsymbol{u})^2 - u^2 + (\theta - 1)(\xi_k^2 - 2)\right] \right. \tag{12}$$

$$\left. + \frac{\boldsymbol{\xi}_k \cdot \boldsymbol{u}}{6}\left[(\boldsymbol{\xi}_k \cdot \boldsymbol{u})^2 - 3u^2 + 3(\theta - 1)(\xi_k^2 - 4)\right]\right\} \tag{13}$$

$$F_k = w_k \rho \left\{ \boldsymbol{\xi}_k \cdot \boldsymbol{g} + (\boldsymbol{\xi}_k \cdot \boldsymbol{g})(\boldsymbol{\xi}_k \cdot \boldsymbol{u}) - \boldsymbol{g} \cdot \boldsymbol{u} + \frac{1}{2\rho}\boldsymbol{a}^{(2)}\left[(\boldsymbol{\xi}_k \cdot \boldsymbol{g})\boldsymbol{\mathcal{H}}^{(2)}(\boldsymbol{\xi}_k) - 2\boldsymbol{g}\boldsymbol{\xi}_k\right]\right\} \tag{14}$$

where

Table 1
Cartesian projections of the vectors $\boldsymbol{\xi}_k$ and their corresponding weights calculated by the *product rule* [2], for the two-dimensional lattice Boltzmann models *HLB*(2; 3, 3) and *HLB*(3; 4, 4).

k	*HLB*(2; 3, 3)			*HLB*(3; 4, 4)		
	$\xi_{k,x}$	$\xi_{k,y}$	w_k	$\xi_{k,x}$	$\xi_{k,y}$	w_k
1	0	0	4/9	$+\sqrt{3-\sqrt{6}}$	$+\sqrt{3-\sqrt{6}}$	$(5+2\sqrt{6})/48$
2	$+\sqrt{3}$	0	1/9	$+\sqrt{3+\sqrt{6}}$	$+\sqrt{3-\sqrt{6}}$	1/48
3	0	$+\sqrt{3}$	1/9	$+\sqrt{3-\sqrt{6}}$	$+\sqrt{3+\sqrt{6}}$	1/48
4	$-\sqrt{3}$	0	1/9	$+\sqrt{3+\sqrt{6}}$	$+\sqrt{3+\sqrt{6}}$	$(5-2\sqrt{6})/48$
5	0	$-\sqrt{3}$	1/9	$-\sqrt{3-\sqrt{6}}$	$+\sqrt{3-\sqrt{6}}$	$(5+2\sqrt{6})/48$
6	$+\sqrt{3}$	$+\sqrt{3}$	1/36	$-\sqrt{3+\sqrt{6}}$	$+\sqrt{3-\sqrt{6}}$	1/48
7	$-\sqrt{3}$	$+\sqrt{3}$	1/36	$-\sqrt{3-\sqrt{6}}$	$+\sqrt{3+\sqrt{6}}$	1/48
8	$-\sqrt{3}$	$-\sqrt{3}$	1/36	$-\sqrt{3+\sqrt{6}}$	$+\sqrt{3+\sqrt{6}}$	$(5-2\sqrt{6})/48$
9	$+\sqrt{3}$	$-\sqrt{3}$	1/36	$-\sqrt{3-\sqrt{6}}$	$-\sqrt{3-\sqrt{6}}$	$(5+2\sqrt{6})/48$
10				$-\sqrt{3+\sqrt{6}}$	$-\sqrt{3-\sqrt{6}}$	1/48
11				$-\sqrt{3-\sqrt{6}}$	$-\sqrt{3+\sqrt{6}}$	1/48
12				$-\sqrt{3+\sqrt{6}}$	$-\sqrt{3+\sqrt{6}}$	$(5-2\sqrt{6})/48$
13				$+\sqrt{3-\sqrt{6}}$	$-\sqrt{3-\sqrt{6}}$	$(5+2\sqrt{6})/48$
14				$+\sqrt{3+\sqrt{6}}$	$-\sqrt{3-\sqrt{6}}$	1/48
15				$+\sqrt{3-\sqrt{6}}$	$-\sqrt{3+\sqrt{6}}$	1/48
16				$+\sqrt{3+\sqrt{6}}$	$-\sqrt{3+\sqrt{6}}$	$(5-2\sqrt{6})/48$

$$\rho \equiv \rho(\boldsymbol{x},t) = \sum_{k=1}^{K} f_k = \sum_{k=1}^{K} f_k^{\mathrm{eq}} \tag{15}$$

$$\boldsymbol{u} \equiv \boldsymbol{u}(\boldsymbol{x},t) = \frac{1}{\rho}\sum_{k=1}^{K} f_k \boldsymbol{\xi} = \frac{1}{\rho}\sum_{k=1}^{K} f_k^{\mathrm{eq}} \boldsymbol{\xi} \tag{16}$$

$$\theta \equiv \theta(\boldsymbol{x},t) = \frac{1}{\rho}\sum_{k=1}^{K} f_k \xi^2 - \frac{1}{2}u^2 = \frac{1}{\rho}\sum_{k=1}^{K} f_k^{\mathrm{eq}} \xi^2 - \frac{1}{2}u^2 \tag{17}$$

are the local particle density, particle velocity and temperature, respectively. We note that in the current LB literature [9, 11–14], the force term

$$F_k^{\mathrm{eq}} = -\frac{1}{k_{\mathrm{B}}T}\,\boldsymbol{g}\cdot(\boldsymbol{\xi}-\boldsymbol{u})\,f^{\mathrm{eq}} \tag{18}$$

is widely used instead of F_k given by Eq. (11). This force term emerges from the approximation

$$\nabla_{\boldsymbol{\xi}} f \simeq \nabla_{\boldsymbol{\xi}} f^{\mathrm{eq}} = -\frac{1}{k_{\mathrm{B}}T}\,(\boldsymbol{\xi}-\boldsymbol{u})\,f^{\mathrm{eq}} \tag{19}$$

that is expected to be appropriate for systems not too far from the equilibrium state.

The local acceleration $\boldsymbol{g}$ that appears in Eqs. (11) and (19) originates within the frame of a mean field theory that accounts for the long-range interparticle interaction and is consistent with Enskog's theory of dense fluids [15,16]. The following expression of $\boldsymbol{g}$ is used in order to simulate the evolution of a van der Waals gas that exhibits surface tension [7–9,11,12,15,16]:

$$\boldsymbol{g} = \frac{1}{\rho}\nabla(p^i - p^w) + \kappa\nabla(\Delta\rho) \qquad p^i = \rho T \qquad p^w = \frac{3\rho T}{3-\rho} - \frac{9}{8}\rho^2 \tag{20}$$

where the parameter κ controls the surface tension. Other models in the literature allows one to account for the non-ideal gas pressure and the presence of surface tension by adding a stress tensor to the Navier–Stokes equations [17], or by considering the pseudopotential method of Shan and Chen [18–20].

2. Numerical schemes

2.1. First-order corner transport upwind

In two-dimensional LB models, the functions f_k, $1 \le k \le K$ are defined in the nodes of a square lattice. The corresponding velocity vectors $\boldsymbol{\xi}_k$ can be built by the *product rule* after separation of variables and use of the Gauss quadratures

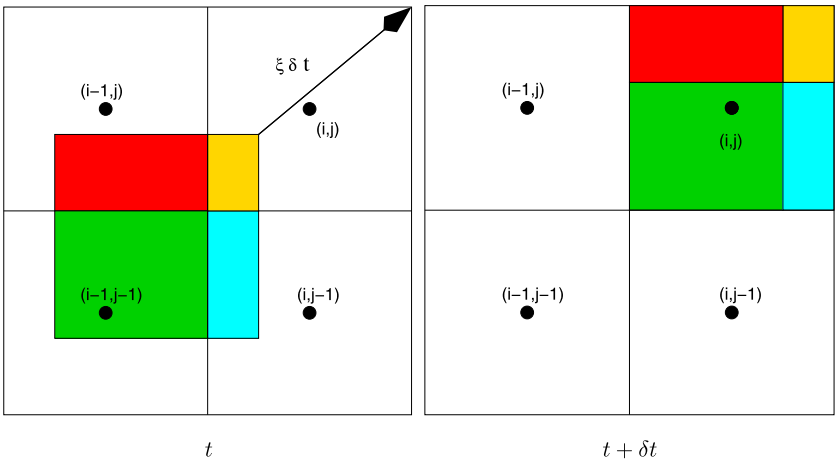

Fig. 1. (Colour online.) The first-order corner transport upwind scheme.

along each Cartesian axis [2]. As seen in Table 1, the velocity vectors $\boldsymbol{\xi}_k$ of the second-order model $HLB(2;3,3)$ are oriented along the Cartesian axes or along the diagonals of the square lattice. This is no longer valid for the third-order model $HLB(3;4,4)$. LB models like $HLB(2;3,3)$, where the velocity set is correlated with the lattice geometry, are called *on-lattice models*. Models like $HLB(3;4,4)$, as well as higher-order models based on Gauss quadratures, which contain velocity vectors whose projections on the Cartesian axes are not expressed by rational numbers, are called *off-lattice models*. Their use is still increasing, especially for the investigation of microfluidics problems involving boundary conditions [4,5,21–25].

The widely-used *collision streaming* scheme [7–9] is very convenient for on-lattice LB models, even if it is generally difficult to be implemented in high-order models that require an elaborated treatment of the momentum set, abscissas, weights, as well as of the boundary conditions [26–32]. Other numerical schemes, including ISLB (interpolation supplemented LB) and FDLB (finite difference LB, with or without flux limiters) exhibit stability problems, especially in the presence of large density gradients [11,33–36].

The first-order corner transport upwind (CTU) scheme, introduced more than two decades ago in the mathematical literature related to hyperbolic equations [37–40], is a natural extension of the well-known first-order upwind scheme introduced by Godunov in the one-dimensional space [39–41]. The CTU scheme is simple enough and is very convenient for solving the lattice Boltzmann evolution equations (10) regardless of the orientation of the velocity vectors $\boldsymbol{\xi}_k$, on square or cubic lattices [38,42]. It is worth to mention here that similar finite-volume schemes, mainly developed for non-uniform meshes, were already used in the so-called *volumetric lattice Boltzmann models* [43–46].

Fig. 1 illustrates the application of the CTU scheme on a $2D$ square lattice when $\xi_{k,x} \geq 0$ and $\xi_{k,y} \geq 0$ (the other cases are handled in a similar manner). The lattice spacing is δs. In this case, the information is transported across the lower left corner of the cell (i,j), which is centered in the lattice node $(x = i\delta s,\ y = j\delta s)$. Let $f^t_{k;(i,j)}$ be the value of f_k at time t in the node (i,j) of the lattice (periodic boundary conditions apply). As a result of the transport process, the value $f^{t+\delta t}_{k;(i,j)}$ of this function at time $t+\delta t$ in the same node (i,j) receives contributions from the four cells that share the corner, i.e. (i,j), $(i-1,j)$, $(i-1,j-1)$ and $(i,j-1)$. According to Fig. 1, these contributions are proportional to the corresponding (coloured) areas in each cell. The CFL condition [40]

$$\max_k \left\{ |\xi_{k,x}|\delta t\,,\ |\xi_{k,y}|\delta t \right\} \leq \delta s \tag{21}$$

ensures that only the nearest neighbours of cell (i,j) contribute to $f^{t+\delta t}_{k;(i,j)}$. Besides the transport process, there are also contributions from the right-hand side of the evolution equation (10) and thus, the new value of the distribution function in the node (i,j) is [40,42]:

$$\begin{aligned} f^{t+\delta t}_{k;(i,j)} = {} & \frac{1}{(\delta s)^2}\Big[f^t_{k;(i,j)}(\delta s - \xi_x\delta t)(\delta s - \xi_y\delta t) + f^t_{k;(i-1,j-1)}(\xi_x\delta t)(\xi_y\delta t) \\ & + f^t_{k;(i-1,j)}(\xi_x\delta t)(\delta s - \xi_y\delta t) + f^t_{k;(i,j-1)}(\delta s - \xi_x\delta t)(\xi_y\delta t)\Big] \\ & - \frac{\delta t}{\tau}\left[f^t_{k;(i,j)} - f^{eq,t}_{k;(i,j)} \right] + \delta t F^t_{k;(i,j)} \end{aligned} \tag{22}$$

2.2. Space derivatives

In each node (i, j) of the square lattice, we use a 25-point stencil to compute the $2D$ derivative operators $\nabla(p^i - p^w)$ and $\nabla(\Delta\rho)$ that appear in Eq. (20). The use of large stencils to compute the space derivatives improves the isotropy of the liquid–vapour interface, as well as the accuracy of the values of the coexistence densities in the phase diagram [10]. Let $q(x, y)$ be the value in the node $(x = i\delta s,\ y = j\delta s)$ of a macroscopic quantity (e.g., the local fluid density ρ or the pressure difference $p^i - p^w$). According to [47], the components of the gradient operator in the node (x, y) are:

$$\partial_x q(x, y) = \frac{1}{(\delta s)^2} \sum_{l,m} \rho(x + l\delta s, y + m\delta s)\, w^{|l|,|m|} c_x^{(l)} + \mathcal{O}[(\delta s)^2] \tag{23a}$$

$$\partial_y q(x, y) = \frac{1}{(\delta s)^2} \sum_{l,m} \rho(x + l\delta s, y + m\delta s)\, w^{|l|,|m|} c_y^{(m)} + \mathcal{O}[(\delta s)^2] \tag{23b}$$

where

$$c_x^{(l)} = l\delta s \quad , \quad c_y^{(m)} = m\delta s \quad , \quad l, m \in \{0, \pm 1, \pm 2\} \tag{24}$$

and [47]

$$\begin{aligned} w^{1,1} &= 2/15 \\ w^{1,0} = w^{0,1} &= 13/30 \\ w^{2,0} = w^{0,2} &= -1/120 \\ w^{2,1} = w^{1,2} &= -1/60 \\ w^{0,0} = w^{2,2} &= 0 \end{aligned} \tag{25}$$

Following [48], the components of the gradient of the Laplacian in the node $(x = i\delta s,\ y = j\delta s)$ are:

$$\partial_x \Delta q(x, y) = \frac{1}{(\delta s)^3} \sum_{l\neq 0, m} \frac{l}{|l|}\, \omega^{|l|,|m|} \rho(x + l\delta s, y + m\delta s) + \mathcal{O}[(\delta s)^2] \tag{26a}$$

$$\partial_y \Delta q(x, y) = \frac{1}{(\delta s)^3} \sum_{l, m\neq 0} \frac{m}{|m|}\, \omega^{|m|,|l|} \rho(x + l\delta s, y + m\delta s) + \mathcal{O}[(\delta s)^2] \tag{26b}$$

where $l, m \in \{0, \pm 1, \pm 2\}$ and [48]

$$\begin{aligned} \omega^{0,0} = \omega^{0,1} = \omega^{0,2} &= 0 \\ \omega^{1,0} &= -5/6 \\ \omega^{1,1} &= -1/6 \\ \omega^{1,2} &= 1/12 \\ \omega^{2,0} = \omega^{2,1} &= 1/6 \\ \omega^{2,2} &= 0 \end{aligned} \tag{27}$$

Both numerical schemes, Eqs. (23) and (26), are not computationally expensive and provide an appropriate degree of isotropy [47–50].

3. GPU simulation results

The evolution equations (10) were solved on a $2D$ lattice with 4096×4096 nodes using a Graphics Processing Unit (GPU) Tesla M2090 with 6 GB memory and 512 cores using the models $HLB(2; 3, 3)$ and $HLB(3; 4, 4)$. These models have 9 and 16 distribution functions defined in each lattice node, respectively. In this paper, we restrict ourselves to the LB simulation of liquid–vapour systems at the Navier–Stokes level. For this reason, we did not consider here larger values of N, as well as of $Q \geq N + 1$, which are required beyond the Navier–Stokes level, i.e. when investigating microfluidics problems [4,5,21–25].

All the distribution functions defined in the node (i, j), $0 \leq i,\ j < 4096$, are processed by a single thread (a sequence of programmed instructions that can be managed independently) at every time step. The architecture of a single GPU allows one to increase the computing speed by launching a very large number of parallel threads [51–53]. The values of the lattice spacing and the time step in Eq. (22) were $\delta s = 1/128$ and $\delta t = 10^{-4}$, respectively. We choose the relaxation time $\tau =$

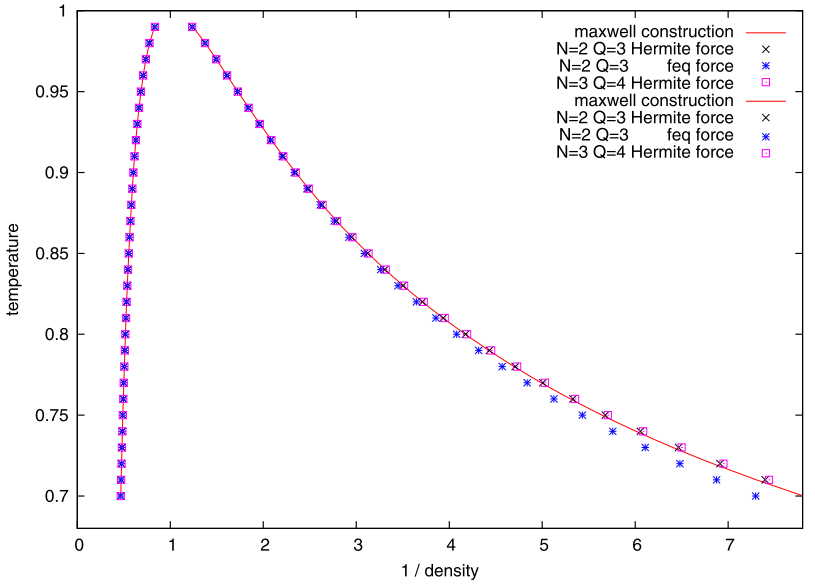

Fig. 2. (Colour online.) Liquid–vapour phase diagram.

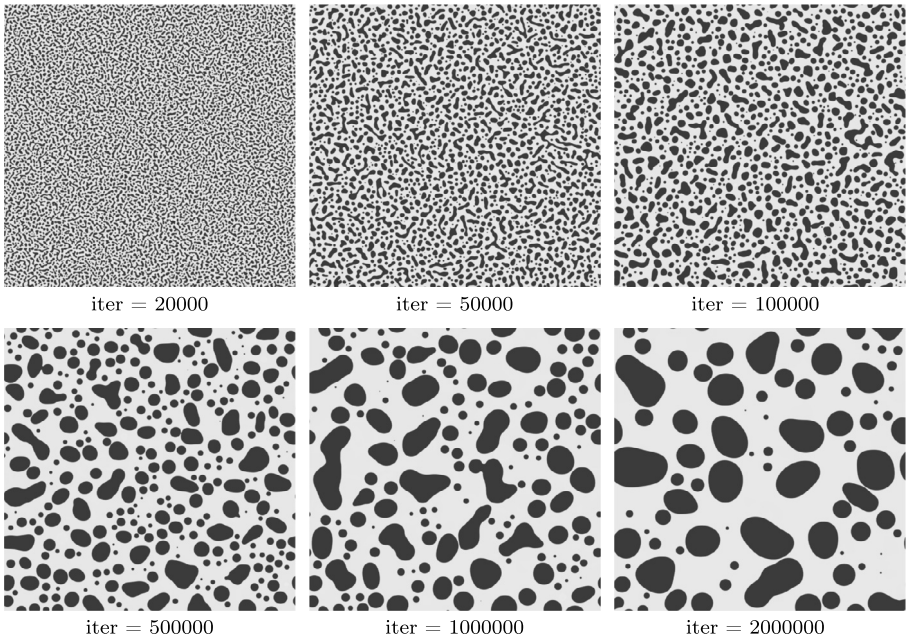

Fig. 3. Evolution of the liquid–vapour phase separation when the mean density of the fluid system is $\rho_{\text{mean}} = 0.90$.

10^{-3} in order to ensure the validity of the Navier–Stokes equations at the macroscopic scale. The value of the constant κ in Eq. (20) was set to $\kappa = 10^{-4}$. During the simulations, we recorded the evolution of the values of the Minkowski functionals (total area $\mathcal{A}$ of the drops, total perimeter $\mathcal{P}$ and number of drops $\mathcal{N}$) calculated using the algorithm described in [54].

The liquid–vapour phase diagram is shown in Fig. 2, where we represented the results obtained with the $HLB(2;3,3)$ model and the two versions of the force term, the "Hermite" force term, Eq. (14) and the "feq" force term, Eq. (18). Also the results obtained with the $HLB(3;4,4)$ model and the "Hermite" force term were represented in this figure. The $HLB(3;4,4)$ results obtained with the "feq" force term were found to be well superposed to the corresponding $HLB(2;3,3)$ results, and therefore were not represented in Fig. 2. In all cases shown in Fig. 2, the values of the liquid density are well superposed to the corresponding values computed using the Maxwell construction. The values of the gas phase density, recovered with the *HLB* models using the "Hermite" force term (14), are always closer to the values computed using the Maxwell construction, while the corresponding values recovered using the f^{eq} force term (18) are farther. Figs. 3 and 4 show the evolution of the liquid–vapour phase separation recovered using the model $HLB(3;4,4)$ and the "Hermite" force term, Eq. (11), on a

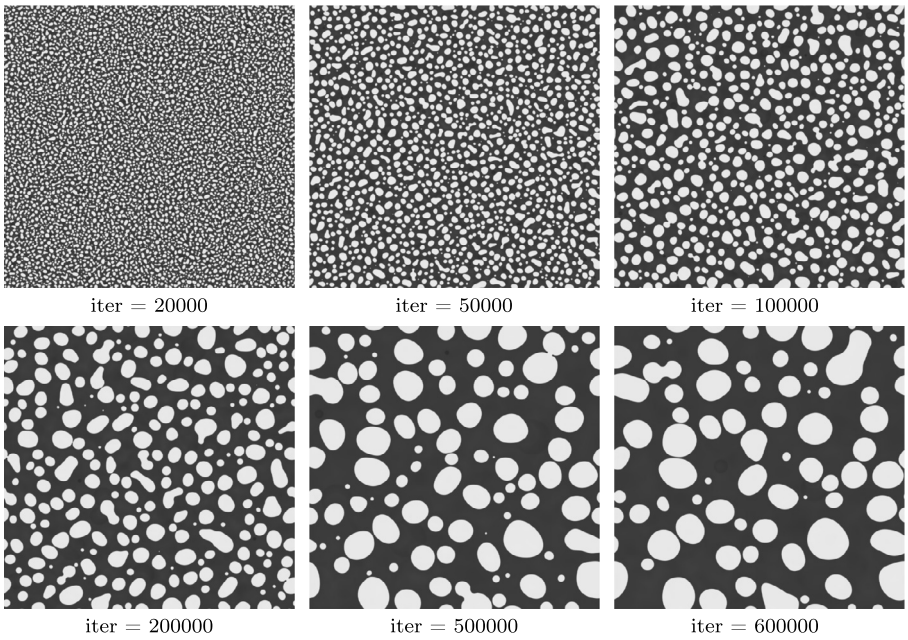

Fig. 4. Evolution of the liquid–vapour phase separation when the mean density of the fluid system is $\rho_{\text{mean}} = 1.30$.

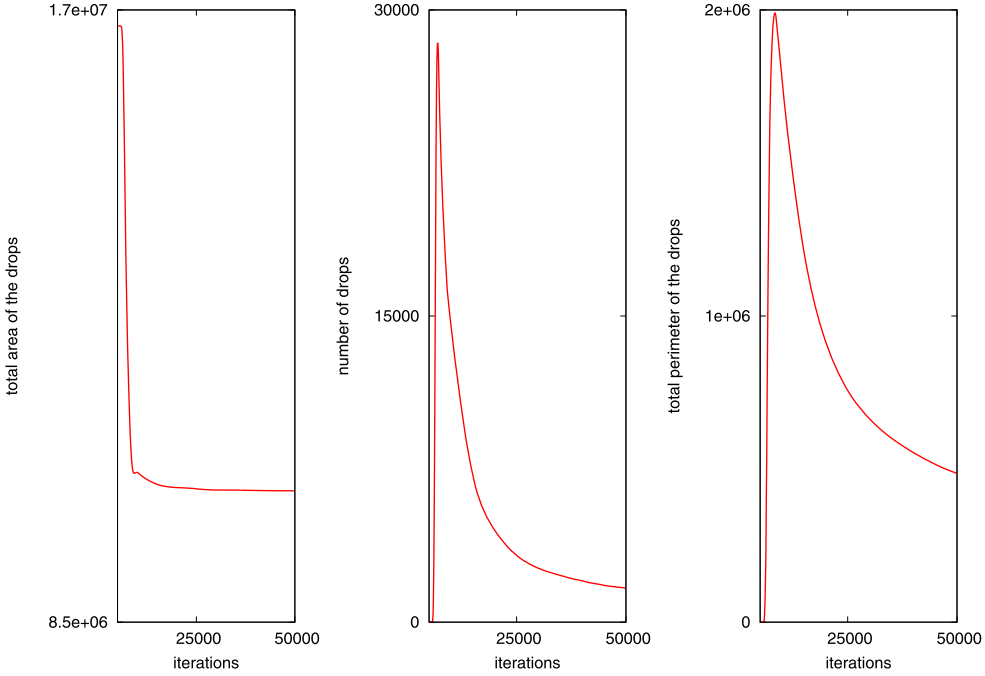

Fig. 5. (Colour online.) Evolution of the Minkowski functionals during the liquid–vapour phase separation when the mean density of the fluid system is $\rho_{\text{mean}} = 0.90$.

square lattice with 4096×4096 nodes. In the initial state (iteration 0, not shown in these figures), the fluid system was quite homogeneous and had the mean density ρ_{mean} (the relative density fluctuations did not exceed 0.001%). Liquid drops or gas bubbles (depending on the value of ρ_{mean}) are formed in the system and the two stages of this process—the early stage spinodal decomposition and the late stage domain growth [55,56]—are clearly seen by following the evolution of the Minkowski functionals (Fig. 5). The number of drops $\mathcal{N}$, as well as the total perimeter $\mathcal{P}$ of the drops, decrease during the late stage domain growth. Fig. 6 shows a log–log plot of the evolution of the mean drop size $1/\mathcal{P}$ that exhibits the crossover between the slopes (growth exponents) 2/3 and /1/2, as already observed in [11,57].

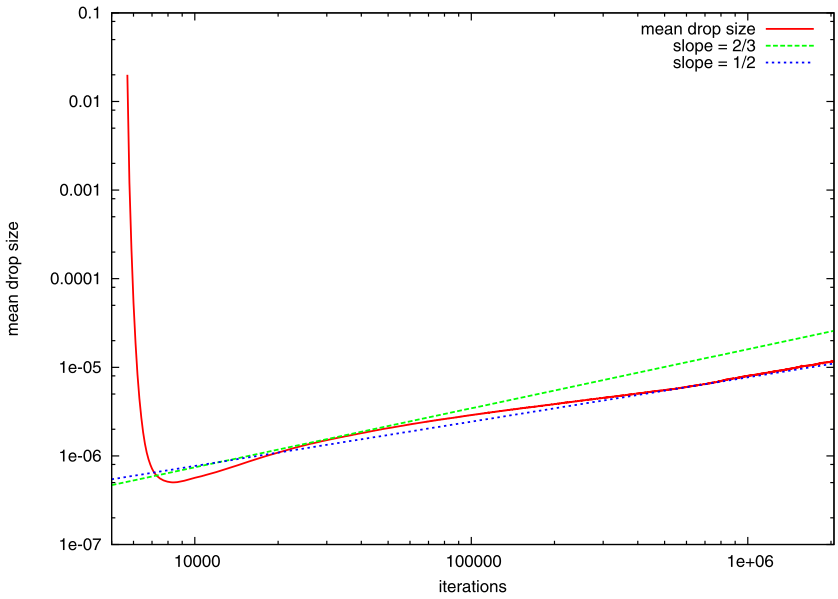

Fig. 6. (Colour online.) Evolution of the mean drop size $1/\mathcal{P}$ during the liquid–vapor phase separation at $\rho_{\text{mean}} = 0.90$ (log–log plot): crossover from the slopes (growth exponents) 2/3 and 1/2.

4. Conclusion

The liquid–vapour phase separation was investigated using the Gauss–Hermite LB models of orders $N \in \{2, 3\}$, with 9 and 16 velocity vectors, respectively. Since the velocity vectors of the model of order $N = 3$ do not lie on the axes or the diagonals of the square lattice used in the LB simulations, the Corner Transport Upwind Scheme was used to evolve the distribution functions in each node of the lattice. The use of the "Hermite" force term, which involves the derivative of the Hermite polynomials, improves the accuracy of the liquid–vapour phase diagram.

Acknowledgements

This work was supported by a grant of the Romanian National Authority for Scientific Research CNCS-UEFISCDI, Project No. PN-II-ID-PCE-2011-3-0516.

References

[1] X. Shan, X. He, Phys. Rev. Lett. 80 (1998) 65.
[2] X. Shan, X. Yuan, H. Chen, J. Fluid Mech. 550 (2006) 413.
[3] B. Piaud, S. Blanco, R. Fournier, V.E. Ambrus, V. Sofonea, Int. J. Mod. Phys. C 25 (2014) 1340016.
[4] V.E. Ambruş, V. Sofonea, Phys. Rev. E 86 (2012) 016708.
[5] V.E. Ambruş, V. Sofonea, Phys. Rev. E 89 (2014) 041301(R).
[6] F.B. Hildebrand, Introduction to Numerical Analysis, 2nd edition, Dover Publications, New York, 1987.
[7] S. Chen, G.D. Doolen, Annu. Rev. Fluid Mech. 30 (1998) 329.
[8] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Clarendon Press, Oxford, 2001.
[9] M.C. Sukop, D.T. Thorne, Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers, Springer, Berlin, 2006.
[10] X. Shan, Phys. Rev. 77 (2008) 066702.
[11] V. Sofonea, A. Lamura, G. Gonnella, A. Cristea, Phys. Rev. E 70 (2004) 046702.
[12] A. Cristea, G. Gonnella, A. Lamura, V. Sofonea, Commun. Comput. Phys. 7 (2010) 350.
[13] X. He, X. Shan, G. Doolen, Phys. Rev. E 57 (1998) R13.
[14] R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous, D. Joseph, Int. J. Multiph. Flow 29 (2003) 117.
[15] L.S. Luo, Phys. Rev. E 62 (2000) 4982.
[16] X. He, G.D. Doolen, J. Stat. Phys. 107 (2002) 309.
[17] B.T. Nadiga, S. Zaleski, Eur. J. Mech. B, Fluids 15 (1996) 885.
[18] X. Shan, H. Chen, Phys. Rev. E 47 (1993) 1815.
[19] L. Chen, Q. Kang, Y. Mu, Y.L. He, W.Q. Tao, Int. J. Heat Mass Transf. 76 (2014) 210.
[20] S. Khajepor, J. Wen, B. Chen, Phys. Rev. E 91 (2015) 023301.
[21] V.E. Ambruş, V. Sofonea, Int. J. Mod. Phys. C 25 (2014) 1441011.
[22] S.H. Kim, H. Pitsch, I.S. Boyd, J. Comput. Phys. 2008 (2008) 8655.
[23] J.P. Meng, Y.H. Zhang, J. Comput. Phys. 230 (2011) 835.
[24] J.P. Meng, Y.H. Zhang, Phys. Rev. E 83 (2011) 036704.
[25] Y. Shi, P.L. Brookes, Y.W. Yap, J.E. Sader, Phys. Rev. E 83 (2011) 045701.
[26] M.O. Deville, T.B. Gatski, Mathematical Modeling for Complex Fluids and Flows, Springer, Berlin, 2012.

[27] P.C. Philippi, L.A. Hegele Jr., L.O.E. dos Santos, R. Surmas, Phys. Rev. E 73 (2006) 056702.
[28] D.N. Siebert, L.A. Hegele Jr., P.C. Philippi, Phys. Rev. E 77 (2008) 026707.
[29] R. Surmas, C.E. Pico Ortiz, P.C. Philippi, Eur. Phys. J. Spec. Top. 171 (2009) 81.
[30] S.S. Chikatamarla, I.V. Karlin, Phys. Rev. E 79 (2009) 046701.
[31] W.P. Yudistiawan, S. Ansumali, I.V. Karlin, Phys. Rev. E 78 (2008) 016705.
[32] W.P. Yudistiawan, S.K. Kwak, D.V. Patil, S. Ansumali, Phys. Rev. E 82 (2010) 046701.
[33] X.Y. He, Int. J. Mod. Phys. C 8 (1997) 737.
[34] X.D. Niu, C. Shu, Y.T. Chew, T.G. Wang, J. Stat. Phys. 117 (2004) 665.
[35] A. Cristea, V. Sofonea, Int. J. Mod. Phys. C 14 (2003) 1251.
[36] V. Sofonea, J. Comput. Phys. 228 (2009) 6107.
[37] P. Colella, J. Comput. Phys. 87 (1990) 171.
[38] R.J. Leveque, SIAM J. Numer. Anal. 33 (1996) 627.
[39] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge, 2001.
[40] J.A. Trangenstein, Numerical Solution of Hyperbolic Partial Differential Equations, Cambridge University Press, Cambridge, 2007.
[41] S.K. Godunov, Mat. Sb. 47 (1959) 271.
[42] D.M. Bond, W. Wheatley, M.N. Macrossan, M. Goldsworthy, J. Comput. Phys. 259 (2014) 175.
[43] F. Nannelli, S. Succi, J. Stat. Phys. 68 (1992) 401.
[44] H. Chen, Phys. Rev. E 58 (1998) 3955.
[45] R. Zhang, H. Chen, Y. Qian, S. Chen, Phys. Rev. E 63 (2001) 056705.
[46] M. Sbragaglia, K. Sugiyama, Phys. Rev. E 82 (2010) 046709.
[47] S. Leclaire, M. El-Hachem, J.Y. Trepanier, M. Reggio, J. Sci. Comput. 59 (2014) 545.
[48] M. Patra, M. Karttunen, Numer. Methods Partial Differ. Equ. 22 (2006) 936.
[49] K.K. Mattila, L.A. Hegele, P.C. Philippi, Sci. World J. 2014 (2014) 142907.
[50] D.N. Siebert, P.C. Philippi, K.K. Mattila, Phys. Rev. E 90 (2014) 053310.
[51] R. Farber, CUDA Application Design and Development, Morgan Kaufmann, Waltham, 2011.
[52] J. Cheng, M. Grossman, T. McKercher, Professional CUDA C Programming, John Wiley and Sons, Inc., Indianapolis, 2014.
[53] CUDA C Programming Guide, http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf.
[54] K. Michielsen, H. De Raedt, Phys. Rep. 347 (2001) 461.
[55] K.R. Mecke, V. Sofonea, Phys. Rev. E 56 (1997) R3761.
[56] V. Sofonea, K.R. Mecke, Eur. Phys. J. B 8 (1999) 99.
[57] Y. Gan, A. Xu, G. Zhang, Y. Li, H. Li, Phys. Rev. E 84 (2011) 046715.