HAL open science

Multidimensional fully adaptive lattice Boltzmann methods with error control based on multiresolution analysis

Thomas Bellotti, Loïc Gouarin, Benjamin Graille, Marc Massot

To cite this version:

Thomas Bellotti, Loïc Gouarin, Benjamin Graille, Marc Massot. Multidimensional fully adaptive lattice Boltzmann methods with error control based on multiresolution analysis. Journal of Computational Physics, 2022, 471, pp.111670. 10.1016/j.jcp.2022.111670 . hal-03158073v2

HAL Id: hal-03158073

https://hal.science/hal-03158073v2

Submitted on 13 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multidimensional fully adaptive lattice Boltzmann methods with error control based on multiresolution analysis

Thomas Bellotti[a], Loïc Gouarin[a], Benjamin Graille[b], Marc Massot[a]

[a]*CMAP, CNRS, École polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau Cedex, France.*
[b]*Institut de Mathématique d'Orsay, Université Paris-Saclay, 91405 Orsay Cedex, France.*

Abstract

Lattice-Boltzmann methods are known for their simplicity, efficiency and ease of parallelization, usually relying on uniform Cartesian meshes with a strong bond between spatial and temporal discretization. This fact complicates the crucial issue of reducing the computational cost and the memory impact by automatically coarsening the grid where a fine mesh is unnecessary, still ensuring the overall quality of the numerical solution through error control. This work provides a possible answer to this interesting question, by connecting, for the first time, the field of lattice-Boltzmann Methods (LBM) to the adaptive multiresolution (MR) approach based on wavelets. To this end, we employ a MR multi-scale transform to adapt the mesh as the solution evolves in time according to its local regularity. The collision phase is not affected due to its inherent local nature and because we do not modify the speed of the sound, contrarily to most of the LBM/Adaptive Mesh Refinement (AMR) strategies proposed in the literature, thus preserving the original structure of any LBM scheme. Besides, an original use of the MR allows the scheme to resolve the proper physics by efficiently controlling the accuracy of the transport phase. We carefully test our method to conclude on its adaptability to a wide family of existing lattice Boltzmann schemes, treating both hyperbolic and parabolic systems of equations, thus being less problem-dependent than the AMR approaches, which have a hard time guaranteeing an effective control on the error. The ability of the method to yield a very efficient compression rate and thus a computational cost reduction for solutions involving localized structures with loss of regularity is also shown, while guaranteeing a precise control on the approximation error introduced by the spatial adaptation of the grid. The numerical strategy is implemented on a specific open-source platform called `SAMURAI` with a dedicated data-structure relying on set algebra.

Keywords: Lattice Boltzmann Method, multiresolution analysis, wavelets, dynamic mesh adaptation, error control, hyperbolic systems of conservation laws, incompressible Navier-Stokes equations
2000 MSC: 76M28,65M50,42C40,
65M12,76-10,35L65,76D05

1. Introduction

Steep fronts and shocks are omnipresent structures in natural phenomena and can be modeled by very diverse mathematical models ranging from hyperbolic systems of conservation laws to hyperbolic-parabolic or parabolic systems. One finds them in a large variety of phenomena including both gas dynamics, when studying the air-flow around solid objects such as airplanes at relatively large speeds, and combustion and detonation waves in various devices or pollutant transport in the atmosphere and even in astrophysics.

In these physical phenomena, there are small areas where all the variation of the solution is concentrated – such as shocks – and a vast remaining part of the system where the solution varies smoothly or remains almost constant in large *plateaux*. Therefore, it is interesting to study how to exploit this property

Email addresses: `thomas.bellotti@polytechnique.edu` (Thomas Bellotti), `loic.gouarin@polytechnique.edu` (Loïc Gouarin), `benjamin.graille@universite-paris-saclay.fr` (Benjamin Graille), `marc.massot@polytechnique.edu` (Marc Massot)

of the system in order to reduce both the computational cost of the resolution and to decrease the memory footprint. This comes from the fact that in the areas where the solution varies gently, we can sample the solution coarsely without giving up on the overall quality and accuracy and we can expect to perform less computations.

Reducing the size of the spatial mesh can basically rely on three approaches. The first one consists in creating a fixed adapted mesh based on some *a priori* knowledge of the solution and which will not evolve with time. This simple approach is not recommended since it is highly problem-dependent, far from being optimal and can hardly provide an estimate on the additional error introduced by the adaptive mesh. There are some situations, namely in aeroacoustics, when the impact on the resolved physics is important and can be hard to recover from [37, 34, 42]. A second approach is the so-called Adaptive Mesh Refinement (AMR) approach [4], where the computational mesh is locally refined or coarsened dynamically in time according to a criterion based on the value of some sensor(s), such as the magnitude of the gradient of the solution. Though this method provides a relatively simple time-adaptive approach, it is built on heuristic arguments and thus prevents, in the vast majority of the cases, from having a direct control on the error. Finally, the last approach is based on the so-called adaptive multiresolution analysis (MR). The discrete solution is decomposed on a local wavelet basis and the coefficients provide a precise measure of the local regularity of the solution, thus supplying essential information on the necessity of refining or coarsening the mesh. This approach guarantees a precise control on the additional error introduced by the mesh adaptation and proves to be fundamentally problem-independent contrarily to AMR. This comes at the price of a more complex implementation and optimization of the procedure and calls for specific data-structures [55].

Multiresolution analysis, stemming from the pioneer works of Daubechies [15] and Mallat [52], has been successfully used in combination with the Finite Volume method by Harten [39, 40] for the solution of conservation laws with shocks, without exploiting its full capabilities in terms of cost reduction, but showing that one can recover an error control on the approximated solution. Cohen *et al.* [10] pushed the method forward by devising a fully adaptive strategy which reduces both the computational cost and the memory demand of the method. Such an approach has had a strong impact for a large spectrum of mathematical models and applications (see the non exhaustive list of examples [7, 60, 46, 13, 8, 12, 21, 28, 23, 19, 22, 56] for hyperbolic, parabolic, mixed and coupled systems). Interestingly, even if such comparisons are a difficult task and the results should be handled carefully, the technique has been compared with the AMR [16] where it proves to yield better compression rates. The major feature of the method is the error control but the difficulties concerning an efficient implementation are already pointed out.

In this work, we concentrate on the class of lattice Boltzmann methods, first introduced by McNamara and Zanetti [53] and which are capable of solving many different types of Partial Differential Equations. In all likelihood, the most well-known lattice Boltzmann scheme is the family of nine-velocities 2D scheme (see for example [45]), used to approximate the solution of the incompressible Navier-Stokes system in the limit of low Mach numbers. Overall, a large variety of such schemes exists in the literature and a general view on the possible approaches and applications is far from the scope of this work. Still, we want to emphasize the fact that the strategy developed in this paper can be *a priori* applied to any lattice Boltzmann scheme.

The inherent structure of the lattice Boltzmann methods comes from a discretization of a mesoscopic equation, with a high level of similarity to the Boltzmann equation, with a small family of discrete velocities which are multiple of the so-called lattice velocity, linked to the space and time discretizations. Therefore, the advection of the particles can be discretized exactly and the collision phase is kept explicit and local by seeing it as a linear relaxation towards the equilibrium distributions, coherently with the so-called BGK approximation of the collision operator as for the Boltzmann equation. The major strengths of the lattice Boltzmann methods are their simplicity and computational performance, since they are fully explicit and they lead to efficient parallelization and implementation on GPUs [61, 58, 50].

Until today, the available mesh adaptation strategies for lattice Boltzmann methods were only based either on a fixed mesh which does not evolve in time [36, 49, 44, 64, 29] or on the AMR approach [14, 30, 32, 33, 31], with the previously highlighted strengths and weaknesses. Due to the special relation between space and time discretization on the uniform lattice, one should pay special attention to the way of performing the time advancement. Besides the way of constructing the mesh, we can identify two main trends:

- Methods using local time steps for each level of refinement, thus needing spatial and temporal interpolations and an adaptation of the collision phase. We can cite those acting on fixed grids [36, 49, 44, 29] with nested patches and [64] with patches communicating only through edges. Rohde *et al.*

[59] employ the previous approach utilizing volumes to simplify the issue of conservation, which is a feature we retain for our strategy. The same method is combined with AMR in [30] with sensors based on the vorticity and on the matching of a free-flow total energy and by [14] with a criterion on the divergence of the velocity field.

The advantage of this procedure is that it minimizes the number of time steps but the shortcomings are the need of an adaptation of the collision phase, with possibly singular parameters, and the need of interpolation which calls for a massive storage of the solution at previous time steps.

- Methods using a global time step given by the finest space step and no need to adapt the collision. This is the strategy we shall employ in the work. We can cite [32, 33], who combine this with the AMR where the sensor is again based on the vorticity or use it [31] to simulate diphasic flows with sensor on the gradient of the phase-field. In these works, the authors employ a Lax-Wendroff scheme for the advection phase where the adaptation to the local level of refinement is done by modifying the CFL number. The collision remains untouched and performed locally. This method is simpler to implement, more flexible and needs less storage than the previous approach.

In a companion paper [3], we have made a first theoretical step in order to bridge the gap, in the one-dimensional context, between multiresolution analysis and lattice Boltzmann schemes, bringing in a proof of concept and the potential ability to address a large variety of schemes. However, several crucial questions remain. First, the extension to multi-dimension is far from obvious and is required in order to check the ability to capture properly the resolution of classical test-cases in the lattice Boltzmann community. Second, the multidimensional extension has to rely on an efficient implementation of the method; dynamic mesh handling, even if especially important to reduce the memory footprint, should remain marginal in terms of computational cost. Since this is strongly problem-dependent as demonstrated in [18], we will focus on algorithms rather than getting into the field of tailored data-structure and implementation optimization.

In this paper, we introduce a strategy to adapt pre-existing multidimensional lattice Boltzmann methods to be used on non-uniform time-evolving spatial meshes built using the multiresolution analysis. We keep the procedure as general as possible, ensuring a straightforward adaptation of the pre-existing lattice Boltzmann strategies to address a wide range of applications involving steep and localized fronts. The developed technique proves, for the first time in the lattice Boltzmann literature, to provide a control on the additional error introduced by the mesh adaptation in time and proves to be successful in coping both with hyperbolic and parabolic equations in several dimensions. Successful means here that we reach very interesting compression rates, while guaranteeing a resolution with a small and controlled error with respect to the simulation on the finest mesh.

Let us insist on the fact that we plainly rely on the original version of any lattice Boltzmann scheme and, combining efficiently with the MR approach, we are able to predict the proper physics with error control for both parabolic and hyperbolic problems. The algorithms are well-tailored and designed in such a way as to avoid the recursive nature of the multiresolution transform. Since the efficient implementation of multiresolution is always a key issue, we rely on a dedicated data-structure and an open-source code `SAMURAI`.[1] Even if the data-structure has been created for the sake of efficient parallelization, this issue is kept out of the scope of the present paper as it is a scientific subject *per se* and has to be properly investigated within the framework of lattice Boltzmann schemes (see [18] and [61]).

The paper is organized as follows: Sections 2, 3 and 4 introduce the basic bricks of the hybrid method, namely the concept of nested lattices, the lattice Boltzmann methods on volumes and the multiresolution analysis on Cartesian grids. Then, these ingredients are put together in the core Section 5, which elucidates how to construct the mesh in time and how to adapt the existing methods. The following Section 6 provides key issues on the implementation of the method and some preliminary paths towards optimization. The new strategy is systematically tested in Section 7 on a vectorial four-velocities scheme for hyperbolic conservation laws, on a nine-velocities scheme for the incompressible Navier-Stokes system of equations and finally on a six-velocities scheme to simulate the transport equation in three dimensions. Here, we show that we actually control the additional error and that the memory needed by the method is effectively reduced. We conclude in Section 8.

[1] https://github.com/hpc-maths/samurai.

2. Space and time discretization

Consider a multidimensional bounded domain $\Omega \subset \mathbb{R}^d$, where the dimension $d = 1, 2, 3$ is fixed from now on. For the sake of presentation, we consider hyper-cubic domains of the form $\Omega = [0, 1]^d$.

2.1. Nested lattices

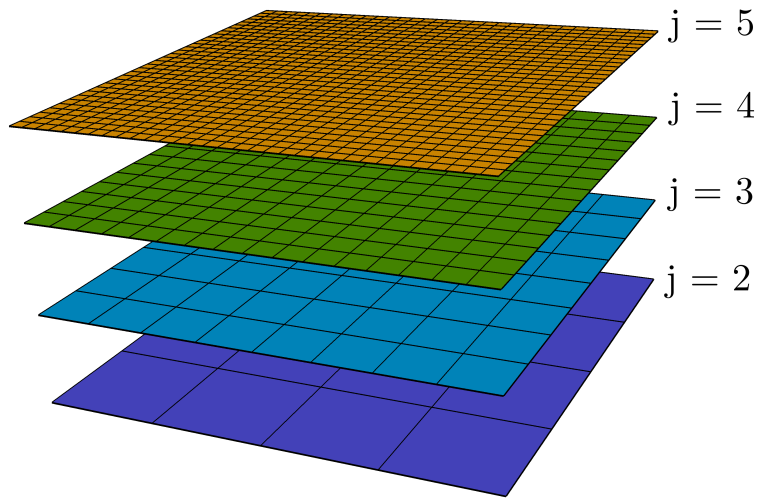

Figure 1: Example of nested dyadic grids for $d = 2$.

As done by [10] and [21], the spatial discretization of the domain Ω is carried out by considering a maximal level of resolution $\overline{J} \in \mathbb{N}$, as well as $L + 1$ nested uni-variate dyadic grids with $L \in \mathbb{N}$. Each lattice is identified by its level $j = \underline{J}, \ldots, \overline{J}$ with $\underline{J} := \overline{J} - L$, going from the coarsest to the finest discretization. The lattices are given by

$$\mathcal{L}_j := \left(C_{j,\boldsymbol{k}}\right)_{\boldsymbol{k}}, \qquad \text{with} \quad C_{j,\boldsymbol{k}} := \prod_{\alpha=1}^{d} \left[2^{-j} k_\alpha, 2^{-j}\left(k_\alpha + 1\right)\right],$$

with the set of admissible indices $\boldsymbol{k} \in \{0, \ldots, N_j - 1\}^d$, setting $N_j = 2^j$. These shall be the sets in which j and $\boldsymbol{k}$ are allowed to vary, unless otherwise stated. As shown in Figure 1, these lattices form a sequence of nested grids where the union of the children cells renders the parent cell

$$\bigcup_{\boldsymbol{\delta} \in \Sigma} C_{j+1, 2\boldsymbol{k}+\boldsymbol{\delta}} = C_{j,\boldsymbol{k}}, \qquad \text{with} \qquad \Sigma := \{0, 1\}^d, \quad j = \underline{J}, \ldots, \overline{J} - 1, \quad \text{and} \quad \boldsymbol{k} \in \{0, \ldots, N_j - 1\}^d. \tag{1}$$

We stress the fact that the nested structure can be seen in terms of trees (a binary tree for $d = 1$, a quadtree for $d = 2$ and an octree for $d = 3$), even if we shall avoid to explicitly construct it in the sequel because of the intrinsic recursive nature of such a structure. Thanks to the spatial discretization we have chosen, each cell at the finest level $\overline{J}$ has a edge length $\Delta x := 1/N_{\overline{J}}$. Thus the length of the edge of the cells at level j and the d-dimensional Lebesgue measures of the cells are

$$\Delta x_j := 2^{\overline{J}-j} \Delta x, \qquad \left|C_{j,\boldsymbol{k}}\right|_d = \left(\Delta x_j\right)^d = 2^{d\left(\overline{J}-j\right)} \left(\Delta x\right)^d,$$

for $j = \underline{J}, \ldots, \overline{J}$ and $\boldsymbol{k} \in \{0, \ldots, N_j - 1\}^d$.

2.2. Acoustic scaling

Once the question of the spatial discretization of Ω is successfully handled, we have to specify how do we intend to discretize the time variable. We have already mentioned that in the standard lattice-Boltzmann methods on uniform grids, the space step Δx and the time step Δt are indeed linked. This is achieved by considering the so-called lattice velocity $\lambda > 0$ and taking the acoustic scaling[2]

$$\lambda = \frac{\Delta x}{\Delta t}. \tag{2}$$

So far, we have seen that we have many spatial scales spanned by j, coexisting on the same domain Ω, at our disposal. The approaches stemming from that of Filippova and Hänel [36] consider as many local time-steps as $L+1$ to preserve the same lattice velocity everywhere. This strategies have the following drawbacks: they oblige one to modify the collision phase of the scheme to recover the right viscosity and they require to synchronize the different levels in time with time interpolations and rescalings the variables. To avoid these complications, we only consider one time scale Δt dictated by the finest spatial scale Δx through (2), as also done by Fakhari *et al.* [32, 33, 31]. However, we shall emphasize the differences with their strategy in the sequel. A unique time-step across levels has notable advantages, in particular the fact that the relaxation parameters do not need to be adjusted, avoiding to deal with singular parameters (infinite relaxation times) and the MRT (Multiple Relaxation Times, [20]) case is very naturally handled. Moreover, we do not have to perform temporal interpolations across the grids and we do not need to store the solution at many different sub time-steps.

3. Lattice Boltzmann schemes on volumes

Being given spatial and temporal discretizations, we now derive the lattice Boltzmann schemes we employ on volumes. Indeed, it is most common to interpret the solutions of a lattice Boltzmann scheme in terms of point values rather than averages on volumes [27, 9]. In this paper, the latter interpretation is utilized to keep the conservation properties unchanged when adapting the mesh and to be coherent with the most common version of the multiresolution procedure.

In this Section – since we do not deal with boundary conditions (treated in Section 5.3) – we temporarily forget to work on a bounded domain Ω by allowing $\boldsymbol{k} \in \mathbb{Z}^d$.

3.1. The origin from the discrete Boltzmann equation

Consider the discrete Boltzmann equation without external forcing with $q \in \mathbb{N}$ discrete velocities spanned by the indices $h = 0, \dots, q-1$ and collision kernel under the MRT approximation

$$\frac{\partial f^h}{\partial t} + \boldsymbol{\xi}^h \cdot \nabla f^h = -\sum_{l=0}^{q-1} \omega_{hl} \left(f^l - f^{l,\mathrm{eq}} \right), \qquad h = 0, \dots, q-1. \tag{3}$$

where $\boldsymbol{\xi}^h = \lambda \boldsymbol{\eta}^h$ is the microscopic velocity with corresponding particle distribution function $f^h = f^h(t, \boldsymbol{x})$, with $\boldsymbol{\eta}^h \in \mathbb{Z}^d$ being the dimensionless "logical" velocity. ω_{hl} are the entries of the matrix of relaxation frequencies denoted $\boldsymbol{\omega}$ and $f^{l,\mathrm{eq}} = f^{l,\mathrm{eq}}(f^0, \dots, f^{q-1})$ are the equilibrium distributions which in general are non linear functions of the conserved moments and thus of the distribution functions.[3]

The velocity space has already been discretized, thus we still have to handle time and space, as fully described in Appendix 1. The main points are the time integration along the characteristics of $\boldsymbol{\xi}^h$ with a trapezoidal rule to approximate the collision term and the averaging on each cell with commutation of the integral and the collision operator. Considering a cell $C_{\bar{J},\boldsymbol{k}}$ for any $\boldsymbol{k} \in \mathbb{Z}^d$ at the finest level of discretization, at the end of the process, the fully discrete scheme reads

$$\overline{f}^h_{\bar{J},\boldsymbol{k}+\boldsymbol{\eta}^h}(t+\Delta t) = \overline{f}^h_{\bar{J},\boldsymbol{k}}(t) - \Delta t \boldsymbol{\omega} (\boldsymbol{I} + \Delta t \boldsymbol{\omega}/2)^{-1} \left(\overline{\boldsymbol{f}}_{\bar{J},\boldsymbol{k}}(t) - \boldsymbol{f}^{\mathrm{eq}} \left(\overline{f}^0_{\bar{J},\boldsymbol{k}}(t), \dots, \overline{f}^{q-1}_{\bar{J},\boldsymbol{k}}(t) \right) \right) |_h, \tag{4}$$

where the bars indicate averages on the considered cell.

[2] A parabolic scaling of type $\Delta t \sim \Delta x^2$ is also possible but not covered here.

[3] The choice $\omega_{hl} = \delta_{hl}/\tau$ with relaxation time $\tau > 0$ gives the well-known BGK model with single relaxation time.

Remark 3.1 (Not a way of approximating the Boltzmann equation)**.** *We emphasize the fact that even if* (4) *is one of the possible space-time discretizations of the discrete Boltzmann equation* (3)*, the link between* (3) *and the continuous Boltzmann equation, where the velocities vary in a large continuous set, is in general far from being clear. It is therefore misleading to consider that* (4) *is a convergent discretization of a mesoscopic level of description describing a specific physics such as the continuous Boltzmann equation, because* q *remains finite whereas* $\Delta x, \Delta t \to 0$*. Our way of proceeding is just a manner of proposing a scheme, which can be seen as an algorithmic recipe for solving a macroscopic level of physical modeling.*

3.2. Collision and transport

The fully discretized lattice Boltzmann scheme (4) is made up of a collision and a stream part. For the sake of clarity and for future use, since they undergo different treatments, we separate them more explicitly. The link with the operator splitting and the fractional step methods has been studied by Dellar [17]. The sub-steps are given by:

- **Collision**, local to each cell and possibly involving the evaluation of non-linear equilibrium functions. This reads

$$\overline{\boldsymbol{f}}^{\star}_{\bar{\jmath},\boldsymbol{k}}(t) = \overline{\boldsymbol{f}}_{\bar{\jmath},\boldsymbol{k}}(t) - \Delta t \boldsymbol{\omega} (\boldsymbol{I} + \Delta t \boldsymbol{\omega}/2)^{-1} \left(\overline{\boldsymbol{f}}_{\bar{\jmath},\boldsymbol{k}}(t) - \boldsymbol{f}^{\mathrm{eq}} \left(\overline{f}^{0}_{\bar{\jmath},\boldsymbol{k}}(t), \dots, \overline{f}^{q-1}_{\bar{\jmath},\boldsymbol{k}}(t) \right) \right).$$

  The sign $\star$ shall denote any post-collisional state. To diagonalize the collision matrix using a change of basis $\boldsymbol{M} \in \mathrm{GL}_q(\mathbb{R})$, it is more common to express it in the space of moments, yielding

$$\begin{aligned} \overline{\boldsymbol{m}}_{\bar{\jmath},\boldsymbol{k}}(t) &= \boldsymbol{M} \overline{\boldsymbol{f}}_{\bar{\jmath},\boldsymbol{k}}(t), \\ \overline{\boldsymbol{f}}^{\star}_{\bar{\jmath},\boldsymbol{k}}(t) &= \boldsymbol{M}^{-1} \left((\boldsymbol{I} - \boldsymbol{S}) \overline{\boldsymbol{m}}_{\bar{\jmath},\boldsymbol{k}}(t) + \boldsymbol{S} \boldsymbol{m}^{\mathrm{eq}} \left(\overline{m}^{0}_{\bar{\jmath},\boldsymbol{k}}(t), \dots \right) \right), \end{aligned} \tag{5}$$

  where $\boldsymbol{I}$ is the $q \times q$ identity, $\boldsymbol{S} = \mathrm{diag}(s_0, \dots, s_{q-1})$ is the matrix of relaxation coefficients such that $s_h \in [0, 2]$ for $h = 0, \dots, q-1$ and $\boldsymbol{m}^{\mathrm{eq}}$ are the moments at equilibrium depending on the conserved moments. Indeed, in every lattice Boltzmann scheme, there is a certain number of moments which are conserved through the collision phase: their relaxation parameter is conventionally set to zero.

- **Stream**, non-local but (intrinsically) linear

$$\overline{f}^{h}_{\bar{\jmath},\boldsymbol{k}+\boldsymbol{\eta}^h}(t + \Delta t) = \overline{f}^{h,\star}_{\bar{\jmath},\boldsymbol{k}}(t), \qquad \text{or} \qquad \overline{f}^{h}_{\bar{\jmath},\boldsymbol{k}}(t + \Delta t) = \overline{f}^{h,\star}_{\bar{\jmath},\boldsymbol{k}-\boldsymbol{\eta}^h}(t). \tag{6}$$

  These equation emphasize, in a very compact and elegant fashion, the upwind nature of the stream phase with respect to the direction of each microscopic velocity. From now and for future use on, given $h = 0, \dots, q-1$, we indicate $h^{\dagger}$ the indices such that $\boldsymbol{\eta}^{h^{\dagger}} = -\boldsymbol{\eta}^h$.

We drop the bar to indicate mean values, assuming that this is understood in the next pages.

Remark 3.2 (Vectorial schemes)**.** *To simulate systems of hyperbolic equations, in particular the Euler system (see Section 7.1), we shall utilize the notion of vectorial scheme, so called in the works of Graille [38] and Dubois [26]. Vectorial schemes are a subclass of lattice Boltzmann schemes introduced originally in [2], [62] and [41] to simulate the Navier-Stokes equations coupled with thermal phenomena. Even if such schemes bear some similarities with discrete velocity kinetic schemes [6] and [1], they are fundamentally different in both the streaming and collision resolution and have been developed in parallel but different communities.*

4. Multiresolution analysis

In the previous Sections, we have presented the volumetric lattice Boltzmann methods on a uniform grid and the framework of nested dyadic grids, without saying how to locally adapt the latter. We answer this question in the present Section, providing the core concepts of multiresolution analysis on Cartesian meshes. Since this procedure is still static with respect to the time variable t, we omit the time for the sake of clarity.

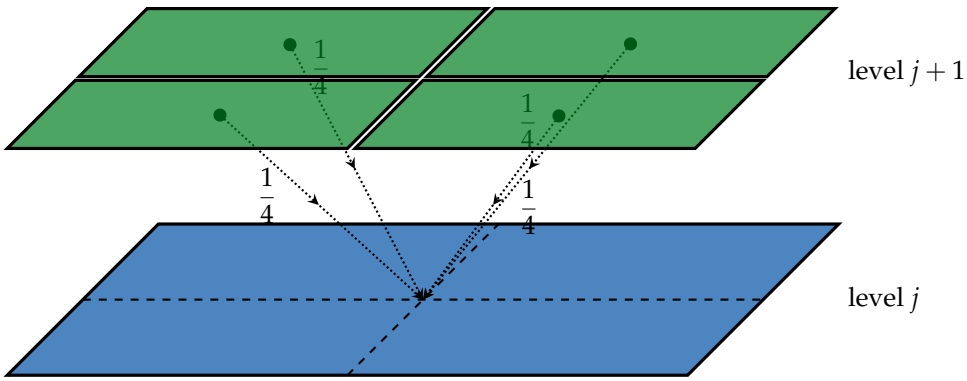

Figure 2: Illustration of the action of the projection operator in the context of $d = 2$. The cell average on the cell at level j is reconstructed by taking the average of the values on its four children at level $j+1$.

4.1. Projection and prediction operators

Multiresolution analysis is made possible by two basic operations, called "projection" and "prediction", allowing to pass from a fine to a coarse representation and *vice-versa* through one-level jumps. We first introduce the projection operator, whose rationale is illustrated on Figure 2 and which is used to transform information known on a fine level $j+1$ to one known on a coarse level j

Definition 4.1 (Projection operator)**.** *Let $j = \underline{J}, \dots, \overline{J} - 1$. The projection operator $\boldsymbol{P}_\triangledown : \mathbb{R}^{2d} \to \mathbb{R}$, taking data on level $j+1$ and yielding data on level j acts in the following way:*

$$f^h_{j,\boldsymbol{k}} = \boldsymbol{P}_\triangledown \left((f^h_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}})_{\boldsymbol{\delta}\in\Sigma} \right) := \frac{1}{|\mathrm{C}_{j,\boldsymbol{k}}|_d} \sum_{\boldsymbol{\delta}\in\Sigma} |\mathrm{C}_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}}|_d \, f^h_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}} = \frac{1}{2^d} \sum_{\boldsymbol{\delta}\in\Sigma} f^h_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}} \, .$$

Given a cell, the projection operator is fully local and is the unique operator conserving the average because it takes the average of the values defined on children cells which are nested inside their parent according to (1).

The prediction operator acts in the opposite direction, thus transforming information known on a coarse level j to one on a finer level $j+1$. This is the crucial ingredient of multiresolution and it is not uniquely defined, because there are many ways of reconstructing the lacking pieces of information "destroyed" by the projection operator. In order to ensure the feasibility of the whole procedure and enforce mass conservation that one expects from a lattice Boltzmann method, some constraints inspired by [10] have to be imposed

Definition 4.2 (Prediction operator)**.** *Let $j = \underline{J}, \dots, \overline{J} - 1$. The prediction operator $\boldsymbol{P}_\triangle : \mathbb{R}^{1+w} \to \mathbb{R}^{2d}$, used to transform information known on a level j to a guessed value on level $j+1$ (denoted by a hat) fulfills the following properties:*

- *It is local, namely, the value predicted on a cell $\mathrm{C}_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}}$ for any $\boldsymbol{\delta} \in \Sigma$ can only depend on the values of cells of level j belonging to a finite stencil $\mathcal{R}(j, \boldsymbol{k})$ of $w \in \mathbb{N}$ elements around its parent $\mathrm{C}_{j,\boldsymbol{k}}$.*
- *It is consistent with the projection operator, so the operator*

$$\boldsymbol{P}_\triangledown \circ \boldsymbol{P}_\triangle : \mathbb{R}^{1+w} \to \mathbb{R},$$

  leaves the value on the parent cell invariant, namely

$$(\boldsymbol{P}_\triangledown \circ \boldsymbol{P}_\triangle)(f^h_{j,\boldsymbol{k}}, \dots, \dots) = f^h_{j,\boldsymbol{k}}. \tag{7}$$

The first point ensures that the operator is local around a given cell, whereas the second[4] guarantees that the discrete integrals are conserved and that the parent necessarily belongs to $\mathcal{R}(j, \boldsymbol{k})$.

[4]In wavelet theory, this consistency property is interpreted by saying that the mother wavelet is oscillating, with vanishing mean.

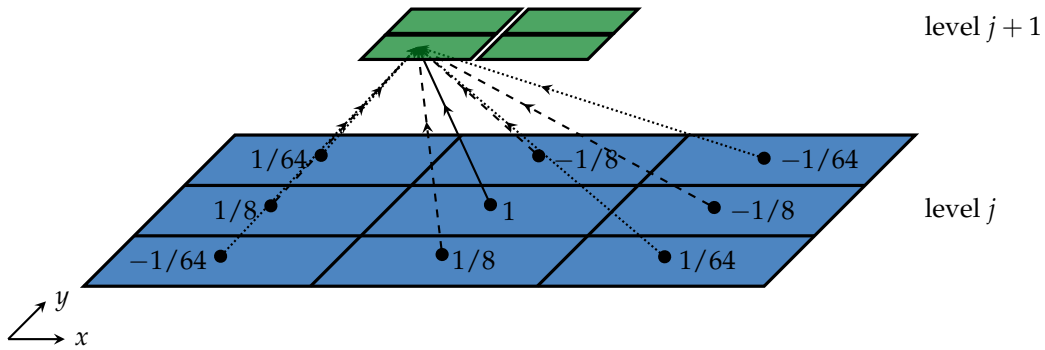

Figure 3: Illustration of the action of the prediction operator in the context of $d = 2$ for $\gamma = 1$. We indicate the value of the weight for each cell in the prediction stencil. In particular, we are predicting on the lower-left sibling and this yields a particular sign of the weights for this cell. The parent cell (weight 1) is connected through a continuous arrow, whereas neighbors along the axis (weight $\pm 1/8$) with a dashed arrow and along the diagonals (weight $\pm 1/64$) with a dotted arrow.

γ	c_1	c_2	c_3
1	$-1/8$	-	-
2	$-22/128$	$3/128$	-
3	$-201/1024$	$11/256$	$-5/1024$

Table 1: Coefficients used in the polynomial centered prediction operators (taken from Duarte [21]). The rationale and the procedure to recover these coefficient are presented in Appendix 2.

Though not compulsory, we consider a linear prediction operator. Using this kind of operator ensures the performance optimization of the whole algorithm as described in Section 6 and guarantees a good generality of the procedure. Let γ be a non-negative integer and consider the prediction operators generated by polynomial centered interpolations which are exact for polynomials up to degree 2γ, thus being of order $\mu := 2\gamma + 1$ and their generalization by tensor product to the multidimensional context by Bihari and Harten [5]:

- For $d = 1$, we utilize

$$\widehat{f}^h_{j+1,2k+\delta} = f^h_{j,k} + (-1)^\delta \, Q^\gamma_1 \left(k; \boldsymbol{f}^h_j\right), \qquad \text{with} \quad Q^\gamma_1 \left(k; \boldsymbol{f}^h_j\right) := \sum_{\alpha=1}^{\gamma} c_\alpha \left(f^h_{j,k+\alpha} - f^h_{j,k-\alpha}\right),$$

  for $\delta = 0, 1$, where the first coefficients are given on Table 1 and recovered as presented in Appendix 2.

- For $d = 2$, we use the same idea than for $d = 1$ along the x and y axis completed by a tensor product term along the diagonals

$$\widehat{f}^h_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}} = f^h_{j,\boldsymbol{k}} + (-1)^{\delta_1} \, Q^\gamma_1 \left(k_1; \boldsymbol{f}^h_{j,(\cdot,k_2)}\right) + (-1)^{\delta_2} \, Q^\gamma_1 \left(k_2; \boldsymbol{f}^h_{j,(k_1,\cdot)}\right) - (-1)^{\delta_1+\delta_2} \, Q^\gamma_2 \left(\boldsymbol{k}; \boldsymbol{f}^h_j\right),$$

  with

$$Q^\gamma_2 \left(\boldsymbol{k}; \boldsymbol{f}^h_j\right) := \sum_{\alpha,\beta=1}^{\gamma} c_\alpha c_\beta \left(f^h_{j,(k_1+\alpha,k_2+\beta)} - f^h_{j,(k_1-\alpha,k_2+\beta)} - f^h_{j,(k_1+\alpha,k_2-\beta)} + f^h_{j,(k_1-\alpha,k_2-\beta)}\right).$$

  for $\boldsymbol{\delta} \in \Sigma$. The way this formula acts for $\gamma = 1$ is illustrated on Figure 3. This will be indeed the choice for all the examples we will show in pages to come.

- For $d = 3$, the strategy follows the same rationale than for the previous point

$$\widehat{f}^h_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}} = f^h_{j,\boldsymbol{k}} + (-1)^{\delta_1} Q^\gamma_1 \left(k_1; \boldsymbol{f}^h_{j,(\cdot,k_2,k_3)}\right) + (-1)^{\delta_2} Q^\gamma_1 \left(k_2; \boldsymbol{f}^h_{j,(k_1,\cdot,k_3)}\right) + (-1)^{\delta_3} Q^\gamma_1 \left(k_3; \boldsymbol{f}^h_{j,(k_1,k_2,\cdot)}\right)$$
$$- (-1)^{\delta_1+\delta_2} Q^\gamma_2 \left((k_1,k_2); \boldsymbol{f}^h_{j,(\cdot,\cdot,k_3)}\right) - (-1)^{\delta_1+\delta_3} Q^\gamma_2 \left((k_1,k_3); \boldsymbol{f}^h_{j,(\cdot,k_2,\cdot)}\right) - (-1)^{\delta_2+\delta_3} Q^\gamma_2 \left((k_2,k_3); \boldsymbol{f}^h_{j,(k_1,\cdot,\cdot)}\right)$$
$$+ (-1)^{\delta_1+\delta_2+\delta_3} Q^\gamma_3 \left(\boldsymbol{k}; \boldsymbol{f}^h_j\right),$$

  for $\boldsymbol{\delta} \in \Sigma$, where we have set

$$Q^\gamma_3 \left(\boldsymbol{k}; \boldsymbol{f}^h_j\right) := \sum_{\alpha,\beta,\tau=1}^{\gamma} c_\alpha c_\beta c_\tau \Big(f^h_{j,(k_1+\alpha,k_2+\beta,k_3+\tau)} - f^h_{j,(k_1-\alpha,k_2+\beta,k_3+\tau)} - f^h_{j,(k_1+\alpha,k_2-\beta,k_3+\tau)}$$
$$- f^h_{j,(k_1+\alpha,k_2+\beta,k_3-\tau)} + f^h_{j,(k_1-\alpha,k_2-\beta,k_3+\tau)} + f^h_{j,(k_1-\alpha,k_2+\beta,k_3-\tau)}$$
$$+ f^h_{j,(k_1+\alpha,k_2-\beta,k_3-\tau)} - f^h_{j,(k_1-\alpha,k_2-\beta,k_3-\tau)} \Big).$$

These operators satisfy the requirements we have highlighted so far in Definition 4.2, namely the locality and the consistency with the projection operator. It should be observed that for $\gamma > 0$, these operators are not convex combinations of the data, so we cannot expect any maximum principle to hold.

4.2. Details and local regularity

For the prediction operator is essentially an interpolation from coarse data obtained by averaging through the projection operator, one understands that the more a function is locally regular, the more the predicted values will be close to the actual values. On the other hand, if the function or its derivatives have abrupt spatial changes, the prediction operator will be less suitable to correctly reconstruct the averages. This is precisely what is encoded in the details, which are a metric to quantify the information loss introduced by the projection operator and which carry essential information on the regularity of the solution. Thus, they let us identify areas of the computation domain where the spatial resolution can be reduced without affecting the quality of the reconstruction.

Definition 4.3 (Details)**.** *Let $h = 0, \dots, q-1$, $j = \underline{J}+1, \dots, \overline{J}$ and $\boldsymbol{k} \in \{0, \dots, N_j - 1\}^d$. The detail $d^h_{j,\boldsymbol{k}}$ for the h-th field on the cell $C_{j,\boldsymbol{k}}$ is the difference between the actual value of the mean and the predicted value on this cell, that is*

$$d^h_{j,\boldsymbol{k}} := f^h_{j,\boldsymbol{k}} - \widehat{f}^h_{j,\boldsymbol{k}}.$$

Without entering into technicalities (see [3] and references therein), if the function has $\nu \geq 0$ bounded derivatives in a suitable neighborhood $\tilde{\Sigma}_{j,\boldsymbol{k}}$ of the cell $C_{j,\boldsymbol{k}}$ depending on the choice of prediction operator (namely on γ), one can show that

$$|d^h_{j,\boldsymbol{k}}| \lesssim 2^{-j\min(\nu,\mu)} |f^h|_{W^{\min(\nu,\mu)}_\infty(\tilde{\Sigma}_{j,\boldsymbol{k}})}, \quad \text{with} \quad |f|_{W^n_\infty(\Omega)} := \max_{|\boldsymbol{\alpha}|\leq n} \|D^{\boldsymbol{\alpha}} f\|_{L^\infty(\Omega)}, \tag{8}$$

being an appropriate Sobolev semi-norm computed on the neighborhood $\tilde{\Sigma}_{j,\boldsymbol{k}}$. The possibility of establishing such control comes from the fact that the weights $(c_\alpha)_{\alpha=1}^{\alpha=\gamma}$ have been chosen (see Appendix 2) so that the prediction operator is exact for polynomial functions of degree up to 2γ. This estimate means that the details decrease with larger level j if the regularity ν is such that $\nu > 0$ (namely if the solution is more than bounded), according to the smoothness of the function. Moreover, close to a jump discontinuity ($\nu = 0$) – the typical situation with Riemann problems – details have constant magnitude throughout the levels. Finally, we emphasize that details on cells having the same parent are redundant as a consequence of the consistency property (7). Hence, the following linear constraint holds[5] for any $h = 0, \dots, q-1$

$$\sum_{\boldsymbol{\delta}\in\Sigma} d^h_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}} = 0, \qquad j = \underline{J}, \dots, \overline{J}-1, \quad \boldsymbol{k} \in \{0, \dots, N_j - 1\}^d. \tag{9}$$

[5]For $d = 1$, only one detail on two is significant; for $d = 2$, only three out of four; for $d = 3$, only seven out of eight.

This calls for the introduction of the set of significant details $\nabla_j \subset \{(j,\boldsymbol{k}) \ : \ \boldsymbol{k} \in \{0,\dots,N_j\}^d\}$ for $j = \underline{J}+1,\dots,\overline{J}$ where for each 2^d cell of level j sharing the same parent, we eliminate one of them in order to avoid the redundancy by (9). We also set $\nabla_{\underline{J}} := \{(\underline{J},\boldsymbol{k}) \ : \ \boldsymbol{k} \in \{0,\dots,N_{\underline{J}}-1\}^d\}$. We also introduce the set $\nabla := \cup_{j=\underline{J}}^{\overline{J}} \nabla_j$. In this way, we have a one-to-one correspondence between data discretized on the finest level $\overline{J}$ and the data at the coarsest level $\underline{J}$ plus the details at each level $j = \underline{J}+1,\dots,\overline{J}$ upon removing the redundancy[6]

$$\boldsymbol{f}^h_{\overline{J}} \qquad \underset{\mathcal{M}_{\mathcal{R}}{}^{-1}}{\overset{\mathcal{M}_{\mathcal{R}}}{\rightleftarrows}} \qquad \left(\boldsymbol{f}^h_{\underline{J}}, \boldsymbol{d}^h_{\underline{J}+1},\dots,\boldsymbol{d}^h_{\overline{J}}\right),$$

where $\boldsymbol{f}^h_j = (f^h_{j,\boldsymbol{k}})_{\boldsymbol{k}\in\{0,\dots,N_j-1\}^d}$ for $j = \underline{J},\dots,\overline{J}$ and $\boldsymbol{d}^h_j = (d_{j,\boldsymbol{k}})_{(j,\boldsymbol{k})\in\nabla_j}$ for $j = \underline{J}+1,\dots,\overline{J}$.

4.3. Tree structure and grading

Definition 4.4 (Tree). *A set of indices $\Lambda \subset \nabla$ is said to represent a tree if the following holds:*

1. $\nabla_{\underline{J}} \subset \Lambda$
2. *If $(j,\boldsymbol{k}) \in \Lambda$ and $j > \underline{J}$, then also $2^d - 2$ siblings of (j,k) belong to Λ.*
3. *If $(j+1, 2\boldsymbol{k}+\boldsymbol{\delta}) \in \Lambda$ for some $\boldsymbol{\delta} \in \Sigma$, then either its parent $(j,\boldsymbol{k}) \in \Lambda$ or its has $2^d - 1$ siblings belonging to Λ.*

Given a tree $\Lambda \subset \nabla$ according to the previous definition, we indicate $R(\Lambda)$ (complete tree) the set of elements in Λ (details cells) completed by their siblings missed due to the construction of ∇_j (non-detail cells). We also introduce the set of leaves $L(\Lambda) \subset \Lambda$, which are the elements of Λ without sons. Adding the non-detail cells to $L(\Lambda)$ we obtain the complete leaves $S(\Lambda)$. As observed by [10] $(C_{j,\boldsymbol{k}})_{(j,\boldsymbol{k})\in S(\Lambda)}$ forms a hybrid partition of the domain Ω.

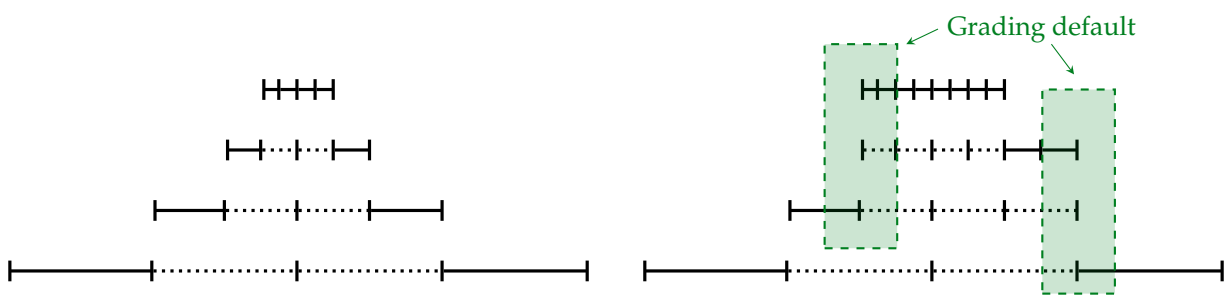

Figure 4: Graded tree (left) *versus* non-graded tree (right) in the case $d = 1$ for $\gamma = 1$. The complete leaves $S(\Lambda)$ are marked with full lines whereas the remaining cells $R(\Lambda) \setminus S(\Lambda)$ of the complete tree are rendered with dotted lines.

Furthermore, Cohen *et al.* [10] (Proposition 2.3 in their work) have also shown that one can perform the multiresolution transform in an optimal way if the resulting tree structure is graded (see Figure 4) with respect to the stencil of the prediction operator. Still, observe that the lack of grading does not prevent one from performing the multiresolution analysis, because the notion of detail and the decay estimates are still available.

Definition 4.5 (Grading). *Let $\Lambda \subset \nabla$ be a tree. Then it is said to be graded with respect to the prediction operator $\boldsymbol{P}_\triangle$ if for every $(j,\boldsymbol{k}) \in R(\Lambda)$ with $j > \underline{J}$ its prediction stencil also belongs to $R(\Lambda)$.*

In the sequel, given a tree structure, the operation yielding the corresponding minimal graded tree shall be indicated by $\mathcal{G}$. Therefore, grading is an important feature because it ensures an efficient implementation of the isomorphic transformation between averages on the complete leaves $S(\Lambda)$, namely $(f^h_{j,\boldsymbol{k}})_{(j,k)\in S(\Lambda)}$ and coarse means $(f^h_{\underline{J},\boldsymbol{k}})_{(j,\boldsymbol{k})\in\nabla_{\underline{J}}}$ plus $(d^h_{j,\boldsymbol{k}})_{(j,\boldsymbol{k})\in\nabla_j}$ details on the tree for $j = \underline{J}+1,\dots,\overline{J}$.

[6] The interested reader can check that both sides of the equation have the same number of elements.

4.4. Mesh coarsening

We have seen that knowing a function discretized on the whole finest level $\bar{J}$ is equivalent to know the means on the entire coarsest level $\underline{J}$ and all the details – upon eliminating their redundancy by (9) – on each level $j = \underline{J}+1, \dots, \bar{J}$. The passage from one representation to the other is performed by the so-called "fast wavelet transform". Apart from this equivalence, the decomposition in terms of details is superior as far as we want to probe the local regularity of the functions as expressed by (8). This can be exploited to coarsen the computational mesh in areas where the solution is strongly regular, still being sure that we can reconstruct information with accuracy within a certain given tolerance $0 < \epsilon \ll 1$. This is done, theoretically, by setting to zero the details which are below a certain value. From the practical point of view, one really eliminates the corresponding cells from the data structure at the end of the process.

Let $\Lambda \subset \nabla$ be a graded tree and $(f^h_{j,\boldsymbol{k}})_{(j,k)\in S(\Lambda)}$ the datum defined on its leaves. In order to yield a tree structure with the following operations, we must treat detail cells having the same parent at once. This is done by considering the same detail for all of them[7]

$$\rangle d^h_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}} \langle_\infty := \max_{\boldsymbol{\pi}\in\Sigma} |d^h_{j+1,2\boldsymbol{k}+\boldsymbol{\pi}}|, \qquad \delta \in \Sigma,$$

for $j = \underline{J}, \dots, \bar{J}-1$ and $\boldsymbol{k} \in \{0, \dots, N_j - 1\}^d$. Thus we consider the coarsening (or threshold) operator $\mathcal{T}_\epsilon$ given by

$$\mathcal{T}_\epsilon(\Lambda) := \nabla_{\underline{J}} \cup \left\{ (j,\boldsymbol{k}) \in \Lambda \setminus \nabla_{\underline{J}} \; : \; \max_{h=0,\dots,q-1} \rangle d^h_{j,\boldsymbol{k}} \langle_\infty \geq \epsilon_j \right\}, \tag{10}$$

where $\epsilon_{\underline{J}+1}, \dots, \epsilon_{\bar{J}}$ is a sequence of non-negative thresholds. We also define the reconstruction operator $\hat{\hat{\ }}$, where $\hat{\hat{\boldsymbol{f}}}^h_{\bar{J}} = (\hat{\hat{f}}^h_{\bar{J},\boldsymbol{k}})_{\boldsymbol{k}\in\{0,\dots,N_{\bar{J}}-1\}^d}$ representing the recursive application of the prediction operator $\boldsymbol{P}_\triangle$ until reaching available data stored on the complete leaves $S(\Lambda)$. Following the proof given by Cohen *et al.* [10], we prove that setting $\epsilon_j := 2^{d(j-\bar{J})}\epsilon$ and considering the operator $\boldsymbol{T}_\Lambda$ setting to zero the details on cells which do not belong to Λ, then

$$\left\| \hat{\hat{\boldsymbol{f}}}^h_{\bar{J}} - \mathcal{M}_{\mathcal{R}}{}^{-1} \boldsymbol{T}_{\mathcal{G}\circ\mathcal{T}_\epsilon(\Lambda)} \mathcal{M}_{\mathcal{R}} \hat{\hat{\boldsymbol{f}}}^h_{\bar{J}} \right\|_{\ell^p} \leq C_{\mathrm{MR}}\epsilon, \tag{11}$$

with $p \in [1,\infty]$, where $C_{\mathrm{MR}} > 0$ depends on γ and p, but not on the number of levels L. In this section, we tried to succinctly present the basic elements without entering into technicalities. The interested reader is referred to our previous contribution [3] and to [10, 55, 21].

5. Adaptive lattice Boltzmann multiresolution scheme

In this central Section of the paper, we put all the previous ingredients together. In particular, we shall answer the question on how to utilize the multiresolution procedure to handle time-dependent problems and on the way of adapting a given lattice Boltzmann scheme under the formalism of Section 3.

5.1. Construction of the time adaptive mesh

At each discrete time t, we possess a solution defined on the complete leaves $S(\Lambda(t))$ of the tree $\Lambda(t) \subset \nabla$. To compute the solution at the next time step $t + \Delta t$, we need to ensure that the computational lattice is refined enough so that an upper-bound similar to (11) still holds for the new solution. Of course, due to the fact that at the moment of constructing the mesh, the new solution is still unknown, we have to devise a heuristics to slightly enlarge the tree $\Lambda(t)$ with the information known at time t. The way of operating is resumed as follows

$$\Lambda(t) \xrightarrow{\mathcal{T}_\epsilon} \mathcal{T}_\epsilon(\Lambda(t)) \xrightarrow{\mathcal{H}_\epsilon} \mathcal{H}_\epsilon \circ \mathcal{T}_\epsilon(\Lambda(t)) \xrightarrow{\mathcal{G}} \mathcal{G} \circ \mathcal{H}_\epsilon \circ \mathcal{T}_\epsilon(\Lambda(t)) =: \Lambda(t+\Delta t),$$

where $\mathcal{T}_\epsilon$ is the threshold operator, which eliminates superfluous cells according to (10), $\mathcal{H}_\epsilon$ is the enlarging operator that is defined in the sequel of the section and finally $\mathcal{G}$ is the grading operator introduced in the previous Section. Again, we stress that in this phase details are computed with the old solution at time t.

[7] Other authors [21] consider a ℓ^2 metric under the form $\rangle d^h_{j+1,2\boldsymbol{k}+\boldsymbol{\delta}} \langle_2 := 2^{-d/2} \sqrt{\sum_{\boldsymbol{\pi}\in\Sigma} |d^h_{j+1,2\boldsymbol{k}+\boldsymbol{\pi}}|^2}$ for $\delta \in \Sigma$.

5.1.1. Propagation of information at finite speed

On one hand, since we expect propagation of information at finite speed *via* the stream phase (6) of the lattice Boltzmann method, we want to ensure that this flux of information is correctly captured by the computational mesh. Inspired by [39], we thus request that $\mathcal{H}_\epsilon$ does the following

$$\text{If} \quad (j, \boldsymbol{k}) \in R(\mathcal{T}_\epsilon(\Lambda(t))), \quad \text{then} \qquad \left(j, \boldsymbol{k} - \boldsymbol{\eta}^h\right) \in R(\mathcal{H}_\epsilon \circ \mathcal{T}_\epsilon(\Lambda(t))), \qquad h = 0, \ldots, q-1. \tag{12}$$

The previous formula (12) stipulates that if a cell at a certain level of refinement j is kept in the structure, then we also keep some of its neighbors at the same level j. The number of kept neighbors is determined by the largest shift associated with the discrete velocities of the scheme. We observe that this procedure is inherent to the hyperbolic equations where it has originally been developed [39, 40] and [10] in the context of Finite Volume schemes. For parabolic systems, where the propagation is done at infinite speed, this procedure cannot guarantee that we are able to capture all the phenomena [56]. Nevertheless, the infinite velocity is intrinsic to the continuous equations and one can still claim that the previous rule works pretty well in practice because the numerical scheme behaves "hyperbolically".

5.1.2. Solution blowup

On the other hand, as the equations we plan to solve are non-linear, and this non-linearity reflects on the collision phase (5) of the lattice Boltzmann method, the regularity of the solution could vary (and decrease) in time. Therefore, especially with hyperbolic systems, the solutions can develop shocks which should be correctly captured by the mesh $\Lambda(t + \Delta t)$. Thus, we enforce that

$$\begin{aligned} \text{If} \quad (j, \boldsymbol{k}) \in R(\mathcal{T}_\epsilon(\Lambda(t))), \quad \underline{J} < j < \overline{J} \quad \text{and} \quad \max_{h=0,\ldots,q-1} \rangle d^h_{j,\boldsymbol{k}}(t) \langle_\infty \geq 2^{\overline{\mu}+d} \epsilon_j, \\ \text{then} \quad (j+1, 2\boldsymbol{k} + \boldsymbol{\delta}) \in R(\mathcal{H}_\epsilon \circ \mathcal{T}_\epsilon(\Lambda(t))), \qquad \boldsymbol{\delta} \in \Sigma, \end{aligned}$$

where $\overline{\mu} \in [0, \mu]$ has to be selected at the beginning of the simulation. Remark that this refinement criterion acts on 2^d siblings at once by refining all of them once the metric on the details is large enough. This refinement criterion means that if the current cell is kept because its detail is not small enough to coarsen it, but moreover the detail is quite large, then we have to refine such cell. This allows, see [3] for more details, to identify areas of the mesh where the solution is undergoing a decrease of smoothness. The rationale is based on estimations like (8) on details at time t to estimate those (unavailable) at time $t + \Delta t$ and the interested reader can look at [3] for details. Since we do not know the number of bounded derivatives $\nu = \nu(t)$ of the solution *a priori*, we have to consider the following parameter $\overline{\mu} := \min(\nu, \mu)$ as an adjustable one. For example, if we know that the solution is going to develop some shock, it could be a wise choice to select $\overline{\mu} = 0$.

By proceeding at enlarging the computational mesh in this way by $\mathcal{H}_\epsilon$, we assume that it is suitable to represent the solution at the new time $t + \Delta t$ within a reasonable tolerance given by ϵ. This assumption is often called Harten's heuristics [39] in the world of Finite Volume. After having applied the operators $\mathcal{T}_\epsilon$ and $\mathcal{H}_\epsilon$, the grid still has to be graded. Thus, applying $\mathcal{G}$, we obtain the graded mesh $\Lambda(t + \Delta t)$ to be used for the new computations.

Once the new mesh $\Lambda(t + \Delta t)$ is created, the old solution is adapted from $S(\Lambda(t))$ to $S(\Lambda(t + \Delta t))$ using the projection operator $\boldsymbol{P}_\triangledown$ when cells are merged by $\mathcal{T}_\epsilon$ and the prediction operator $\boldsymbol{P}_\triangle$ when they are created either by $\mathcal{H}_\epsilon$ or $\mathcal{G}$.

5.2. Adapting the lattice Boltzmann method

Once we have a new mesh $\Lambda(t + \Delta t)$ and the solution at time t defined on $S(\Lambda(t + \Delta t))$ at our disposal, we have to adapt the reference lattice Boltzmann method given by (5) and (6) in order to be deployed only on the complete leaves of such a tree. Given a cell $C_{j,\boldsymbol{k}}$ at level j, we consider the set $\mathcal{B}_{j,\boldsymbol{k}}$ of virtual cells at the finest level of refinement $\overline{J}$ covering $C_{j,\boldsymbol{k}}$ (recall that the grids are nested) and defined by

$$\mathcal{B}_{j,\boldsymbol{k}} := \left\{ \boldsymbol{k} 2^{\overline{J}-j} + \boldsymbol{\delta} \quad : \quad \boldsymbol{\delta} \in \left\{0, \ldots, 2^{\overline{J}-j} - 1\right\}^d \right\}. \tag{13}$$

5.2.1. Collision

As seen before, the collision (5) is local to each cell at the finest level $\bar{J}$, thus its generalization to non-uniform meshes is straightforward. Indeed, consider any complete leaf $C_{j,\boldsymbol{k}}$ with $(j,\boldsymbol{k}) \in S(\Lambda(t+\Delta t))$, then we update with

$$\begin{aligned} \boldsymbol{m}_{j,\boldsymbol{k}}(t) &= \boldsymbol{M}\boldsymbol{f}_{j,\boldsymbol{k}}(t), \\ \boldsymbol{f}^{\star}_{j,\boldsymbol{k}}(t) &= \boldsymbol{M}^{-1}\left((\boldsymbol{I}-\boldsymbol{S})\boldsymbol{m}_{j,\boldsymbol{k}}(t) + \boldsymbol{S}\boldsymbol{m}^{\mathrm{eq}}\left(m^0_{j,\boldsymbol{k}}(t),\dots\right)\right). \end{aligned} \tag{14}$$

As we fully discussed in [3], one should be aware that this can potentially add an error to the solution because we are implicitly assuming that

$$\boldsymbol{m}^{\mathrm{eq}}\left(m^0_{j,\boldsymbol{k}}(t),\dots\right) = \frac{1}{2^{d(\bar{J}-j)}}\sum_{\bar{\boldsymbol{k}}\in\mathcal{B}_{j,\boldsymbol{k}}} \boldsymbol{m}^{\mathrm{eq}}\left(\hat{\hat{m}}^0_{\bar{J},\bar{\boldsymbol{k}}}(t),\dots\right), \tag{15}$$

which is true only if the equilibrium functions are linear. Otherwise, we can only hope to have the equality plus an error of the order of ϵ. The previous formula means that colliding using the data directly available on the cell $C_{j,\boldsymbol{k}}$ is equivalent to doing it with the reconstructed data on the cells at the finest level $\mathcal{B}_{j,\boldsymbol{k}}$ covering $C_{j,\boldsymbol{k}}$ followed by an averaging step. Performing the collision at the finest level $\bar{J}$ by using the reconstruction operator as on the right-hand side of (14) and (15) (what [43] call "original adaptive scheme" for the computation of source terms for Finite Volume scheme) clearly guarantees excellent results. However, this would rely on the recursive nature of adaptive multiresolution and would yield an explosion of the complexity of the algorithm in most cases where a high compression rate can be reached. This holds even when memoization techniques are employed to reduce the number of evaluations due to the recursive structure of the reconstruction operator. We verified both with 1D and 2D tests that for the problems we analyzed in [3] and in this paper, the use of collision operator on the mere complete leaves (14) has a marginal impact on the accuracy of the adaptive method.

To sum up, performing the collision only on the complete leaves of the tree reduces the computational time associated with the collision phase because one performs a number of change of variables through $\boldsymbol{M}$ and evaluation of the non-linear equilibria equal to the cardinality of $S(t+\Delta t)$ instead of $2^{d\bar{J}}$.

5.2.2. Advection

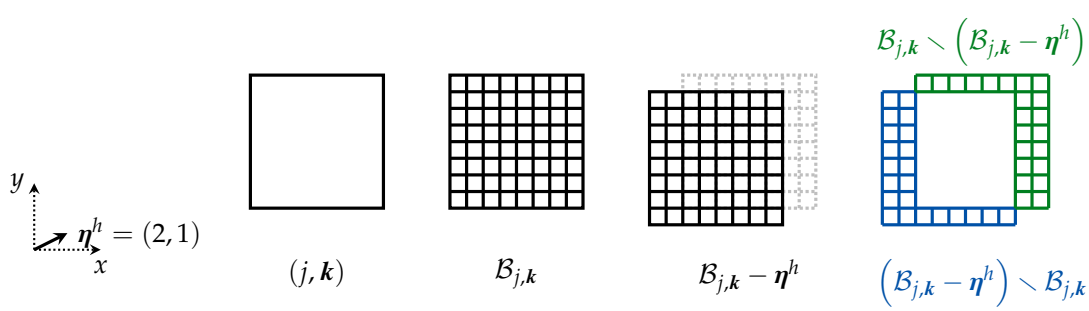

Figure 5: Example of the sets needed for the adaptive stream phase for the $d=2$ case. We consider a leaf $(j,\boldsymbol{k})$ which in this example is at level $j=\bar{J}-3$ and a logical velocity $\boldsymbol{\eta}^h=(2,1)$ (just for illustrative purposes).

Again, consider a complete leaf $C_{j,\boldsymbol{k}}$ with $(j,\boldsymbol{k}) \in S(\Lambda(t+\Delta t))$ and assume to work on the advection of the population $h=0,\dots,q-1$. We again recover the indices $\mathcal{B}_{j,\boldsymbol{k}}$ of the virtual cells at the finest level of resolution $\bar{J}$ covering $C_{j,\boldsymbol{k}}$, see (13). Let $\boldsymbol{w}\in\mathbb{Z}^d$, then $\mathcal{B}_{j,\boldsymbol{k}}-\boldsymbol{w}$ indicates the element-wise subtraction of $\boldsymbol{w}$. Once we have this, the adaptive stream phase is given by

$$f^h_{j,\boldsymbol{k}}(t+\Delta t) = f^{h,\star}_{j,\boldsymbol{k}}(t) + 2^{d(j-\bar{J})}\left(\sum_{\bar{\boldsymbol{k}}\in\mathcal{E}^h_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\bar{J},\bar{\boldsymbol{k}}}(t) - \sum_{\bar{\boldsymbol{k}}\in\mathcal{A}^h_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\bar{J},\bar{\boldsymbol{k}}}(t)\right), \tag{16}$$

with (Figure 5)

$$\mathcal{E}^h_{j,\boldsymbol{k}} := \left(\mathcal{B}_{j,\boldsymbol{k}} - \boldsymbol{\eta}^h\right) \setminus \mathcal{B}_{j,\boldsymbol{k}}, \qquad \mathcal{A}^h_{j,\boldsymbol{k}} := \mathcal{B}_{j,\boldsymbol{k}} \setminus \left(\mathcal{B}_{j,\boldsymbol{k}} - \boldsymbol{\eta}^h\right).$$

The updated average on cell $\boldsymbol{k}$ at level j is obtained by the post-collisional average on the same cell completed by the reconstruction of the post-collisional averages on some shifted virtual cells at the finest level of refinement $\overline{J}$, according to the velocity h we are considering. Here the first summed term corresponds to an incoming pseudo-flux, whereas the second one is the outgoing one. We mimic the way of proceeding by [10] performing the so-called "exact flux reconstruction" for Finite Volume schemes via the double hat operator. Conversely, the AMR approach from Fakhari *et al.* [32, 31] relies on a Lax-Wendroff scheme with direct evaluation on some ghost (or halo) cells at the same level of resolution j. This procedure is surely cheaper than (16) but cannot ensure any control on the error because the mesh is generated by a heuristic criterion which is not correlated to the way of locally reconstructing the solution *via* a cascade of interpolations.

It is important to observe that once we consider a cell $C_{j,\boldsymbol{k}}$ adjacent to the boundary of the domain $\partial\Omega$, some virtual cell indexed by $\mathcal{E}^h_{j,\boldsymbol{k}}$ could lie outside the domain. This calls for the enforcement of some boundary condition as explained in the following Section 5.3.

It can be easily shown that our approach, devised through a different way of reasoning, is indeed a CTU (Corner Transport Upwind method) Finite Volume discretization by Colella [11] (also see Leveque [48]) of the advection phase, with reconstruction of a piece-wise constant representation of the solution at the finest level made possible by the multiresolution. At finest level, the advection phase is given by the translation of data in the opposite direction of the velocity field. Let us illustrate the construction of (16): following (6)

$$f^h_{\overline{J},\overline{\boldsymbol{k}}}(t+\Delta t) = f^{h,\star}_{\overline{J},\overline{\boldsymbol{k}}-\boldsymbol{\eta}^h}(t), \qquad \overline{\boldsymbol{k}} \in \mathcal{B}_{j,\boldsymbol{k}}.$$

Unfortunately, the right-hand side of these equations is not always well defined because the cell could fail to be a complete leaf of the tree. Nevertheless, we are allowed to reconstruct data on these cells using multiresolution with a precision controlled by ϵ

$$f^h_{\overline{J},\overline{\boldsymbol{k}}}(t+\Delta t) \simeq \hat{\hat{f}}^{h,\star}_{\overline{J},\overline{\boldsymbol{k}}-\boldsymbol{\eta}^h}(t), \qquad \overline{\boldsymbol{k}} \in \mathcal{B}_{j,\boldsymbol{k}}.$$

We come back to level j by $\overline{J} - j$ applications of the projection operator $\boldsymbol{P}_\triangledown$ and consistency, yielding

$$\begin{aligned}
f^h_{j,\boldsymbol{k}}(t+\Delta t) &\simeq 2^{d(j-\overline{J})} \sum_{\overline{\boldsymbol{k}}\in\mathcal{B}_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\overline{J},\overline{\boldsymbol{k}}-\boldsymbol{\eta}^h}(t) \\
&= 2^{d(j-\overline{J})} \left(\sum_{\overline{\boldsymbol{k}}\in\mathcal{B}_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\overline{J},\overline{\boldsymbol{k}}}(t) + \sum_{\overline{\boldsymbol{k}}\in\mathcal{E}^h_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\overline{J},\overline{\boldsymbol{k}}}(t) - \sum_{\overline{\boldsymbol{k}}\in\mathcal{A}^h_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\overline{J},\overline{\boldsymbol{k}}}(t) \right) \\
&= f^{h,\star}_{j,\boldsymbol{k}}(t) + 2^{d(j-\overline{J})} \left(\sum_{\overline{\boldsymbol{k}}\in\mathcal{E}^h_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\overline{J},\overline{\boldsymbol{k}}}(t) - \sum_{\overline{\boldsymbol{k}}\in\mathcal{A}^h_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\overline{J},\overline{\boldsymbol{k}}}(t) \right).
\end{aligned}$$

If the implementation of the reconstruction operator is optimized, one can expect a gain in computational time when using (16) on the adaptive grid because, as one can see from Figure 5, the sets $\mathcal{E}^h_{j,\boldsymbol{k}}$ and $\mathcal{A}^h_{j,\boldsymbol{k}}$ only involve few cells at the finest level $\overline{J}$ close to the edges of each leaf.

5.3. Boundary conditions

We are left to describe our implementation of suitable boundary conditions for the problem. We consider a cell $C_{j,\boldsymbol{k}}$ touching the boundary $\partial\Omega$, so that $|\overline{C_{j,\boldsymbol{k}}} \cap \partial\Omega|_{d-1} > 0$ and a velocity field $\boldsymbol{\eta}^h$ with information coming from outside the domain, so $\boldsymbol{\eta}^h \cdot \boldsymbol{n} < 0$, where $\boldsymbol{n}$ is the normal vector to $\partial\Omega$. For illustrative purposes, we consider three types of boundary conditions which – for the reference scheme – correspond to

$$f^h(t+\Delta t) = \begin{cases} f^{h,\star}(t), & \text{(Copy)}, \\ f^{h^\dagger,\star}(t), & \text{(Bounce Back)}, \\ -f^{h^\dagger,\star}(t), & \text{(Anti Bounce Back)}, \end{cases} \tag{17}$$

where we recall that $\boldsymbol{\eta}^{h^\dagger} = -\boldsymbol{\eta}^h$ yielding the opposite dimensionless velocity. This given the name "bounce back" to this kind of conditions. One can handle non-homogeneous conditions or more intricate ones in the same way. Observe that, in the advection phase (16), we cannot close the system due to the lack of some pieces of information, indicated without the reconstruction sign

$$f^h_{j,\boldsymbol{k}}(t+\Delta t) = f^{h,\star}_{j,\boldsymbol{k}}(t) + 2^{d(j-\bar{J})}\left(\sum_{\bar{\boldsymbol{k}}\in\mathcal{E}^h_{j,\boldsymbol{k}}\cap\{0,\dots,N_{\bar{J}}-1\}^d} \hat{\hat{f}}^{h,\star}_{\bar{J},\bar{\boldsymbol{k}}}(t) - \sum_{\bar{\boldsymbol{k}}\in\mathcal{A}^h_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\bar{J},\bar{\boldsymbol{k}}}(t)\right) + 2^{d(j-\bar{J})}\sum_{\bar{\boldsymbol{k}}\in\mathcal{E}^h_{j,\boldsymbol{k}}\setminus\{0,\dots,N_{\bar{J}}-1\}^d} f^{h,\star}_{\bar{J},\bar{\boldsymbol{k}}}(t)\,.$$

Having $\mathcal{A}^h_{j,\boldsymbol{k}} \subset \mathcal{B}^h_{j,\boldsymbol{k}} \subset \{0,\dots,N_{\bar{J}}-1\}^d$, all the quantities appearing in the outgoing pseudo-fluxes are known since they come from inside the domain. Using (17) and presenting the "bounce back" case only for the sake of presentation, this corresponds to

$$f^h_{j,\boldsymbol{k}}(t+\Delta t) = f^{h,\star}_{j,\boldsymbol{k}}(t) + 2^{d(j-\bar{J})}\left(\sum_{\bar{\boldsymbol{k}}\in\mathcal{E}^h_{j,\boldsymbol{k}}\cap\{0,\dots,N_{\bar{J}}-1\}^d} \hat{\hat{f}}^{h,\star}_{\bar{J},\bar{\boldsymbol{k}}}(t) - \sum_{\bar{\boldsymbol{k}}\in\mathcal{A}^h_{j,\boldsymbol{k}}} \hat{\hat{f}}^{h,\star}_{\bar{J},\bar{\boldsymbol{k}}}(t)\right) + 2^{d(j-\bar{J})}\sum_{\bar{\boldsymbol{k}}\in\left(\mathcal{E}^h_{j,\boldsymbol{k}}\setminus\{0,\dots,N_{\bar{J}}-1\}^d\right)+\boldsymbol{\eta}^h} f^{h^\dagger,\star}_{\bar{J},\bar{\boldsymbol{k}}}(t)\,, \tag{18}$$

where all the sums pertain to quantities inside Ω, so the quantities on the second line of (18) can be evaluated either using the usual reconstruction operator or by direct evaluation, a cheaper alternative. We implement the latter method, which consists in taking the value directly available on the corresponding leaf.

6. Implementation

As already emphasized in the course of the paper, the structure of the multiresolution is intrinsically recursive, leading to important slow-downs when implementing the reconstruction operator in this fashion to obtain the solution on the finest mesh. This is especially evident for large problems with $d = 2, 3$ (see [10, 55]). In this Section, we provide an interesting idea on how to speed-up the computation of the reconstruction operator for the sole purpose of the lattice Boltzmann scheme in order to guarantee the proper accurate evaluation of the advection without relying on the full reconstruction at the finest level. Besides, we describe the overall structure of the numerical solver.

6.1. A non-recursive implementation of the reconstruction operator

To make computations feasible for large problems, we follow the idea of [10], claiming that in the univariate case, the recursive application of a linear prediction operator can be condensed into the computation of the powers of a given matrix at the beginning of the simulation, based on the assumption that the fluxes of the Finite Volume method involve only adjacent cells. For $d = 1$ and $\gamma = 1$, this reads

$$\begin{pmatrix} \hat{\hat{f}}^{h,\star}_{\bar{J},2^{\bar{J}-j}k-2} \\ \hat{\hat{f}}^{h,\star}_{\bar{J},2^{\bar{J}-j}k-1} \\ \hat{\hat{f}}^{h,\star}_{\bar{J},2^{\bar{J}-j}k} \\ \hat{\hat{f}}^{h,\star}_{\bar{J},2^{\bar{J}-j}k+1} \end{pmatrix} = \begin{pmatrix} 1/8 & 1 & -1/8 & 0 \\ -1/8 & 1 & 1/8 & 0 \\ 0 & 1/8 & 1 & -1/8 \\ 0 & -1/8 & 1 & 1/8 \end{pmatrix}^{\bar{J}-j} \begin{pmatrix} f^{h,\star}_{j,k-2} \\ f^{h,\star}_{j,k-1} \\ f^{h,\star}_{j,k} \\ f^{h,\star}_{j,k+1} \end{pmatrix}. \tag{19}$$

Clearly, this method works far from the boundary $\partial\Omega$ and in areas where the local refinement level of the leaves is constant. Since for $d > 1$ we have constructed the prediction operator by tensor product, the

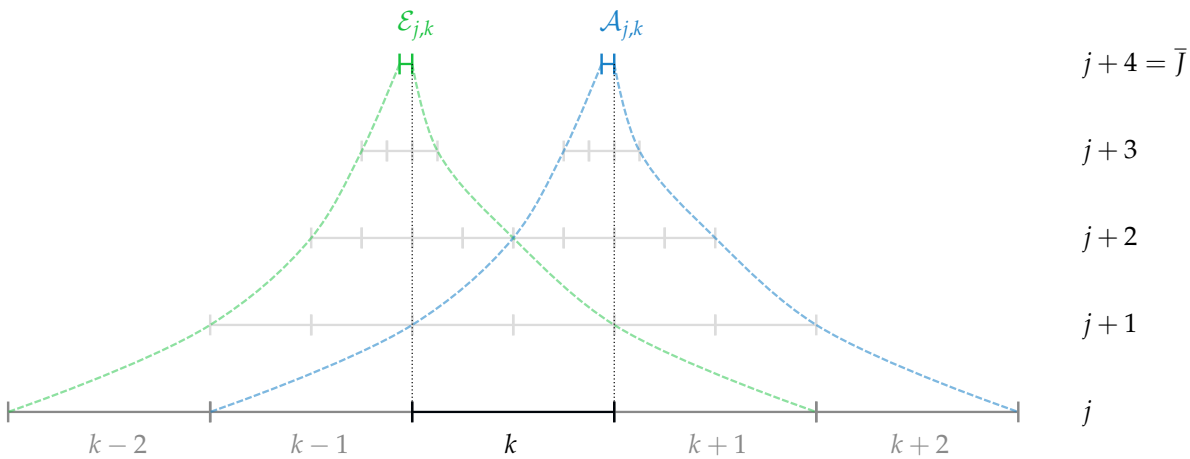

Figure 6: Example of non-recursive implementation of the reconstruction operator for $d = 1$ and $\gamma = 1$ for the velocity λ. The cell on which one updates the solution is four level far from the finest level and indexed by k. The green cell on top corresponds to the cell at the finest level $\bar{J}$ giving the incoming pseudo-flux for the cell k at level j, whereas that in blue corresponds to that of the outgoing one. The prediction operator is progressively applied spanning the intermediate (non existing) cells in pale grey inside the green (resp. blue) funnel, until reaching cells at the same level j in black and grey.

matrices involved in this framework shall just be the Kronecker product of that for $d = 1$ in (19) d-times with itself.

We start to describe the procedure for larger spatial dimensions d by selecting a complete leaf $C_{j,\boldsymbol{k}}$ and assuming that it is surrounded by enough leaves of the same level. At the beginning of the numerical simulation, we can once for all compute by recursion (we also did it analytically), for any level j and velocity h, the set of logical shifts $\Xi_j^h \subset \mathbb{Z}^d$ and real weights $(w_{j,\boldsymbol{\delta}}^h)_{\boldsymbol{\delta}\in\Xi_j^h}$ so that

$$\sum_{\bar{\boldsymbol{k}}\in\mathcal{E}_{j,\boldsymbol{k}}^h} \hat{\hat{f}}_{\bar{J},\bar{\boldsymbol{k}}}^{h,\star}(t) = \sum_{\boldsymbol{\delta}\in\Xi_j^h} w_{j,\boldsymbol{\delta}}^h \, f_{j,\boldsymbol{k}+\boldsymbol{\delta}}^{h,\star}(t)\,, \tag{20}$$

and the same for $\mathcal{A}_{j,\boldsymbol{k}}^h$, transforming the recursive left-hand side of (20) into a right-hand side made up of linear combinations of data known on the leaves with previously computed weights, in analogy with (19). An illustration of such a process in a one-dimensional setting is given in Figure 6.

On the other hand, if the surrounding leaves at the same level are not enough (we could, for example, fall on some ghost or halo cell[8]), we are not sure that the value we retrieve is accurate enough according to the multiresolution analysis. Let us study this more in detail, considering $\boldsymbol{\delta} \in \Xi_j^h$. There are two possibilities:

- $C_{j,\boldsymbol{k}+\boldsymbol{\delta}}$ is a complete leaf, thus everything is fine, because multiresolution allows us to employ this value in a recursion formula to reconstruct at finest level without adding any detail.
- $C_{j,\boldsymbol{k}+\boldsymbol{\delta}}$ is not a complete leaf. There are two situations (and not more thanks to the construction of the mesh), one of which requires particular care:
  - It intersects a leaf at the coarser level $j-1$. The value we retrieve is fine because multiresolution guarantees that quantities computed by applying the prediction operator without adding details are fine if their respective cell is situated on top of a leaf.
  - It intersects a leaf at the finer level $j+1$. This is the critical situation, because the retrieved value is not accurate enough to be employed in reconstructions, according to the multiresolution analysis and we have to add the proper detail information in order to preserve our target of error control.

[8]We added 2γ of them in each Cartesian and diagonal direction to ensure the feasibility of any operation concerning the prediction operator.

Comparing for $d = 2$, $\gamma = 1$ and a D2Q9 scheme to the Lax-Wendroff advection phase by [32, 33, 31], which has to recover three values of the solution at the considered level j in each velocity direction, multiply each of them by a weight and then summing them, it can be shown that our method using (20) needs to recover at most 25 values – whatever the velocity satisfying $\max(|\eta_x^h|, |\eta_y^h|) \leq 2$, thus for a large family of schemes – multiply each of them by a weight and then summing them. Even if this should be quantified precisely in terms of computational cost for a given problem, a given implementation and a given architecture, the complexity of the algorithms are sensibly at the same level. The gain comes once considering that we can deal with a very large class of schemes achieving a control on the error which is not obtained by the AMR procedures and that current investigations using formal expansions indicate that our adaptive method reproduces the behavior of the reference method on the finest grid in terms of both error control and equivalent equations up to order three, instead of two for the Lax-Wendroff approach.

6.2. *Overall solver*

To summarize, we present the overall procedure in a synthetic fashion. In the actual algorithm, we swap the collision and the advection phase as in Algorithm 1. This does not change the result since we start with initial data at the equilibrium.

Algorithm 1: Overall solver.

```
Initial time : t = 0;
Full finest mesh : Λ(0) = {(J̄, k) : k ∈ {0, ..., N_J̄ − 1}^d};
Initial datum at equilibrium : f_{J̄,k}(0) = M^{-1} m^eq_{J̄,k}(0);
while t ≤ T do
    Mesh adaptation : Λ(t + Δt) = G ∘ H_ε ∘ T_ε(Λ(t)) computing the details using P_▽ and P_△;
    The data structure on S(Λ(t)) is updated on S(Λ(t + Δt));
    for (j, k) ∈ S(Λ(t + Δt)) do
        if (j, k) internal to Ω then
            Advect using (16) and optimizing using (20) when possible ;
        else
            Advect using (18) and optimizing using (20) when possible;
        end
        Collide using (14);
    end
end
```

Remark 6.1. *In this work, we consider problems of moderate size where starting from the full finest mest* $\Lambda(0) = \{(\bar{J}, \boldsymbol{k}) \ : \ \boldsymbol{k} \in \{0, \ldots, N_{\bar{J}-1}\}^d\}$ *is not an issue. However, as emphasized by Cohen* et al. *[8] for Finite Volume, the initial datum could not to fit into memory for large realistic problems, especially when* q*, the number of discrete velocities, is large. We warn the reader that in this case, two approaches are customary. The first one is to start from a heuristically adapted mesh* $\Lambda(0) \neq \{(\bar{J}, \boldsymbol{k}) \ : \ \boldsymbol{k} \in \{0, \ldots, N_{\bar{J}-1}\}^d\}$ *which is coarsened in areas the initial solution is very smooth. The second one is to initialize the problem performing the multiresolution analysis on a less refined grid and then to refine every remaining cell a certain number of times and reinitializing the solution on the new cells, see [24, 21].*

The code is sequentially implemented in `C++` using a code called `SAMURAI`[9] (**S**tructured **A**daptive mesh and **MU**lti-**R**esolution based on **A**lgebra of **I**ntervals) which is currently under development and than can handle general problems involving dynamically refined Cartesian meshes, stemming both from MR and AMR, in an efficient way. The data structure `SAMURAI` relies on intervals of contiguous cells along each Cartesian direction gathered in different sets. The availability of inter-sets operations allows one to perform all the operations involved in multiresolution and to easily write the corresponding adaptive numerical

[9]The code with the test cases as well as its documentation can be found at `https://github.com/hpc-maths/samurai`.

scheme (lattice Boltzmann, Finite Volume, ...). Our implementation relies on the library `xtensor`[10] for the data storage and in order to take advantage of the lazy evaluations of expressions.

7. Validation and numerical experiments

To validate our approach, we investigate two types of system for $d = 2$, namely the full Euler system – taken as the prototype of non-linear hyperbolic system – on one hand and the incompressible Navier-Stokes equations on the other hand, which represent the archetypal non-linear parabolic system. Finally, for $d = 3$, we simulate the advection equation.

For the first test, the purpose is twofold: show that we can ensure a precise error control [3] using the procedure described in Section 5, which is tailored to fit hyperbolic systems; show that in problems involving steep fronts, we can achieve interesting compression rates and thus potentially a reduction of the computation time. For this test case, we employ a quite unsophisticated vectorial four-velocities scheme (D2Q4). Using vectorial five-velocities schemes (D2Q5) or a nine-velocities scheme (D2Q9) is straightforward.

The second test aims at showing that our procedure can be coupled with the most well-known lattice Boltzmann scheme – namely the nine-velocities scheme (D2Q9) – to tackle a problem which is not typically addressed by multiresolution, because of the smoothness of the solutions induced by the important role of the diffusion. We compare the adaptive scheme with the reference scheme using integral quantities such as the drag coefficient and the Strouhal number.

For the third test, we employ a six-velocities scheme (D3Q6) and we aim at providing a first example of three-dimensional computation.

All the tests are performed using $\gamma = 1$.

7.1. *Systems of hyperbolic conservation laws*

The non-isothermal Euler system is probably the most emblematic example of system of hyperbolic conservation laws. This kind of problem naturally develops weak solutions with shocks and therefore constitutes the ideal setting to test the efficiency and the reliability of multiresolution.

7.1.1. *Problem*

In the context of $d = 2$, we aim at finding an approximation of the weak entropic solution of the system

$$\begin{cases} \partial_t \boldsymbol{u} + \partial_x \Phi_x(\boldsymbol{u}) + \partial_y \Phi_y(\boldsymbol{u}) = 0, & t \in [0, T], \quad \boldsymbol{x} \in \mathbb{R}^2, \\ \boldsymbol{u}(t = 0, \boldsymbol{x}) = \boldsymbol{u}_0(\boldsymbol{x}), & \boldsymbol{x} \in \mathbb{R}^2, \end{cases} \tag{21}$$

where $\boldsymbol{u} : [0, T] \times \mathbb{R}^2 \to \mathbb{R}^m$, for $m \geq 1$, that we assume to be hyperbolic [63]. In this paper, we consider some of the test cases introduced by Lax and Liu [47] for the complete Euler system ($m = 4$), given by

$$\boldsymbol{u} = (\rho, \rho u, \rho v, E)^T, \quad \Phi_x(\boldsymbol{u}) = \Big(\rho u, \rho u^2 + p, \rho uv, u(E + p)\Big)^T, \quad \Phi_y(\boldsymbol{u}) \quad = \Big(\rho v, \rho uv, \rho v^2 + p, v(E + p)\Big)^T,$$

with pressure law given by $E = p/(\gamma_{\text{gas}} - 1) + \rho(u^2 + v^2)/2$, selecting $\gamma_{\text{gas}} = 1.4$. The computational domain is $\Omega = [0, 1]^2$ and we enforce the copy boundary conditions for each variable. The initial datum, given as a function of the density, the velocities and the pressure, is

$$\boldsymbol{u}_0(x, y) = \begin{cases} (\rho_{\text{UR}}, u_{\text{UR}}, v_{\text{UR}}, p_{\text{UR}})^T, & x > 1/2, \quad y > 1/2, \\ (\rho_{\text{UL}}, u_{\text{UL}}, v_{\text{UL}}, p_{\text{UL}})^T, & x < 1/2, \quad y > 1/2, \\ (\rho_{\text{LL}}, u_{\text{LL}}, v_{\text{LL}}, p_{\text{LL}})^T, & x < 1/2, \quad y < 1/2, \\ (\rho_{\text{LR}}, u_{\text{LR}}, v_{\text{LR}}, p_{\text{LR}})^T, & x > 1/2, \quad y < 1/2. \end{cases}$$

As test cases, we consider Configuration 3 and Configuration 12 by [47].

[10] https://xtensor.readthedocs.io

7.1.2. Numerical scheme

We employ a batch of m D2Q4 schemes with only one conserved moment, coupled through their equilibri in order to achieve the solution of the whole system, see Remark 3.2. The base D2Q4 scheme is obtained by selecting $q = 4$ for the following choice of discrete velocities

$$\boldsymbol{\xi}^h = \lambda \left(\cos\left(\frac{\pi}{2}h\right), \sin\left(\frac{\pi}{2}h\right) \right)^T, \qquad h = 0, 1, 2, 3,$$

regardless of which scheme $i = 1, \dots, m$ we are looking at. The change of variable yielding the moments is fixed as well

$$\boldsymbol{M} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \lambda & 0 & -\lambda & 0 \\ 0 & \lambda & 0 & -\lambda \\ \lambda^2 & -\lambda^2 & \lambda^2 & -\lambda^2 \end{pmatrix}, \qquad \boldsymbol{S}_i = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & s_i^q & 0 & 0 \\ 0 & 0 & s_i^q & 0 \\ 0 & 0 & 0 & s_i^{xy} \end{pmatrix}, \qquad i = 1, \dots, m,$$

with the relaxation matrix depending on the component of the system we are considering with $s_i^q, s_i^{xy} \in (0, 2)$ for $i = 1, \dots, m$. We can show, using the equivalent equations [25], that selecting the equilibri as

$$m_i^{0,\text{eq}} = u_i, \qquad m_i^{1,\text{eq}} = \Phi_{x,i}(\boldsymbol{u}), \qquad m_i^{2,\text{eq}} = \Phi_{y,i}(\boldsymbol{u}), \qquad m_i^{3,\text{eq}} = 0,$$

the scheme is consistent with system (21) up to order Δt. We have also tested our strategy with the D2Q4 twisted scheme, see Février [35], where the velocities are not parallel to the axis and are given by

$$\boldsymbol{\xi}^h = \lambda \left(\cos\left(\frac{\pi}{2}h + \frac{\pi}{4}\right), \sin\left(\frac{\pi}{2}h + \frac{\pi}{4}\right) \right)^T, \qquad h = 0, 1, 2, 3,$$

observing good behaviors like for the standard D2Q4. We do not present such tests. For the examined configurations, we found that $\lambda = 5$ and

$$s_i^q = \begin{cases} 1.9, & i = 1 \\ 1.75, & i = 2, 3, 4. \end{cases} \qquad \text{and} \qquad s_i^{xy} = 1,$$

provide adequate performances and a reasonable amount of numerical diffusion.

7.1.3. Points of emphasis

- The first important point is to verify that we control the additional error. In our previous contribution [3], we have shown that under certain assumptions on the lattice Boltzmann scheme, we are able to control the additional error introduced by the mesh adaptation

$$E[m^h](t) = \frac{\sum_{\boldsymbol{k} \in \{0,\dots,N_{\overline{J}}-1\}^d} \Delta x \left| \widehat{\widehat{m}}^h_{\overline{J},\boldsymbol{k}}(t) - m^{\text{REF},h}_{\overline{J},\boldsymbol{k}}(t) \right|}{\sum_{\boldsymbol{k} \in \{0,\dots,N_{\overline{J}}-1\}^d} \Delta x \left| m^{\text{REF},h}_{\overline{J},\boldsymbol{k}}(t) \right|},$$

  where the solution denoted by "REF" is the one by the reference scheme on a uniform mesh at finest level $\overline{J}$ and h spans the conserved moments. We are looking for an upper bound with $C(T) > 0$ of the kind

$$E[m^h](T) \leq C(T)\, \epsilon, \tag{22}$$

  where $T > 0$ is the final time. Observe that as we are computing the difference with the reference scheme, the solution of the latter depends on the maximum level of resolution $\overline{J}$. This should be taken into account when comparing results for different $\overline{J}$.

- The second important axis of analysis is the gain in terms of computational time and memory impact thanks to the joint work of the adaptive scheme with multiresolution. The memory occupation rate and the mesh occupation rate at time t, given by

$$\text{MemOR}(t) = \frac{\#(\text{total cells in } \Lambda(t + \Delta t))}{\#(\text{total cells in the full mesh})}, \qquad \text{MeshOR}(t) = \frac{\#(S(\Lambda(t + \Delta t))}{2^{\overline{J}}},$$

are taken as metrics as far as the memory impact is concerned. Observe that occupation rates much smaller than one are good and correspond to high compression rates (one has CR $= 1 -$ OR). Since the parallelization and the optimization of the computational code is not the central topic of this work and is still undergoing major improvements which shall be a subject on their own, we do not perform an extensive analysis, which could follow the guidelines of [16], of the gain in term of computational time. However, notice that since the adaptive scheme only works on the complete leaves of the tree, MeshOR(t) can be reasonably considered an indicator of the computational cost of the method, apart from the time spent adapting the mesh. This will need to be confirmed in future works. However, at each time step, since the number of collisions to perform is equal to $\#(S(\Lambda(t+\Delta t)))$ and $\#(\mathcal{E}^h_{j,\boldsymbol{k}}), \#(\mathcal{A}^h_{j,\boldsymbol{k}}) \sim 2^{(d-1)(\bar{J}-j)} \ll 2^{d(\bar{J}-j)}$ if the level j is far enough from $\bar{J}$, corresponding to the number of reconstruction to be done for the stream phase, we believe that the number of operations to perform in one time step can be made significantly smaller than for the reference scheme. This is particularly flagrant when handling large three-dimensional problems like in Section 7.3.

7.1.4. Results and discussion

General remarks. The structure of the solution and the additional error for Configuration 3 are given in Figure 7 at final time $T = 0.3$; those for Configuration 12 are given in Figure 8 at final time $T = 0.25$. In the former case, we remark that the four shocks, where all the conserved moments are discontinuous, are well resolved and finely meshed, so that we can observe some hydrodynamic instabilities typical of such systems[11] (see [51] for example). In the latter case, the two shocks propagating towards the upper-right corner are followed by the finest discretization of the mesh, whereas we observe a coarsening of one level (for $\epsilon = 5 \cdot 10^{-3}$) close to the static contact discontinuities. This phenomenon shall be clarified in a moment with a finer analysis. Overall, this qualitative analysis allows us to conclude that the adaptive lattice Boltzmann scheme succeeds in reproducing the expected behavior of the solution [47, 51] of the Euler system and that the adaptive grid follows the shock structures propagating with finite velocity.

Error control. According to the results shown in Figure 9 and 10, respectively for Configuration 3 and 12, we verify that the upper bound (22) with ϵ is verified for these test cases. This is in agreement with the theoretical analysis made in [3] and holds for any choice of finest resolution $\bar{J}$.

Besides corroborating the theoretical analysis in a multidimensional setting, this confirms that, as far as we are dealing with hyperbolic problems, the enlargement strategy devised in Section 5.1 is capable of ensuring that the adaptive mesh correctly follows the temporal evolution of the solution.

Compression. Still looking at Figure 9 and 10, we observe that the occupation rates, namely MemOR(t) and MeshOR(t) become more interesting as one approaches larger maximum levels of resolution $\bar{J}$. This is in accordance with the intuition that since a fine sampling of the solution is needed only close to the shocks, the number of needed leaves shall grow in $\bar{J}$ more slowly than $2^{d\bar{J}}$, also because shock are $(d-1)$-dimensional entities.

As far as time is concerned, after an important initial growth guided by the refinement criterion $\mathcal{H}_\epsilon$, the trend of the occupation rates eventually stabilizes, especially for Configuration 3, where the secondary structures close to the hydrodynamic instabilities induced by the contact discontinuities do not grow too much in size as time advances. For Configuration 12, the occupation rates grow linearly even close to the final time because of the expanding curvilinear front linking the main shocks and the contact discontinuities.

Finally, the occupation rates are less interesting once we decrease ϵ. In these tests, we deliberately used unrealistically small ϵ only aiming at showing convergence. We conclude that the choice of ϵ should be the result of an arbitration between a desired target error and performances.

Coupling between MR and LBM. Role of $\overline{\mu}$. As it has been hinted while discussing Figure 8, we remark that the larger errors for Configuration 12 are situated close to the static contact discontinuities. This also holds, but less spectacularly, for Configuration 3 in the area where contact discontinuities are present causing the

[11] With the limitation of dealing with a low order scheme.

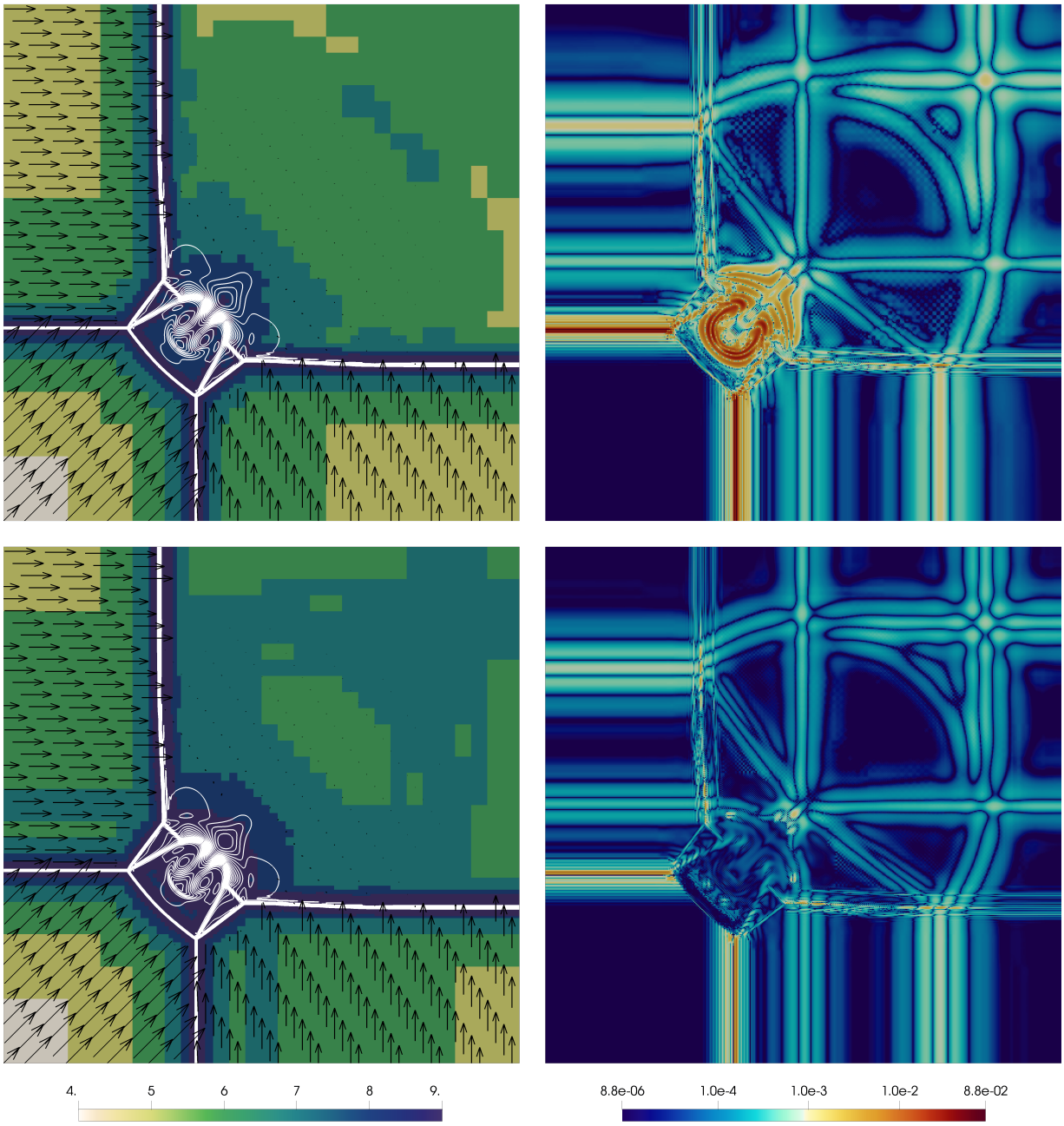

Figure 7: Configuration 3. Top $\epsilon = 0.005$, bottom $\epsilon = 0.001$ and $\overline{\mu} = 0$. On the left, level j (colored), contours of the density field (white) and velocity field (black). On the right: local relative error of the adaptive method with respect to the reference method. Time $T = 0.3$, $\underline{J} = 2$ and $\overline{J} = 9$.

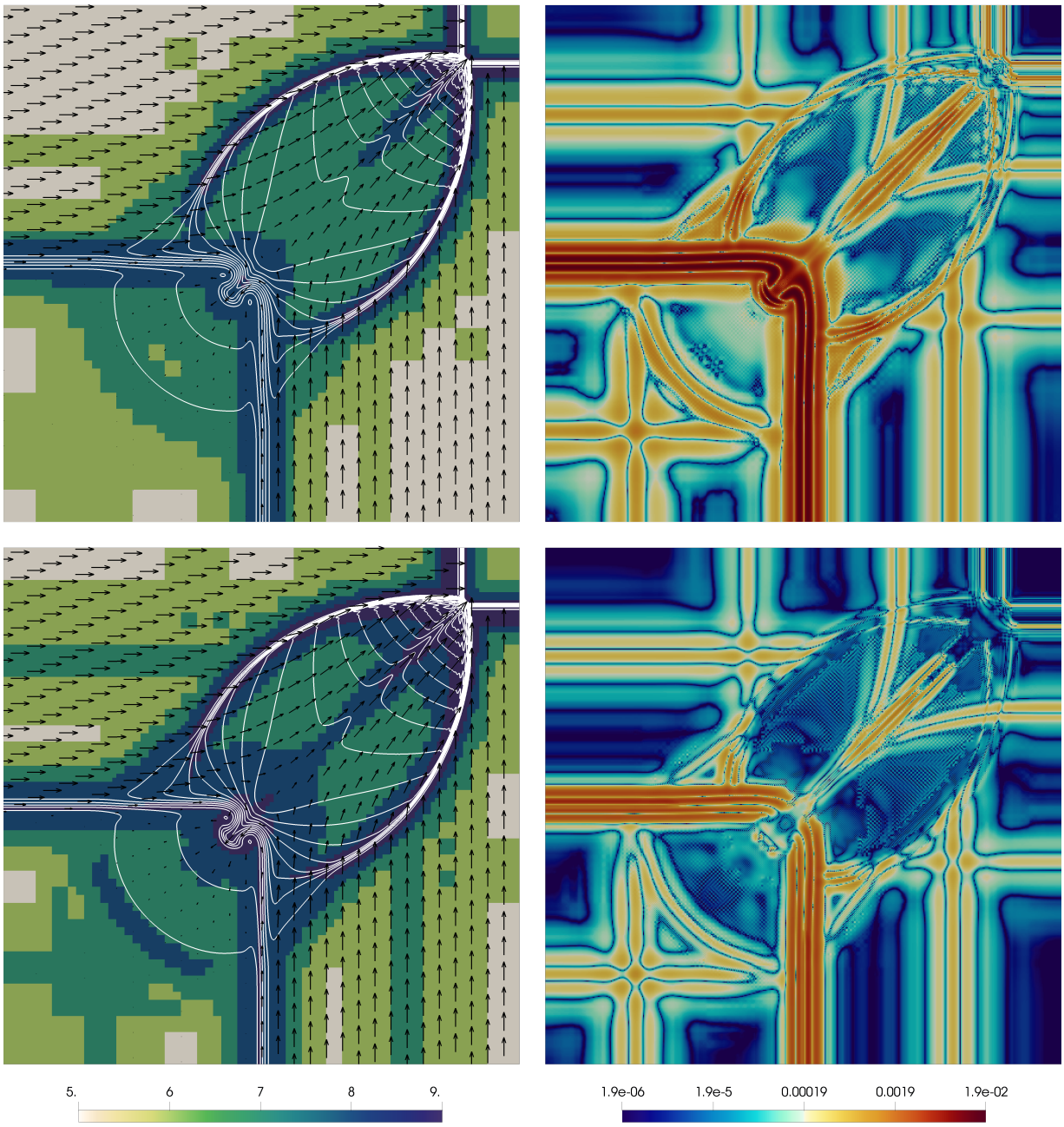

Figure 8: Configuration 12. Top $\epsilon = 0.005$, bottom $\epsilon = 0.001$ and $\overline{\mu} = 0$. On the left, level j (colored), contours of the density field (white) and velocity field (black). On the right: local relative error of the adaptive method with respect to the reference method. Time $T = 0.25$, $\underline{J} = 2$ and $\overline{J} = 9$.

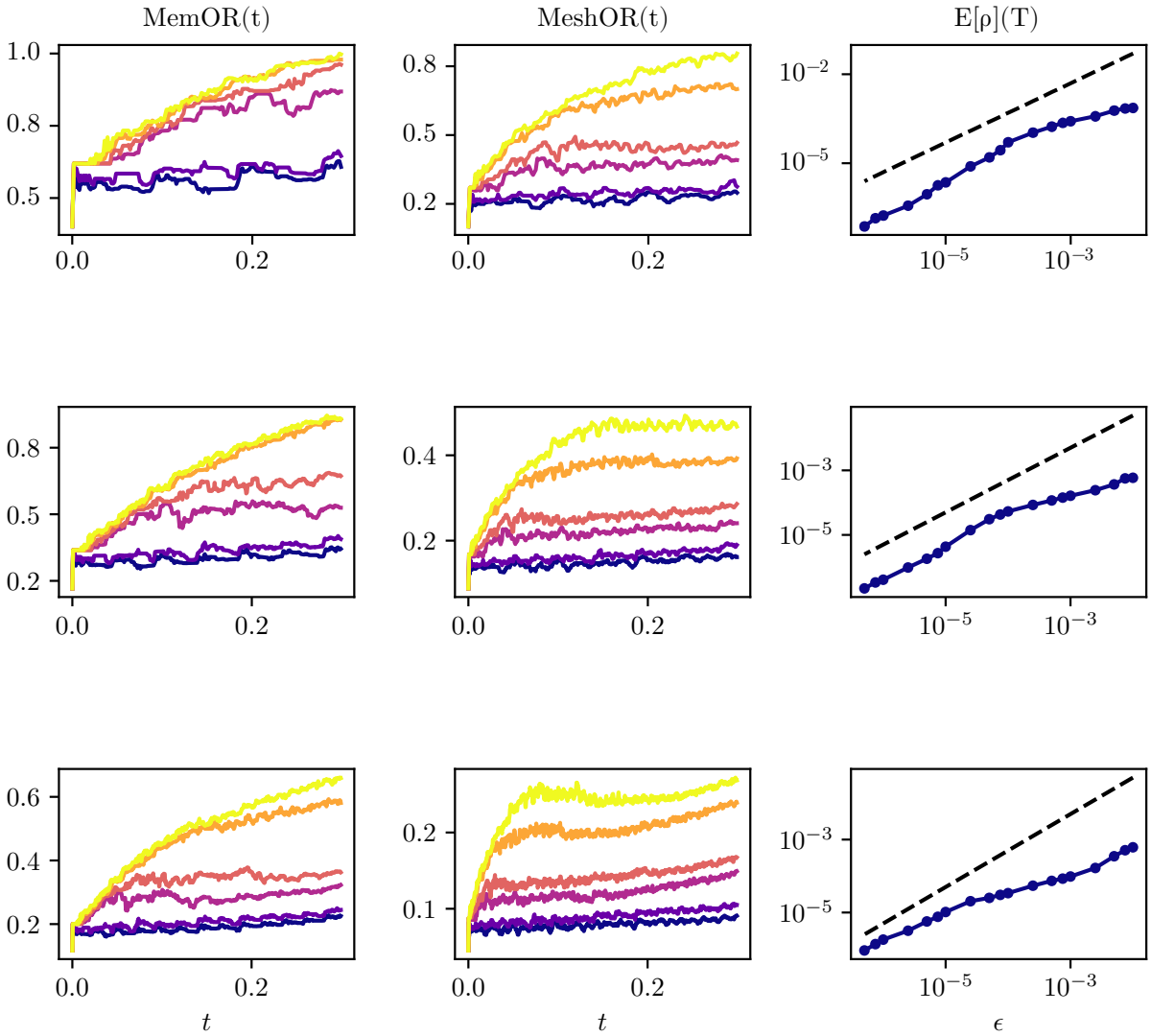

Figure 9: Configuration 3 with $\underline{J} = 2$ and $\overline{\mu} = 0$. From top to bottom $\overline{J} = 7, 8$ and 9. From left to right: memory occupation and mesh occupation as functions of time and additional error as function of ϵ. For the two plots on the left: ■ $\epsilon = 10^{-2}$, ■ $\epsilon = 5 \cdot 10^{-3}$, ■ $\epsilon = 10^{-3}$, ■ $\epsilon = 5 \times 10^{-4}$, ■ $\epsilon = 10^{-4}$, ■ $\epsilon = 5 \times 10^{-5}$. The dashed black line gives the slope ϵ.

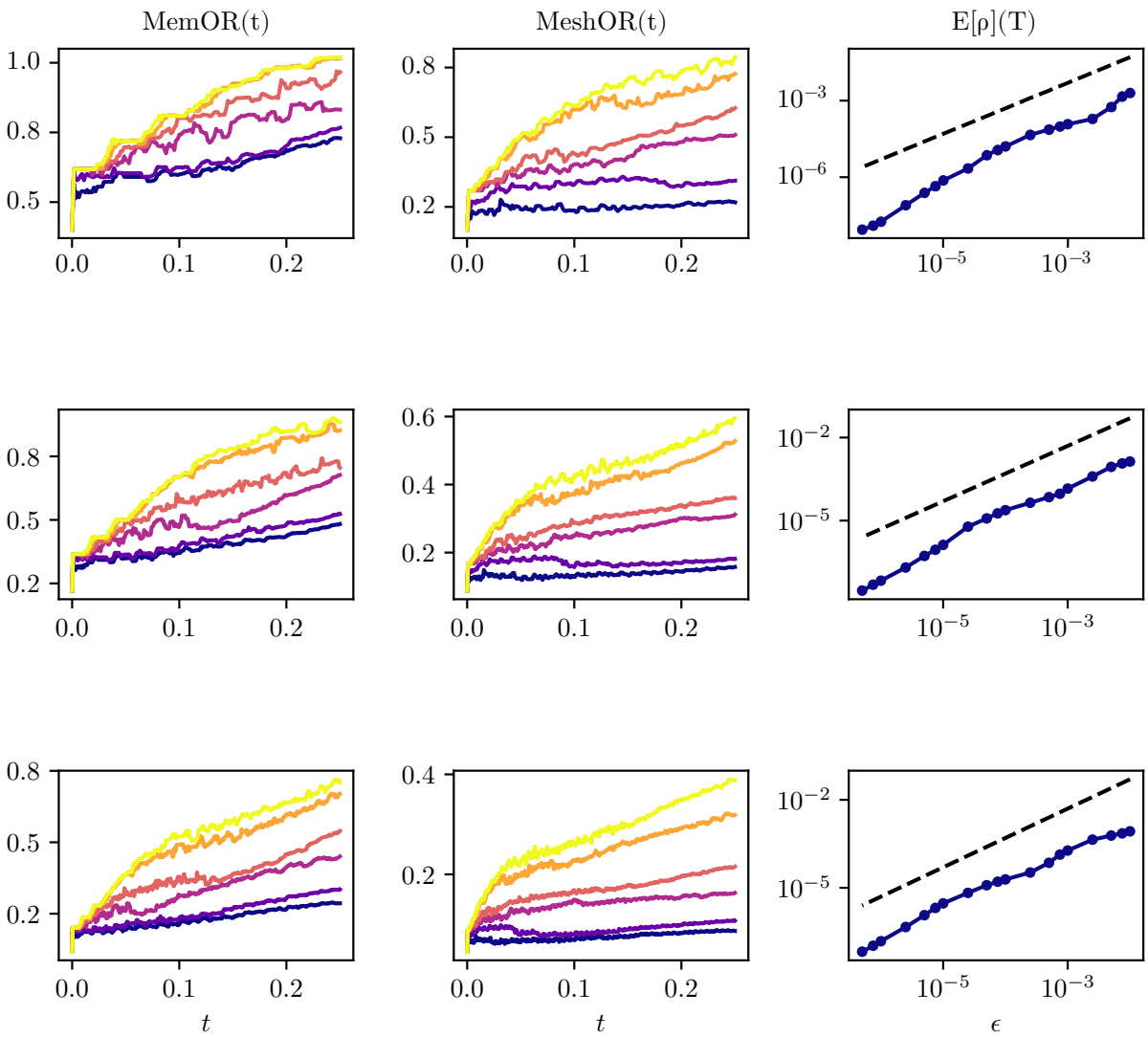

Figure 10: Configuration 12 with $\underline{J} = 2$ and $\overline{\mu} = 0$. From top to bottom $\overline{J} = 7, 8$ and 9. From left to right: memory occupation and mesh occupation as functions of time and additional error as function of ϵ. For the two plots on the left: ■ $\epsilon = 10^{-2}$, ■ $\epsilon = 5 \cdot 10^{-3}$, ■ $\epsilon = 10^{-3}$, ■ $\epsilon = 5 \times 10^{-4}$, ■ $\epsilon = 10^{-4}$, ■ $\epsilon = 5 \times 10^{-5}$. The dashed black line gives the slope ϵ.

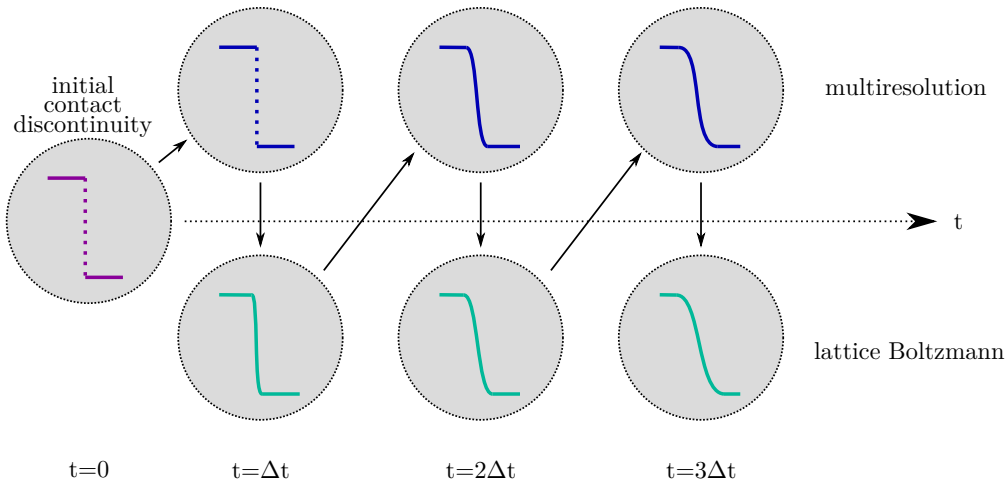

Figure 11: Coupling between the lattice Boltzmann scheme and the multiresolution to amplify the errors on the contact discontinuities.

hydrodynamic instabilities to appear. This is an interesting coupling phenomenon between the poor behavior of the reference scheme on the contact discontinuities – which is inherent to this class of vectorial schemes (see Graille [38]) – and the multiresolution. Indeed, the reference scheme smears the contact discontinuities (decreases the magnitude of the details in these areas) from the very beginning and then the multiresolution adaptation coarsens the mesh causing a local accumulation of error in time. This pattern is schematized in Figure 11.

We also verified that the fact of performing the collision on the complete leaves without reconstruction, see Section 5.2.1, has a negligible impact on this particular phenomenon, even if the equilibrium functions are strongly non-linear.

As Figure 8 and 7 show, this problem is clearly alleviated by decreasing the quality factor ϵ, from 0.005 to 0.001. Still, it is more interesting to study how the regularity guess $\overline{\mu}$ for the solution influences the behavior of the hybrid scheme with respect to this issue. To study this for Configuration 12, we fixed $\epsilon = 0.001$ and varied $\overline{\mu} = 0, 1$ and 2, as shown in Figure 12. We observe that this parameter, involved in the refinement process $\mathcal{H}_\epsilon$, does not affect the structures which are already well refined after the coarsening process $\mathcal{T}_\epsilon$, namely the shocks. Close to these structures, if we assume that (8) is sharp, the details do not decrease with j and thus precision is ensured if ϵ is reasonably small. On the other hand, we observe that a wise choice of $\overline{\mu}$ is effective in diminishing the coupling effect between the multiresolution and the lattice Boltzmann scheme in smearing contact discontinuities (consider that the color-scale is logarithmic), without having to drastically reduce ϵ causing a degradation of the performances of the algorithm. Since $\overline{\mu}$ represents the number of bounded derivatives of the expected solution of the problem, the advice for solutions developing shocks and contact discontinuities (thus only L^∞ solutions) is to set $\overline{\mu} = 0$. This has proved to allow for a reduction of the artificial smearing of the contact discontinuities by the numerical method.

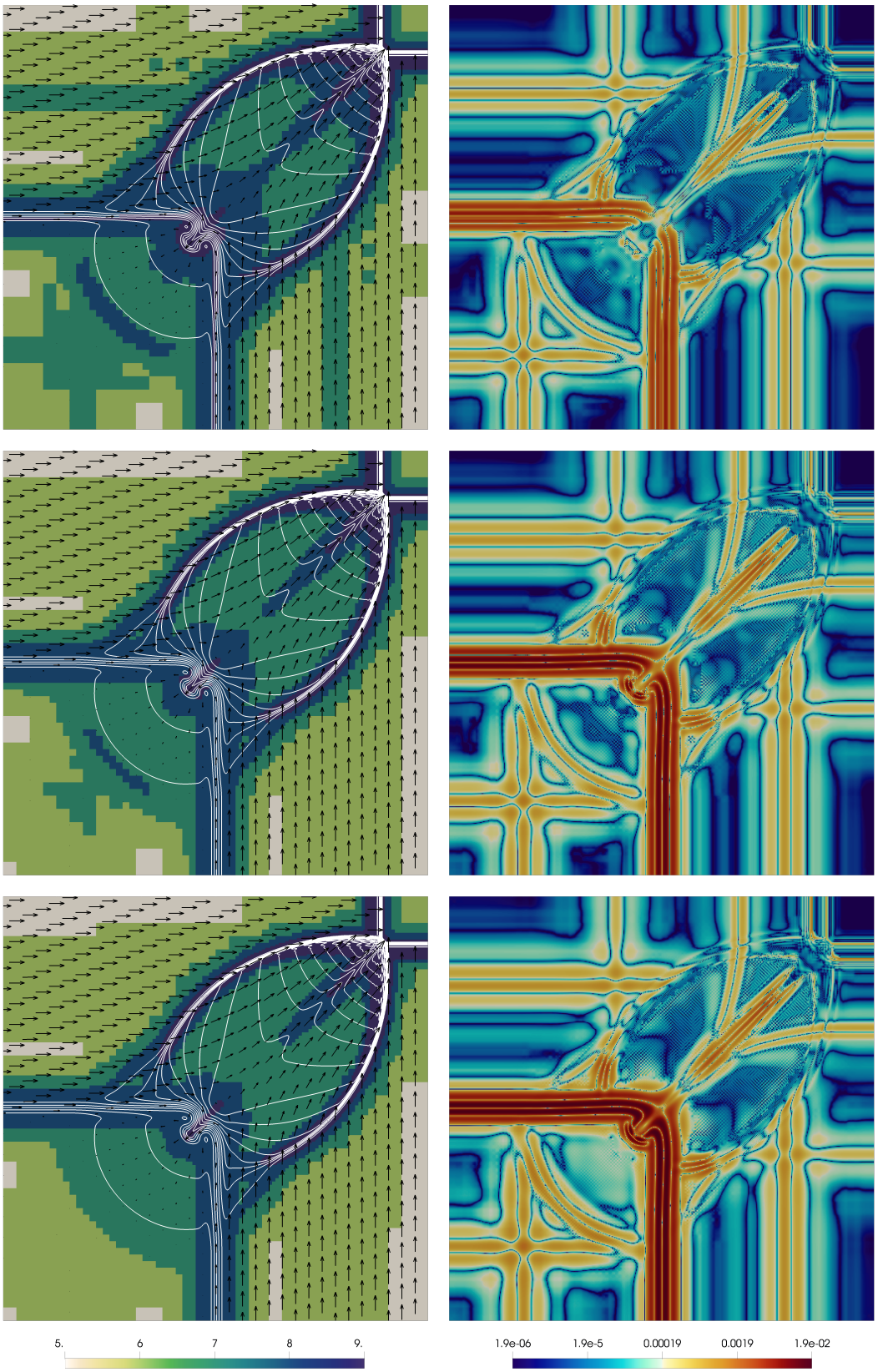

Figure 12: Configuration 12. For all the simulations, time $T = 0.25$, $\underline{J} = 2$ and $\overline{J} = 9$, $\epsilon = 0.001$. On the left, level j (colored), contours of the density field (white) and velocity field (black). On the right: local relative error of the adaptive method with respect to the reference method. From top to bottom $\overline{\mu} = 0, 1$ and 2.

7.2. Incompressible Navier-Stokes system

The simulation of the incompressible Navier-Stokes equations is probably the most well-known application of the lattice Boltzmann schemes. Though the study of this problem with lattice Boltzmann schemes is limited to low Mach and Reynolds numbers, thus limiting the interest of multiresolution, we still perform this test to showcase the applicability of our strategy to this class of problems.

7.2.1. Problem

We consider the problem of a flow around an obstacle occupying the open set $\Theta \subset \mathbb{R}^2$, with flow supposed to be incompressible and considering a Newtonian fluid

$$\begin{cases} \nabla \cdot \boldsymbol{u} = 0, & t \geq 0, \quad \boldsymbol{x} \in \mathbb{R}^2 \setminus \Theta, \\ \rho_0 \left(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) = -\nabla p + \nabla \cdot \left(2\mu_{\text{visc}} \dfrac{\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T}{2}\right), & t \geq 0, \quad \boldsymbol{x} \in \mathbb{R}^2 \setminus \Theta, \\ \boldsymbol{u} = \boldsymbol{u}_0 = (u_0, 0), & t = 0, \quad \boldsymbol{x} \in \mathbb{R}^2 \setminus \Theta, \\ \boldsymbol{u} = 0, & t \geq 0, \quad \boldsymbol{x} \in \partial\Theta. \end{cases} \tag{23}$$

The leading dimensionless quantity in this problem, determining the flow regime, is the Reynolds number

$$\text{Re} = \frac{\rho_0 u_0 L_\Theta}{\mu_{\text{visc}}},$$

given the characteristic length of the obstacle $L_\Theta \sim (|\Theta|_2)^{1/2}$. It is well-known that for $\text{Re} > 90$, the flow passes from being fully laminar to a periodic regime where vortices periodically shed. This phenomenon is known as von Kármán vortex street.

7.2.2. Numerical scheme

The numerical scheme we test is the popular D2Q9 scheme by Lallemand and Luo [45][12], with $q = 9$ velocities given by

$$\boldsymbol{\xi}^h = \begin{cases} (0,0)^T, & h = 0, \\ \lambda \left(\cos\left(\dfrac{\pi}{2}(h-1)\right), \sin\left(\dfrac{\pi}{2}(h-1)\right)\right)^T, & h = 1,2,3,4, \\ \lambda \left(\cos\left(\dfrac{\pi}{2}(h-5) + \dfrac{\pi}{4}\right), \sin\left(\dfrac{\pi}{2}(h-5) + \dfrac{\pi}{4}\right)\right)^T, & h = 5,6,7,8, \end{cases}$$

with relaxation $\boldsymbol{S} = \text{diag}\,(0,0,0,s_1,s_1,s_1,s_1,s_2,s_2)$ and change of basis given by

$$\boldsymbol{M} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & \lambda & 0 & -\lambda & 0 & \lambda & -\lambda & -\lambda & \lambda \\ 0 & 0 & \lambda & 0 & -\lambda & \lambda & \lambda & -\lambda & -\lambda \\ -4\lambda^2 & -\lambda^2 & -\lambda^2 & -\lambda^2 & -\lambda^2 & 2\lambda^2 & 2\lambda^2 & 2\lambda^2 & 2\lambda^2 \\ 0 & -2\lambda^3 & 0 & 2\lambda^3 & 0 & \lambda^3 & -\lambda^3 & -\lambda^3 & \lambda^3 \\ 0 & 0 & -2\lambda^3 & 0 & 2\lambda^3 & \lambda^3 & \lambda^3 & -\lambda^3 & -\lambda^3 \\ 4\lambda^4 & -2\lambda^4 & -2\lambda^4 & -2\lambda^4 & -2\lambda^4 & \lambda^4 & \lambda^4 & \lambda^4 & \lambda^4 \\ 0 & \lambda^2 & -\lambda^2 & \lambda^2 & -\lambda^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda^2 & -\lambda^2 & \lambda^2 & -\lambda^2 \end{pmatrix}.$$

The moments at the equilibrium are, indicating $c_s = \lambda/\sqrt{3}$, $u_x = q_x/\rho$, $u_y = q_y/\rho$ and $|\boldsymbol{u}|^2 = u_x^2 + u_y^2$

$$\boldsymbol{m}^{\text{eq}} = \left(\rho, q_x, q_y, 3(-2c_s^2\rho + |\boldsymbol{q}|/\rho), -3c_s^2 q_x, -3c_s^2 q_y, 9(c_s^4 - c_s^2|\boldsymbol{q}|/\rho), (q_x^2 - q_y^2)/\rho, q_x q_y/\rho\right)^T.$$

[12] Many other schemes are proposed in the literature and we tested some of them with similar results.

Assuming to be in the low-Mach setting, namely $|\boldsymbol{u}_0|/c_s \ll 1$, it can be shown [35] that we obtain a quasi-incompressible regime ($\rho \simeq \rho_0$) and that the equivalent equations are, neglecting parasitic terms proportional to the cube of the velocity

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = O(\Delta t^2), \\ \partial_t (\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = -\nabla (c_s^2 \rho) + c_s^2 \Delta t \nabla \cdot \left(2\rho\sigma_2 \left(\dfrac{\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T}{2} \right) + \rho\sigma_1 (\nabla \cdot \boldsymbol{u}) \boldsymbol{I} \right) + O(\Delta t^2), \end{cases}$$

with $\sigma_k := \left(\dfrac{1}{s_k} - \dfrac{1}{2} \right)$. Thus, to enforce the diffusivity μ_{visc}, one has to set s_2

$$s_2 = \left(\frac{1}{2} + \frac{\mu_{\text{visc}}}{c_s^2 \Delta t \rho_0} \right)^{-1}.$$

Special care must be devoted to the representation of the obstacle Θ. Usually, one employs bounce back boundary condition to enforce zero velocity on $\partial\Theta$ and there exist a vast specialized literature on this matter. However, in this work, we do not want to adapt the domain and its discretization to fit $\partial\Theta$ and so we proceed in the following way: we mesh the entire space $\Omega = [0,2] \times [0,1]$ without considering the obstacle and we apply the adaptive scheme as always. At the end of each time step, for the leaves $C_{j,\boldsymbol{k}}$ intersecting Θ, we estimate $|C_{j,\boldsymbol{k}} \cap \Theta|_d$ and we set

$$\boldsymbol{f}_{j,\boldsymbol{k}}(t+\Delta t) = \frac{|C_{j,\boldsymbol{k}} \cap \Theta|_d}{|C_{j,\boldsymbol{k}}|_d} \boldsymbol{M}^{-1} \left(\boldsymbol{m}^{\text{eq}}|_{\substack{\rho=\rho_0 \\ u_x=0 \\ u_y=0}} \right) + \left(1 - \frac{|C_{j,\boldsymbol{k}} \cap \Theta|_d}{|C_{j,\boldsymbol{k}}|_d} \right) \boldsymbol{f}_{j,\boldsymbol{k}}(t+\Delta t),$$

following the direction of Mohamad and Succi [54]. For the external boundaries, we impose a bounce back boundary condition with the given velocity $\boldsymbol{u}_0$ on the left, top and bottom boundary and a copy boundary condition on the outlet.

7.2.3. Points of emphasis

Due to the important role of the viscosity in the whole domain and not only where the mesh is more refined, we cannot compare the solution at each time frame because the von Kármán instability is going to develop differently (see comments). Thus, we analyze some integral quantities [30] like the drag coefficient C_D, the lift coefficient C_L and the Strouhal number, given by

$$C_D = \frac{2F_x}{\rho_0 u_0^2 L_\Theta}, \qquad C_L = \frac{2F_y}{\rho_0 u_0^2 L_\Theta}, \qquad \text{St} = \frac{L_\Theta \omega}{u_0},$$

where $\boldsymbol{F} = (F_x, F_y)$ is the force acting on the obstacle and ω is the shedding frequency of the vortices. In particular, the shedding frequency is computed using the fast Fourier transform on the available lift coefficient. We are interested in comparing these dimensionless quantities between the reference method on a uniform mesh and the adaptive method on the evolving adaptive mesh, as well as the occupation rates introduced in the previous Section.

7.2.4. Results and discussion

We test the method for $\text{Re} = 1200$ (we also tested it for other moderate Reynolds numbers, obtaining analogous results) with $\lambda = 1$, fixing $\underline{J} = 2$ and using $\overline{J} = 7, 8$ and 9 with $\epsilon = 0.00075, 0.000375$ and 0.000175 respectively and selecting $\overline{\mu} = 1$. The reason why we have chosen to decrease ϵ as $\overline{J}$ increases shall be clear in a moment.

On Figure 13 we observe an excellent agreement on the value of the integral quantities between the adaptive method and the reference method. The discrepancies are obviously reduced as $\overline{J}$ increases because of the variation of the respective threshold parameter ϵ. This comes from the fact that the most important contribution to these quantities comes from the area around the obstacle Θ, where the flow regime can be considered to be, to some extent, highly inertial (or hyperbolic). For this kind of regime, previous works

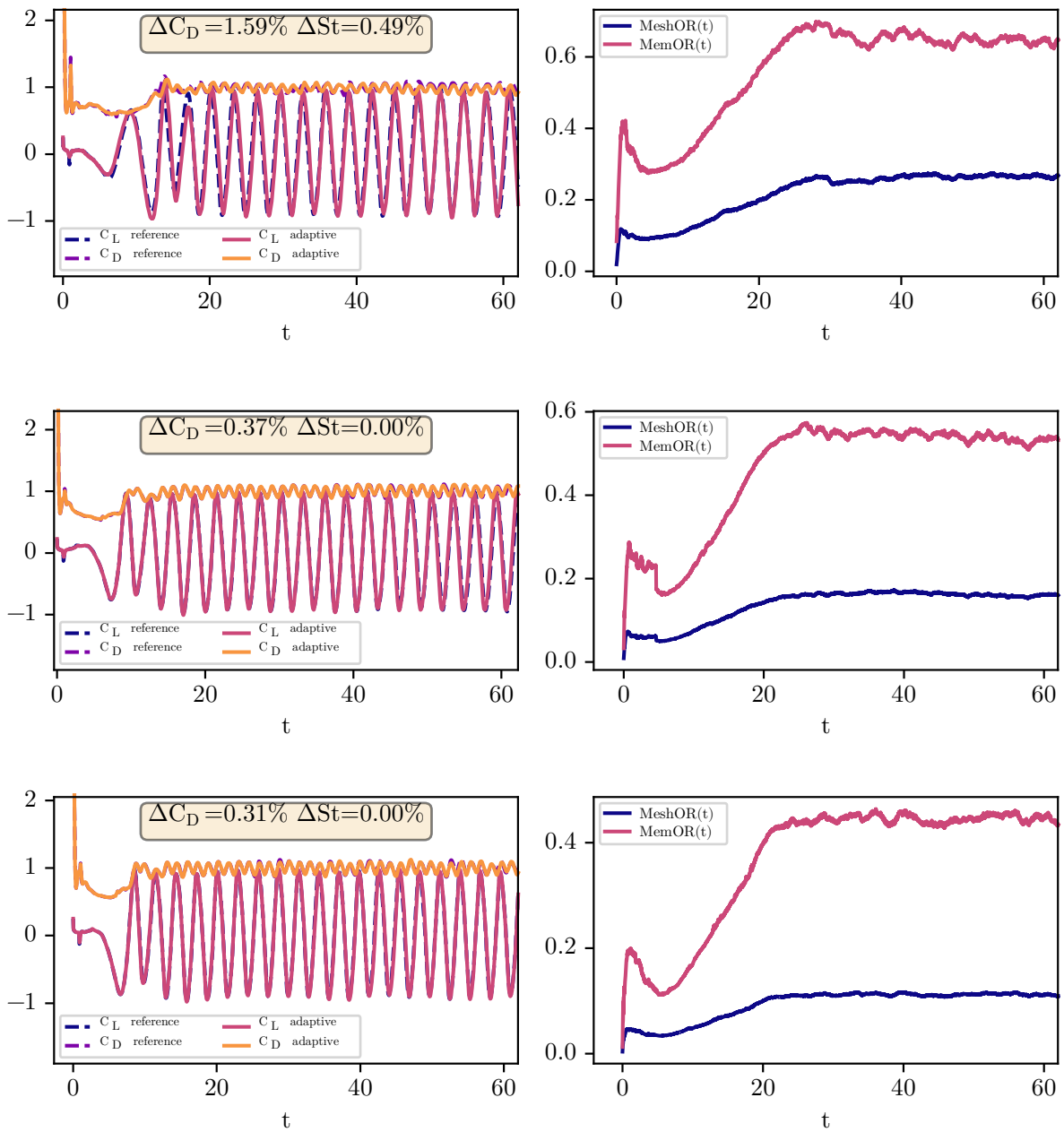

Figure 13: On the left, time behavior of the drag and lift coefficient for the reference and the adaptive scheme and on the right, mesh and memory occupation rates, for Re $= 1200$, $\underline{J} = 2$ and $\overline{\mu} = 1$. From top to bottom: $(\overline{J}, \epsilon) = (7, 0.00075)$, $(8, 0.000375)$ and $(9, 0.000175)$. The drag coefficient has been normalized with its average for the reference scheme and the lift coefficient with the maximum for the reference scheme.

[10] and [3] and the previous Section have shown that the Harten heuristics is respected with our choice of $\mathcal{T}_\epsilon$ and $\mathcal{H}_\epsilon$. The occupation rates are interesting and become better and better with $\bar{J}$ as we also observed for the solution of the Euler system, despite the fact that we reduced the threshold parameter ϵ as we increased $\bar{J}$. The initial growth of these rates followed by a decrease is due to some initial acoustic waves quickly propagating radially in the domain which are eventually damped by the external boundary conditions and are inherent to the lattice Boltzmann method and its way of treating boundary conditions. These waves do not affect computations on a longer time scale. The values of the occupation rates stabilize after the complete onset of the instability followed by the periodic regime, that is for $t > 20$, because the number of vortices present in the domain is roughly the same for every time.

On the opposite side, far from the obstacle and close to the outlet, the regime can be considered to be mostly diffusive (or parabolic) and here the Harten heuristics could be violated. The purpose behind the division of the parameter ϵ by two at each time we increased the maximum level $\bar{J}$ was to try to follow the vortices until they reach the outlet with the finest possible resolution. As shown in Figures 14, 15 and 16, this attempt of following the vortices until the outlet with the finest resolution $\bar{J}$ was only partially successful. More investigations are needed to clarify the role of the "copy" boundary conditions coupled with the multiresolution procedure. This is not surprising by looking at (8) and by concluding that the exact solution of the Navier-Stokes equations for this Reynolds number must have more than only one bounded derivative (*i.e.* more than just bounded vorticity), because the details scale of a factor larger than two at each change of level. This is related to the nature of the solution and we refer to [57] for a detailed study of the use of multiresolution in order to resolve the incompressible Navier-Stokes equations. It is also worthwile mentioning that the D2Q9 scheme for the Navier-Stokes system under acoustic scaling $\Delta t \sim \Delta x$ is not converging for $\Delta x \to 0$: therefore one should be really careful once comparing the results for different $\bar{J}$, as we did.

To conclude, this test shows that our method is effective when applied to parabolic problems solved with the lattice Boltzmann method. This reflects on the excellent results concerning the integral quantities. Current investigations being conducted by our team using formal expansions [25] show that our adaptive method does not modify (for $\gamma \geq 1$) the viscosity of the flow, thus confirming once more that it is suitable to simulate the Navier-Stokes equations. Furthermore, the method also preserves higher order terms and thus is less likely to modify the stability properties of the reference scheme.

7.3. Three dimensional advection equation

7.3.1. Problem

The only test we show for $d = 3$ aims at solving the transport equation with constant velocity $\boldsymbol{V} = (V_x, V_y, V_z)^T$ given by

$$\begin{cases} \partial_t u + \partial_x(V_x u) + \partial_y(V_y u) + \partial_z(V_z u) = 0, & t \in [0,T], \quad \boldsymbol{x} \in \mathbb{R}^3, \\ u(t=0,\boldsymbol{x}) = \chi_{\sqrt{x^2+y^2+z^2} \leq 0.15}(\boldsymbol{x}), & \boldsymbol{x} \in \mathbb{R}^3. \end{cases}$$

7.3.2. Numerical scheme

We employ a D3Q6 scheme, thus $q = 6$, with one conserved moment with choice of velocities

$$\boldsymbol{\xi}^h = \begin{cases} \lambda\left(\cos\left(\pi h\right), 0, 0\right)^T, & h = 0,1, \\ \lambda\left(0, \cos\left(\pi h\right), 0\right)^T, & h = 2,3, \\ \lambda\left(0, 0, \cos\left(\pi h\right)\right)^T, & h = 4,5, \end{cases}$$

with relaxation matrix $\boldsymbol{S} = \mathrm{diag}(0, s_1, s_1, s_1, s_2, s_2)$ and change of basis given by

$$\boldsymbol{M} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \lambda & -\lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & -\lambda \\ \lambda^2 & \lambda^2 & -\lambda^2 & -\lambda^2 & 0 & 0 \\ \lambda^2 & \lambda^2 & 0 & 0 & -\lambda^2 & -\lambda^2 \end{pmatrix}.$$

The vector of the equilibria is $\boldsymbol{m}^{\mathrm{eq}} = (u, V_x u, V_y u, V_z u, 0, 0)^T$. For this problem, the relaxation parameters are $s_1 = 1.4$ and $s_2 = 1$. We take $\lambda = 1$.

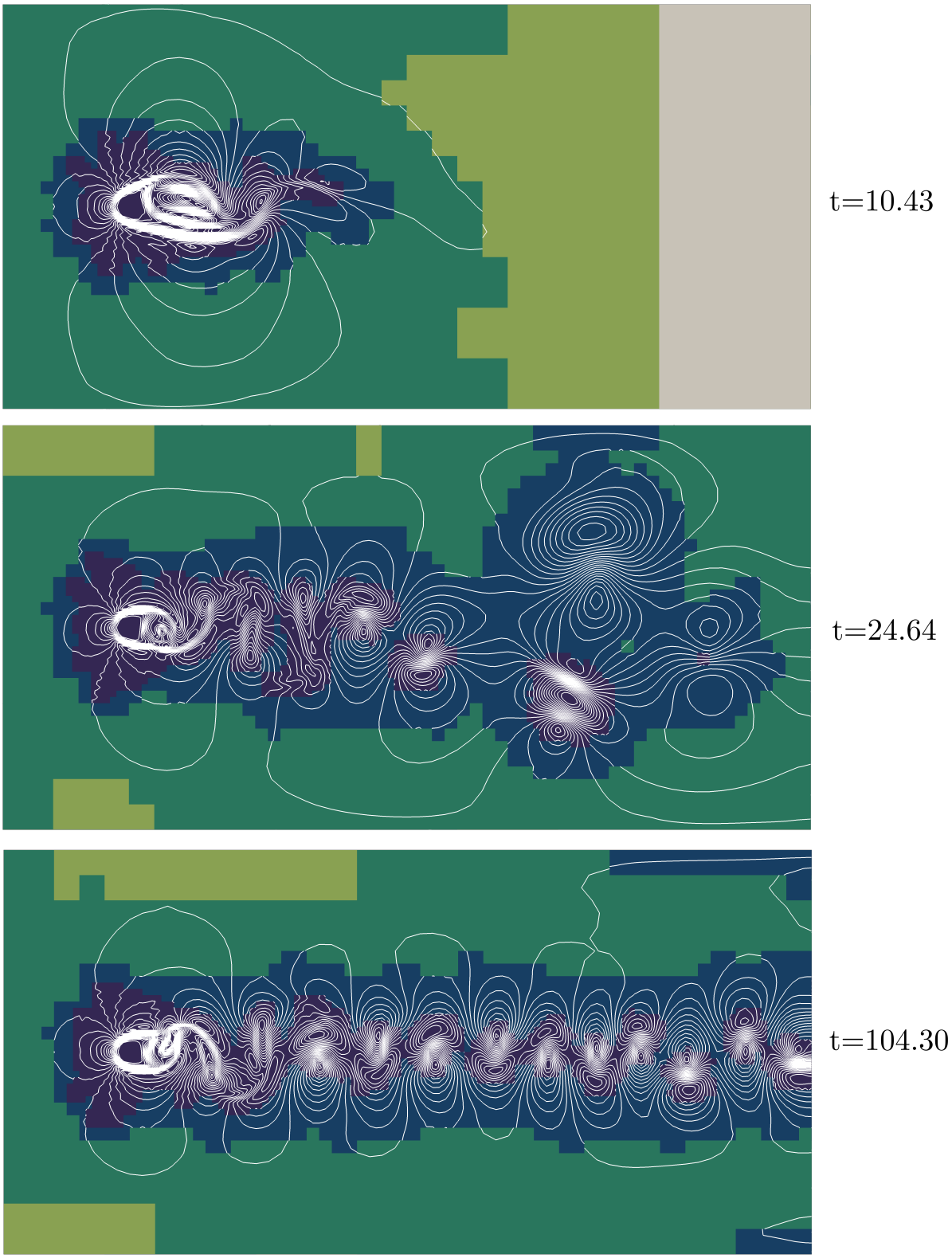

Figure 14: Snapshots of the solution of the adaptive scheme for Re $= 1200$, $\underline{J} = 2$, $\overline{J} = 7$, $\overline{\mu} = 1$ and $\epsilon = 0.00075$. The colors represent the levels of the mesh and the white contours are that of the velocity modulus.

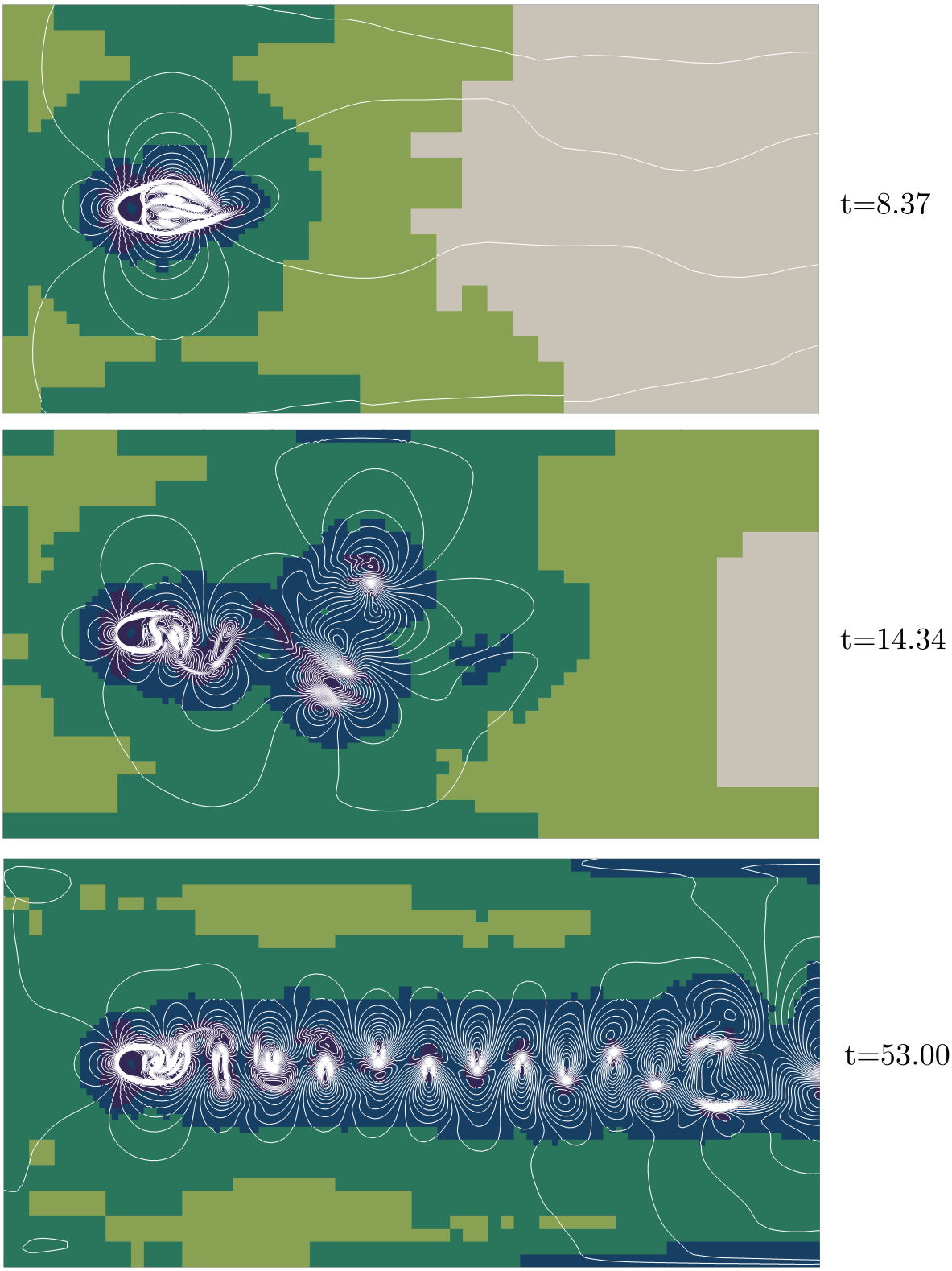

Figure 15: Snapshots of the solution of the adaptive scheme for Re $= 1200$, $\underline{J} = 2$, $\overline{J} = 8$, $\overline{\mu} = 1$ and $\epsilon = 0.000375$. The colors represent the levels of the mesh and the white contours are that of the velocity modulus.

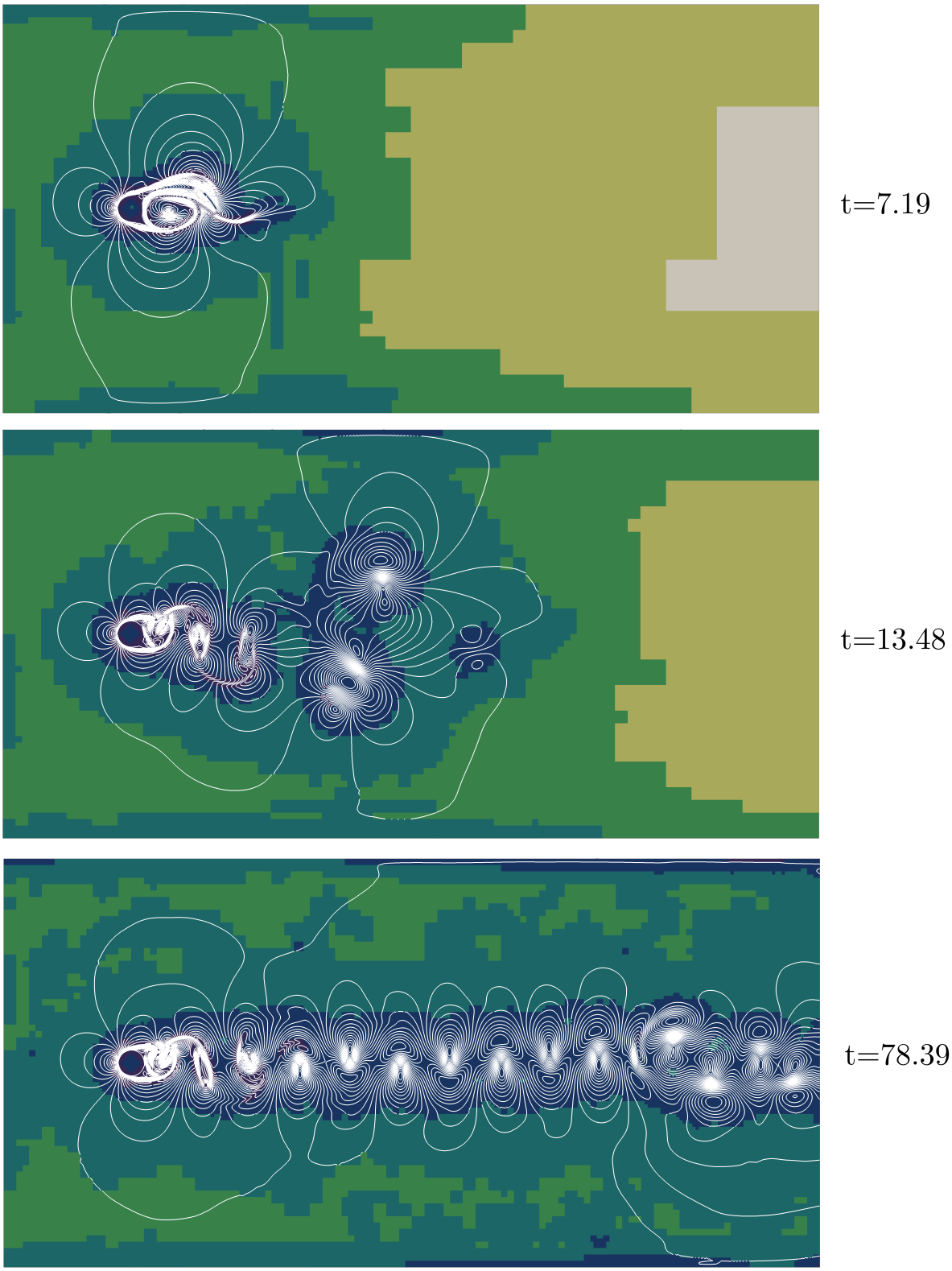

Figure 16: Snapshots of the solution of the adaptive scheme for Re $= 1200$, $\underline{J} = 2$, $\overline{J} = 9$, $\overline{\mu} = 1$ and $\epsilon = 0.0001875$. The colors represent the levels of the mesh and the white contours are that of the velocity modulus.

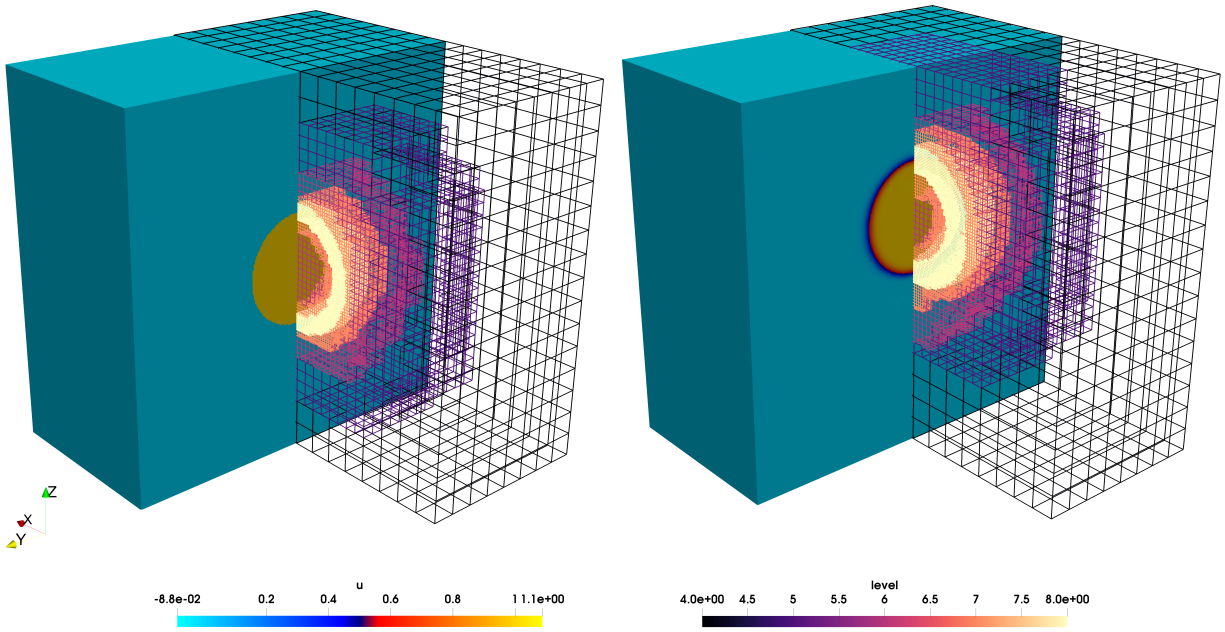

Figure 17: Snapshots at times $t = 0$ (left) and $t = 0.3125$ (right) of the solution of the adaptive scheme for the 3D advection equation. The domain has been cut so that the left half of each snapshots shows the value of the conserved variable u whereas the right half represents the structure of the mesh with the corresponding local level of resolution.

7.3.3. Results and discussion

We carry on the simulation until $T = 0.78125$ with $\underline{J} = 1$, $\overline{J} = 8$, $\overline{\mu} = 2$ and $\epsilon = 0.001$. Some snapshots of the solutions and the mesh at different time steps are provided in Figure 17: one can appreciate the significantly hollow mesh produced by multiresolution. The time behavior of the occupation rates is presented in Figure 18, corresponding to an excellent compression rate of around 97%. This shows that our method is capable of coping with this three dimensional problem properly and to achieve really interesting occupation rates which are crucial to conduct large simulations for 3D problems. Indeed, the full problem would have needed $256^3 \sim 16$ millions cells, which is quite demanding for a sequential code. Thanks to multiresolution, we have made the problem easily treatable with at most $0.03 \times 256^3 \sim 50000$ cells involved (for all sizes). This means that the adaptive mesh occupies 3% of its original size, which is quite impressive and shows the interest of the method to be used in 3D problems.

8. Conclusions and future developments

In this paper, we have introduced a novel way of performing mesh adaptation on multidimensional lattice Boltzmann schemes in order to reduce their memory footprint particularly on problems with steep fronts. This approach, based on adaptive multiresolution, modifies the mesh according to the local regularity of the solution and adapts the advection phase to reduce the total number of operations, while doing the collision only on the leaves of the adaptive tree. Numerical experiments have shown that this technique can be combined with a wide range of existing schemes, both to simulate non-linear hyperbolic and non-linear parabolic systems. Therefore, the technique is problem independent and provides an effective cost reduction still being able to control the quality of the solution by means of a single threshold parameter ϵ.

Several questions remain open and give room for improvement in future research works. First, one has to reach a real performance enhancement coming from dedicated forms of data storage and the parallelization of the code on various architectures. Second, it would be interesting to investigate how to handle complex geometries, which are common in real problems, possibly using the numerous high-order boundary conditions present in the literature. Detailed comparisons with the existing AMR approaches, even if always a difficult task, could also be conducted with the tools we have developed. Finally, we are

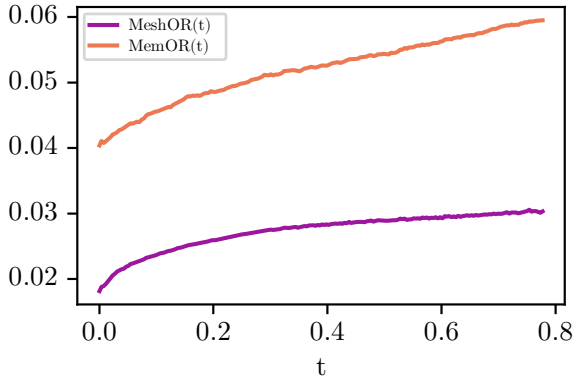

Figure 18: Temporal variation of the occupation rates for the adaptive scheme simulating the 3D advection equation.

currently investigating the possibility of constructing a more reliable collision phase following the lines by Hovhannisyan and Müller [43].

Acknowledgments

The authors deeply thank Laurent Séries and Christian Tenaud for fruitful discussions on multiresolution and the referee for having provided valuable inputs and suggestions to improve this work. Thomas Bellotti is supported by a PhD funding (year 2019) from the Ecole polytechnique.

References

[1] Aregba-Driollet, D., and Natalini, R. Discrete kinetic schemes for multidimensional systems of conservation laws. *SIAM Journal on Numerical Analysis 37*, 6 (2000), 1973–2004.

[2] Bartoloni, A., Battista, C., Cabasino, S., Paolucci, P., Pech, J., Sarno, R., Todesco, G., Torelli, M., Tross, W., Vicini, P., et al. Lbe simulations of rayleigh-benard convection on the ape100 parallel processor. *International Journal of Modern Physics C 4*, 05 (1993), 993–1006.

[3] Bellotti, T., Gouarin, L., Graille, B., and Massot, M. Multiresolution-based mesh adaptation and error control for lattice boltzmann methods with applications to hyperbolic conservation laws. *SIAM J. Scientific Computing* (2021). Submitted - available on HAL : https://hal.archives-ouvertes.fr/hal-03148621 and ArXiv : https://arxiv.org/abs/2102.12163.

[4] Berger, M. J., Colella, P., et al. Local adaptive mesh refinement for shock hydrodynamics. *Journal of computational Physics 82*, 1 (1989), 64–84.

[5] Bihari, B. L., and Harten, A. Multiresolution schemes for the numerical solution of 2-d conservation laws i. *SIAM Journal on Scientific Computing 18*, 2 (1997), 315–354.

[6] Bouchut, F. Construction of bgk models with a family of kinetic entropies for a given system of conservation laws. *Journal of Statistical Physics 95*, 1 (1999), 113–170.

[7] Bramkamp, F., Lamby, P., and Müller, S. An adaptive multiscale finite volume solver for unsteady and steady state flow computations. *Journal of Computational Physics 197*, 2 (2004), 460–490.

[8] Bürger, R., Ruiz, R., Schneider, K., and Sepúlveda, M. A. Fully adaptive multiresolution schemes for strongly degenerate parabolic equations with discontinuous flux. *Journal of Engineering Mathematics 60*, 3-4 (2008), 365–385.

[9] CAETANO, F., DUBOIS, F., AND GRAILLE, B. A result of convergence for a mono-dimensional two-velocities lattice boltzmann scheme. *arXiv preprint arXiv:1905.12393* (2019).

[10] COHEN, A., KABER, S., MÜLLER, S., AND POSTEL, M. Fully adaptive multiresolution finite volume schemes for conservation laws. *Mathematics of Computation 72*, 241 (2003), 183–225.

[11] COLELLA, P. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics 87*, 1 (1990), 171–200.

[12] COQUEL, F., NGUYEN, Q. L., POSTEL, M., AND TRAN, Q. H. Local time stepping applied to implicit-explicit methods for hyperbolic systems. *Multiscale Modeling & Simulation 8*, 2 (2010), 540–570.

[13] COQUEL, F., POSTEL, M., POUSSINEAU, N., AND TRAN, Q.-H. Multiresolution technique and explicit–implicit scheme for multicomponent flows. *Journal of Numerical Mathematics 14*, 3 (2006), 187–216.

[14] CROUSE, B., RANK, E., KRAFCZYK, M., AND TÖLKE, J. A lb-based approach for adaptive flow simulations. *International Journal of Modern Physics B 17*, 01n02 (2003), 109–112.

[15] DAUBECHIES, I. Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics 41*, 7 (1988), 909–996.

[16] DEITERDING, R., DOMINGUES, M. O., GOMES, S. M., AND SCHNEIDER, K. Comparison of adaptive multiresolution and adaptive mesh refinement applied to simulations of the compressible euler equations. *SIAM Journal on Scientific Computing 38*, 5 (2016), S173–S193.

[17] DELLAR, P. J. An interpretation and derivation of the lattice boltzmann method using strang splitting. *Computers & Mathematics with Applications 65*, 2 (2013), 129–141.

[18] DESCOMBES, S., DUARTE, M., DUMONT, T., GUILLET, T., LOUVET, V., AND MASSOT, M. Task-based adaptive multiresolution for time-space multi-scale reaction-diffusion systems on multi-core architectures. *SMAI J. Comput. Math. 3* (2017), 29–51.

[19] DESCOMBES, S., DUARTE, M., DUMONT, T., LAURENT, F., LOUVET, V., AND MASSOT, M. Analysis of operator splitting in the nonasymptotic regime for nonlinear reaction-diffusion equations. Application to the dynamics of premixed flames. *SIAM J. Numer. Anal. 52*, 3 (2014), 1311–1334.

[20] D'HUMIERES, D. Multiple–relaxation–time lattice boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 360*, 1792 (2002), 437–451.

[21] DUARTE, M. *Adaptive numerical methods in time and space for the simulation of multi-scale reaction fronts*. PhD thesis, Ecole Centrale Paris, 2011. https://tel.archives-ouvertes.fr/tel-00667857.

[22] DUARTE, M., BONAVENTURA, Z., MASSOT, M., AND BOURDON, A. A numerical strategy to discretize and solve the Poisson equation on dynamically adapted multiresolution grids for time-dependent streamer discharge simulations. *J. Comput. Phys. 289* (2015), 129–148.

[23] DUARTE, M., DESCOMBES, S., TENAUD, C., CANDEL, S., AND MASSOT, M. Time–space adaptive numerical methods for the simulation of combustion fronts. *Combustion and Flame 160*, 6 (2013), 1083–1101.

[24] DUARTE, M., MASSOT, M., DESCOMBES, S., TENAUD, C., DUMONT, T., LOUVET, V., AND LAURENT, F. New resolution strategy for multiscale reaction waves using time operator splitting, space adaptive multiresolution, and dedicated high order implicit/explicit time integrators. *SIAM Journal on Scientific Computing 34*, 1 (2012), A76–A104.

[25] DUBOIS, F. Third order equivalent equation of lattice boltzmann scheme. *Discrete & Continuous Dynamical Systems-A 23*, 1&2 (2009), 221.

[26] DUBOIS, F. Simulation of strong nonlinear waves with vectorial lattice boltzmann schemes. *International Journal of Modern Physics C 25*, 12 (2014), 1441014.

[27] DUBOIS, F., AND LALLEMAND, P. On lattice boltzmann scheme, finite volumes and boundary conditions. *Progress in Computational Fluid Dynamics, an International Journal 8*, 1-4 (2008), 11–24.

[28] DUMONT, T., DUARTE, M., DESCOMBES, S., DRONNE, M.-A., MASSOT, M., AND LOUVET, V. Simulation of human ischemic stroke in realistic 3D geometry. *Commun. Nonlinear Sci. Numer. Simul. 18*, 6 (2013), 1539–1557.

[29] DUPUIS, A., AND CHOPARD, B. Theory and applications of an alternative lattice boltzmann grid refinement algorithm. *Physical Review E 67*, 6 (2003), 066707.

[30] EITEL-AMOR, G., MEINKE, M., AND SCHRÖDER, W. A lattice-boltzmann method with hierarchically refined meshes. *Computers & Fluids 75* (2013), 127–139.

[31] FAKHARI, A., GEIER, M., AND LEE, T. A mass-conserving lattice boltzmann method with dynamic grid refinement for immiscible two-phase flows. *Journal of Computational Physics 315* (2016), 434–457.

[32] FAKHARI, A., AND LEE, T. Finite-difference lattice boltzmann method with a block-structured adaptive-mesh-refinement technique. *Physical Review E 89*, 3 (2014), 033310.

[33] FAKHARI, A., AND LEE, T. Numerics of the lattice boltzmann method on nonuniform grids: standard lbm and finite-difference lbm. *Computers & Fluids 107* (2015), 205–213.

[34] FENG, Y., GUO, S., JACOB, J., AND SAGAUT, P. Grid refinement in the three-dimensional hybrid recursive regularized lattice boltzmann method for compressible aerodynamics. *Phys. Rev. E 101* (Jun 2020), 063302.

[35] FÉVRIER, T. *Extension et analyse des schémas de Boltzmann sur réseau: les schémas à vitesse relative*. PhD thesis, 2014.

[36] FILIPPOVA, O., AND HÄNEL, D. Grid refinement for lattice-bgk models. *Journal of Computational physics 147*, 1 (1998), 219–228.

[37] GENDRE, F., RICOT, D., FRITZ, G., AND SAGAUT, P. Grid refinement for aeroacoustics in the lattice boltzmann method: A directional splitting approach. *Phys. Rev. E 96* (Aug 2017), 023311.

[38] GRAILLE, B. Approximation of mono-dimensional hyperbolic systems: A lattice boltzmann scheme as a relaxation method. *Journal of Computational Physics 266* (2014), 74–88.

[39] HARTEN, A. Adaptive multiresolution schemes for shock computations. *Journal of Computational Physics 115*, 2 (1994), 319–338.

[40] HARTEN, A. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Communications on Pure and Applied Mathematics 48*, 12 (1995), 1305–1342.

[41] HE, X., CHEN, S., AND DOOLEN, G. D. A novel thermal model for the lattice boltzmann method in incompressible limit. *Journal of computational physics 146*, 1 (1998), 282–300.

[42] HORSTMANN, J. *Hybrid numerical method based on the lattice Boltzmann approach with application to non-uniform grids*. PhD thesis, Université de Lyon, 2018.

[43] HOVHANNISYAN, N., AND MÜLLER, S. On the stability of fully adaptive multiscale schemes for conservation laws using approximate flux and source reconstruction strategies. *IMA journal of numerical analysis 30*, 4 (2010), 1256–1295.

[44] KANDHAI, D., SOLL, W., CHEN, S., HOEKSTRA, A., AND SLOOT, P. Finite-difference lattice-bgk methods on nested grids. *Computer Physics Communications 129*, 1-3 (2000), 100–109.

[45] LALLEMAND, P., AND LUO, L.-S. Theory of the lattice boltzmann method: Dispersion, dissipation, isotropy, galilean invariance, and stability. *Physical Review E 61*, 6 (2000), 6546.

[46] LAMBY, P., MÜLLER, S., AND STIRIBA, Y. Solution of shallow water equations using fully adaptive multiscale schemes. *International journal for numerical methods in fluids 49*, 4 (2005), 417–437.

[47] LAX, P. D., AND LIU, X.-D. Solution of two-dimensional riemann problems of gas dynamics by positive schemes. *SIAM Journal on Scientific Computing 19*, 2 (1998), 319–340.

[48] LEVEQUE, R. J. *Numerical methods for conservation laws*, vol. 3. Springer, 1992.

[49] LIN, C.-L., AND LAI, Y. G. Lattice boltzmann method on composite grids. *Physical Review E 62*, 2 (2000), 2219.

[50] LIN, L.-S., CHANG, H.-W., AND LIN, C.-A. Multi relaxation time lattice boltzmann simulations of transition in deep 2d lid driven cavity using gpu. *Computers & Fluids 80* (2013), 381–387.

[51] LISKA, R., AND WENDROFF, B. Comparison of several difference schemes on 1d and 2d test problems for the euler equations. *SIAM Journal on Scientific Computing 25*, 3 (2003), 995–1017.

[52] MALLAT, S. G. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence 11*, 7 (1989), 674–693.

[53] MCNAMARA, G. R., AND ZANETTI, G. Use of the boltzmann equation to simulate lattice-gas automata. *Physical review letters 61*, 20 (1988), 2332.

[54] MOHAMAD, A., AND SUCCI, S. A note on equilibrium boundary conditions in lattice boltzmann fluid dynamic simulations. *The European Physical Journal Special Topics 171*, 1 (2009), 213–221.

[55] MÜLLER, S. *Adaptive multiscale schemes for conservation laws*, vol. 27. Springer Science & Business Media, 2012.

[56] N'GUESSAN, M., MASSOT, M., SERIES, L., AND TENAUD, C. High order time integration and mesh adaptation with error control for incompressible navier–stokes and scalar transport resolution on dual grids. *Journal of Computational and Applied Mathematics 387* (2021), 112542.

[57] N'GUESSAN, M.-A. *Space adaptive methods with error control based on adaptive multiresolution for the simulation of low-Mach reactive flows*. PhD thesis, Université Paris-Saclay, 2020. https://tel.archives-ouvertes.fr/tel-02895792.

[58] OBRECHT, C., KUZNIK, F., TOURANCHEAU, B., AND ROUX, J.-J. Scalable lattice boltzmann solvers for cuda gpu clusters. *Parallel Computing 39*, 6-7 (2013), 259–270.

[59] ROHDE, M., KANDHAI, D., DERKSEN, J., AND VAN DEN AKKER, H. E. A generic, mass conservative local grid refinement technique for lattice-boltzmann schemes. *International journal for numerical methods in fluids 51*, 4 (2006), 439–468.

[60] ROUSSEL, O., AND SCHNEIDER, K. An adaptive multiresolution method for combustion problems: application to flame ball–vortex interaction. *Computers & Fluids 34*, 7 (2005), 817–831.

[61] SCHÖNHERR, M., KUCHER, K., GEIER, M., STIEBLER, M., FREUDIGER, S., AND KRAFCZYK, M. Multi-thread implementations of the lattice boltzmann method on non-uniform grids for cpus and gpus. *Computers & Mathematics with Applications 61*, 12 (2011), 3730–3743.

[62] SHAN, X. Simulation of rayleigh-bénard convection using a lattice boltzmann method. *Physical review E 55*, 3 (1997), 2780.

[63] TORO, E. F. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[64] YU, D., MEI, R., AND SHYY, W. A multi-block lattice boltzmann method for viscous fluid flows. *International journal for numerical methods in fluids 39*, 2 (2002), 99–120.

Appendix 1

In this appendix, we provide the details on how recovering the fully discretized volumetric lattice Boltzmann scheme (4) starting from the discrete form of the Boltzmann equation (3). Integrating this last equation on the characteristic of the velocity $\boldsymbol{\xi}^h$ gives

$$\begin{aligned}\int_t^{t+\Delta t}\left(\frac{\partial f^h}{\partial t}+\boldsymbol{\xi}^h\cdot\nabla f^h\right)\left(s,\boldsymbol{x}+s\boldsymbol{\xi}^h\right)\mathrm{d}s &= -\sum_{l=0}^{q-1}\omega_{hl}\int_t^{t+\Delta t}\left(f^l-f^{l,\mathrm{eq}}\right)\left(s,\boldsymbol{x}+s\boldsymbol{\xi}^h\right)\mathrm{d}s,\\ &= f^h(t+\Delta t,\boldsymbol{x}+\Delta t\boldsymbol{\xi}^h)-f^h(t,\boldsymbol{x}),\\ &= -\frac{\Delta t}{2}\sum_{l=0}^{q-1}\omega_{hl}\left(f^l-f^{l,\mathrm{eq}}\right)\left(t+\Delta t,\boldsymbol{x}+\Delta t\boldsymbol{\xi}^h\right)\\ &\quad -\frac{\Delta t}{2}\sum_{l=0}^{q-1}\omega_{hl}\left(f^l-f^{l,\mathrm{eq}}\right)(t,\boldsymbol{x}),\end{aligned}$$

where the collision terms has been approximated by a trapezoidal rule, see Dellar [17]. Calling $\tilde{f}^h = f^h + \frac{\Delta t}{2}\sum_{l=0}^{q-1}\omega_{hl}(f^l - f^{l,\mathrm{eq}})$ to keep the method explicit and rearranging gives

$$\tilde{f}^h\left(t+\Delta t,\boldsymbol{x}+\Delta t\boldsymbol{\xi}^h\right)=\tilde{f}^h(t,\boldsymbol{x})-\Delta t\sum_{l=0}^{q-1}\omega_{hl}\left(f^l-f^{l,\mathrm{eq}}\right)(t,\boldsymbol{x}).$$

We are left to dealing with the last term on the right hand side of the previous equation

$$f^l-f^{l,\mathrm{eq}}=\tilde{f}^l-\frac{\Delta t}{2}\sum_{m=0}^{q-1}\omega_{lm}(f^m-f^{m,\mathrm{eq}})-f^{l,\mathrm{eq}},$$

which also reads, gathering the collision frequencies in a matrix $\boldsymbol{\omega}$ with entries ω_{ij}

$$(\boldsymbol{I}+\Delta t\boldsymbol{\omega}/2)(\boldsymbol{f}-\boldsymbol{f}^{\mathrm{eq}})=\tilde{\boldsymbol{f}}-\boldsymbol{f}^{\mathrm{eq}},\qquad \text{thus}\quad \boldsymbol{f}-\boldsymbol{f}^{\mathrm{eq}}=(\boldsymbol{I}+\Delta t\boldsymbol{\omega}/2)^{-1}(\tilde{\boldsymbol{f}}-\boldsymbol{f}^{\mathrm{eq}})$$

Therefore, the semi-discretized numerical scheme reads

$$\begin{aligned}\tilde{f}^h\left(t+\Delta t,\boldsymbol{x}+\Delta t\boldsymbol{\xi}^h\right) &= \tilde{f}^h(t,\boldsymbol{x})-\Delta t\boldsymbol{\omega}(\boldsymbol{f}-\boldsymbol{f}^{\mathrm{eq}})|_h(t,\boldsymbol{x}),\\ &= \tilde{f}^h(t,\boldsymbol{x})-\Delta t\boldsymbol{\omega}(\boldsymbol{I}+\Delta t\boldsymbol{\omega}/2)^{-1}(\tilde{\boldsymbol{f}}-\boldsymbol{f}^{\mathrm{eq}})|_h(t,\boldsymbol{x}).\end{aligned}$$

Ahead, the tilde sign is understood. For the discretization of the spatial variable, consider a cell $C_{\bar{J},\boldsymbol{k}}$ for any $\boldsymbol{k}\in\mathbb{Z}$ at the finest level of discretization. Taking the average yields

$$\begin{aligned}\frac{1}{|C_{\bar{J},\boldsymbol{k}}|_d}\int_{C_{\bar{J},\boldsymbol{k}}}f^h\left(t+\Delta t,\boldsymbol{x}+\Delta t\boldsymbol{\xi}^h\right)\mathrm{d}\boldsymbol{x} &= \frac{1}{|C_{\bar{J},\boldsymbol{k}}|_d}\int_{C_{\bar{J},\boldsymbol{k}}}f^h(t,\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\\ &\quad -\frac{\Delta t\boldsymbol{\omega}(\boldsymbol{I}+\Delta t\boldsymbol{\omega}/2)^{-1}}{|C_{\bar{J},\boldsymbol{k}}|_d}\int_{C_{\bar{J},\boldsymbol{k}}}(\boldsymbol{f}-\boldsymbol{f}^{\mathrm{eq}})(t,\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\ \ |_h.\end{aligned}$$

A change of variable in the first integral provides, indicating the averages with a bar

$$\overline{f}^h_{\bar{J},\boldsymbol{k}+\boldsymbol{\eta}^h}(t+\Delta t)=\overline{f}^h_{\bar{J},\boldsymbol{k}}(t)-\Delta t\boldsymbol{\omega}(\boldsymbol{I}+\Delta t\boldsymbol{\omega}/2)^{-1}\left(\overline{\boldsymbol{f}}_{\bar{J},\boldsymbol{k}}(t)-\frac{1}{|C_{\bar{J},\boldsymbol{k}}|_d}\int_{C_{\bar{J},\boldsymbol{k}}}\boldsymbol{f}^{\mathrm{eq}}\left(f^0(t,\boldsymbol{x}),\ldots,f^{q-1}(t,\boldsymbol{x})\right)\mathrm{d}\boldsymbol{x}\right)|_h.$$

We commute the integral operator and the equilibrium operator, thus yielding the fully discrete scheme

$$\overline{f}^h_{\bar{J},\boldsymbol{k}+\boldsymbol{\eta}^h}(t+\Delta t)=\overline{f}^h_{\bar{J},\boldsymbol{k}}(t)-\Delta t\boldsymbol{\omega}(\boldsymbol{I}+\Delta t\boldsymbol{\omega}/2)^{-1}\left(\overline{\boldsymbol{f}}_{\bar{J},\boldsymbol{k}}(t)-\boldsymbol{f}^{\mathrm{eq}}\left(\overline{f}^0_{\bar{J},\boldsymbol{k}}(t),\ldots,\overline{f}^{q-1}_{\bar{J},\boldsymbol{k}}(t)\right)\right)|_h.$$

Appendix 2

In this appendix, we briefly recall how the prediction coefficient presented in Table 1 are recovered, following [10] and [21]. In a one-dimensional setting, for a cell $C_{j,k}$, one considers the local reconstruction polynomial of degree 2γ written in the canonical basis

$$\pi^h_{j,k}(x) = \sum_{m=0}^{2\gamma} A^{h,m}_{j,k} x^m.$$

The coefficients $(A^{h,m}_{j,k})^{m=2\gamma}_{m=0} \subset \mathbb{R}$ are obtained enforcing the linear constraints

$$\frac{1}{\Delta x_j} \int_{C_{j,k+\delta}} \pi^h_{j,k}(x) \mathrm{d}x = f^h_{j,k+\delta}, \qquad \delta = -\gamma, \dots, \gamma.$$

This signifies that the means of the reconstruction polynomial $\pi^h_{j,k}$ on the 2γ neighbors of $C_{j,k}$, plus the cell itself, must yield the actual mean value on these cells. This gives a linear system with matrix $\boldsymbol{T} \in \mathrm{GL}_{2\gamma+1}(\mathbb{R})$

$$\boldsymbol{T}(A^{h,m}_{j,k})^{m=2\gamma}_{m=0} = (f^h_{j,k+\delta})^{\delta=+\gamma}_{\delta=-\gamma}. \tag{24}$$

Notice that $\boldsymbol{T}$ does not depend on j, nor on k, nor on h. Once the system is solved and thus the reconstruction polynomial is known, the prediction is obtained by averaging the local reconstruction polynomial over the siblings $C_{j+1,2k}$ and $C_{j+1,2k+1}$

$$\widehat{f}^h_{j+1,2k+\delta} = \frac{1}{\Delta x_{j+1}} \int_{C_{j+1,2k+\delta}} \pi^h_{j,k}(x) \mathrm{d}x = f^h_{j,k} + (-1)^\delta \, Q^\gamma_1 \left(k; \boldsymbol{f}^h_j\right), \qquad \delta = 0, 1,$$

with $Q^\gamma_1 \left(k; \boldsymbol{f}^h_j\right) := \sum_{\alpha=1}^{\gamma} c_\alpha \left(f^h_{j,k+\alpha} - f^h_{j,k-\alpha}\right)$, where one can check that the weights are those given in Table 1.

For $d > 1$ the previous procedure is generalized by tensor product [5], thus considering an analogous of system (24) where the matrix is the Kronecker product (d-times) of $\boldsymbol{T}$ with itself.