Advances in multi-domain lattice Boltzmann grid refinement

D. Lagrava[a,*], 0. Malaspinas[b,a], J. Latt[a], B. Chopard[a]

[a] *Centre Universitaire d'Informatique, Université de Genève 7, route de Drize, CH-1227 Switzerland*
[b] *Institut Jean le Rond d'Alembert, UMR 7190, Université Pierre et Marie Curie - Paris 6, 4 place Jussieu - case 162, F-75252, France*

Abstract

Grid refinement has been addressed by different authors in the lattice Boltzmann method community. The information communication and reconstruction on grid transitions is of crucial importance from the accuracy and numerical stability point of view. While a decimation is performed when going from the fine to the coarse grid, a reconstruction must performed when going form the coarse to the fine grid. In this paper we analyze these two steps. We first show that for the decimation operation, a simple copy of the information from the fine to the coarse grid is not sufficient to guarantee the stability of the numerical scheme at high Reynolds numbers, but that a filtering operation must be added. Then we demonstrate that to reconstruct the information, a local cubic interpolation scheme is mandatory in order to get a precision compatible with the order of accuracy of the lattice Boltzmann method.

These two fundamental extra-steps are validated on two classical 2D benchmarks, the 2D circular cylinder and the 2D dipole-wall collision. The latter is especially challenging from the numerical point of view since we allow strong gradients to cross the refinement interfaces at a relatively high Reynolds number of 5000. A very good agreement is found between the single grid and the refined grid cases.

The proposed grid refinement strategy has been implemented in the parallel open-source library Palabos.

Keywords: Lattice Boltzmann method, grid refinement, Palabos

1. Introduction

The lattice Boltzmann method (LBM) is a clever discretization of the Boltzmann equation which has become a widely used numerical tool for the computational fluid dynamics.

*Corresponding author

In order to simulate many of the flows of engineering interest, the computational power needed exceeds by far the available resources (even on the most modern hardware) because of the wide range of scales involved in fluid dynamics. Nevertheless the smallest scales are often localized in a relatively limited computational area and therefore a great amount of computational power can be spared by refining locally the mesh.

In methods like the LBM that use conformal meshes, the refinement operation induces a strong discontinuity in the physical quantities at the grid transition and can therefore be a very complicated process.

Unlike classical computational fluid dynamics solver, there exists a relatively limited amount of papers treating grid refinement in the LBM community. One can distinguish two major approaches, one of which is based on a volumetric representation (cell-centered approach) of the flow variables [19, 10, 15, 4] while the other uses of point-wise interpretation (cell-vertex approach) [23, 9, 24, 8, 5].

Independently of the approach used, the grid transition treatment is of crucial importance as far as accuracy and stability are concerned. The two-way coupling between a coarse and a fine grid involves two fundamental operations that are the decimation and the reconstruction steps. When going from the fine to the coarse grid the amount of information represented in the fine grid must be reduced, while it must be "increased" when going from the coarse to the fine grid. In this work we use the cell-vertex approach and show that for the decimation process we cannot simply copy the information from the fine to the coarse grid, but must add a filtering step in order to increase the numerical stability of the scheme. Secondly, in the reconstruction operation we show that a local cubic interpolation is mandatory to preserve the order of accuracy of the LBM.

We propose two 2D benchmarks to show the consistency of our method and show that one can use local refinement without loss of accuracy with respect to a single grid approach. This implementation is also publicly available through the open-source LBM library Palabos [16].

The paper is organized as follows. We first give a very brief introduction to the LBM in Sec. 2. Then, in Sec. 3 the basic concepts of grid refinement are first revisited, for reasons of self-consistency and then a novel coupling algorithm between grid is proposed. Our approach is then validated in Sec. 4 and finally a conclusion is given in Sec. 5.

2. Lattice Boltzmann Method

The lattice Boltzmann method (LBM) is now a widely used method for computational fluid dynamics and only the basic concepts are given in this section. For further details the reader is referred to [6, 22, 21, 1].

The LBM scheme for the BGK model (for Bhatnagar, Gross, Krook [2]) is given by

$$f_i(\boldsymbol{x} + \boldsymbol{\xi}_i \delta t, t + \delta t) = f_i(\boldsymbol{x}, t) - \omega \left(f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t) \right), \tag{1}$$

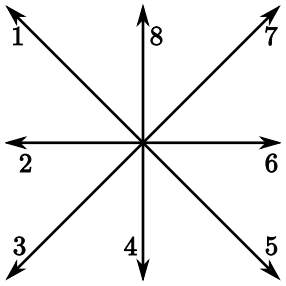

Figure 1: The D2Q9 lattice with the vectors representing the microscopic velocity set $\boldsymbol{\xi}_i$. A rest velocity $\boldsymbol{\xi}_0 = (0, 0)$ is added to this set.

where f_i is the is the discrete probability distribution to find a particle with velocity $\boldsymbol{\xi}_i$ at position $\boldsymbol{x}$ at time t, f_i^{eq} is the discrete Maxwell–Boltzmann equilibrium distribution function, ω the relaxation frequency, and δt the discrete time step. In this paper we are interested in the two dimensional weakly compressible case only, therefore the discrete velocity set is given by the D2Q9 lattice (see Fig. 1). This lattice is defined by

$$\begin{aligned} c_s^2 = 1/3,\ w_0 = 4/9,\ w_j = 1/9,\ j = \{2,4,6,8\},\ w_k = 1/36,\ k = \{1,3,5,7\} \\ \{\boldsymbol{\xi}_i\}_{i=0}^{8} = \{(0,0),(-1,1),(-1,0),(-1,-1),(0,-1), \\ (1,-1),(1,0),(1,1),(0,1)\}. \end{aligned} \tag{2}$$

The equilibrium distribution is given by

$$f_i^{eq} = w_i \rho \left(1 + \frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{c_s^2} + \frac{1}{2c_s^4} \boldsymbol{Q}_i : \boldsymbol{u}\boldsymbol{u} \right), \tag{3}$$

where ρ is the density, $\boldsymbol{u}$ is the macroscopic velocity field, $\boldsymbol{Q}_i = \boldsymbol{\xi}_i\boldsymbol{\xi}_i - c_s^2\boldsymbol{I}$, c_s and w_i the lattice speed of sound and the lattice weights respectively. The density and the velocity fields are computed by the distribution function through the relations

$$\rho = \sum_{i=0}^{q-1} f_i = \sum_{i=0}^{q-1} f_i^{eq}, \tag{4}$$

$$\rho \boldsymbol{u} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{eq}, \tag{5}$$

where q is the number of discrete velocities. Doing a multi-scale Chapman–Enskog (CE) expansion (see [3, 6] for more details) one can show that the LBM BGK scheme is asymptotically equivalent to the weakly compressible Navier–

Stokes equations

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

$$\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} = -\frac{1}{\rho}\boldsymbol{\nabla} p + \nu \boldsymbol{\nabla} \cdot (2\boldsymbol{S}), \tag{7}$$

with p being the pressure, $\boldsymbol{S}$ the strain tensor and ν the kinematic viscosity defined by

$$p = c_s^2 \rho \tag{8}$$

$$\boldsymbol{S} = \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^{\mathrm{T}}\right), \tag{9}$$

$$\nu = c_s^2(1/\omega - 1/2). \tag{10}$$

The CE expansion is done under the assumption that f_i is given by a small perturbation of the equilibrium distribution

$$f_i = f_i^{eq} + \varepsilon f_i^{(1)} + \mathcal{O}(\varepsilon^2), \tag{11}$$

where $\varepsilon \ll 1$ can be identified with the Knudsen number (see [12]).
One can also show that $f^{(1)}$ is given by

$$\varepsilon f_i^{(1)} = \frac{w_i}{2c_s^4}\boldsymbol{Q}_i : \boldsymbol{\Pi}^{(1)}, \tag{12}$$

where the tensor $\boldsymbol{\Pi}^{(1)} \equiv \sum_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i \varepsilon f_i^{(1)}$ is related to the strain rate tensor $\boldsymbol{S}$ through the relation

$$\boldsymbol{\Pi}^{(1)} = -\frac{2c_s^2\rho}{\omega}\boldsymbol{S}. \tag{13}$$

Therefore the f_i can be approximated by

$$f_i = w_i \rho \left(1 + \frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{c_s^2} + \frac{1}{2c_s^4}\boldsymbol{Q}_i : \boldsymbol{u}\boldsymbol{u}\right) - \frac{w_i \rho}{c_s^2 \omega}\boldsymbol{Q}_i : \boldsymbol{S}. \tag{14}$$

For implementation purposes all the above quantities are expressed in "lattice units", $\delta x = \delta t = 1$, and a time-step is decomposed into two parts that are applied successively on the whole computational domain. The two-steps are called the "collide-and-stream" operation.

1. The collision, which modifies *locally* the value of the populations according to
$$f_i^{\mathrm{out}}(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) - \omega\left(f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)\right). \tag{15}$$
2. The streaming, which moves the populations to their neighbors according to their microscopic velocity
$$f_i(\boldsymbol{x} + \boldsymbol{\xi}_i, t + 1) = f_i^{\mathrm{out}}(\boldsymbol{x}, t). \tag{16}$$

3. The grid refinement algorithm

In this section, we first present the basic concepts of grid refinement in the LBM framework in Subsecs. 3.1 and 3.2. Then we propose to analyze more deeply the coupling between the grids in Subsecs. 3.3 and 3.4. Finally we summarize the algorithm in Subsec. 3.5.

3.1. Basic concepts of grid refinement

In this section we go through the basic concepts of grid refinement. The discussion hereafter follows closely the one presented in reference [20], while Subsec. 3.2 is inspired by [8, 5]. To keep the discussion simple the approach is demonstrated for a two dimensional case, but can be straightforwardly generalized in three dimensions.

There exists two grid refinement techniques, the multi-grid and multi-domain. In the first case, the coarse grid is present all over the simulation domain, even in the places where there exist refined patches (see Fig 2). On the other hand, when defining a multi-domain refinement, the regions where refined patches are inserted are taken off the coarse grid (see Fig. 3).

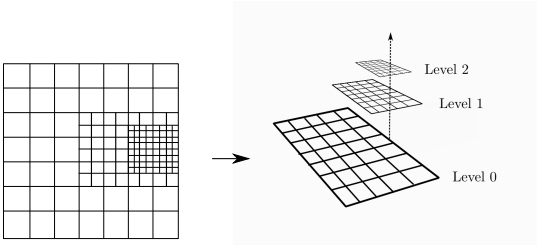

Figure 2: A schematic view the multi-grid approach.

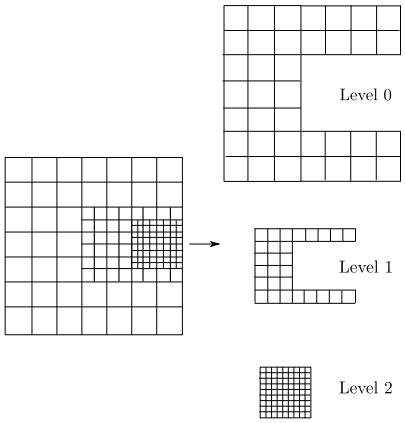

Figure 3: A sketch of the multi-domain approach.

Each of these techniques has its advantages and disadvantages. In our case we chose the multi–domain approach in order to have better CPU performances and higher memory savings. However the actual implementation and the coupling between grids for this approach is more complex.

When using multi-resolution approaches, a communication between the grids is needed. In the case of multi-domain methods the communication is done on the boundaries connecting the grids. The coupling is made in two directions: from coarse to fine and from fine to coarse grids.

On the boundaries of each refinement level, after a "collide-and-stream" operation (see Sec. 2) there will be some missing information (some populations f_i are unknown on the coarse and on the fine grids) that one needs to reconstruct. The algorithm applied is discussed in Subsec. 3.3 and 3.4.

For the sake of clarity, let us call C the ensemble of coarse sites and F the ensemble of all fine sites. Let us now define $\boldsymbol{x}_{f\to c}$ the fine sites that are contained in F and C where the coupling from fine to coarse is performed and $\boldsymbol{x}_{c\to f}$ all the sites contained in F and C where the coupling goes from coarse to fine (see Fig. 4). Let us also define $\boldsymbol{x}^c_{f\to c} = \{\boldsymbol{x} | \boldsymbol{x} \in \boldsymbol{x}_{f\to c} \text{ and } \boldsymbol{x} \notin F\}$, $\boldsymbol{x}^c_{c\to f} = \{\boldsymbol{x} | \boldsymbol{x} \in \boldsymbol{x}_{c\to f} \text{ and } \boldsymbol{x} \notin F\}$ and $\boldsymbol{x}^f_{c\to f} = \{\boldsymbol{x} | \boldsymbol{x} \in \boldsymbol{x}_{c\to f} \text{ and } \boldsymbol{x} \notin \boldsymbol{x}^c_{c\to f}\}$.

The coupling proposed in this work requires the grids to overlap themselves by a domain of at least one coarse cell width, as shown in the one-dimensional example in Fig. 5. Let us now analyze what is happening on Fig. 5. After a

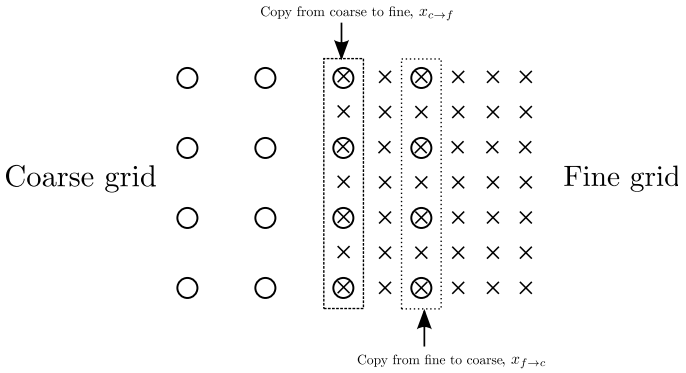

Figure 4: Complete sketch of the places where the copies are performed.

coarse grid collide-and-stream operation, all the sites have the necessary information, except for the last site of the grid (labeled "unknown value on Fig. 5), where the are missing populations. It is therefore impossible to perform the coupling on this coarse site to the fine grid site. However, all the other sites have all the needed information, thus becoming good candidates to provide the information to the fine grid. If we apply the same reasoning to the fine grid, it becomes clear that it is necessary to implement a redundant area between the grids, so that the copies are performed over complete sites. The complete two dimensional example is found in Fig. 4

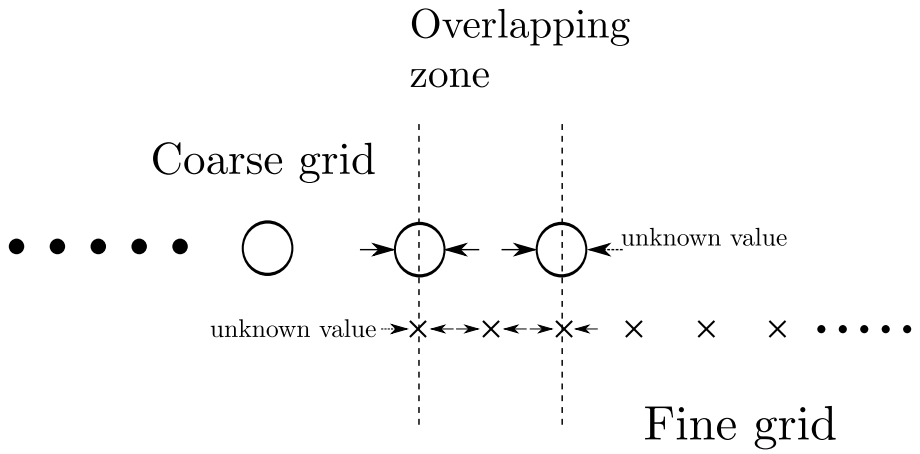

Figure 5: Overlapping zone between coarse and fine grids

We have chosen to make this overlapping area as small as possible, namely one coarse site, or equivalently, two fine sites. However, when dealing with turbulent flows, it might be desirable to have a more important area between the grids, in order to allow a gentler transition of the information.

3.2. Rescaling of physical quantities

In the LBM a regular conformal grid is used. Therefore an abrupt transition occurs when refining the computational domain. In our of implementation each resolution level possesses its own "lattice units". This change of scales induces a need for a rescaling of the physical quantities between the grids. In the following, we will work in lattice units, the c subscript stands for coarse grid units, while f for fine grid units.

In our case we chose to refine the grids by a factor two only. Thus defining δx_c and δx_f the spatial discretization of the coarse and fine grids respectively one has the following relation between them

$$\delta x_f = \delta x_c/2. \tag{17}$$

This will remain true for the rest of this document.

Once the spatial refinement has been chosen, one still has the freedom to chose the temporal refinement. In the LBM two popular scalings exist: the convective and the diffusive scaling. While in the diffusive case the temporal scale is proportional to the square of the spatial scale ($\delta t \sim \delta x^2$), in the convective case the temporal scale is proportional to the spatial scale ($\delta t \sim \delta x$). Each one of these scalings has its advantages and inconveniences. While the diffusive scaling removes the compressibility error terms that may appear when simulating incompressible fluid flows, the convective scaling has a much greater numerical efficiency. In the present work we chose to use the latter. The convective scaling

has as a consequence that the ratio of the spatial and temporal discretizations is a constant, and therefore

$$\delta t_f/\delta x_f = \delta t_c/\delta x_c = \text{const.} \tag{18}$$

Therefore the temporal loop in the fine grid must do twice more iterations than the coarse one. Another consequence of the convective scaling, is that the velocity and the pressure (and also the density, see Eq. (8) with c_s being a constant of the lattice) in lattice units are continuous fields on the grid transition while the viscosity must be rescaled as discussed now. Let the Reynolds number related to the coarse or fine grid as

$$\text{Re}_n = U_n L_n/\nu_n, \tag{19}$$

where n stands for c or f, U_n, L_n and ν_n being respectively the characteristic velocity, the characteristic length-scale and the viscosity of the n grid. Writing the U_c, L_c, U_f and L_f in terms of the physical characteristic velocity and length-scale, U and L, one has

$$U_n = U\delta t_n/\delta x_n, \quad L_n = L/\delta x_n, \tag{20}$$

and imposing that the Reynolds number is independent of the grid one gets

$$\text{Re}_c = \text{Re}_f \Leftrightarrow \frac{UL\delta t_c}{\delta x_c^2 \nu_c} = \frac{UL\delta t_f}{\delta x_f^2 \nu_f}. \tag{21}$$

Finally remembering Eq. (18) one finds for the rescaling of the viscosity

$$\nu_f = \frac{\delta x_c}{\delta x_f}\nu_c. \tag{22}$$

As a consequence, by using the relation between the relaxation frequency and the viscosity of Eq. (9), one gets that ω_f is given by

$$\omega_f = \frac{2\delta x_f \omega_c}{\delta x_f \omega_c + 2\delta x_c - \delta x_c \omega_c}, \tag{23}$$

$$\omega_f = \frac{2\omega_c}{4 - \omega_c}, \tag{24}$$

The rescaling of the distribution function f_i now needs to be discussed. The algorithm that is used here is the one proposed by Dupuis et *al.* [8]. It must be noted that this algorithm is the same as the algorithm of Filippova et *al.* [9], with the exception that applying the rescaling before collision avoids the restriction over the value of the relaxation frequency.

The basic ideas of the algorithm are explained in the following. As noted in Eq. (14) each $f_{i,n}$ can be written as

$$\begin{aligned} f_{i,n} &= f_i^{eq}(\rho_n, \boldsymbol{u}_n) + f_{i,n}^{neq}(\boldsymbol{\nabla u}), \\ f_{i,n} &= f_i^{eq}(\rho, \boldsymbol{u}) + f_{i,n}^{neq}(\boldsymbol{\nabla u}), \end{aligned} \tag{25}$$

where in the second line we used that $\rho_f = \rho_c = \rho$ and $\boldsymbol{u}_f = \boldsymbol{u}_c = \boldsymbol{u}$ are expressed in the same units independently if computed from $f_{i,f}$ or $f_{i,c}$. Since f_i^{eq} only depends on ρ and $\boldsymbol{u}$ and, as discussed above, both are continuous between the grids f_i^{eq} does not need any rescaling.

On the other hand, the non-equilibrium part $f_i^{neq} = f_i - f_i^{eq}$ is proportional to the gradient of the velocity, it is therefore necessary to rescale it when communicating it between grids with different resolution. To determine this scaling, let us note by $f_{i,c}^{neq}$ the non-equilibrium part of the coarse grid and $f_{i,f}^{neq}$ the one from the fine grid. The continuity of the f_i^{neq} quantities read

$$f_{i,f}^{neq} = \alpha f_{i,c}^{neq}, \tag{26}$$

where α is the factor to be determined to impose the continuity of the non-equilibrium distribution functions. At the leading order one has $f_i^{neq} \cong \epsilon f_i^{(1)}$ and using Eq. (12) and (13) in the previous equation one has

$$\begin{aligned} \frac{1}{\omega_f}\boldsymbol{Q}_i : \boldsymbol{S}_f &= \alpha \frac{1}{\omega_c}\boldsymbol{Q}_i : \boldsymbol{S}_c \\ \frac{1}{\delta t_f \omega_f}\boldsymbol{Q}_i : \boldsymbol{S} &= \alpha \frac{1}{\delta t_c \omega_c}\boldsymbol{Q}_i : \boldsymbol{S} \\ \alpha &= \frac{\delta t_c}{\delta t_f}\frac{\omega_c}{\omega_f}, \end{aligned} \tag{27}$$

where $\boldsymbol{S}$ is the strain rate tensor in physical units, while $\boldsymbol{S}_c$ and $\boldsymbol{S}_f$ are the same tensor, but in coarse and fine lattice units respectively. Finally we find the following relation

$$f_{i,c}^{neq} = \frac{\delta t_f \omega_f}{\delta t_c \omega_c} f_{i,f}^{neq} = \frac{2\omega_f}{\omega_c} f_{i,f}^{neq}, \tag{28}$$

where we used that $\delta t_f = \delta t_c/2$.

Finally, in order to reconstruct the fine and coarse populations from their corresponding partners, we can use the following equations

$$f_{i,f}(\boldsymbol{x}_{c\to f}) = f_i^{eq}(\rho(\boldsymbol{x}_{c\to f}), \boldsymbol{u}(\boldsymbol{x}_{c\to f})) + \frac{\omega_c}{2\omega_f} f_{i,c}^{neq}(\boldsymbol{x}_{c\to f}) \tag{29}$$

and

$$f_{i,c}(\boldsymbol{x}_{f\to c}) = f_i^{eq}(\rho(\boldsymbol{x}_{f\to c}), \boldsymbol{u}(\boldsymbol{x}_{f\to c})) + \frac{2\omega_f}{\omega_c} f_{i,f}^{neq}(\boldsymbol{x}_{f\to c}). \tag{30}$$

This rescaling between the grids allows for a continuous transition of the physical quantities at the grids interface. We are now going to discuss in more details the actual coupling procedure between the coarse and fine grids.

3.3. Coupling from the fine to the coarse grid

The easiest part of the coupling is to go from the fine to the coarse grid. As the fine grid has more sites than the coarse one (see Fig. 4). The necessary

steps are: restrict the values, rescale them and copy them to the coarse grid. The restriction operation can be a simple copy from the corresponding site, or something more complicated, as a low-pass filter. This latter can be justified by the fact that the fine grid contains information about scales that cannot be resolved by the coarse grid and thus must be eliminated.

The proposed coupling is over the sites marked as $\boldsymbol{x}_{f\to c}$ is expressed by the following equation

$$f_{i,c}(\boldsymbol{x}^c_{f\to c},t) = f_i^{eq}\left(\rho_f(\boldsymbol{x}^c_{f\to c},t),\boldsymbol{u}_f(\boldsymbol{x}^c_{f\to c},t)\right) + \frac{2\omega_f}{\omega_c}\overline{f}^{neq}_{i,f}(\boldsymbol{x}^c_{f\to c},t). \tag{31}$$

where $\rho_f = \sum_i f_{i,f}$ and $\boldsymbol{u}_f = \sum_i \boldsymbol{\xi}_i f_{i,f}$, and $\overline{f}^{neq}_{i,f}(\boldsymbol{x}^c_{f\to c},t)$ is the result of applying the restriction to the incoming fine grid values. It must be noted that ρ and $\boldsymbol{u}$ are calculated over the fine grid and by continuity of this fields due to the convective rescaling, no other operation over them needs to be performed.

If we perform a simple decimation, then it is clear that the restriction operation is given by

$$\overline{f}^{neq}_{i,f}(\boldsymbol{x}^c_{f\to c},t) = f^{neq}_{i,f}(\boldsymbol{x}^c_{f\to c},t) \tag{32}$$

However, as we will show later, this is not enough. We propose a filtering operation that is inspired by [17]. The aim of this operation is to remove the scales of the fine grid that are not resolved by the coarse grid. We use a filter width corresponding to the coarse grid resolution. A simple box filter is used here but more complex filters can also be applied.

There exists many different ways of applying a filter in the lattice Boltzmann framework. One can filter the complete distribution function or only ρ and $\boldsymbol{u}$ (see [18]). In our case, we chose to apply the filter only on the non-equilibrium part of the populations f_i^{neq}, because when the filtering was done on f_i, or only on ρ and $\boldsymbol{u}$ there was a too strong dissipation added by the filter and therefore we noticed a decrease of the accuracy. In practice one simply does an averaging over all the q lattice directions, thus obtaining the following restriction

$$\overline{f}^{neq}_{i,f}(\boldsymbol{x}^c_{f\to c},t) = \frac{1}{q}\sum_{i=0}^{q-1} f^{neq}_{i,f}(\boldsymbol{x}^c_{f\to c}+\boldsymbol{\xi}_i,t) \tag{33}$$

As a final remark, we have chosen not to filter ρ and $\boldsymbol{u}$. The main reason for this being that the filtering of this quantities results in an artificial increase of viscosity around the refinement interface. This can, of course, have undesired results such as loss of accuracy and modification of the expected behavior of the system.

3.4. Coupling from the coarse to the fine grid

As shown in Fig. 6, the fine grid possesses sites that do not have a corresponding site in the coarse one. Thus when performing the copy from the coarse to the fine grid, it is necessary to estimate the missing information in

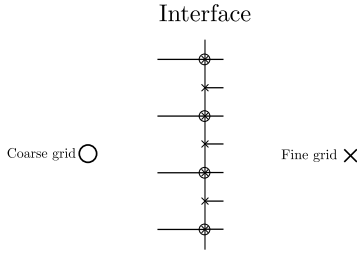

Figure 6: Unknown sites in the fine grid.

such sites. We have chosen to apply an interpolation in order to complete the missing informations. The details of the interpolation are explained further in this document, as we have found that it is a vital part of the algorithm.

The coupling over the sites $\boldsymbol{x}_{c\to f}$ is given by two different operations. If a point has a corresponding coarse site in $\boldsymbol{x}_{c\to f}$ (i.e. if a computational node has both a coarse and a fine site, or in a mathematical notation if $\boldsymbol{x}_f \in \boldsymbol{x}^c_{c\to f}$) then

$$f_{i,f}(\boldsymbol{x}^c_{c\to f}) = f^{eq}_i(\rho_c(\boldsymbol{x}^c_{c\to f}), \boldsymbol{u}_c(\boldsymbol{x}^c_{c\to f})) + \frac{\omega_c}{2\omega_f} f^{neq}_{i,c}(\boldsymbol{x}^c_{c\to f}) \tag{34}$$

where $\rho_c = \sum_i f_{i,c}$, $\boldsymbol{u}_c = \sum_i \boldsymbol{\xi}_i f_{i,c}$, and $f^{neq}_{i,c}$ are computed from the populations of the coarse grid.

However, if the fine site does not correspond to a coarse site in $\boldsymbol{x}_{c\to f}$ (i.e. the computational node contains only a site of the fine domain), we will use the following formula

$$f_{i,f}(\boldsymbol{x}^f_{c\to f}) = f^{eq}_i(\overline{\rho}_c, \overline{\boldsymbol{u}}_c) + \frac{\omega_c}{2\omega_f} \overline{f^{neq}_{i,c}} \tag{35}$$

where $\overline{\rho}_c$, $\overline{\boldsymbol{u}}_c$ and $\overline{f^{neq}_{i,c}}$ are interpolated from the values where the fine and coarse sites are coincident.

The fine grid can resolve more scales than the coarse one, it might be necessary to try recreate these smaller scales when transferring the information from the coarse to the fine grid. In order to solve this issue, we implemented the approximate deconvolution approach proposed in [17]. However in the benchmarks we performed we did not notice a significant gain when using this method and therefore for the sake of clarity we will not present it here. Nevertheless we think that in three dimensional cases and/or with a higher Reynolds number (where very small structures should be present) such an operation might be of the uttermost importance.

3.5. Grid coupling algorithm

We present a detailed version of the coupling algorithm that we implemented. All the operations are explained in detail further in this document.

Let us suppose that the system is at time t. Both grids are complete, i.e. all information needed on every site is given. A complete time iteration, for the

convective scheme explained above, consists of one iteration of the coarse grid and two iterations of the fine grid. The details of these iterations are given now.

1. A "collide-and-stream" operation is performed on the coarse grid bringing it to time $t+\delta t_c$. At this point the populations at $\boldsymbol{x}_{f\to c}$ that were supposed to be streamed from the fine grid are unknown.
2. A "collide-and-stream" cycle is performed on the fine grid bringing it at time $t+\delta t_c/2$. The grid lacks information on the grid refinement boundary sites $\boldsymbol{x}_{c\to f}$. One then performs a double interpolation, one in time and one in space. First the values of ρ_c, $\boldsymbol{u}_c$ and $f_{i,c}^{neq}$ of the coarse sites at $\boldsymbol{x}_{c\to f}$ and are interpolated at time $t+\delta t_c/2$, by a linear scheme (which is of second order accuracy at $\delta t_c/2$). Then the values $\rho_c(t+\delta t/2)$, $\boldsymbol{u}_c(t+\delta t/2)$ and $f_{i,c}^{neq}(t+\delta t/2)$ are interpolated in space according to the discussion of Subsec. 3.6 by using a local cubic scheme. All the populations at $\boldsymbol{x}_{c\to f}$ (and not only the missing ones) are reconstructed following Eqs. (34) and (35). At this point all the fine sites are complete.
3. A second "collide-and-stream" operation is performed on the fine grid, bringing it to time $t+\delta t_c$. At this point we have the information from the coarse grid to complete the fine grid at $\boldsymbol{x}_{c\to f}$, and therefore no time interpolation is necessary. However, a space interpolation must be performed for $\rho_c(t+\delta t_c)$, $\boldsymbol{u}_c(t+\delta t_c)$, and $f_{i,c}^{neq}(t+\delta t_c)$ as in the previous step. Then the populations of the fine grid are all (and not only the missing ones) replaced at position $\boldsymbol{x}_{c\to f}$ according to Eqs. (34) and (35) again.
4. All the populations of the coarse grid at $\boldsymbol{x}_{f\to c}$ are replaced following Eqs. (31) and (33)

$$f_{i,c}(\boldsymbol{x}_{f\to c},t) = f_i^{eq}\left(\rho_f(\boldsymbol{x}_{f\to c},t),\boldsymbol{u}_f(\boldsymbol{x}_{f\to c},t)\right) + \frac{2\omega_f}{\omega_c}\frac{1}{q}\sum_i \overline{f}_{i,f}^{neq}(\boldsymbol{x}_{f\to c}+\boldsymbol{\xi}_i,t). \tag{36}$$

   At this point both grid were brought to time $t+\delta t_c$, are complete and ready for a new iteration.

3.6. Interpolation scheme

As pointed out in the preceding subsection, the values of ρ_c, $\boldsymbol{u}_c$ and $f_{i,c}^{neq}$ on nodes that do not contain both fine and coarse grids sites must be interpolated, and are noted by $\overline{\rho}_c$, $\overline{\boldsymbol{u}}_c$ and $\overline{f^{neq}}_{i,c}$. In 2D, this interpolation is performed over a line, thus over sites parallel to the refinement interface. The problem becomes then a 1D interpolation of a function g which is known at the coarse sites. In the following, we show our interpolation schemes for g.

The two easiest symmetric ways of computing the interpolation are using two or four neighbors as shown on Fig. 7. If we use two points, it is sufficient to perform a mean of them:

$$g(x) = \frac{g(x+h)+g(x-h)}{2} \tag{37}$$

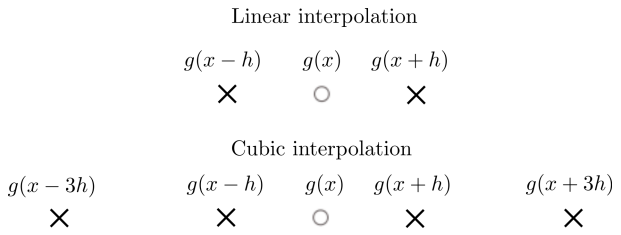

Figure 7: Possible interpolation schemes. Order one (top) and order three (bottom).

As the point where we interpolate is exactly in the middle of the two others, an analysis of the order of interpolation shows us that this is in fact a second order method in h.

On the other hand, by using four points, we find the following equation:

$$g(x) = \frac{9}{16}(g(x+h) + g(x-h)) - \frac{1}{16}(g(x+3h) + g(x-3h)) \tag{38}$$

By using Taylor expansions for each term, this interpolation can be proved to be of fourth order.

The only drawback for this interpolation is the cost in parallel execution. Every point must ensure access to two neighbors, this means that when parallelizing the code, more communications have to be done between the distributed blocks.

In addition, as the first and last site do not have as many neighbors as needed, we need to use a non-centered third order scheme as depicted is Fig. 8 and given by the following formula

$$g(x) = \frac{3}{8}g(x-h) + \frac{3}{4}g(x+h) - \frac{1}{8}g(x+3h). \tag{39}$$

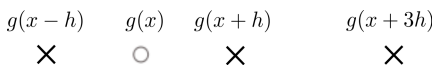

Figure 8: Asymmetric second order interpolation for sites that do not have enough neighbors.

Finally, let us note that there are no other interpolation cases in our two dimensional implementation, because the outermost fine grid points always have corresponding coarse sites. In particular, cases like the one depicted in Fig. 9 are forbidden.

3.7. Is second order enough?

Interpolation is a key part of grid refinement. In this section, we show that even for a simple Poiseuille flow, the second order interpolation does not conserve the mass.

We fix a flow at $Re = u_{\max}N/\nu = 100$, with $u_{\max} = 0.01$ the maximum value of the velocity in lattice units, N the width of the channel in coarse lattice units

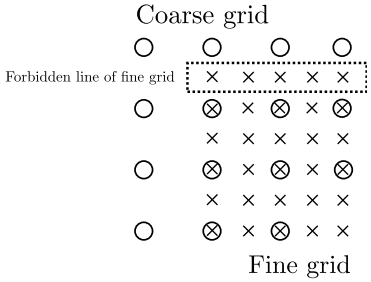

Figure 9: Forbidden case to avoid complicated interpolations.

and ν the kinematic viscosity. The length of the channel is $4N$ (see Fig. 10). We set the reference length to have $N = 30$ lattice sites. We prescribe the analytical solution of the velocity on both the inlet and the outlet and the horizontal walls have a zero velocity boundary condition. This setup is depicted in Fig. 10. The

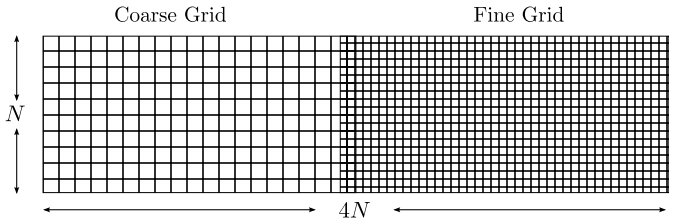

Figure 10: Setup for the channel flow

channel is divided in two equal parts. The first one is covered by a coarse grid and the second part is covered by a grid two times finer. A linear pressure drop is expected in the direction of the flow. As can be seen in Fig. 11 a clear discontinuity of the pressure appears on the refinement interface in the case of second order interpolation, while it remains completely smooth in the cubic interpolation case.

Even though the pressure slope is correct, the mass loss is clearly visible in the pressure jump right where the interface is located which leads, in more complicated cases, to numerical instabilities and to the introduction of spurious error terms. In Fig. 12 we plotted the value of this pressure jump with respect to the resolution. One can see that the magnitude of the jump is of order one in space.

This error can be explained in the following way. On the one hand, the linear interpolation is locally of order two in space (since the interpolated point lies exactly in the middle of the two known points, see Fig. 7). This error gives rise to an order one error globally as errors are summed over the entire line. On the other hand, the LBM time-space integration is third order accurate locally, and second order globally (it is obtained through a trapezoidal integration). Therefore the linear interpolation gives rise to error terms incompatible with

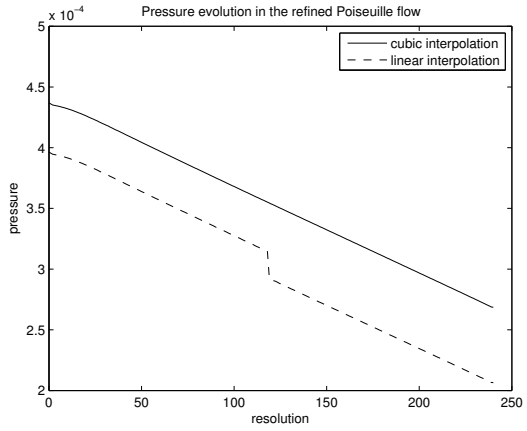

Figure 11: Pressure plot along a horizontal line for the proposed refined Poiseuille flow.

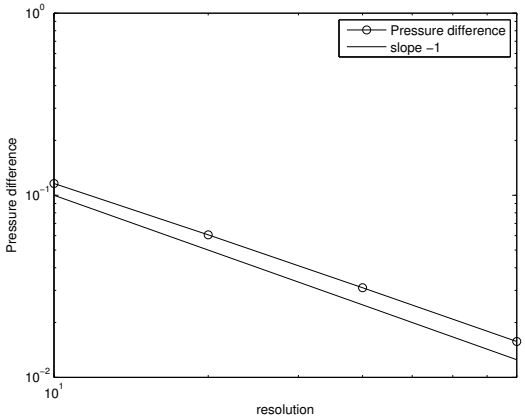

Figure 12: The pressure jump magnitude verus the resolution of the coarse grid.

the error expected from the LBM.

4. Numerical experiments and results

The grid refinement technique, presented in the previous section, has been tested on two 2D problems. First, we study the unsteady flow past a circular cylinder at Re = 100 which is a well documented problem (see [23] for example). Then, the case of the dipole–wall collision, which is a 2D turbulent problem that has been already studied and benchmarked for a LBM code in [13].

4.1. Flow past a cylinder

This flow has been extensively studied. In particular we find a set of tests in [23] where a set of values have been obtained using a wide range of methods. Here we are interested in the unsteady flow past a circular cylinder at $\mathrm{Re} = u_{\mathrm{mean}} N/\nu = 100$, with u_{mean} the mean velocity of the inlet flow, N the diameter of the cylinder and ν the kinematic viscosity, and will compare our result for the maximum drag C_d and lift C_l coefficients

$$C_d = \frac{2F_d}{\rho u_{\mathrm{mean}}^2 N} \ , \ C_l = \frac{2F_l}{\rho u_{\mathrm{mean}}^2 N}, \tag{40}$$

F_d and F_l are the drag and lift forces respectively, which are given by the equations

$$F_d = \int_S (\rho\nu \frac{\partial \boldsymbol{v_t}}{\partial \boldsymbol{n}} n_y - p n_x) dS \ , \ F_l = \int_S (\rho\nu \frac{\partial \boldsymbol{v_t}}{\partial \boldsymbol{n}} n_x + p n_y) dS, \tag{41}$$

where S is the cylinder surface, $\boldsymbol{n} = (n_x, ny)$ is the normal vector of the surface S, and $\boldsymbol{v}_t$ is the tangential velocity on S. In the LBM, this forces can be easily computed by the momentum exchange method as presented in [14].

The details of the geometry are shown in Fig. 13. In this figure all the quantities are given with respect to the diameter of the cylinder N. At the inlet we impose a Poiseuille profile and a zero velocity gradient at the outlet. Furthermore near the outlet we filtered the solution using the method proposed in [18] to increase the numerical stability of the boundary condition. The horizontal walls have zero velocity. We use a lattice velocity $u_{\mathrm{mean}} = 0.00333$ and the cylinder diameter is fixed to have N lattice sites. First, we used a uniform

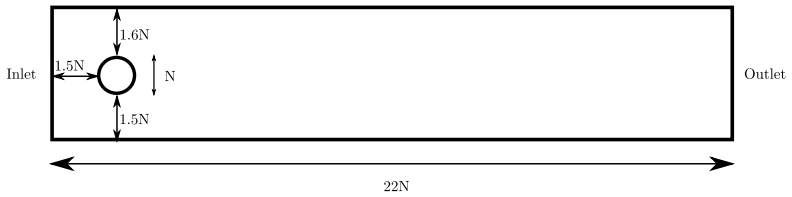

Figure 13: Geometry for the flow past a cylinder.

single-level grid to find a resolution at which we find values for the maximum of the drag and lift coefficients that are close enough to the values in the reference [23]. We found that $N = 80$ was enough.

Based on this, we implemented a four level refined grid where the finest resolution has $N_f = 80$. In this case the coarsest grid will only have $N_c = 10$. For accuracy reasons in this benchmark, we have decided to use the incompressible BGK model [11]. The only difference introduced by this model is the computation of the equilibrium distribution which is now given by

$$f_i^{eq} = w_i \left(\rho + \rho_0 \left(\frac{\boldsymbol{x}_i \cdot \boldsymbol{u}}{c_s^2} + \frac{1}{2c_s^4} \boldsymbol{Q}_i : \boldsymbol{uu} \right) \right), \tag{42}$$

where here ρ_0 is a constant and was chosen to be equal to one here. Since the discussion we performed in Sec. 3 is generic and does not really depend on the exact form of the equilibrium distribution there is no change in the refinement strategy.

The refinement used for our simulation is depicted in Fig. 14. In this picture, N refers to the total diameter of the cylinder in coarse units. The results

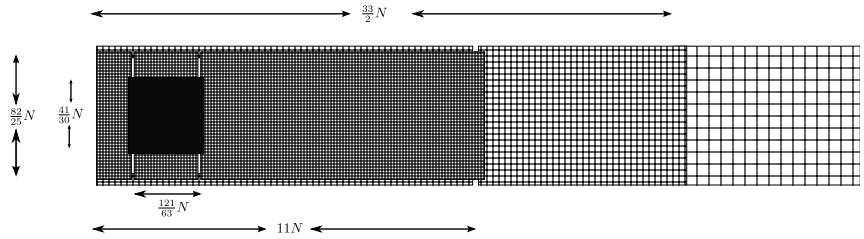

Figure 14: Proposed refinement for the 2D flow past a cylinder.

obtained for the drag and lift for the refined case match the accuracy with three significants digits (which is also the level of accuracy given in Ref. [23])

$$C_d = 3.24, \quad C_l = 0.982. \tag{43}$$

Furthermore, these values are in good agreement with those found in [23], where $C_d = 3.22 - 3.24$ and $C_l = 0.99 - 1.01$. The results for the uniform grid are obtained with about $5900N^2$ points, whereas there are only about $1000N^2$ points in the refined case, representing roughly five time less points. We did not perform any advanced technique to find the optimal position of our refined grids, it is therefore possible that one can get similar results with even less points.

In order to compare the CPU performances, we used a single CPU to simplify the comparison procedure. When benchmarking the uniform grid against the refined one for several values of N, we find that a mean speed-up of 5, thus proving that the overhead introduced by the coupling between the different grid levels does not have a significant impact in the overall performances.

4.2. Dipole–wall collision

This benchmark, based on Refs. [7] and [13], analyzes the time evolution of a self-propelled dipole confined within a 2D box. The geometry of the box is a square domain $[-1, 1] \times [-1, 1]$ and is surrounded by no-slip walls. The initial condition describes two counter-rotating monopoles, one with positive core vorticity at the position (x_1, y_1) and the other one with negative core vorticity at (x_2, y_2). This is obtained with an initial velocity field $\boldsymbol{u}_0 = (u_x, u_y)$ which reads as follows in dimensionless variables :

$$u_x = -\frac{1}{2} \|\omega_e\| (y - y_1) \mathrm{e}^{-(r_1/r_0)^2} + \frac{1}{2} \|\omega_e\| (y - y_2) \mathrm{e}^{-(r_2/r_0)^2}, \tag{44}$$

$$u_y = +\frac{1}{2} \|\omega_e\| (x - x_1) \mathrm{e}^{-(r_1/r_0)^2} - \frac{1}{2} \|\omega_e\| (x - x_2) \mathrm{e}^{-(r_2/r_0)^2}. \tag{45}$$

Here, $r_i = \sqrt{(x-x_i)^2 + (y-y_i)^2}$, defines the distance to the monopole centers. The parameter r_0 labels the diameter of a monopole and ω_e its core vorticity.

The average kinetic energy of this system at a given time is defined by the expression

$$\langle E \rangle (t) = \frac{1}{2} \int_{-1}^{1} \int_{-1}^{1} \|\boldsymbol{u}\|^2 (\boldsymbol{x}, t) d^2x, \tag{46}$$

and the average enstrophy by

$$\langle \Omega \rangle (t) = \frac{1}{2} \int_{-1}^{1} \int_{-1}^{1} \omega^2 (\boldsymbol{x}, t) d^2x, \tag{47}$$

where $\omega = \partial_x u_y - \partial_y u_x$ is the flow vorticity.

The dipole described by Eqs. (44) and (45), under the actions of viscous forces, develops a net momentum in the positive x-direction and is self-propelled towards the right wall. When the dipole collides with the wall a maxima of enstrophy is achieved.

In order to obtain really accurate results, the pressure field must be initialized from the velocity field, using the Poisson equation

$$\boldsymbol{\nabla}^2 p = (\boldsymbol{\nabla} \boldsymbol{u}) : (\boldsymbol{\nabla} \boldsymbol{u})^T \tag{48}$$

where the column symbol stands for the full index contraction. This equation is solved with a Gauss-Seidel iterative method.

In this test-case, we are aiming at obtaining the maximum values of the enstrophy which have been computed in [7] with a spectral method. We intend to use local refinement in the areas where we know the important physics happen, namely close to the wall where the small structures are formed. It is expected to obtain accurate values comparable to those obtained with a uniform grid. We study two different Reynolds numbers , $\mathrm{Re} = 625$ and $\mathrm{Re} = 5000$.

For $\mathrm{Re} = 625$, we use three levels of refinement as shown in the Fig. 15. The pathway of the dipole is refined. However, we know where it will collide, so this part is further refined. We find that this strategy agrees with the physical phenomenon.

This example is particularly challenging from the numerical point of view since strong velocity gradients are crossing the different refinement interfaces. We can compute the number of points in the refined domain and compare it against the number of points in the non-refined grid. Let us call N the resolution of the coarsest grid. The finer grid would therefore have a resolution of $4N$ or a total amount of points of $16N^2$. When using a local refinement strategy as the one proposed in this benchmark, one obtains $\frac{9}{2}N^2$ points. For any length N we observe that we are using almost four times less points in the refined case.

When performing a one CPU benchmark between a uniform grid and our refined version for several values of N, we find a mean speed-up of roughly three. Once again it is interesting to see that the performance gain is close to the memory saving that has been computed.

When analyzing the error for our refined grids, we compare the values obtained with the most accurate value of the enstrophy found by a spectral method,

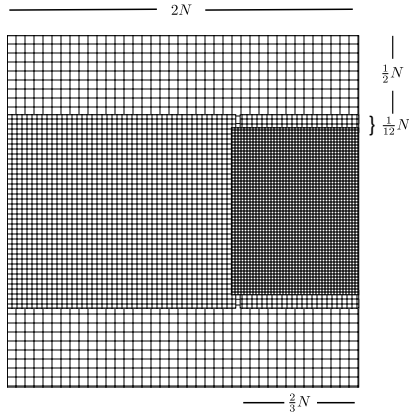

Figure 15: Refinement for the dipole-wall collision

which is 933.6. When decreasing N and simulating again, we find that the solutions tends to this value with second order slope. This can be seen in Fig. 16. This is really an interesting fact, as we are preserving the second order accuracy of the LBM with the grid refinement technique. It is important to note

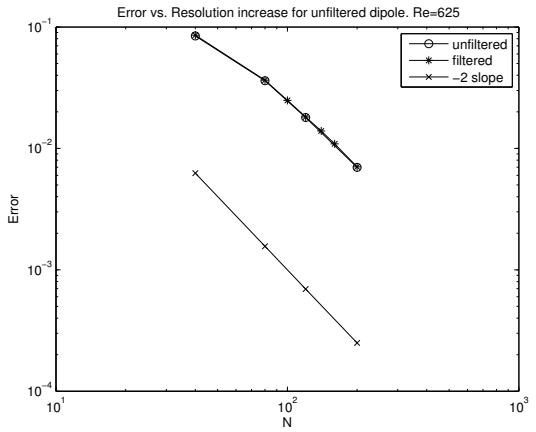

Figure 16: Evolution of error for the filtered and unfiltered simulations.

that the case $Re = 625$ can be done without the filtering method presented in Subsec. 3.3. However, when dealing with the case $Re = 5000$, it is imperative to use the filtering operation, as the small structures from the fine grid pollute the coarser grids, triggering numerical instability. This can be seen in Fig. 17, where, on the left, the non-filtered case shows signs of numerical instability (and finally diverges) while the filtered one remains stable.

For the case Re = 5000 the most accurate value obtained for the enstrophy in [7] (with an spectral method code) is 5536. When using six levels of refinement

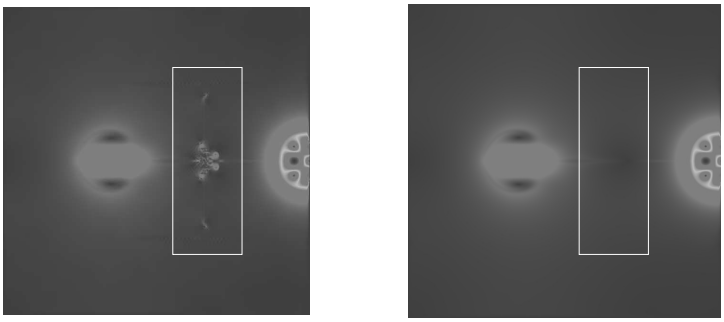

Figure 17: Problems at $Re = 5000$ for non-filtered simulation on the left. On the right the filtered version, which shows no perturbation.

(Fig. 18) and setting $N = 100$ for the coarsest grid we have found 5500 for the maximum of the enstrophy. When using a uniform grid, this same result would require a resolution of approximately $N = 3200$, which makes the solution of the problem really difficult to achieve, thus proving the importance of grid refinement.

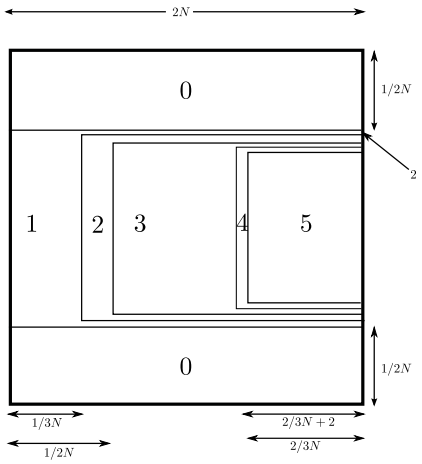

Figure 18: Six level grid refinement to achieve a good enstrophy value.

5. Conclusions

In this paper we proposed a detailed description of grid-refinement for the lattice Boltzmann method. First we have revisited the basics of the subject trying to provide a self-consistent presentation of grid-refinement in the cell-vertex case.

The order of accuracy of the interpolation needed at refinement interfaces was shown to be of at least third order in order to give satisfactory results.

We also introduced an advanced technique for the coupling between different grid levels thanks to a filtering of the non-equilibrium parts of the distribution function. This approach, inspired by classical computational fluid mechanics solvers, allowed an enhanced stability of the refinement algorithm at high Reynolds numbers. Furthermore, the accuracy of our algorithm at moderate Reynolds numbers was shown to be not reduced with respect to its non filtered counterpart.

Our approach was validated on two two-dimensional benchmarks: the flow past a circular cylinder and the dipole-wall collision. The effectiveness of the filtering was especially visible on the $\mathrm{Re} = 5000$ dipole-wall collision case, which was intractable without the filtering technique.

We have also performed comparisons between the CPU performance of our grid refinement against a uniform grid. We found that the speed-up obtained show a good consistency with respect to the memory savings.

The implementation presented is built on top of the Palabos open-source library, and it is available for testing purposes.

The next step is to port this approach in three dimensions. Because of the different nature of 2D and 3D turbulence, it is likely that extra care will be due for the treatment of 3D, high-Reynolds grid refinement. The 2D considerations of the present paper represent however the fundamental steps to a general understanding of grid refinement with lattice Boltzmann.

References

[1] C. K. Aidun and J. R. Clausen. Lattice-Boltzmann Method for Complex Flows. *Annu. Rev. Fluid Mech.*, 42(1):439–472, 2010.

[2] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94(3):511–525, May 1954.

[3] S. Chapman and T. G. Cowling. *The mathematical theory of nonuniform gases*. Cambridge University Press, Cambridge, 1960.

[4] Hudong Chen, Olga Filippova, J. Hoch, K. Molvig, Rick Shock, Chris Teixeira, and Raoyang Zhang. Grid refinement in lattice boltzmann methods based on volumetric formulation. *Physica A: Statistical Mechanics and its Applications*, 362(1):158–167, 2006.

[5] B. Chopard. Basic of Grid Refinement in Lattice Boltzmann. In S. Ubertini, G. Bella, S. A. Orszag, and S. Succi, editors, *Lectures on Lattice Boltzmann Methods for complex fluid flows*, page 120. Science4 Press, 2009.

[6] B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, 1998.

[7] H.J.H. Clercx and C.-H. Bruneau. The normal and oblique collision of a dipole with a no-slip boundary. *Computers & Fluids*, 35(3):245–279, March 2006.

[8] Alexandre Dupuis and Bastien Chopard. Theory and applications of an alternative lattice boltzmann grid refinement algorithm. *Physical Review E*, 67(6):066707, June 2003.

[9] Olga Filippova and Dieter Hänel. Grid refinement for Lattice-BGK models. *Journal of Computational Physics*, 147(1):219–228, November 1998.

[10] R K Freitas, M. Meinke, Lattice-Boltzmann Method, and Grid Refinement. Turbulence simulation via the Lattice-Boltzmann method on hierarchically refined meshes. pages 1–12, 2006.

[11] X. He and L.-S. Luo. Lattice Boltzmann model for the Incompressible Navier–Stokes Equation. *J. Stat. Phys.*, 88:927–944, 1997. 10.1023/B:JOSS.0000015179.12689.e4.

[12] K. Huang. *Statistical Mechanics*. J. Wiley, New York, 1987.

[13] Jonas Latt and Bastien Chopard. A benchmark case for lattice Boltzmann: Turbulent dipole-wall collision. *International Journal of Modern Physics C*, 18(04):619, 2007.

[14] Renwei Mei, Dazhi Yu, Wei Shyy, and Li-Shi Luo. Force evaluation in the lattice Boltzmann method involving curved geometry. *Physical Review E*, 65(4):041203, April 2002.

[15] Kannan N Premnath, Martin J Pattison, and Sanjoy Banerjee. Dynamic subgrid scale modeling of turbulent flows using Lattice-Boltzmann method. *Physica A*, page 318, January 2009.

[16] The Palabos project. `http://www.lbmethod.org/palabos`.

[17] Patrick Quéméré, Pierre Sagaut, and Vincent Couailler. A new multidomain/multiresolution method for largeeddy simulation. *International Journal for Numerical Methods in Fluids*, 36(4):391–416, June 2001.

[18] D. Ricot, S. Marié, P. Sagaut, and C. Bailly. Lattice Boltzmann method with selective viscosity filter. *J. Comp. Phys.*, 228(12):4478–4490, 2009.

[19] M. Rohde, D. Kandhai, J. J. Derksen, and H. E. A. van den Akker. A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes. *International Journal for Numerical Methods in Fluids*, 51:439–468, June 2006.

[20] Pierre Sagaut, Sebastien Deck, and Marc Terracol. *Multiscale And Multiresolution Approaches in Turbulence*. Imperial College Press, June 2006.

[21] X. Shan, X.-F. Yuan, and H. Chen. Kinetic theory representation of hydrodynamics: a way beyond the Navier Stokes equation. *J. Fluid Mech.*, 550:413–441, March 2006.

[22] S. Succi. *The lattice Boltzmann equation for fluid dynamics and beyond.* Oxford University Press, Oxford, 2001.

[23] S. Turek and M. Schfer. Recent benchmark computations of laminar flow around a cylinder. 1996.

[24] Dazhi Yu, Renwei Mei, Li-Shi Luo, and Wei Shyy. Viscous flow computations with the method of lattice boltzmann equation. *Progress in Aerospace Sciences*, 39(5):329–367, 2003.