TASK QUARTERLY 8 No 1 (2004), 5–15

THE LATTICE BOLTZMANN METHOD FOR FLUID MIXING: A COMPARISON WITH THE FINITE ELEMENT METHOD

NADANIELA EGIDI, LUCIANO MISICI,
RICCARDO PIERGALLINI AND FRANCESCA TOSI

Dipartimento di Matematica e Informatica,
Università di Camerino,
Madonna delle Carceri, 62032 Camerino, Italy
{*nadaniela.egidi, luciano.misici*}*@unicam.it*

(Received 15 September 2003; revised manuscript received 26 September 2003)

Abstract: In this article we study three-dimensional mixing of an incompressible viscous fluid subjected to the force of rotating blades in a vessel, with a low Reynolds number. The results obtained with the lattice Boltzmann method are compared with the ones previously obtained using the finite element method. All the qualitative and quantitative parameters of the two simulations agree.

Keywords: Lattice Boltzmann method, incompressible viscous fluid mixing, Reynolds number, numerical simulation

1. Introduction

Lattice Boltzmann models (LBM) [1, 2] are revolutionary models in Computational Fluid Dynamics (CFD). They can be seen as an evolution of Cellular Automata (CA) approaches, computational methods offering flexibility and efficiency in modelling complex phenomena capable of providing simple models of complex systems [3, 4]. The most important feature of LBMs is their ability to simulate the collective behavior of the systems considered by describing microscopic interactions of particles through simple laws. They allow a drastic reduction of the computational time due to the simplicity of evolution rules and an easily implementable parallelization, based on the classical technique of domain decomposition.

Unlike conventional methods, which solve the discretized macroscopic Navier-Stokes equations, the philosophy of LBM is to construct simple models that observe these macroscopic equations and describe the microscopic details of a phenomenon. LBMs achieve this by replacing the explicit interactions of particles with a law that describes the relaxation of the system towards an ideal local equilibrium distribution. This can be formally obtained by partitioning the real domain with a regular and symmetric lattice and applying the equation of motion at each node of the grid.

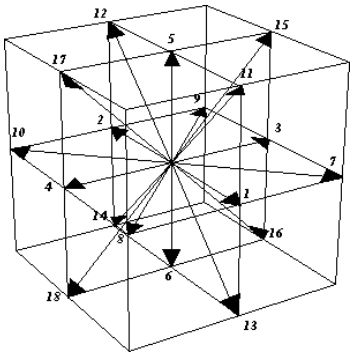

Figure 1. Lattice velocities of D3Q19

D3Q19 is the lattice that shows more stability and, therefore, the one we use in the simulation.

In the D3Q19 model we have a regular grid with 18 links at each grid node. At each node we have 19 possible velocities $\boldsymbol{v}_i$ whose directions are showed in Figure 1. Namely, let:

$$\boldsymbol{e}_i = \begin{cases} (0,0,0), & i=0, \\ (\pm 1,0,0) \text{ or } (0,\pm 1,0) \text{ or } (0,0,\pm 1), & i=1,\ldots,6, \\ (\pm 1,\pm 1,0) \text{ or } (0,\pm 1,\pm 1) \text{ or } (\pm 1,0,\pm 1), & i=7,\ldots,18, \end{cases} \tag{1}$$

then $\boldsymbol{v}_i = c\boldsymbol{e}_i$ is the velocity of each particle along each node direction, where $c = \delta_x/\delta_t$ is the lattice velocity defined using the lattice space-step δ_x and the lattice time-step δ_t. For each $i = 0,\ldots,18$ we denote by $f_i(\boldsymbol{x},t)$ the distribution function of the mass having velocity $\boldsymbol{v}_i$ starting from node of coordinates $\boldsymbol{x} = (x,y,z)$ at time t. From these microscopic quantities we can obtain the following macroscopic variables:

- mass density

$$\rho = \sum_{i=0}^{18} f_i(\boldsymbol{x},t), \tag{2}$$

- hydrodynamic momentum

$$\boldsymbol{j} = \rho \boldsymbol{u} = \sum_{i=0}^{18} \boldsymbol{v}_i f_i(\boldsymbol{x},t), \tag{3}$$

where $\boldsymbol{u}(\boldsymbol{x},t)$ is the fluid velocity in position $\boldsymbol{x}$ at time t.

The evolution step $\mathcal{E}$ can be obtained through a composition of two phases, collision $\mathcal{C}$ and streaming $\mathcal{S}$, that is:

$$\mathcal{E} = \mathcal{S} \circ \mathcal{C}. \tag{4}$$

Using the BGK (Bhatnagar, Gross and Krook, 1954) equation [5], the evolution step $\mathcal{E}$ can be espressed in the following simpler manner:

$$f_i(\boldsymbol{x} + \boldsymbol{v}_i\delta_t, t+\delta_t) = f_i(\boldsymbol{x},t) - \omega(f_i(\boldsymbol{x},t) - f_i^{(eq)}(\boldsymbol{x},t)), \tag{5}$$

where ω is the relaxation parameter and $f_i^{(eq)}$ is the equilibrium distribution.

Passing from the discrete setting to the continuous one, we let $[0,T]$ be the observation time interval and Ω – the simulation domain. From the second-order

expansion of the equilibrium distribution by the lattice gas method of Frisch *et al.* [6] it is possible to obtain the equations of motion:

- continuity

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{6}$$

- Navier-Stokes

$$\frac{\partial (\rho \boldsymbol{u})}{\partial t} + \rho \boldsymbol{u} \cdot (\nabla \boldsymbol{u}) = -\nabla p + \frac{\rho}{\mathrm{Re}} \nabla^2 \boldsymbol{u}, \tag{7}$$

where $t \in [0,T]$, $\boldsymbol{x} = (x,y,z) \in \Omega$, $\boldsymbol{u}(\boldsymbol{x},t) = (u_1(\boldsymbol{x},t), u_2(\boldsymbol{x},t), u_3(\boldsymbol{x},t))$ is the fluid velocity in $\Omega \times [0,T] \subset \mathbb{R}^3 \times \mathbb{R}$, Re is the adimensional Reynolds number that characterizes the fluid, and p is pressure.

We notice that the derivation of (6) and (7) from (2), (3) and (5) makes sense only for a low Mach number $\mathrm{Ma} := U/c_s$, where $p = c_s^2 \rho$ is the pressure of an ideal gas, $c_s = c/\sqrt{3} = \delta_x/(\delta_t \sqrt{3})$ is the speed of sound.

The Reynolds number is defined as:

$$\mathrm{Re} = \frac{UL}{\nu}, \tag{8}$$

where ν is the cinematic viscosity, while U and L are respectively the characteristic speed and length of the stream.

The kinematic viscosity ν is related with the parameters of LBM by:

$$\frac{1}{\nu} = \frac{3}{\delta_t c^2} \left(\frac{1}{\omega} - \frac{1}{2} \right)^{-1}. \tag{9}$$

Assuming that the fluid is incompressible (*i.e.* ρ is constant versus time) Equations (6) and (7) become:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{10}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot (\nabla \boldsymbol{u}) = -\frac{\nabla p}{\rho} + \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{u}. \tag{11}$$

Conditions (10) and (11) are obtained assuming that the microscopic quantities $f_i(\boldsymbol{x},t)$ respect the conservation of mass, ρ, and momentum, $\boldsymbol{j}$, during the simulation. These assumptions are the base of the method and are fulfilled if we use the following expression for the equilibrium distribution in Equation (5) [1]:

$$\begin{array}{ll} f_i^{(eq)} = \frac{1}{3}\rho \left[1 - \frac{3}{2}\boldsymbol{u}^2\right], & i = 0, \\ f_i^{(eq)} = \frac{1}{18}\rho \left[1 + 3(\boldsymbol{v}_i \cdot \boldsymbol{u}) + \frac{9}{2}(\boldsymbol{v}_i \cdot \boldsymbol{u})^2 - \frac{3}{2}\boldsymbol{u}^2\right], & i = 1,\ldots,6, \\ f_i^{(eq)} = \frac{1}{36}\rho \left[1 + 3(\boldsymbol{v}_i \cdot \boldsymbol{u}) + \frac{9}{2}(\boldsymbol{v}_i \cdot \boldsymbol{u})^2 - \frac{3}{2}\boldsymbol{u}^2\right], & i = 7,\ldots,18. \end{array} \tag{12}$$

2. Mathematical formulation of fluid mixing

2.1. Domain

In our simulation we suppose that Equations (10) and (11) are satisfied in the region $\Omega \times [0,T]$, where $\Omega \in \mathbb{R}^3$ is defined as:

$$\Omega = T_C \setminus C, \tag{13}$$

with

$$T_C = \left\{ (x,y,z) \in \mathbb{R}^3 : 0 < z < z_s, x^2 + y^2 < \left(\frac{r_s - r_i}{z_s} z + r_i \right)^2 \right\} \tag{14}$$

and

$$C = \left\{ (x,y,z) \in \mathbb{R}^3 : z_m \le z \le z_M, x^2 + y^2 \le (r_C)^2 \right\}. \tag{15}$$

In the latter definitions z_s is the height of the frustum of the right circular cone T_C, r_i and r_s are its lower and upper base radii, respectively, while r_C is the radius of the circular cylinder C and $(z_M - z_m)$ is its height.

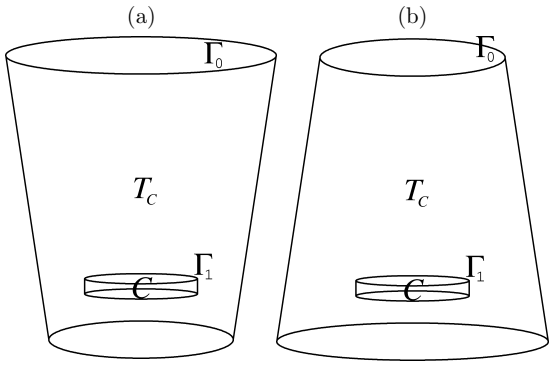

Figure 2. Geometries used in the mixing simulation

Moreover, we have:

$$\begin{gathered} r_i > 0, \qquad r_s > 0, \\ z_s > z_M > z_m > 0, \\ \frac{r_s - r_i}{z_s} z_m + r_i > r_C > 0. \end{gathered} \tag{16}$$

Now we can choose $r_i < r_s$ or $r_i > r_s$, which yields geometries shown in Figure 2a or 2b.

2.2. Boundary conditions

Our aim is a simulation of fluid motion in $\Omega = T_C \setminus C$ under the action of the force of blades rotating inside C. So, we have to impose on $\partial\Omega = \Gamma_1 \cup \Gamma_0 = \Gamma$, where $\Gamma_1 = \partial C$ and $\Gamma_0 = \partial T_C$, the following Dirichlet boundary conditions:

$$\boldsymbol{u}(\boldsymbol{x},t) = \begin{cases} \boldsymbol{g}(\boldsymbol{x},t), & \forall \boldsymbol{x} \in \Gamma_1, \forall t, \\ 0, & \forall \boldsymbol{x} \in \Gamma_0, \forall t. \end{cases} \tag{17}$$

Γ_0 is treated as a solid boundary, so we suppose to have no-slip conditions. Instead, on Γ_1, the particles are forced by the rotating blades using a particular $\boldsymbol{g}$ function that we assume to be initially null and has to fulfil:

$$\int_{\partial\Omega} \boldsymbol{g} \cdot \boldsymbol{n} dS = 0. \tag{18}$$

Below, we give a precise mathematical formulation of the $\boldsymbol{g}$ function which simulates the flux in C, the region containing the rotating blades. We can express this function as:

$$\boldsymbol{g}(\boldsymbol{x},t) = (-yf(\theta), xf(\theta), -(z - z_0)f'(\theta) + h(\theta)\varphi(\rho)), \tag{19}$$

where $\boldsymbol{x}$ is particle position expressed as (x,y,z) and (ρ,θ,z) in Cartesian and cylindrical coordinates, respectively, and $z_0=(z_m+z_M)/2$. In definition (19) we have:

$$\begin{cases} f(\theta)=s(t)\left(\omega_f+\sum_{i=0}^{3}f_i(\theta,t)\right), \\ h(\theta)=s(t)\left(\sum_{i=0}^{3}h_i(\theta,t)\right), \end{cases} \tag{20}$$

where

$$f_i(\theta,t)=l\left(0,\omega_b-\omega_f,\frac{\pi}{4},\theta-\theta_i(t)\right),$$

$$h_i(\theta,t)=\begin{cases} l(0,v_i,\pi,\theta-\theta_i(t)), & \text{if} \quad \sin(\theta-\theta_i(t))<0, \\ l(0,v_i,\frac{\pi}{4},\theta-\theta_i(t)), & \text{if} \quad \sin(\theta-\theta_i(t))\geq 0, \end{cases}$$

$$\varphi(\rho)=\begin{cases} \frac{\left(1-\cos\left(\frac{\pi\rho}{l_b}\right)\right)}{2}, & \text{if} \quad \rho<l_b, \\ 1, & \text{if} \quad \rho\geq l_b. \end{cases}$$

Moreover, ω_b is the angular speed of the blades, l_b is the blades' length, $\omega_f l_b$ is the drag velocity of the fluid, $v_i=(-1)^i\tilde{s}$. In our simulation we fixed $\omega_b=8\pi$, $\omega_f=2\pi$, $l_b=r_C$ and $\tilde{s}=5$.

To complete the definition of $\boldsymbol{g}$, we introduce the following periodic function:

$$l(l_1,l_0,A,a)=\begin{cases} l_1+\frac{l_0-l_1}{2}\left(\cos\left(\frac{2\pi a}{A}\right)+1\right), & \text{if} \quad \cos(a)\geq\cos(A/2), \\ l_1, & \text{if} \quad \cos(a)<\cos(A/2). \end{cases} \tag{21}$$

(Please note that if $a=0$ then $l=l_0$.)

The initial position of each blade is given by $\theta_i=(i\pi)/2$, $i=0,\ldots,3$, while at time t their position is given by $\theta_i(t)=\theta_i+\omega_b t$.

The function $s(t)$,

$$s(t)=\begin{cases} \frac{1-\cos(2\pi t)}{2}, & \text{if} \quad t\leq 0.5, \\ 1, & \text{if} \quad t>0.5, \end{cases} \tag{22}$$

allows us to progress from the initial condition with null velocity to the final stationary velocity at the time $t=0.5$. In fact, using $s(t)$ when $t\geq 0.5$, the drag velocity is effectively $\omega_f l_b$, to which we add the radial velocity along the blades. For further explanations about all mathematical properties of the $\boldsymbol{g}$ function see [7].

3. The setting for LB simulation

To apply the LB method to our domain we approximate it considering a parallelepiped

$$P=\left\{(x,y,z)\in\mathbb{R}^3:-r_m\leq x,y\leq r_m,0\leq z\leq z_s\right\}\supset\Omega, \tag{23}$$

circumscribed to Ω, as shown in Figure 3. We suppose $z_s=n_z\cdot\delta_x$, $2r_m=n_x\cdot\delta_x$ and $r_m=\max\{r_i,r_s\}$. In P we consider a cubic and homogeneous lattice grid made up of $(n_x+1)\times(n_y+1)\times(n_z+1)$ nodes, where $n_y=n_x$.

In particular, the nodes of the lattice are in:

$$\begin{aligned} G=\{(x_i,y_j,z_k):\, & x_i=-r_m+i\delta_x, \quad y_j=-r_m+j\delta_x, \quad z_k=k\delta_x, \\ & i=0,\ldots,n_x, \quad j=0,\ldots,n_y, \quad k=0,\ldots,n_z\}. \end{aligned} \tag{24}$$

Thus (x_i,y_j,z_k) are the coordinates of the node (i,j,k). Using this notation we can say that (i,j,k) is an internal node if $(x_i,y_j,z_k)\in\Omega$; otherwise we say that (i,j,k) is a boundary node.

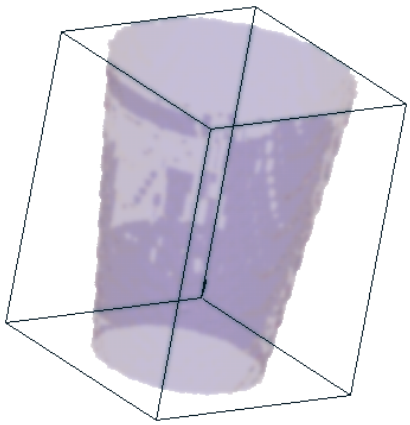

Figure 3. Parallelepiped containing the domain

Additionally, we subdivide the temporal interval $[0,T]$ of our simulation in N_T time-steps, so that $\delta_t = \frac{T}{N_T}$.

For $t=0$, we put $f_i(\boldsymbol{x},0) = f_i^{eq}(\boldsymbol{x},0)$. Then the evolution takes place using Equations (5) and (12) at each lattice node, except for the boundary ones.

The boundary nodes are handled differently, depending on whether they belong to Γ_0 or C. For the nodes in Γ_0, the collision is replaced by the classical bounce-back rule (see [1]), which leaves unchanged the velocity modulus, but in the opposite direction. This means that in Γ_0 particle velocity is null ($\boldsymbol{u}=0$). At the nodes in C, the particles are forced by rotating blades. This is implemented using Equations (5) and (12), but in Equation (12) we use the $\boldsymbol{g}$ function instead of $\boldsymbol{u}$.

The details given in the previous paragraphs are enough to build an algorithm that lets us solve the mixing problem and obtain a simulation of the process. This simulation was carried out for a testing period of $T=14$ seconds and by choosing $N_T = 14000$. Considering the domain, we partitioned P by setting $n_x = n_y = 44$ and $n_z = 56$. Moreover, we made the simulation in a region with the following real dimensions: $\min\{r_i,r_s\} = 0.45$, $\max\{r_i,r_s\} = 0.66$, $z_s = 1.68$, $l_b = r_C = 0.275$, $z_m = 0.27$ and $z_M = 0.36$. The fluid submitted to rotating blades is characterized by a Reynolds number of $\mathrm{Re} = 250$. This value determines unambiguosly the parameter ω as we decided to measure lengths in meters [m] and time in seconds [s], obtaining $L = 0.1\,\mathrm{m}$ and $U = 0.1\,\mathrm{m/s}$.

As a result of our LB simulation we know the velocity field at each lattice node for each time-step. The velocity at a generic point of Ω is computed by interpolation, while we remember that in C the velocity field is given by the $\boldsymbol{g}$ function.

We are thus able to follow a particle and calculate its trajectories during motion. In fact, let $\boldsymbol{\xi}^q(t)$ be the position of a generic particle q at time t and $\boldsymbol{\xi}_0^q$ be its initial position. To obtain the trajectory of particle q we have to solve the following differential equations:

$$\begin{cases} \frac{d\boldsymbol{\xi}^q}{dt} = \boldsymbol{v}(\boldsymbol{\xi}^q,t), \quad \forall t \in [0,T], \\ \boldsymbol{\xi}^q(0) = \boldsymbol{\xi}_0^q, \end{cases} \tag{25}$$

where $\boldsymbol{v}(\boldsymbol{\xi}^q,t)$ is the velocity of particle q that at time t is in $\boldsymbol{\xi}^q$ and is given by:

$$\boldsymbol{v}(\boldsymbol{x},t)=\begin{cases}\boldsymbol{u}(\boldsymbol{x},t), & \forall t\in[0,T],\forall \boldsymbol{x}\in\Omega,\\ \boldsymbol{g}(\boldsymbol{x},t), & \forall t\in[0,T],\forall \boldsymbol{x}\in C.\end{cases} \tag{26}$$

To solve these equations we use a second-order Adams-Bashforth method, subsetting the temporal domain $[0,T]$ in N_T sub-intervals, so that:

$$\Delta t=\frac{10T}{N_T},\qquad t_n=n\Delta t,\qquad 0\le n\le\frac{N_T}{10}, \tag{27}$$

obtaining

$$\boldsymbol{\xi}^q(t_{n+1})=\boldsymbol{\xi}^q(t_n)+\frac{\Delta t}{2}(3\boldsymbol{v}(\boldsymbol{\xi}^q(t_n),t_n)-\boldsymbol{v}(\boldsymbol{\xi}^q(t_{n-1}),t_{n-1})),\quad \forall 1\le n\le\frac{N_T}{10}. \tag{28}$$

We decide to follow only a part of the total fluid and to calculate the trajectories of these particles during motion. Thus we know the position of each particle forced by the blades.

We denote with Q the set of particles we follow. At the time $t=0$ these particles are in $S\cap G$, where:

$$S=\{\boldsymbol{x}\in\Omega\cup C:\frac{z_m+z_M}{2}-0.135\le z\le\frac{z_m+z_M}{2}+0.135,\quad 0.0\le x,y\le r_m\}. \tag{29}$$

The mixing phenomenon can be shown to represent the iso-surface of the function $\delta: G\longrightarrow \mathbb{R}$, defined as:

$$\delta(g)=\sum_{q\in Q}\delta_q(g), \tag{30}$$

with

$$\delta_q(g)=\begin{cases}\exp\frac{d_{g,q}}{d_{g,q}-\varepsilon}, & \text{if}\quad d_{g,q}<\varepsilon\\ 0, & \text{if}\quad d_{g,q}\ge\varepsilon\end{cases}, \tag{31}$$

where $d_{g,q}=d(g,\boldsymbol{\xi}^q)=\boldsymbol{\xi}^q-g$ is the distance of the position of particle q from point g, and $\varepsilon=2\delta_x$. Thus, we obtain the qualitative measure of motion. To obtain its quantitative measure, we define the following set:

$$\begin{aligned}B_s=\Bigg\{&\left(id_1\cos\left(\frac{k\pi}{8}\right),\ id_1\sin\left(\frac{k\pi}{8}\right),\ jd_2+\frac{d_2}{2}\right):\ 0\le k\le 7,\\ &3+6k\le i\le 50,\quad 3+12(s-1)\le j\le 3+12s-1\Bigg\},\end{aligned} \tag{32}$$

where $s=1,\ldots,12$, $d_1=\frac{r_m}{75}$ and $d_2=\frac{z_s}{150}$. Setting P_s as the set of particles that are initially $(t=0)$ in B_s, $\forall s=1,\ldots,12$, we can consider a further split of $\Omega\cup C$ in a N_b^{tot} of little boxes (in our simulation $N_b^{tot}=1100$) which measure $0.13\times0.13\times2d_2$. Each box is occupied if there is at least one particle of P_{tot}, with $P_{tot}=\cup_{s=1}^{12}P_s$; otherwise the box is empty. Setting $N_b(t)$ as the total number of occupied boxes, we can define the dispersion index $\sigma(t)$ as follows:

$$\sigma(t)=\frac{N_b(t)}{N_b^{tot}}. \tag{33}$$

We can consider a further subdivision of $\Omega\cup C$ in a total number L_{tot} of layers L_k (in our simulation $L_{tot}=72$) defined as follows:

$$L_k=\left\{\boldsymbol{x}\in\Omega\cup C:\left(2k+\frac{3}{2}\right)d_2\le z<\left(2k+\frac{7}{2}\right)d_2\right\}\quad k=1,\ldots,L_{tot}. \tag{34}$$

Putting $N_s(t,k)$ as the number of particles of P_s that are in L_k at time t, we can define the diffusion index, $\rho_s(t,k)$, as:

$$\rho_s(t,k) = \frac{N_s(t,k)}{10V_k}, \tag{35}$$

where V_k is the volume of each layer, L_k.

4. Experimental results and conclusion

In our analysis we chose all the geometrical and temporal quantities so as to be able to make a comparison with the numerical simulations of the same mixing problems solved using the finite element method (FEM) in [7]. We have thus obtained the qualitative comparisons shown in Figure 5 and the quantitative ones given by the dispersion index shown in Figure 4 and the diffusion index shown in Figure 6. Similar results have been obtained for the domain shown in Figure 2b. The following figures show that the qualitative and quantitative results obtained using the LBM and the FEM are very similar. We have noted that the dispersion index obtained with the LBM is much greater than the one obtained by the FEM, due to intrinsic properties of the lattice Boltzmann method. The fundamental difference between the two methods is greatly reduced computational time, due to the simplicity of the LBM updating rules. In fact, both methods were implemented in Fortran 90 and run in WS Digital 500

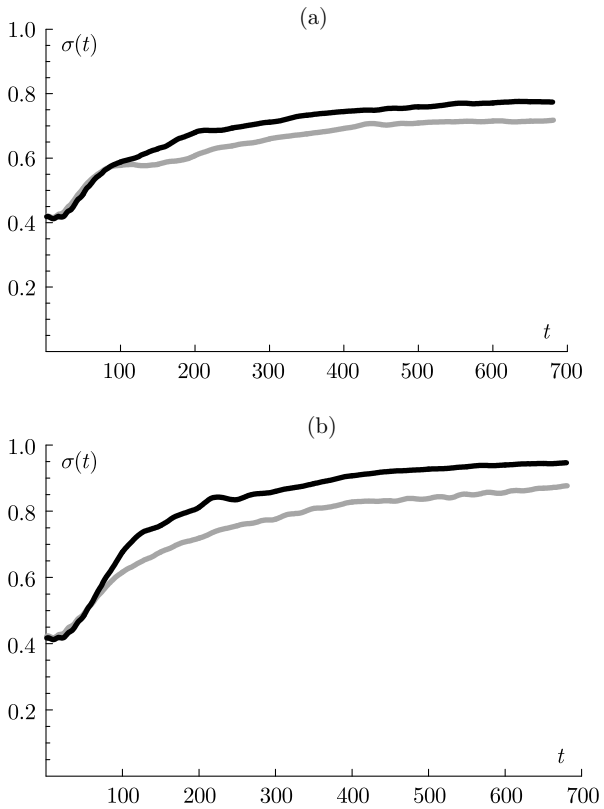

Figure 4. Dispersion index, $\sigma(t)$; the grey and black curves are obtained by using the finite element method and the lattice Boltzmann method, respectively

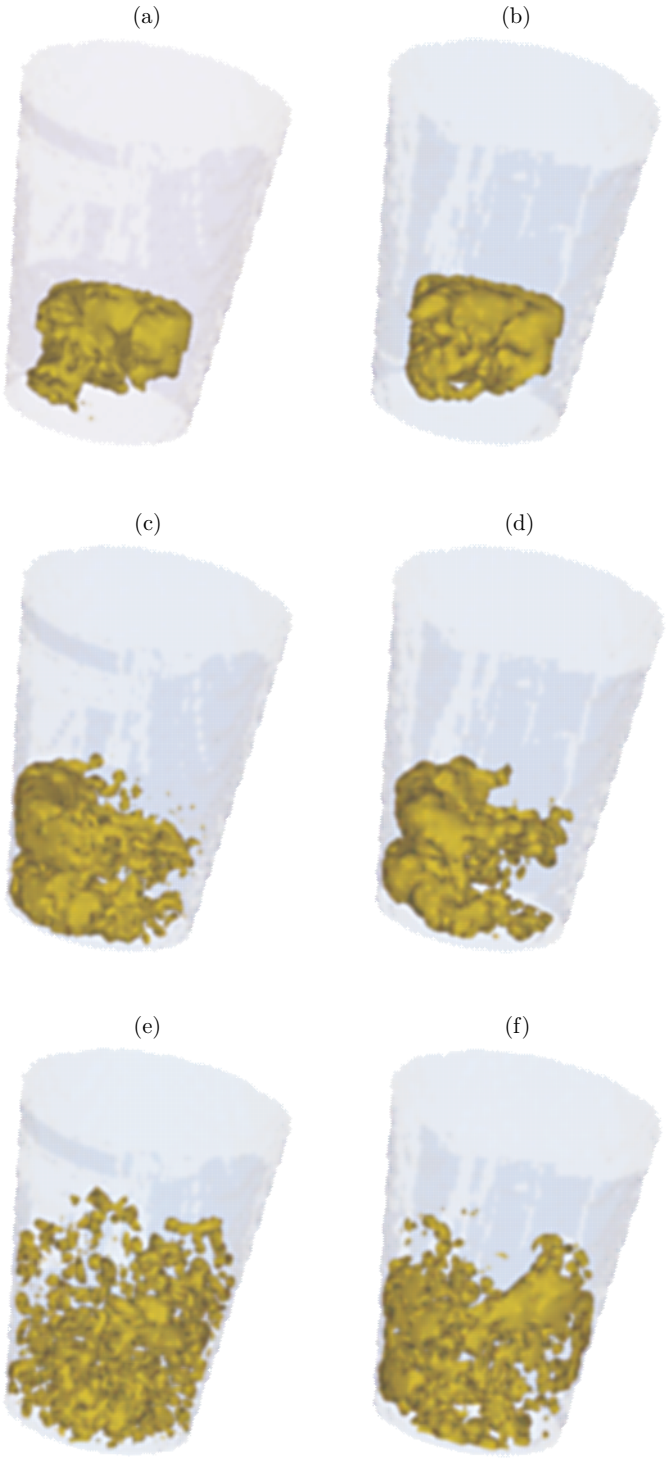

Figure 5. Isosurface $\delta(g) = 0.61$ at different times for the domain shown in Figure 2a: (a) LBM, $t = 0.5$s; (b) FEM, $t = 0.5$s; (c) LBM, $t = 1$s; (d) FEM, $t = 1$s; (e) LBM, $t = 3$s; (f) FEM, $t = 3$s

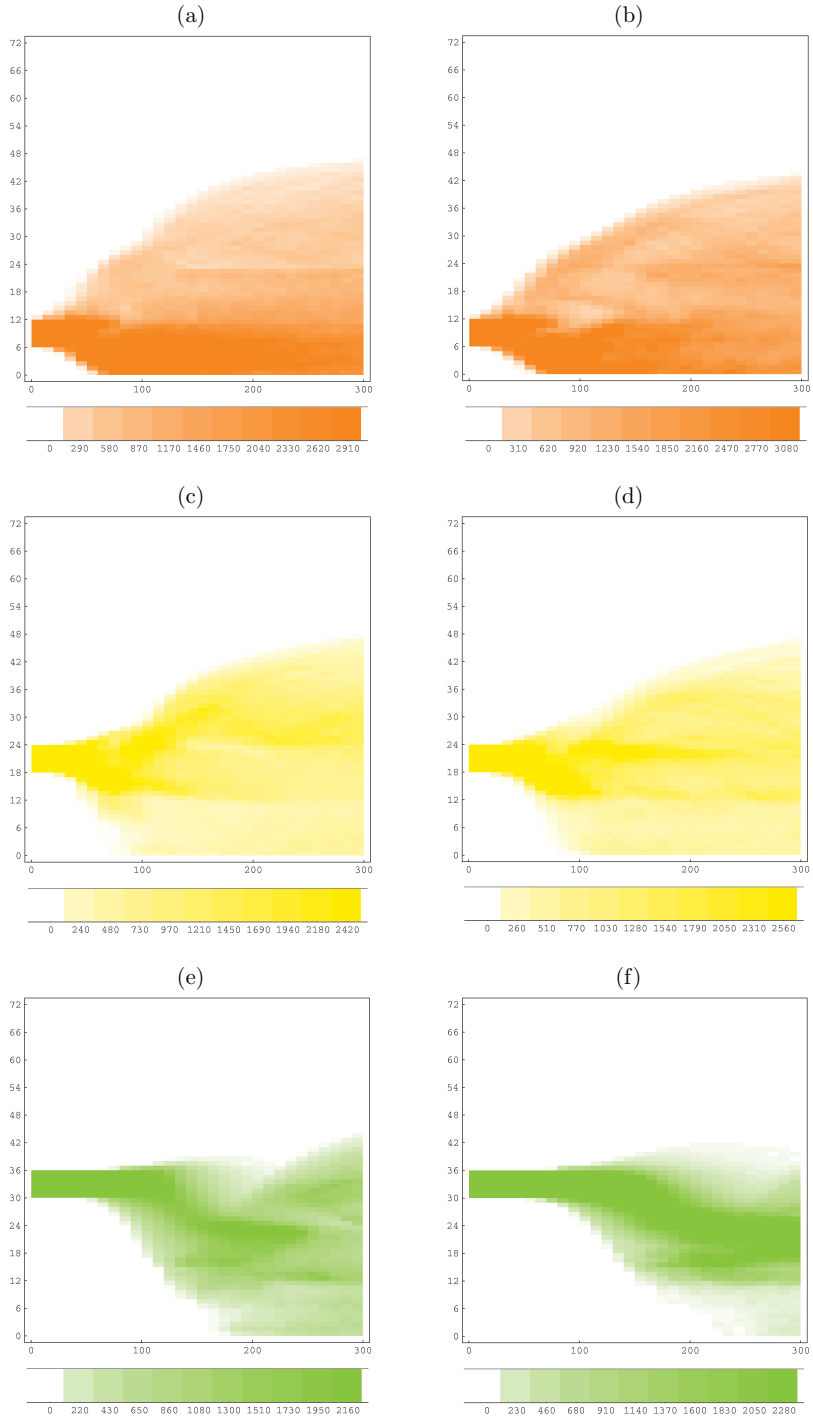

Figure 6. Diffusion index for the domain shown in Figure 2a, represented as a density function of time in the x axis ($100 = 1\,\mathrm{s}$) and of the layers in the y axis: (a) LBM, $\rho_2(t,k)$; (b) FEM $\rho_2(t,k)$; (c) LBM, $\rho_4(t,k)$; (d) FEM $\rho_4(t,k)$; (e) LBM, $\rho_6(t,k)$; (f) FEM $\rho_6(t,k)$

AU/256Mb. To obtain the illustrated results using the FEM we needed 28 minutes of CPU for each simulation describing 0.01 seconds of fluid motion, while using the LBM we needed only 4s of CPU for each simulation describing 0.01s of fluid motion.

Our next goal is to make the parallelism for the complete simulation by utilizing the lattice Boltzmann method for two miscible identical fluids with different colors.

We can conclude that all the parameters of the two simulations for the same problem, using the lattice Boltzmann method and the finite element method, agree, but in velocity field evaluation the computational time of the lattice Boltzmann method is smaller than that of the finite element method.

References

[1] Wolf-Gladrow D A 2000 *Lattice-Gas Cellular Automata and Lattice Boltzmann Models. An Introduction*, Springer
[2] McNamara G and Zanetti G 1988 *Phys. Rev. Lett.* **61** 2332
[3] Wolfram S 1986 *J. Statistical Phys.* **45** 471
[4] Frisch U, Hasslacher B and Pomeau Y 1986 *Phys. Rev. Lett.* **56** 1505
[5] Bhatnagar P, Gross E P and Krook M K 1954 *Phys. Rev.* **94** (3) 511
[6] Frisch U, d'Humières D, Hasslacher B, Lallemand P, Pomeau Y and Rivet J P 1987 *Complex Systems* **1** 649
[7] Egidi N, Misici L and Piergallini R 2001 *TASK Quart.* **5** (1) 71