Theory and applications of an alternative lattice Boltzmann grid refinement algorithm

Alexandre Dupuis and Bastien Chopard
CUI-Computer Science Department, University of Geneva, 24, rue Général-Dufour, CH-1211 Geneva, Switzerland
(Received 6 August 2002; published 20 June 2003)

This contribution proposes an alternative lattice Boltzmann grid refinement algorithm that overcomes the drawbacks that plague existing approaches. We demonstrate that this algorithm is accurate and applicable for all values of the relaxation time. We also show that this algorithm can significantly speed up the flow settlement process. By using a hierarchy of grid levels, the stationary regime can be approached up to a thousand times faster than with a single grid resolution.

DOI: 10.1103/PhysRevE.67.066707 PACS number(s): 02.70.−c, 47.10.+g, 47.11.+j

I. INTRODUCTION

The lattice Boltzmann (LB) models are rather different numerical techniques aimed at modeling a physical system in terms of the dynamics of fictitious particles [1,2]. This method is now considered as a serious alternative to standard computational fluid dynamics [3]. The main idea of this approach is to model the physical reality at a mesoscopic level: the generic features of microscopic processes can be expressed through simple rules, from which the desired macroscopic behavior emerges as a collective effect of the interactions between the many elementary components.

Originally, the LB models are built on regular lattices. In order to resolve the fluid flow with enough accuracy, one is tempted to use a very fine lattice, thus requiring more computing resources. However, with irregular flows, high grid resolution is only needed in some specific regions. Refining the lattice locally, where more precision is needed, may thus represent a significant improvement. In order to connect the two scales, an algorithm allowing us to pass on the results simulated on the fine lattice to the coarse one and vice versa is necessary. Existing published algorithms present some weaknesses. Here, we propose an alternative, more general and yet simpler technique which we validate with a simple flow.

We then apply our lattice refinement technique to speed up the flow settlement process. Settling a flow is definitely a complicated and a necessary step [4,5], as the time needed to reach a stationary regime from an arbitrary initial condition is generally not negligible. Hence, decreasing the duration of the transient stage is an issue of great interest.

The paper is organized as follows. First, in Sec. II, we recall the fundamentals of the LB models. Section III starts to describe existing refinement algorithms and then presents our approach that is compared to the existing ones. Then, we present in Sec. IV an algorithm allowing us to speed up the flow settlement process. Depending on the desired accuracy, a CPU gain of about 1000 can be achieved. Finally, we draw some conclusions.

II. LATTICE BOLTZMANN MODELS

A LB fluid is described by density distribution functions $f_i(\mathbf{r},t)$ giving the probability that a fictitious fluid particle with velocity $\mathbf{v}_i$ enters the lattice site $\mathbf{r}$ at a discrete time t. At each time step, the pseudoparticles entering the same site collide, i.e., the corresponding density distributions interact. Then, the resulting distribution functions are moved to the neighboring sites, mimicking the motion of the pseudoparticles.

The admissible velocities $\mathbf{v}_i$, of component $v_{i\alpha}$, are dependent on the lattice topology. Usually, i runs between 0 and z, where z is the lattice coordination number (i.e., the number of lattice links). By convention $\mathbf{v}_0=\mathbf{0}$ and f_0 represents the density distribution of particles at rest. For many lattice topologies the set of vectors $\mathbf{v}_i$ can be divided into slow and fast velocities: slow velocities correspond to a jump to a nearest neighbor site while fast velocities imply a jump to a second nearest neighbor.

As explained above, the dynamics of a LB model alternates between collision and propagation phases. It is sometimes necessary to distinguish the density distributions before collision and after collision. For instance, some quantities such as the stress tensor take different values if measured before or after the collision process. Thus, we define $f_i^{\mathrm{in}}=f_i$ as the precollision value and f_i^{out} as the new value after the collision process. After the propagation step, the postcollision values become precollision distributions at the nearest neighbors,

$$f_i^{\mathrm{in}}(\mathbf{r}_1+\Delta t\mathbf{v}_i,t+\Delta t)=f_i^{\mathrm{out}}(\mathbf{r},t). \tag{1}$$

In the so-called Bhatnagar-Gross-Krook (BGK) model [1,6,7], the collision step is computed as

$$f_i^{\mathrm{out}}(\mathbf{r},t)=f_i^{\mathrm{in}}(\mathbf{r},t)+\frac{1}{\tau}[f_i^{\mathrm{eq}}(\mathbf{r},t)-f_i^{\mathrm{in}}(\mathbf{r},t)]+\frac{\Delta t}{v^2C_2}\mathbf{v}_i\cdot\mathbf{G}. \tag{2}$$

These two equations define the dynamics of a LB fluid. The quantity Δt is the time step of the simulation, $\mathbf{G}$ is a possible body force, τ is the relaxation time, and f_i^{eq} is the local equilibrium which is a function of the density $\rho=\Sigma_{i=0}^{z}m_i f_i^{\mathrm{in}}$ and the fluid velocity $\mathbf{u}$ defined through the relation $\rho\mathbf{u}=\Sigma_{i=0}^{z}m_i f_i^{\mathrm{in}}\mathbf{v}_i$. The quantities m_i are weights associated with the lattice directions and C_2 is a geometrical constant defined as $\Sigma_{i=1}^{z}m_i v_{i\alpha}v_{i\beta}=v^2C_2\delta_{\alpha\beta}$.

TABLE I. Constants of the most common $DdQ(z+1)$ lattices, where d is the spatial dimension and z is the number of link, $\mathbf{v}_i$ is the velocity on link i and m_i are the weights associated with each link.

	Slow velocities		Fast velocities					
Models	$\lvert\mathbf{v}_i\rvert$	m_i	$\lvert\mathbf{v}_i\rvert$	m_i	C_0	C_2	C_4	c_s^2
$D2Q7$	v	1			6	3	$\frac{3}{4}$	$\frac{1}{4}$
$D2Q9$	v	4	$\sqrt{2}v$	1	20	12	4	$\frac{1}{3}$
$D3Q15$	v	1	$\sqrt{3}v$	$\frac{1}{8}$	7	3	1	$\frac{1}{3}$
$D3Q19$	v	2	$\sqrt{2}v$	1	24	12	4	$\frac{1}{3}$

It can be shown (see, for instance, Refs. [2,8,6]) that Eqs. (1) and (2) reproduce a hydrodynamic behavior if the local equilibrium functions are chosen as follows (Greek indices label spatial coordinates):

$$f_i^{\mathrm{eq}}=\rho\left[\frac{1}{C_2}\frac{c_s^2}{v^2}+\frac{1}{C_2}\frac{v_{i\alpha}u_\alpha}{v^2}+\frac{1}{2C_4v^4}\sum_{\alpha\beta}\left(v_{i\alpha}v_{i\beta}-v^2\frac{C_4}{C_2}\delta_{\alpha\beta}\right)u_\alpha u_\beta\right],$$
$$f_0^{\mathrm{eq}}=\rho\left[1-\frac{C_0}{C_2}\frac{c_s^2}{v^2}+\left(\frac{C_0}{2C_2}-\frac{C_2}{2C_4}\right)\frac{u^2}{v^2}\right]. \tag{3}$$

Table I gives the values of the coefficients C_k and the weights m_i for a few standard lattice topologies noted $DdQ(z+1)$, where d is the spatial dimension. The quantity v gives the speed unit that corresponds to the modulus of the slow velocities, i.e., $v=\Delta\mathbf{r}/dt$, where $\Delta\mathbf{r}$ is the lattice spacing.

Then Eqs. (1) and (2) are equivalent to the continuity equation and Navier-Stokes equation with speed of sound c_s and viscosity

$$\nu=\Delta t v^2\frac{C_4}{C_2}\left(\tau-\frac{1}{2}\right). \tag{4}$$

The two free parameters are c_s and τ. An obvious constraint on these parameters is that the f_i's and the viscosity remain positive, which implies that $\tau>0.5$ and $c_s^2<(C_2/C_0)v^2$. A commonly chosen value for c_s is $c_s^2=v^2(C_4/C_2)$.

A relaxation time τ close to 0.5 implies a small viscosity but also, if the lattice spacing is not fine enough, some numerical instabilities. To solve the problem, one can have recourse to a turbulence model which, roughly, computes the effects of the unresolved scales on some physical quantities (see Refs. [9,4,1] for details). To simulate low viscosities, one can also consider refining the lattice.

III. A MULTIGRID ALGORITHM

A local lattice refinement consists in refining the lattice spacing ($\Delta\mathbf{r}$) in order to simulate smaller scales or, simply, to get more precision. Computational time being related to the lattice size, an improvement consists in refining the lattice locally by defining a coarse lattice everywhere and by defining a fine lattice where extra accuracy is needed. One consequently needs an algorithm to connect coarse and fine lattices.

A. Existing approaches

A well-known grid refinement model is the one due to Filippova and Hänel [10]. This scheme proposes the following relations between the fields of the fine and the coarse lattice:

$$f_i^{\mathrm{out},c}=f_i^{\mathrm{eq},f}+(f_i^{\mathrm{out},f}-f_i^{\mathrm{eq},f})\frac{n_{\mathrm{ref}}(\tau_c-1)}{\tau_f-1},$$
$$f_i^{\mathrm{out},f}=\tilde{f}_i^{\mathrm{eq},c}+(\tilde{f}_i^{\mathrm{out},c}-\tilde{f}_i^{\mathrm{eq},c})\frac{\tau_f-1}{n_{\mathrm{ref}}(\tau_c-1)}, \tag{5}$$

where $\tilde{f}_i$ denotes the spatially and temporally interpolated value of the coarse grid fields. The indices c and f indicate quantities belonging to the coarse or the fine lattice, respectively. Finally n_{ref} is the ratio between the coarse and the fine lattice spacing, i.e., $n_{\mathrm{ref}}=\Delta\mathbf{r}_c/\Delta\mathbf{r}_f$.

Another approach to grid refinement is the one by Lin and Lai [11]. It proposes a simpler algorithm without considering a rescaling of the f_i's. The authors argue that the fields f_i are interchangeable after the streaming step.

Two major remarks on the existing models presented above have to be made. First, Eqs. (5) of the Filippova model present a singularity when $\tau=1$, which reduces the generality of the model. Also, the proposed transformation is unnecessarily complicated (see below).

Second, Lin's model is inaccurate as it considers the fields as interchangeable. This is not the case (although the difference between coarse and fine fields is small), due to the nonequilibrium part of the distribution function.

The grid refinement technique we propose in the following section offers an accurate scheme that does not neglect the nonequilibrium parts of the f_i's. In addition, it is simpler than Filippova's one and has no singularity for $\tau=1$.

B. Our grid refinement algorithm

In order to have the same molecular velocities (v) on different lattices, we choose to keep constant the ratio $\Delta\mathbf{r}/\Delta t$. This choice implies that the time step Δt changes from one grid level to the other but that $\mathbf{u}$ is identical on all lattices.

On the other hand, this specific choice requires to modify the relaxation time τ in order to keep the viscosity constant across the grids. Hence, the relaxation time on a fine lattice is computed using Eq. (4) as

$$\tau_f = \frac{\Delta \mathbf{r}_c}{\Delta \mathbf{r}_f}\left(\tau_c - \frac{1}{2}\right) + \frac{1}{2}, \tag{6}$$

where indices c and f denote coarse and fine quantities, respectively.

We now describe our algorithm. A field can be decomposed into its equilibrium and nonequilibrium parts [2] as

$$f_i^{\mathrm{in}} = f_i^{\mathrm{eq}} + f_i^{\mathrm{neq}}. \tag{7}$$

Let us examine them. First, Eqs. (3) show that the equilibrium part is a function of ρ and $\mathbf{u}$. It is neither dependent on $\Delta \mathbf{r}$ nor on Δt and is, neglecting discretization errors, identical on lattices with different resolution. Second, the nonequilibrium part can be written as [2]

$$f_i^{\mathrm{neq}} = \frac{\Delta t \tau}{C_2 v^2}\left(\sum_{\gamma\delta} v_{i\gamma} v_{i\delta} \partial_\gamma \rho u_\delta - c_s^2 \,\mathrm{div}\,(\rho \mathbf{u})\right) \equiv \Delta t \tau C(\rho, \mathbf{u}), \tag{8}$$

where $C(\rho,\mathbf{u})$ is a function of ρ, $\mathbf{u}$ and their derivatives only. The quantities depending on ρ and $\mathbf{u}$ are identical on lattices of different scales as ρ and $\mathbf{u}$ are lattice blind. Consequently, only the coefficient $\Delta t \tau$ in front of the nonequilibrium part has to be rescaled when one wishes to connect different lattices.

It follows that one can express a relation between fields of different lattices as

$$f_i^{\mathrm{eq},f} = f_i^{\mathrm{eq},c} = f_i^{\mathrm{eq}}. \tag{9}$$

Considering Eq. (8) and the fact that $C(\rho,\mathbf{u})$ is identical on both lattices, the following relation between nonequilibrium distributions can be written:

$$f_i^{\mathrm{neq},f} = \frac{\Delta t_f \tau_f}{\Delta t_c \tau_c} f_i^{\mathrm{neq},c} = \frac{\tau_f}{n_{\mathrm{ref}} \tau c} f_i^{\mathrm{neq},c}. \tag{10}$$

Combining Eqs. (9) and (10), one can easily express how to transform coarse to fine fields:

$$f_i^{\mathrm{in},c} = f_i^{\mathrm{eq}} + f_i^{\mathrm{neq},c} = f_i^{\mathrm{eq}} + f_i^{\mathrm{neq},f} \frac{n_{\mathrm{ref}} \tau_c}{\tau_f}$$

$$= f_i^{\mathrm{eq}} + (f_i^{\mathrm{in},f} - f_i^{\mathrm{eq}}) \frac{n_{\mathrm{ref}} \tau_c}{\tau_f}. \tag{11}$$

Similarly one can express the fine field transformation. Hence, fields are transformed by the following relations:

$$f_i^{\mathrm{in},c} = f_i^{\mathrm{eq}} + (f_i^{\mathrm{in},f} - f_i^{\mathrm{eq}}) \frac{n_{\mathrm{ref}} \tau_c}{\tau_f},$$

$$f_i^{\mathrm{in},f} = \tilde{f}_i^{\mathrm{eq}} + (\tilde{f}_i^{\mathrm{in},c} - \tilde{f}_i^{\mathrm{eq}}) \frac{\tau_f}{n_{\mathrm{ref}} \tau_c}, \tag{12}$$

where $\tilde{f}_i$ denotes the spatially and temporally interpolated value of the coarse grid fields. In our approach we use the same interpolation scheme as proposed by Lin [11]. We refer the reader to this reference in which this rather technical aspect of the method is well explained.

In summary, our approach rescales the incoming fields while the one by Filippova and Hänel rescales the outcoming fields. Ours has the advantage of being more general than the one of Filippova and Hänel [10] as the singularity arising when $\tau = 1$ is not present anymore. Moreover, the collision operator is applied also on the boundary of finer lattices, which is not the case in the Filippova model. Finally, our method is obviously more accurate than the simple one proposed by Lin and Lai [11], which does not consider the nonequilibrium part of the distribution.

The following section is devoted to the validation of our approach and to the numerical comparison of the three techniques presented here.

C. Validation

1. Field decomposition

We start by highlighting the field decomposition into an equilibrium and a nonequilibrium part. For that, we consider a $D2Q9$ lattice with a BGK (LBGK) model Poiseuille flow on a channel of length L and diameter $D = 1.0$, accelerated by a constant body force.

Space is discretized on a longitudinally periodic lattice of size $N_x \times N_z = 11 \times 33$ so that $L = N_x \Delta \mathbf{r}$ and $D = (N_z - 1)\Delta \mathbf{r}$. The fact that the number of lattice points N_x and N_z do not contribute the same way to the physical size L and D is because the x axis is periodic while the z axis is closed by two walls.

We now consider a refinement factor $n_{\mathrm{ref}} = 2$, that is, a lattice of size 22×65. Since the system is periodic along the x axis, and the dynamics invariant under a horizontal translation, N_x is irrelevant, we may as well consider a system of length $L/2$. Thus, in order to save CPU time, we again take $N_x = 11$.

The flow is settled by imposing a constant body force $\mathbf{G} = (G_x, 0)$. With the definition

$$g_i = \frac{m_i \Delta t}{v^2 C_2} \mathbf{G} \cdot \mathbf{v}_i,$$

we choose $g_i^c = 3.75 \times 10^{-4} (m_i/36) v_{ix}$ and $g_i^f = 1.875 \times 10^{-4} (m_i/36) v_{ix}$.

At the upper and lower walls, we use the so-called Inamuro nonslip boundary conditions [12]. We set the viscosity $\nu = 0.005$ and the maximum velocity $U_c = 0.1$. With relation (4), one can then easily compute relaxation times which are equal to $\tau^c = 0.98$ and $\tau^f = 1.46$ for the coarse and the fine lattice, respectively.

The nine fields f_i are measured at the point $(x = 0, y = D/4)$ common to both lattices. As expected, their equilibrium parts are identical up to the discretization errors (i.e., $|f_i^{\mathrm{eq},f} - f_i^{\mathrm{eq},c}| < 10^{-8}$). Thus, we focus our attention on the nonequilibrium part of the fields which is computed as f_i^{neq}

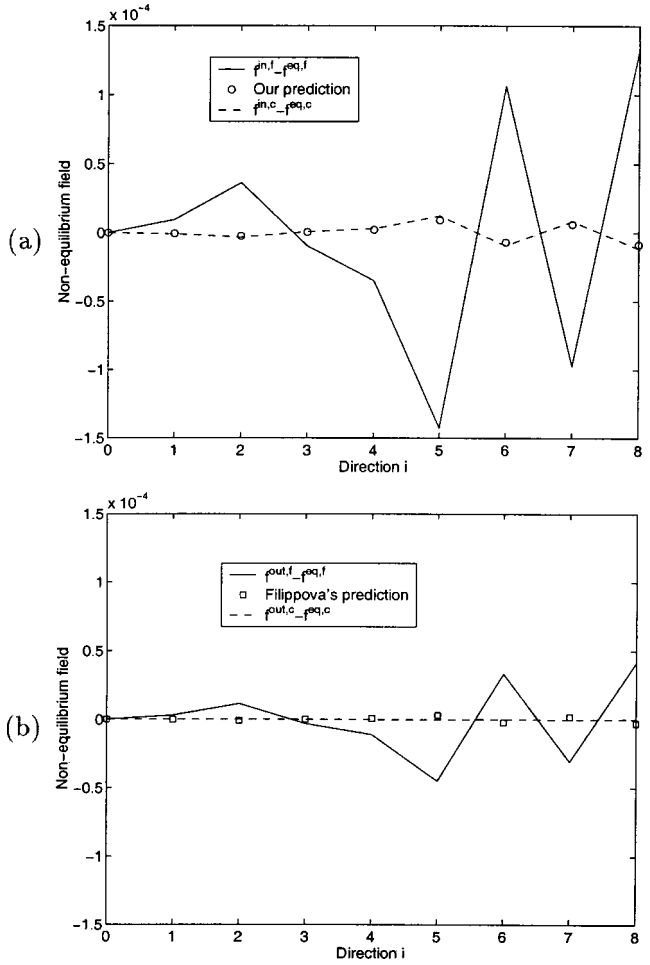

FIG. 1. Prediction of coarse from fine nonequilibrium fields using (a) our transformation and (b) Filippova's transformation equations, see Eqs. (12) and (5), respectively. Applying transformations (12) and (5) on the solid lines, one obtains the dots (circles and squares). If the proposed transformations are correct one should obtain the dashed lines. We observe accurate transformation in both cases.

$=f_i^*-f_i^{\mathrm{eq}}$, where the symbol $*$ stands for *in* or *out* and indicates if this part is computed considering the incoming or the outcoming fields, i.e., using Filippova's model or ours. We then talk about incoming or outcoming nonequilibrium part. Note that when nothing is specified the incoming nonequilibrium part is considered.

We simulate the stationary flow on the coarse and on the fine lattice. The two sets of transformation equations (12) and (5) are used to switch from coarse to fine fields. The results simulated on the fine lattice are used to check the accuracy of each method. The results are presented in Fig. 1. Notice that both transformation equations, Filippova's and ours, produce accurate results. In the following section, we will see that Filippova's scheme no longer works for τ close to 1. Finally, we observe that the nonequilibrium part is rather small. It represents a small percentage of the total, i.e., $f_i^{\mathrm{neq}}/f_i \approx 10^{-4} \ll 1$. It is probably the reason why Lin and Lai neglect this part.

From this first validation, we conclude that the nonequilibrium part is nonzero and depends on $\Delta\mathbf{r}$ and Δt. Hence, one has to rescale it to use the values of one lattice to set the values on another. Thus we see that the Lin model presented above misses some aspects of the LBGK models as it does not consider a rescaling process. Depending on the simulation (e.g., if gradients are large), it can be prejudicial to use this simplistic model.

2. Local refinement of a Poiseuille flow

We continue our validation by considering a local lattice refinement. Consider again a LBGK $D2Q9$ Poiseuille flow on a coarse $N_x \times N_z = 10 \times 17$ lattice which is longitudinally periodic. The coarse lattice is locally refined by considering a 10×15 patch that refines the first seven sites of the coarse lattice along z direction. The flow is settled by applying a constant body force and the Inamuro nonslip boundary condition [12]. The diameter of the channel $D=1.0$, the velocity in the center of the channel is $U_c=0.1$. In order to highlight the singularity around $\tau_c=1$ in the Filippova model, we choose $\tau_c = 1.0 \pm \epsilon_1$.

Using the above algorithms we simulate the flows on both lattices. Missing fields of the fine lattice on the top layer (i.e., $z=14\times\Delta\mathbf{r}_f$) are then determined by the coarse ones.

With the considered Inamuro's boundary conditions, simulation results on any (complete) lattice differ from the theoretical ones only by the numerical precision of the computer (considering a Poiseuille flow, recall that Inamuro's condition produces exact results). Hence, the results of the coarse lattice can be considered as exact. This is not the case for the results of the fine lattice as it is not complete. So in order to check the accuracy of the three algorithms (ours, Filippova's, and Lin's) we measure an error (E) on the fine lattice as

$$E = \left(\frac{\sum_z \left[u_x(z) - U_x(z) \right]^2}{\sum_z U_x^2(x)} \right)^{1/2}, \tag{13}$$

where $u_x(z)$ is the simulated velocity and

$$U_x(z) = 4U_c \frac{z\Delta\mathbf{r}_f}{D} \left(1 - \frac{z\Delta\mathbf{r}_f}{D} \right)$$

is the theoretical velocity. Figure 2 presents this error for $\tau_c = 1 \pm \epsilon_1$. We observe that the Filippova model and our approach have the same error that is close to the computational numerical error. However, for $1-1\times10^{-3} \leqslant \tau_c \leqslant 1+1\times10^{-3}$ the Filippova model blows up and is consequently unreliable. On the other hand, we note that the Lin model, which does not rescale the fields, has a larger error.

Grid refinement techniques require an increase of the computer memory. It can be estimated easily. If the area which is refined is of size $N\times M$, then $(z+1)\times n_{\mathrm{ref}}\times N \times M$ new fields must be defined. In our current implementation, the fine and coarse grids are defined as two logically distinct data structures. A local and adaptative refinement is also possible but it requires to define hierarchical data struc-

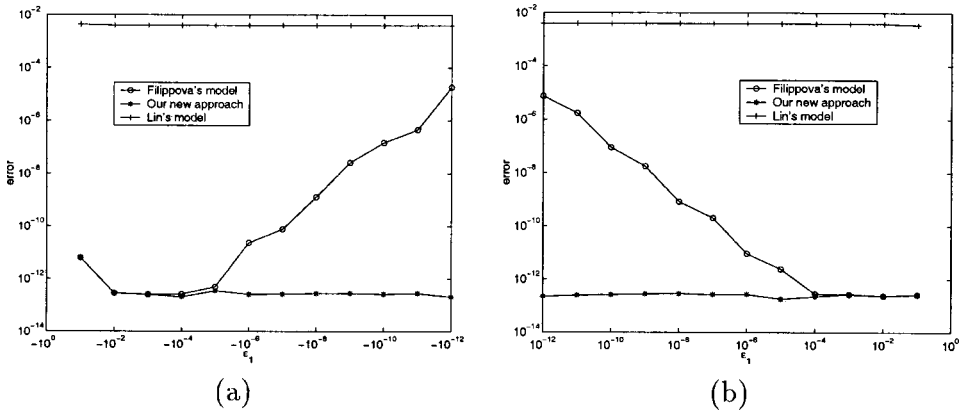

FIG. 2. Error defined by Eq. (13) on the fine lattice considering $\tau_c = 1 \pm \epsilon_1$ for the three presented algorithms. The error is presented for (a) negative and (b) positive values of ϵ_1.

tures to embed the finer grid in the coarser one. An even more difficult programming problem arises when parallelization is considered on a dynamical grid. These questions are not further discussed here as this paper focuses on conceptual issues.

IV. FLOW SETTLEMENT ACCELERATION BY USING A MULTIGRID APPROACH

A. Description

We now describe a useful utilization of the grid refinement techniques: the acceleration of the flow settlement. Indeed, a non-negligible part of the computation time is devoted to settle the flow. Thus, one is interested in reducing this time by accelerating the flow settlement process.

Note that the finer the lattice, the longer the time to settle the flow (this time goes as R_e^2 [4]). So considering a lattice L_0 with a spacing $\Delta \mathbf{r}_0 = \Delta \mathbf{r}$, we propose to settle the flow on a lattice L_1 twice as coarse as L_0, i.e., with a lattice spacing $\Delta \mathbf{r}_1 = 2\Delta \mathbf{r}$, and use the flow in L_1 as an initial condition for L_0. To settle the flow on L_1, one can again consider a coarser lattice L_2, and so on. Hence, we have a hierarchical process iterated ℓ times which turns out to accelerate the flow settlement process.

Let us present the way to connect two successive lattices. We learned from the experiments presented above that only the nonequilibrium part has to be rescaled. However, considering that the connection between lattices L_i and L_{i+1} is made once, it is unnecessary to proceed to this rescaling. Indeed, the equilibrium part is a sufficient approximation to start the computation on next level. Hence, fine fields common to both lattices are set with coarse fields without modification and the others are spatially interpolated.

A stopping criterion is needed to terminate automatically the process on lattice L_i and to start the process on lattice L_{i-1}. There are various ways to interrupt the process when convergence is reached. For instance, a simple one consists in stopping the flow if the difference between two given successive values in time is smaller than a given value ϵ_{ref}. The quantity ϵ_{ref} requires a special attention as its value may dramatically change the speedup (see below). On the other hand, if a specific characteristic of the flow is known before the simulation, one can use it to impose the stopping criterion.

Our way to accelerate the flow settlement process is summarized in algorithm 4.1.

Algorithm 4.1. Algorithm for the acceleration of the flow settlement process.

(1) Allocate memory space for lattice L_ℓ (the coarser one).
(2) Initialize L_ℓ, e.g., with the equilibrium distribution functions.
(3) Repeat collision-propagation steps until the stopping criterion is reached.
(4) For $i = \ell - 1$ down to 0, (a) allocate memory space for lattice L_i, (b) initialize the density distributions using data from lattice L_{i+1}, (c) spatially interpolate the missing density distributions, (d) deallocate memory space for lattice L_{i+1}, (e) repeat collision-propagation steps on lattice L_i until the stopping criterion is reached.
(5) Continue the simulation with an established flow on lattice L_0.

B. Application

We apply this hierarchical process to settle a LBGK $D2Q9$ Poiseuille flow on a longitudinally periodic $10 \times N_z$ lattice. The flow is settled by imposing a constant body force and the Inamuro nonslip boundary condition. We set the viscosity to $\nu = 0.005$ and the maximum velocity to $U_c = 0.1$.

One can determine the benefit of considering such a hierarchical algorithm by computing a speedup. The speedup of the flow settlement process is defined as

$$S_\ell = \frac{T_0}{T_\ell},$$

where T_i is the computation time necessary to reach a stationary state when starting the computation at refinement level i. Then considering a refinement at level ℓ, the quantity S_ℓ indicates how many times the flow settlement process goes faster than with the finer grid L_0.

The stopping criterion we consider for this application consists in computing the difference between the theoretical and simulated velocity profiles and stopping the process when this difference is smaller than ϵ_{ref}. Effects of using other criteria are investigated in Ref. [4].

Figure 3 reports the evolution of the velocity in the middle of a channel of height $N_z = 257$ when zero and seven

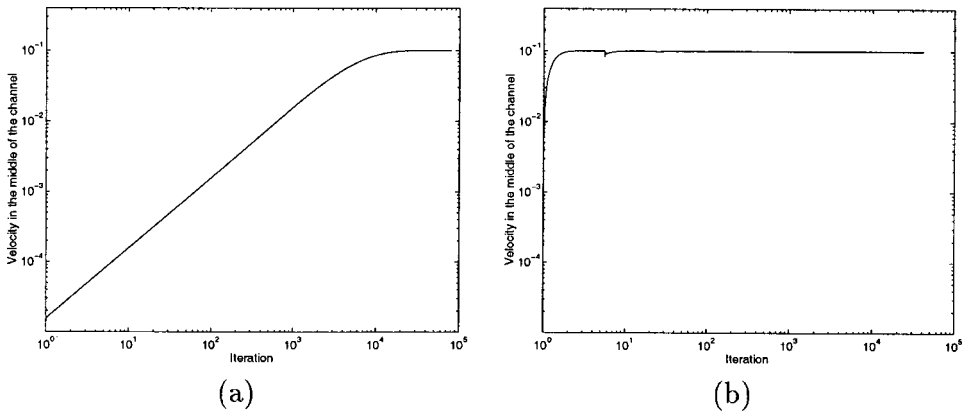

FIG. 3. Evolution of the velocity in the middle of a LBGK $D2Q9$ 10×257 channel. The stopping criterion allows the simulation to reach the steady state ($\epsilon_{\mathrm{ref}}=10^{-16}$). (a) Zero and (b) seven refinement levels have been considered. Iterations are measured in the common unit Δt.

refinement levels are considered. The quantity ϵ_{ref} is first set to 10^{-16}. We notice the exponential form of the curves and the cusps indicating that a new finer lattice is considered and initialized by the previous one, without rescaling the nonequilibrium parts.

Figure 3(a) indicates that, after $\approx 3\times10^4$ iterations, the middle velocity has almost reached its steady value. However, 5×10^4 iterations are needed to perform the simulation. Figure 3(b) shows that the first 3×10^4 iterations needed in (a) are avoided by the acceleration process. However, one still needs to perform the second part to satisfy the rather demanding stopping criterion we are considering. In this case, the speedup is only around 2, as only the first part, representing half of the number of iterations, can be avoided.

However, in many applications, one can be satisfied by a flow that has not completely reached its steady state. For instance, one may typically consider to deal with a flow that has reached 99% of its steady state. Speedups are expected to be much higher when one considers a higher value of ϵ_{ref}.

Figure 4 reports speedups for which two lattice heights and two values of ϵ_{ref} have been considered. We observe a significant speedup enhancement from 200 up to 2000. In this two-dimensional (2D) application, only one dimension is refined, as explained above. The curves behave as $2^{l^2}_{\mathrm{ref}}$ for low refinement levels (the power of 2 is because not only the number of iterations are doubled but also the computational time needed to perform an iteration). This relation does not stand for larger ℓ as the overhead due to lattices $L_{1\ldots\ell-1}$ becomes higher. However, we note that the curves saturate only at the end. This implies that it is better to consider the largest possible number of refinement levels.

We also remark that the lower the quantity ϵ_{ref} the higher the speedups. This is due to the fact that a low value of ϵ_{ref} implies a complete or a partial avoidance of the last iterations needed for machine precision. Consequently, the proportion of iterations which can be suppressed becomes larger as we accept less accuracy.

Figure 4 also highlights size effects by showing a speedup four times bigger when the height is doubled (factor of 4 between dashed and solid curves). This factor of 4 in the CPU is due to the fact that there are twice the number of sites to update and, also, because the number of iterations needed to converge doubles.

The above results indicated that even larger speedups can be expected in 3D applications.

V. CONCLUSION

In this contribution, we began by reviewing the existing lattice Boltzmann grid refinement algorithms. We highlighted their weaknesses and proposed an alternative one, more general and yet simpler. Through experiments and theoretical arguments, we concluded that our algorithm always produces correct results.

We also applied our multigrid techniques in order to speed up the flow settlement process, which is often a time consuming process. By choosing an appropriate stopping criterion, we could gain a factor between 200 and 2000. We argued that this speedup should be higher when considering 3D lattices.

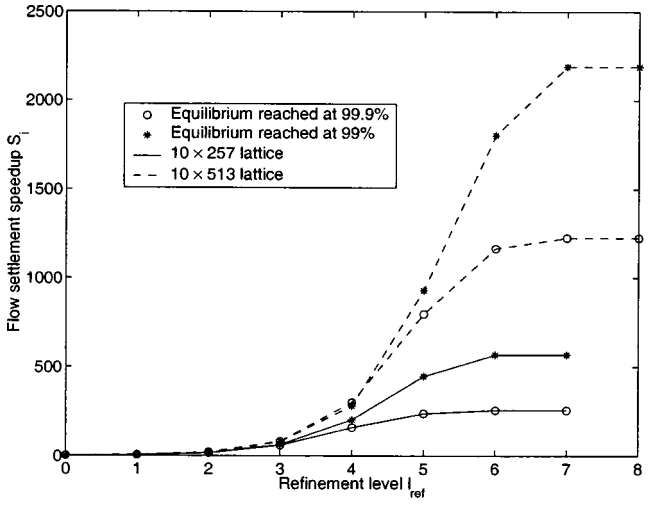

FIG. 4. Speedups of the flow settlement process of a LBGK $D2Q9$ Poiseuille flow on a $10\times N_z$ lattice. $N_z=257$ (solid) and $N_z=513$ (dashed) have been considered. The stopping criterion is satisfied when the difference between theoretical and simulated profiles is smaller than $\epsilon_{\mathrm{ref}}=10^{-2}$ (○) and $\epsilon_{\mathrm{ref}}=10^{-3}$ (+).

[1] S. Succi, *The Lattice Boltzmann Equation, For Fluid Dynamics and Beyond* (Oxford University Press, Oxford, 2001).
[2] B. Chopard and M. Droz, *Cellular Automata Modeling of Physical Systems* (Cambridge University Press, Cambridge, 1998).
[3] *Proceeding of the Seventh International Conference on the Discrete Simulation of Fluid Dynamics*, edited by B. Boghosian *et al.*, special issue of Int. J. Mod. Phys. C **9** (1998).
[4] A. Dupuis, Ph.D. thesis, University of Geneva, 2002 (unpublished), see http://cui.unige.ch/spc/PhDs/aDupuisPhD/
[5] M. Bernaschi and S. Succi, Int. J. Mod. Phys. B **17**, 1 (2003).
[6] B. Chopard, A. Dupuis, A. Masselot, and P. Luthi, Adv. Complex Syst. **5**, 1 (2002).
[7] S. Chen and G. D. Doolen, Annu. Rev. Fluid Mech. **30**, 329 (1998).
[8] Y. H. Qian, S. Succi, and S. A. Orszag, in *Recent Advances in Lattice Boltzmann Computing*, edited by Dietrich Stauffer, Annual Reviews of Computational Physics Vol. III (World Scientific, Singapore 1996), pp. 195–242.
[9] A. Dupuis and B. Chopard, J. Comput. Phys. **178**, 161 (2002).
[10] O. Filippova and D. Hänel, J. Comput. Phys. **147**, 219 (1998).
[11] C. L. Lin and Y. Lai, Phys. Rev. E **62**, 2219 (2000).
[12] T. Inamuro, M. Yoshino, and F. Ogino, Phys. Fluids **7**, 2928 (1995).