A hybrid a posteriori MOOD limited lattice Boltzmann method to solve compressible fluid flows – LBMOOD

Ksenia Kozhanova, Song Zhao, Raphaël Loubère, Pierre Boivin

A hybrid *a posteriori* MOOD limited Lattice Boltzmann method to solve compressible fluid flows – LBMOOD

Ksenia Kozhanova[*b], Song Zhao[b], Raphaël Loubère[a], Pierre Boivin[b]

[a] *Université de Bordeaux, Institut de Mathématiques de Bordeaux, (IMB), CNRS, Bordeaux INP, UMR 5251, France*
[b] *Aix Marseille Université, M2P2, UMR7340, Centrale Marseille, France*

Abstract

In this paper we blend two Lattice-Boltzmann (LB) numerical schemes with an *a posteriori* Multi-dimensional Optimal Order Detection (MOOD) paradigm to solve hyperbolic systems of conservation laws in 1D and 2D. The first LB scheme is robust to the presence of shock waves but lacks accuracy on smooth flows. The second one has a second-order of accuracy but develops non-physical oscillations when solving steep gradients. The MOOD paradigm produces a hybrid LB scheme via smooth and positivity detectors allowing to gather the best properties of the two LB methods within one scheme. Indeed, the resulting scheme presents second order of accuracy on smooth solutions, essentially non-oscillatory behavior on irregular ones, and, an 'almost fail-safe' property concerning positivity issues. The numerical results on a set of sanity test cases and demanding ones are presented assessing the appropriate behavior of the hybrid LBMOOD scheme in 1D and 2D.

Keywords: Lattice Boltzmann Method (LBM), MOOD, Hyperbolic system of conservation laws, Shock, Hydrodynamics.

1. Introduction

Historically the Lattice Boltzmann (LB) method [1, 2] has been designed to solve fluid flow problems relying on a kinetic description, namely the Boltzmann equation. The Boltzmann equation describes the dynamics of a gas at a meso-scopic scale, which by integration leads to the equations of fluid dynamics at macroscopic scale, namely Navier-stokes or Euler inviscid partial differential equations (PDEs). While classical numerical methods, e.g. Finite Difference, Finite Element or Finite Volume (FV) methods, are usually designed to solve directly the macro-scale fluid dynamics equations, LB methods focus on the kinetic version of the PDEs. This kinetic version can be split into a local non-linear collision stage and a non-local linear advection stage [2], the latter being exactly solved by LB method.

The generic working variable for LB approach is a distribution function while classical methods usually solve for

[*]Corresponding author
Email addresses: ksenia.kozhanova@univ-amu.fr (Ksenia Kozhanova[*]), song.zhao@univ-amu.fr (Song Zhao), raphael.loubere@math.u-bordeaux.fr (Raphaël Loubère), pierre.boivin@univ-amu.fr (Pierre Boivin)

the macroscopic conservative variables such as density, momentum and energy. The distribution function models the probability density of finding particles at given position, velocity and time. Its first moments are indeed the macroscopic conserved variables. These fundamental differences render LB methods particularly appealing to solve the weakly compressible Navier-Stokes equations in an efficient manner[3]. However, for compressible flows, for which conservation of total energy becomes a necessary condition to capture the correct weak solutions, classical LB methods present some issues due to intrinsic isotropy errors [1, 4].

Several approaches exist to advance LB methods to fully compressible flows [5]. The most intuitive one is to increase the number of lattice velocities . However, while it has been employed in recent studies [6], it has several known drawbacks such as high computational cost, particularly in 3D, sophisticated ways to deal with boundary conditions and stability issues due to the increased number of non-hydrodynamic modes. Another approach consists in solving the energy equation separately where two routes are possible. The first one introduces the double distribution function [7] where, yet again, stability ambiguity arises. The second one, the so-called hybrid LB method [8–10], resolves the energy equation and, hence, the temperature fluctuations using a finite difference scheme coupled with LBM. Additionally, the in-variance Galilean error is addressed by adding appropriate forcing terms.

The hybrid LB methods for compressible flows can be based on coupling the LB scheme with explicit discrete energy equation solved by finite difference scheme, in which case new instabilities arise due to the strong coupling instabilities [11, 12]. An important question is, thus, how to avoid such a coupling and obtain better stability properties while retaining the accuracy of the scheme. This can be addressed by, for instance, solving a simple advection equation describing the evolution of the characteristic variables of hyperbolic system of Euler equations. A hybrid LBM can benefit from such an approach thanks to the entropy-based models, where entropy is indeed a characteristic variable of Euler equations. The advantage of this technique is its linear independence from the rest of the system modelled by LBM, which allows in return to control the entropy without degrading the stability and accuracy of the original LB method. However, explicit discretization of characteristic equation does not lead to a conservative system. This latter issue has been solved in [13] by deriving a fully conservative model for the total energy. The resulting scheme benefits from linear equivalence of conservative total energy equation and non-conservative characteristic entropy-based equation. This allows to keep the advantages of the linearly decoupled from LBM entropy-based model. However, while this hybrid LB method undoubtedly led to improved solutions and correct jump relations across shock-waves, e.g. detonations [14], some stability issues for higher Mach numbers remain a challenge. Innovative stabilisation strategy avoiding the need of the shock sensors, which have been used up to this date in classical LB schemes, is required.

The difficulty when solving hyperbolic PDEs is the creation in finite time of discontinuous solutions, usually due to the presence or creation of shock waves or contacts. Conservative numerical schemes are mandatory in order to capture the correct wave speeds, and, ultimately converge towards a weak solution of the system of PDEs under consideration. However, when high accurate schemes are considered, that is beyond first order, unavoidably, one faces the creation of spurious numerical oscillations due to Gibbs phenomenon in presence of steep gradients. Therefore

any high order scheme requires to be supplemented with an additional dissipation operator. Most of the time, the extra dissipation consists in adding a large amount of numerical viscosity in the vicinity of steep gradients. Such a procedure is generally referred to as a 'limiter' and is embedded into the numerical method. Philosophically, any effective limiting procedure answers two questions: (i) where in the computational domain is the solution presenting behaviors that would require extra dissipation? and (ii) how much extra dissipation should be supplemented?

At first glance classical limiters seem to have different forms, for instance the artificial viscosity for Lagrangian staggered schemes [15], slope/flux limiter for finite volume/finite element second-order schemes [16], (Weighted) Essentially-Non-Oscillatory procedure for finite difference/volume schemes [17, 18], etc. In reality almost all limiters resort to "sensors" to answer question (i) and mimic the dissipation of a first order accurate scheme to tackle (ii). Starting from the solution at time t^n, any numerical scheme evolves the state variables during $\Delta t > 0$ up $t^{n+1} = t^n + \Delta t$. The limiters usually act on the states variables at t^n, and must anticipate possible spurious oscillations or troubles which may manifest at time t^{n+1}. Unfortunately, this task is complex because the system of PDEs as well as the numerical method, are often non-linear. Even the notion of 'spurious oscillations or troubles' is not clearly defined especially at the beginning of their development. Moreover, the limiter can fail because there exists no mechanism to recover from possible mistakes. The consequence of this complex task of anticipation usually manifests as a precaution principle: the sensors are too sensitive, and, the amount of extra-dissipation is excessive, hence degrading the solution[1].

A new type of limiting procedure called MOOD (Multi-dimensional Optimal Order Detection) has emerged to solve this state of affair [19–21]. This procedure is quoted as being an *a posteriori* paradigm, because it relies on (a) the detection of a problematic situation on a candidate solution at time t^{n+1} computed with an accurate scheme, and, (b) the subsequent re-computation with a more dissipative scheme of the troubled cells. Because it is simpler to detect a troubled cell instead of anticipating its future occurrence, MOOD allowed to secure numerical finite volume/difference, Smoothed Particle Hydrodynamics methods or Discontinuous Galerkin schemes in an efficient way [22–24]. Moreover, this approach allows the scheme to recover from positivity issues, occurrence of `NaN` values, etc. because of the *a posteriori* test and re-computation. In fact, the re-computation stage uses a 'parachute' (i.e ultra-robust) numerical scheme as a last resort scheme for extreme situations [22]. As such, numerical methods which are *a posteriori* limited by a MOOD paradigm are fail-safe ones. As far as we know MOOD limiting procedure has never been coupled with LB methods for solving compressible fluid flows.

Consequently, in this article we present a novel way to design a hybrid method from the LB family. Our approach relies on the high accurate LB scheme for compressible flows described in [10, 13] supplemented with an *a posteriori* MOOD limiting using a low order LB method as a fail-safe/low-order scheme. In this article we only focus on the compressible fluid flow model for this proof of concept, knowing that more complex models and more advanced set

[1] For instance solving an isentropic vortex with a second order finite volume scheme with classical minmod slope limiter leads to a first-order error in L^∞ norm, and strictly less than second-order in L^2 norm, although no limiting is required for such a smooth solution. If the limiter is turned off a plain second order of accuracy is retrieved in both norms.

of LB methods could be used in future studies.

Following this introduction we present in Section 2 the discretized LB method dedicated to solve compressible fluid flows. Then, in Section 3 this numerical method is further blended with a low-order LB method via the *a posteriori* MOOD approach. Numerical experiments are carried out and presented in the Section 4 where sanity checks and more advances ones are proposed. A discussion (Section 5) follows and some trivial and non-trivial extensions are proposed. Section 6 concludes this article.

2. Lattice-Boltzmann method for compressible flows

In this section we briefly introduce Lattice-Boltzmann (LB) method focusing on systems of partial differential equations (PDEs) modeling the compressible fluid flows. The goal of the study is to provide a stable, accurate, non-oscillating and admissible numerical solution by means of Lattice-Boltzmann type of scheme.

2.1. Target macroscopic equations

This study focuses on the PDEs governing compressible gas dynamics written with Einstein's notation

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_\beta}{\partial x_\beta} = 0, \tag{1}$$

$$\frac{\partial \rho u_\alpha}{\partial t} + \frac{\partial \rho u_\alpha u_\beta + p\delta_{\alpha\beta}}{\partial x_\beta} = \frac{\partial \tau_{\alpha\beta}}{\partial x_\beta}, \tag{2}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \rho u_\beta (E + p/\rho)}{\partial x_\beta} = \frac{\partial \tau_{\alpha\beta} u_\alpha}{\partial x_\beta}. \tag{3}$$

Here t refers to time, $\boldsymbol{x} = x_\alpha$ is the space vector, $\boldsymbol{u} = u_\alpha$ is the fluid velocity vector with $._\alpha$ referring to their α^{th} component, ρ is the mixture density and p it the thermodynamic pressure. E is the total energy (sum of internal energy e and kinetic energy $u_\alpha^2/2$), $\tau_{\alpha\beta}$ is the viscous stress tensor, and $\delta_{\alpha\beta}$ is the Kronecker symbol, i.e 1 if $\alpha = \beta$, 0 otherwise.

The system is closed upon choosing an equation of state (EOS), for instance a complete one such as the perfect gas EOS

$$p = (\gamma - 1)\rho e = k\rho T, \tag{4}$$

with γ the ratio of specific heats depending on the kind of gas considered, T its temperature and k is the Boltzmann constant ($k = 1.38 \times 10^{-23}$ J K^{-1}). The sound-speed is defined as $c_s^2 = \gamma p/\rho$ for this EOS. The viscous stress tensor takes the form

$$\tau_{\alpha\beta} = \mu\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha} - \delta_{\alpha\beta}\frac{2}{3}\frac{\partial u_\gamma}{\partial x_\gamma}\right), \tag{5}$$

which is proportional to the dynamic viscosity $\mu > 0$ which is a physical parameter.

The present study focuses on the Euler limit of the above system of PDEs, that is when $\tau_{\alpha\beta} \to 0$, hence all computations are carried out with a negligible viscosity $\mu = 10^{-8}$ Pa.s. The system of PDEs then becomes of hyperbolic type with the possible creation of discontinuous solution in finite time, which is a difficult feature to capture.

2.2. Isothermal Lattice-Boltzmann method

2.2.1. Boltzmann equation

The Lattice-Boltzmann method builds upon a specific space, time and velocity discretization of the Boltzmann Equation (BE). In the absence of external forces, it reads

$$\frac{\partial f}{\partial t} + \xi_\alpha \frac{\partial f}{\partial x_\alpha} = \Omega(f), \tag{6}$$

where $f(\boldsymbol{\xi}, \boldsymbol{x}, t)$ is the probability density function of finding particles with velocity $\boldsymbol{\xi} = \xi_\alpha$ at position x_α at time t. The left hand-side of (6) indicates that particles travel with their own speed, $\boldsymbol{\xi}$. The right-hand-side $\Omega(f)$ is the so-called collision operator, taking into account particle collisions. Collisions tend to bring back the distribution f towards the Maxwell-Boltzmann distribution function (also called Maxwellian function)

$$f^{eq} = \frac{\rho}{(2\pi T)^{D/2}} e^{\frac{-\|\xi - u\|^2}{2T}}, \tag{7}$$

where D is the spatial dimension.

The simplest collision model is the BGK (Boltzmann-Gross-Bathnagar) model [25]

$$\Omega(f) = \frac{1}{\tau}(f^{eq} - f), \tag{8}$$

where $\tau > 0$ is a characteristic time between collisions, related to the fluid viscosity μ. It can be shown via Chapman-Enskog expansion [26] that the first moments of the above system are the Navier-Stokes equations (1-3) since the first three moments of f are related to the macroscopic variables

$$\int f(\boldsymbol{\xi}, t, \boldsymbol{x}) d\boldsymbol{\xi} = \rho(t, \boldsymbol{x}), \quad \int \boldsymbol{\xi} f(\boldsymbol{\xi}, t, \boldsymbol{x}) d\boldsymbol{\xi} = \rho(t, \boldsymbol{x}) \boldsymbol{u}(t, \boldsymbol{x}), \quad \int |\boldsymbol{\xi}|^2 f(\boldsymbol{\xi}, t, \boldsymbol{x}) d\boldsymbol{\xi} = \rho(t, \boldsymbol{x}) E(t, \boldsymbol{x}). \tag{9}$$

2.2.2. Velocity discretization

To construct a Lattice-Boltzmann model, the BE equation is first discretized in a reduced-order velocity space ($\boldsymbol{\xi}_i$), so that $f_i(\boldsymbol{x}, t) \equiv f(\boldsymbol{\xi}_i, \boldsymbol{x}, t)$, and

$$\frac{\partial f_i}{\partial t} + \xi_{i,\alpha} \frac{\partial f_i}{\partial x_\alpha} = \Omega(f_i). \tag{10}$$

The main ingredient in the Lattice-Boltzmann model is to choose an appropriate basis for the velocity space. Given that f^{eq} is a Maxwellian, a Hermite polynomial basis is chosen. This velocity basis is called the Lattice.

Anticipating on the space/time discretization step, the system is normalized in a way such as each discrete velocity corresponds exactly to the distance to the next neighboring cell to the time-step. A number of classical lattices DnQm is available, where n is the spatial dimension (D in (7)), and m is the basis dimension. Popular examples include D1Q3, D2Q9, D3Q19, D3Q27, with the first two being illustrated in Fig. 1. The present work uses the D2Q9 basis and the interested reader is referred to [2] for a detailed description of the normalization and velocity discretization steps.

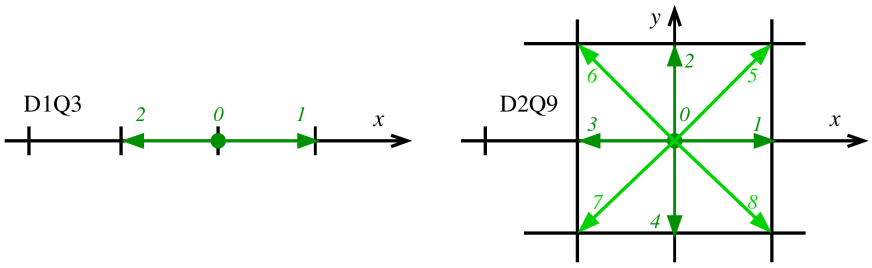

Figure 1: Examples of lattices for LB method in 1D, D1Q3 (left) and 2D, D2Q9 (right).

Finally, the normalized system of equation reads

$$\frac{\partial f_i}{\partial t} + c_{i,\alpha}\frac{\partial f_i}{\partial x_\alpha} = \Omega(f_i), \tag{11}$$

where $\boldsymbol{c}_i = c_{i,\alpha}$ is the ith discrete velocity component of the selected lattice. Note that f_i has now Lattice units [2], even though the same notation is classically retained.

2.2.3. *Space/time discretization*

Contrary to the Navier-Stokes equations, the convection velocity $\boldsymbol{c_i}$ of distribution function f_i is constant, so the convection term is linear. Furthermore, it is normalized such as f_i is advected *exactly* to a neighboring cell during one time-step $\Delta t > 0$. As illustrated in Fig. 1, space and time discretizations are closely related with the lattice. Integration along a characteristic line of the left-hand side of (11) therefore yields exactly

$$f_i(t + \Delta t, \boldsymbol{x}) = f_i^{coll}(t, \boldsymbol{x} - \boldsymbol{c}_i\Delta t), \tag{12}$$

which corresponds to the *streaming step* of any Lattice-Boltzmann algorithm. f_i^{coll} is called the post-collision distribution, to be defined hereafter. The exactness of this integration provides Lattice-Boltzmann methods with excellent dissipation properties compared to classical finite differences solvers [27].

Via a trapezoidal rule, integration of the collision term – see [2, 28] for a step-by-step proof – yields the fundamental formula

$$f_i^{coll}(\boldsymbol{x}, t) = f_i^{eq}(\boldsymbol{x}, t) + (1 - \frac{\Delta t}{\tau})f_i^{neq}(\boldsymbol{x}, t) + \frac{\Delta t}{2}F_i(\boldsymbol{x}, t), \tag{13}$$

where the definition of f_i^{neq} is dependent on the collision model. In the simple case of BGK [25],

$$f_i^{neq}(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t) + \frac{\Delta t}{2}F_i(\boldsymbol{x}, t), \tag{14}$$

but more advanced alternatives, such as multiple relaxation [29] and regularized [30, 31] models do exist. Contrary to the streaming step (12), the *collision step* is non-linear, but, fortunately it is a local operation.

In (13), the relaxation time $\tau > 0$ depends on the fluid dynamic viscosity μ as

$$\tau = \frac{\mu}{\rho c_s^2} + \frac{\Delta t}{2}, \tag{15}$$

and F_i is a forcing term, which may contain volume forces (e.g. gravity) and correcting terms [32]. Note that this term was omitted up to equation (11) for the sake of clarity. Also remark that in the case of Euler limit, $\mu \simeq 0$, hence $\tau \simeq \frac{\Delta t}{2}$. The definition of Δt can be found in Appendix A (A.2).

A generic LB algorithm consists of a succession of *Stream & Collide* steps, respectively described in (12) and (13). Macroscopic mass and momentum are reconstructed after each streaming step as

$$\rho(\boldsymbol{x}, t + \Delta t) = \sum_{i=1}^{m} f_i(\boldsymbol{x}, t + \Delta t), \qquad \rho u_\alpha(\boldsymbol{x}, t + \Delta t) = \sum_{i=1}^{m} c_{i\alpha} f_i(\boldsymbol{x}, t + \Delta t). \tag{16}$$

2.3. Collision models and its stability properties

Better stability properties can be achieved by reconsidering the collision step. Indeed, the collision model (13) can be decomposed into two steps (neglecting the forcing term F_i):

1. a pre-collision regularisation

$$f_i^{reg} = f_i^{eq} + f_i^{neq}, \tag{17}$$

where the computation of f^{eq} depends on the macroscopic variables, ρ, u_x, u_y and, as a result, three moments of discrete f_i are involved. On the other hand, f^{neq} depends on six variables, i.e. ρ, u_x, u_y, a_{xx}, a_{xy}, a_{yy} where six independent moments of discrete f_i are involved and

$$a_{ij} = \sum_k (f_k - f_k^{eq}) \mathcal{H}_{ij} \tag{18}$$

where $\mathcal{H}_{ij}$ are Hermite polynomials. The reader is referred to [33] for more details.

2. a BGK collision

$$f_i = f_i^{req} - \frac{\Delta t}{\tau}(f_i^{reg} - f_i^{eq}), \tag{19}$$

The regularisation strategy leads to the reduction in the system rank which yields to the reduction of the modes number, i.e. six modes. The reduction of the system rank results in further decrease of modes and, hence, better stability properties [34]. This can be simply achieved by setting the variable $\tau = \Delta t$ where now the rank of the system is reduced and the number of remaining modes is three. This allows to obtain a substantial stability gain as a result of this modes filtering. The resulting collision model is then reduced to

$$f_i^{coll}(x, t) = f_i^{eq}(x, t) + \frac{\Delta t}{2} F_i(x, t). \tag{20}$$

Hence, our collision model can be changed by setting $\tau = \Delta t$ to add extra dissipation to the solution where discontinuities are present by filtering out the non-physical modes. This concept has been analysed in details in [33] and, moreover, applied in a more traditional shock sensor framework in [35].

2.4. Extension to compressible flows

The above LB scheme based on the D2Q9 lattice solves the 2D isothermal Navier-Stokes equations (1-2) in the weakly compressible regime. The corresponding equation of state is $p = \rho c_s^2$.

Solving the full compressible set of equations is not possible with the above algorithm because the D2Q9 lattice quadrature order is too low [2]. This can be illustrated in a simple way: the total energy equation corresponds to the second-order moment of f_i, and has a flux related to the third-order moment [28]. Since $c_{i,\alpha} = \{-1, 0, 1\}$ for all nearest neighbor lattices (D1Q3, D2Q9, D3Q19, D3Q27), the third-order moment $\sum_i c_{i,\alpha} c_{i,\alpha} c_{i,\alpha} = \sum_i c_{i,\alpha}$ depends on the first-order moment (corresponding to the mass flux).

There are essentially two possibilities to tackle the full set of compressible equations (1-3):

- Extend the lattice quadrature order, e.g. D2Q37 and D3Q343. The advantage is that the required changes are minimal, but the numerical cost may dramatically increase.
- Solve for the energy equation aside, e.g. not as the second-order of f_i. This can be achieved either with a second distribution function (double distribution methods), or a scalar (hybrid methods).

For a detailed review of these alternatives, the reader is referred to [5]. The present paper pertains to the hybrid method category, but the authors expect the MOOD strategy presented in section 3 to apply to other compressible LB methods through minor modifications.

The hybrid LB method employed in this work allows to rewrite the evolution equations for the mass and momentum in conservative form as well as the evolution for the total energy, as has been shown in [13]. These evolution equations are

$$\rho(\boldsymbol{x}, t+\Delta t) = \rho(\boldsymbol{x}, t) - \frac{\Delta t}{\Delta x}\left[F^{\rho,LB}_{+\frac{\alpha}{2}}(\boldsymbol{x}, t) - F^{\rho,LB}_{+\frac{\alpha}{2}}(\boldsymbol{x} - c_\alpha \Delta x, t)\right], \tag{21}$$

$$(\rho u_\alpha)(\boldsymbol{x}, t+\Delta t) = (\rho u_\alpha)(\boldsymbol{x}, t) - \frac{\Delta t}{\Delta x}\left[F^{\rho u_\alpha,LB}_{+\frac{\alpha}{2}}(\boldsymbol{x}, t) - F^{\rho u_\alpha,LB}_{+\frac{\alpha}{2}}(\boldsymbol{x} - c_\alpha \Delta x, t)\right], \tag{22}$$

$$(\rho E)(\boldsymbol{x}, t+\Delta t) = (\rho E)(\boldsymbol{x}, t) - \frac{\Delta t}{\Delta x}\left[F^{\rho E,FV}_{+\frac{\alpha}{2}}(\boldsymbol{x}, t) - F^{\rho E,FV}_{+\frac{\alpha}{2}}(\boldsymbol{x} - c_\alpha \Delta x, t)\right], \tag{23}$$

where the LB fluxes are defined in Appendix B and the definition for the total energy flux in (23) can be found in [13] equation 35.

2.5. LB algorithm for compressible flows

The LB model for compressible flows is summarized in Table 3.4. Derivation details may be found in the reference papers [10] for the LB algorithm, and [13] for the numerical scheme specifically developed for the energy equation. The consistency with the full Navier-Stokes equations (1-3) is proved in [28].

The hybrid algorithm of LB method for compressible flows is summarised below. At time t the conserved fields f_i, ρ, ρu_α and ρE are available along with the knowledge of ρ, T at time $t - \Delta t$.

1. Compute f_i^{coll} at time t by using the collision model detailed in Appendix A The computations of the correction term F_i require the knowledge of ρ and T at the time $t - \Delta t$.
2. Perform the streaming step in order to obtain the values of f_i at time $t + \Delta t$.
3. Compute the LB fluxes $F^{\rho}_{\Delta x/2}$, $F^{\rho}_{\Delta y/2}$, $F^{\rho u_\alpha}_{\Delta x/2}$, $F^{\rho u_\alpha}_{\Delta y/2}$ from the knowledge of f^{coll} as detailed in Appendix B.
4. Compute the macroscopic variables ρ and ρu_α at time $t + \Delta t$. These can be obtained by either computing the moments of f_i or by using the LB fluxes. Both methods are equivalent as has been demonstrated in [13].
5. Compute the total energy flux (as per equation 35 in [13]). The total energy is then transported according to (23).

All these steps being done, $f_i, \rho, \rho u_\alpha, \rho E$ at time $t + \Delta t$ are available as well as the values of ρ and T at time t. The procedure is then repeated till the final time.

While recent advances in LB method for compressible flows indeed led to the improvement of the solution in terms of the correct jumps relations recovery across the shock waves, notably in recent works [13] and [14], the stabilisation of the method for the applications where flow discontinuities are observed remains a challenge. In what follows, the aim is to define an appropriate framework to apply LBM to fully compressible flows with strong shock waves. Firstly, a LBM collision model with better stability properties is recalled to be an alternative for the areas of the solution where more dissipation is required. Secondly, an appropriate detection technique of these critical areas is defined aiming to restore the stability in the zones of strong discontinuities while not degrading the accuracy in the smooth regions. Hence, the proposed novel stabilisation method for LBM is based on the detection-*a posteriori* correction idea rather than traditional shock-sensor techniques.

3. An *a posteriori* limited LBM-MOOD algorithm

3.1. The MOOD philosophy

The Multi-dimensional Optimal Order Detection [19–22] approach has been chosen to accommodate for both the collision models where it is necessary. Originally proposed for the finite volume schemes, this paradigm can be ultimately used for any numerical scheme where certain problematic areas of the solution require a change of the scheme to fix, for instance, spurious numerical phenomena.

The MOOD technique is based on the *a posteriori* detection and correction of the "troubled" cells of the solution by using an appropriate scheme order. The algorithm first runs one iteration of the solution by using the "best" available scheme, usually the more accurate one prone to instability. The obtained solution is then analysed by the set of physical and numerical admissibility criteria. If one of them fails, the troubled cells are recomputed by using the next "best" numerical scheme. Then the obtained solution is again analyzed, with, possibly some new troubled cells to be corrected. This approach is almost fail-free thanks to the availability of the "worst case scenario" scheme which is usually a first-order accurate numerical scheme carrying good stability and robustness properties.

3.2. Admissibility criteria

The detection of problematic parts of the solution is needed due to the possible presence of non-physical oscillations caused by Gibbs phenomena. Hence, an extra dissipation might be required in those parts of the domain. The process of such a detection results in marking the troubled (problematic/bad) cells and applying a more dissipative scheme for these cells. The detection criteria are twofold.

The first one, the *Physical Admissibility Detection* (PAD), is based on the physics involved into the system of PDEs solved. For Euler or Navier-Stokes equations, we test the solution for the positivity of density ρ and internal energy e. Notice that this is sufficient to ensure the numerical state to remain in the admissible set.

The second part, the *Numerical Admissibility Detection* (NAD) is meant to detect the spurious oscillations or other numerical issues. The base of NAD is the Discrete Maximum Principle (DMP) for cell k which can be represented in the form of two inequalities as

$$\min_{k\in S}(\alpha_k^n) - \Delta \leq \alpha_k^{n+1} \leq \max_{k\in S}(\alpha_k^n) + \Delta, \tag{24}$$

where S is a set containing the current cell k and its neighbors, α is the tested variable (we test density, velocity and pressure variables) and parameter Δ is a small number meant to regulate the admissibility of small undershoots and overshoots in order to have a higher accuracy with smooth extrema. This parameter is computed as

$$\Delta = \max\left(\epsilon_0, \epsilon\left(\max_{k\in S}(\alpha_k^n) - \min_{k\in S}(\alpha_k^n)\right)\right), \tag{25}$$

where we set ϵ_0=10^{-4} which plays the role of an absolute small number and ϵ=10^{-3}. The parameter Δ can be interpreted as an option to allow an appearance of new extremes smaller than one thousandth of the local jump at the current time step n in the immediate neighbourhood of current cell k.

3.3. Re-computation/Correction

Any cell which has not passed one of the admissibility criteria is flagged as troubled/bad and its index is stored in set $\mathcal{B}$. Moreover, once a bad cell is flagged, in fact the vicinity of this cell is possibly also troubled. Therefore the direct neighbors of a bad cells are also flagged and recomputed. Obviously more neighbors could be considered hence increasing the dissipation if needed. The numerical section will present some tests where this neighborhood vary. In general, only the direct neighbors of a bad cell are added to $\mathcal{B}$.

3.4. LBMOOD algorithm for compressible flows

The resulting LBMOOD algorithm for compressible flows requires nothing than adding two extra steps inside the time loop of hybrid LB algorithm. That is, once the solution of total energy at time $t + \Delta t$ has been computed in step 5 of Subsection 2.5, the following steps are added.

6. Run the solution through the MOOD PAD and NAD criteria. Obtain the matrix $\mathcal{B}$ of cells to be corrected.

Description/Stage	Input	Output	Equation
Equilibrium update	ρ , u, T	f_i^{eq}	(A.1)
Force Update	ρ , u, T	$F_i(t,x)$	(A.7)
Non-equilibrium update	f_i^{eq}, $F_i(t,x)$	f_i^{neq}	(A.3)
Collision	f_i^{eq}, f_i^{neq}, $F_i(t,x)$	$f_i^{coll}(t+\Delta t,x)$	(13)
Streaming	$f_i^{coll}(t,x-c_i\Delta t)$	$f_i(t+\Delta t,x)$	(12)
Macroscopic variables update	f_i	ρ , u	(16)
FV Energy coupling	ρ , u, T	T	(23)
MOOD detection	ρ , u, T	$\mathcal{B}$	(24)

Table 1: General LBMOOD structure and main stages.

7. Recompute the solution for the cells in $\mathcal{B}$ following the steps 1-5 of Subsection 2.5 by using the LB scheme with collision model as per (20).

Thus, one time iteration of LBMOOD method consists in steps 1-7. The general structure of the method with references to necessary equations are summarised in Table 1.

4. Numerical tests

In this section we present the numerical experiments and evidences assessing that the *a posteriori* limited LBMOOD scheme is a viable option to maintain accuracy, robustness and ensure important physics constraints like positivity for inviscid gas-dynamics. The following 2D schemes are tested:

LBM the classical D2Q9 LBM scheme with collision according to (13).

LBM0 the dissipative D2Q9 LBM scheme with collision model as per (20).

LBMOOD the *a posteriori* blended scheme with these two methods.

The methodology of testing relies on a first series of 1D tests:

- Advection of a 1D profiles. These simulations test the ability to *advect or maintain stationary profiles*.
- Sod and modified Sod 1D shock tubes. These classical shock tubes challenge the *Essentially Non Oscillatory* character of the schemes and their ability to capture shocks and rarefaction waves without spurious oscillations.

Each of the previous 1D tests have an analytical solution, allowing error computation and visual comparisons. Mesh convergence studies will be presented to assess that the properties of the LBMOOD scheme is relatively independent from the number of cells. In a second series of test, classical 2D benchmarks are simulated, namely

- Entropy spot and isentropic vortex. These test the *Accuracy and experimental orders of convergence* because smooth exact solutions can be derived.

- Cylindrical Sod shock tube. This is an explosion like problem which generates a radial solution presenting cylindrical separated simple waves (rarefaction, contact and shock).

- 2D Riemann problems. Several 4 state configurations are simulated to assess the ability of the scheme to capture complex flow structures for which we can only compare to reference solutions.

4.1. 1D tests

The initial data (domain, density, velocity component u and pressure, and final time) are gathered in Table 2 for all 1D tests. γ is set to 7.5. Initial time is set to $t^0 = 0$. The boundary conditions are set to wall type. Exact solutions can be analytically computed and are superimposed on the numerical results, see [36] for details. Only the density variable is plotted.

1D contact discontinuity. The first test is a simple contact discontinuity in advection which should not generate any spurious secondary waves. We plot the numerical density for 400 (left panel) and 800 (middle panel) cells in figure 2 when LBM0, LBM and LBMOOD schemes are used. The right-most panel presents the mesh convergence for LBMOOD scheme for $N = 50 \times 2^k$ cells and $k = 0, 1, 2, 3, 4$. The dissipative LBM0 scheme produces smooth solutions (green symbols) while LBM generates spurious numerical oscillations (blue symbols). LBMOOD reduces their amplitude without entirely damping all of them. For LBMOOD scheme, when a bad cell is detected, then

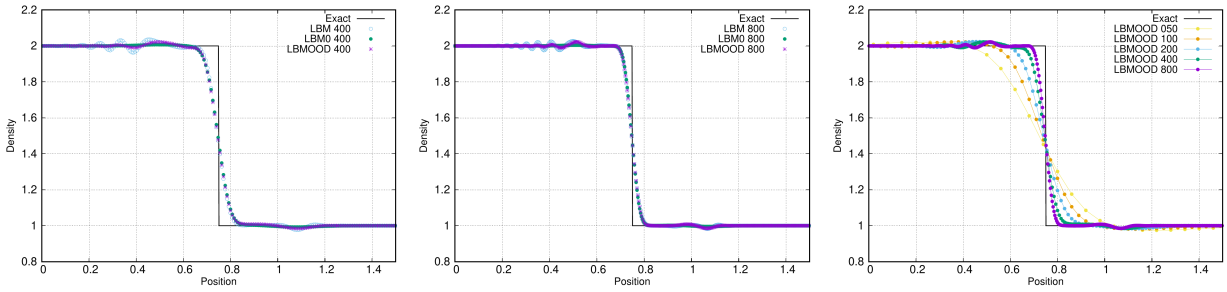

Figure 2: Contact discontinuity problem — Numerical density for LBM0, LBM and LBMOOD schemes with 400 (left) and 800 (middle) cells. The right panels present the mesh convergence for LBMOOD scheme for 5 successively refined meshes.

this only cell is recomputed with LBM0 scheme, restricting as such the dissipation. However we could spread the dissipation locally and correct also its closest neighbors. As such 3, 5 or 8 cells could be affected by the correction. Figure 3 presents a zoom on the numerical results when such neighborhood increase is applied. The results do not improve when a large neighborhood is to be considered. Because the cost increases with larger neighbor, it does not justify such an increase in neighborhood size. For now on, only the detected cell will be recomputed with LBM0 scheme.

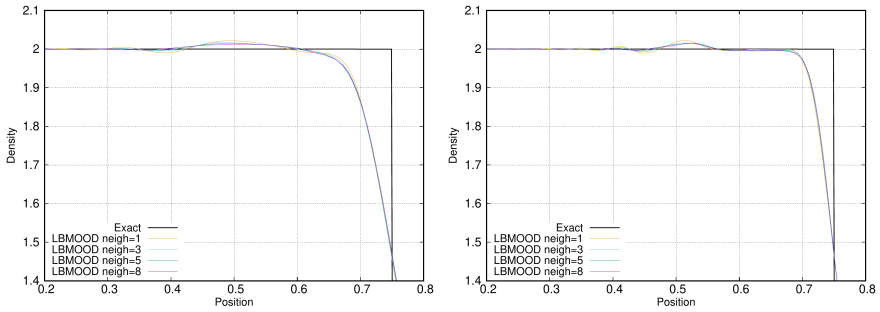

Figure 3: Contact discontinuity problem — Numerical density for LBMOOD scheme with 400 (left) and 800 (right) cells —- Comparison of different correction neighborhoods with increasing spreading.

Problem	ρ_L	u_L	p_L	ρ_R	u_R	p_R	t_{final}	Ω	$x_{\text{ddc}}(t^0)$
Sod 1D	1.0	0.0	1.0	0.125	0.0	0.1	0.2	[0,1]	0.5
Contact 1D	2.0	1.0	1.0	1.0	1.0	1.0	0.25	[-1,1.5]	0.5
Sod2 1D	1.0	0.0	1.0	0.125	0.0	0.01	0.2	[0,1]	0.5

Table 2: Initial left and right states for the density ρ, velocity u and the pressure p for the 1D problems. The discontinuity is located at position x_{ddc} at initial time $t^0 = 0$. The final simulation times t_{final} are also given.

1D planar Sod shock tubes. Then we run the 1D classical planar Sod shock tube and a modified version (Sod2) to assess the ability of the methods to capture 1D simple waves accurately. In Figures 4 we compare the results of the LBM0, LBM and LBMOOD schemes when 400 or 800 cells are employed (left and right panels). The results for Sod problem are plotted on top panels, while those for Sod2 are displayed on bottom ones. We observe that LBM0 scheme results are more diffused, LBM ones are more accurate but present overshoots in the vicinity of the shock wave. LBMOOD produces numerical solutions which are a good compromise between these two inappropriate behaviors. Namely the oscillations have been almost entirely damped without sacrificing the accuracy anywhere else. On the right panels of figure 4 we present the mesh convergence of LBMOOD scheme results when the mesh is successively refined by a factor 2, that is with $N = 50 \times 2^k$ cells and $k = 0, 1, 2, 3, 4$. Clearly, when the mesh is refined, the solution tends towards the exact solution without spurious behaviors. Next, in figure 5, one presents in red the bad cells which are detected by LBMOOD scheme using 50 (left), 100 (middle) and 200 (right) cells as a function of position and time iteration. At first glance we observe that a small amount of cells need extra dissipation. The red bad cells are updated with the robust LBM0 scheme while the blue ones rely on the accurate LBM. From these figures we can observe that the troubled cells mainly follow the displacement of the discontinuities: head of the rarefaction, contact and shock waves. The contact displacement seems to generate spurious numerical effects which require dissipation, hence the relative large number of troubled red cells in this area. However this region in between the contact and shock is essentially constant which minor spurious waves. Therefore the correction has no bad (nor good) effect on the numerical solution because LBM0 and LBM, both can capture constant states. At last, in table 3, we gather the

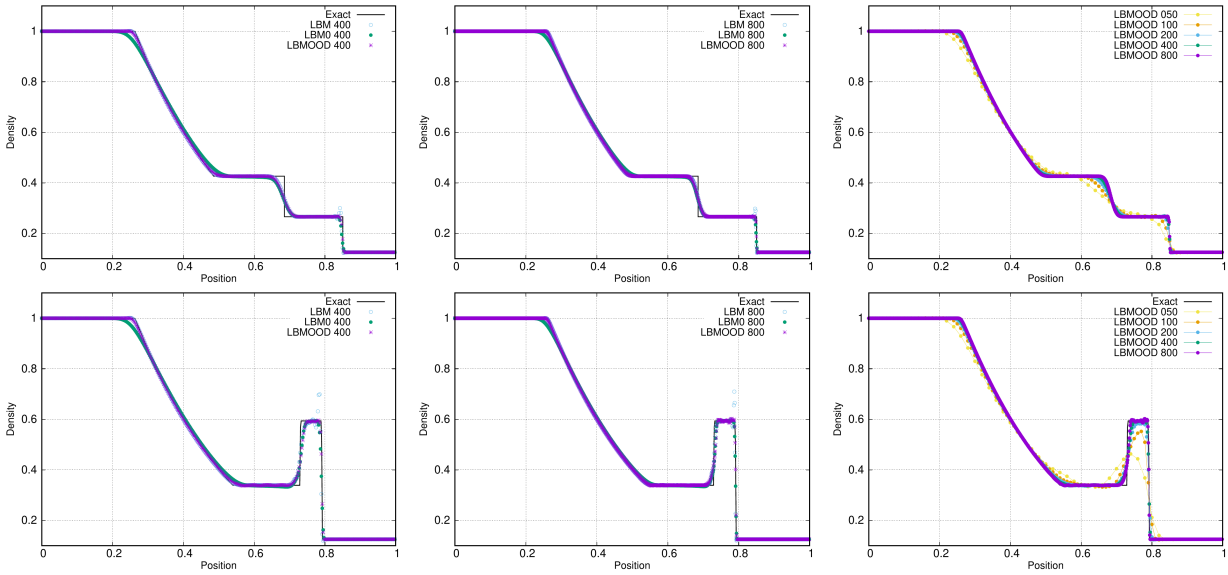

Figure 4: Sod (top line) and Sod2 (bottom line) 1D problems — Numerical density for LBM0 (green), LBM (blue) and LBMOOD (purple) schemes with 400 (left panels) and 800 (middle panels) cells. The right-most panels present the mesh convergence for LBMOOD scheme for 5 successively refined meshes.

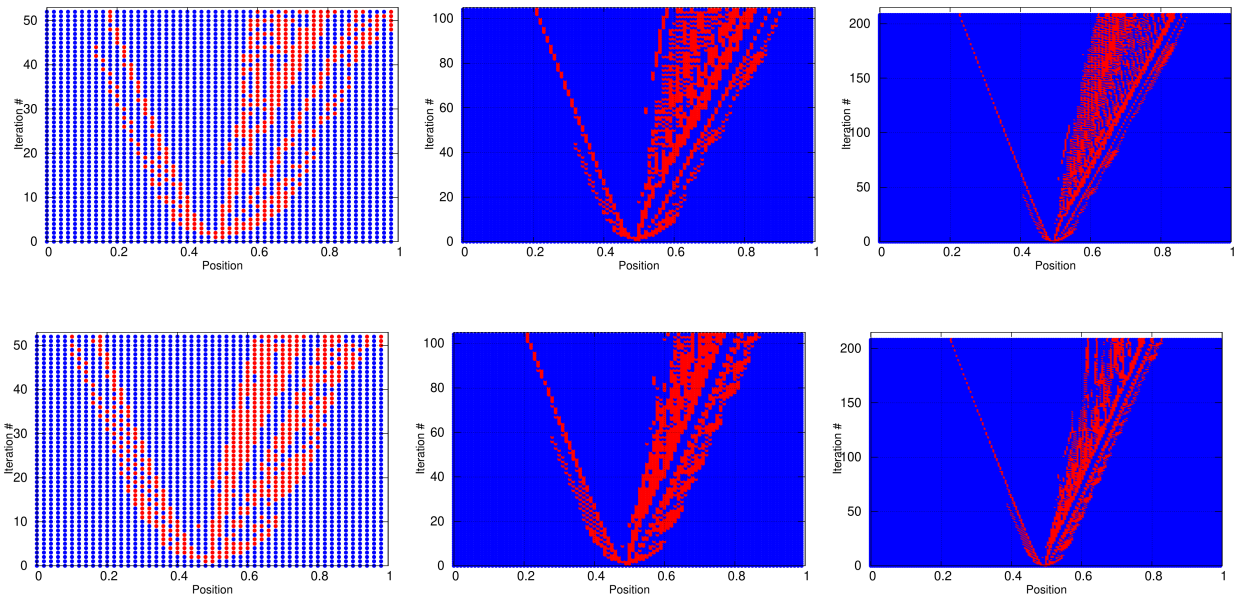

Figure 5: Sod (top line) and Sod2 (bottom line) 1D problems solved by LBMOOD scheme using 50 (left), 100 (middle) and 200 (right) cells — Bad cells (red) detected by as a function of position (horizontal) and iteration number (vertical) — Red/bad cells are updated with LBM0 scheme, while blue/good ones with LBM.

percentage of troubled cells detected by LBMOOD scheme over an entire simulation P. This percentage of troubled cells for the entire simulation is computed as $P = \frac{100}{N_t N_c} \sum_{n=0}^{N_t} N_b^n$, where N_c is the number of cells, $N_t \in \mathbb{N}^*$ the number of time iterations, and, for each time step $t^n > 0$, N_b^n is the amount of troubled cells. This percentage is computed for all 1D simulations presented in this section and graphically represented also in figure 6. It is obvious that the percentage of bad cells tends to negligible amount when the mesh gets refined. As such it is clear that the extra-cost brought by MOOD approach becomes smaller and smaller.

Value of P. % of troubled cells integrated over the simulation					
Problems	50× 50	100× 100	200× 200	400× 400	800× 800
Contact	25.03%	16.72%	7.08%	2.42%	1.58%
Sod	16.80%	12.14%	8.84%	6.48%	4.26%
Sod2	21.13%	12.43%	7.01%	4.51%	3.02%

Table 3: Percentage of troubled cells for the 1D test cases when 50×50 up to 800×800 cells are employed.

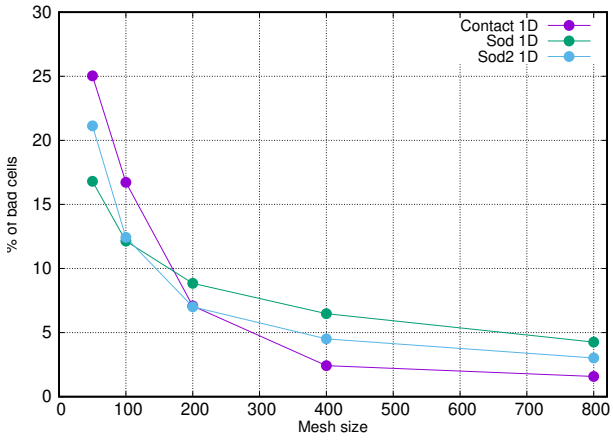

Figure 6: Curves from the data in Table 3 — Percentage of troubled cells for the 1D test cases when 50×50 up to 800×800 cells are employed.

4.2. 2D tests

In this section 2D test cases are simulated, and, LBMOOD is set up to employ the lattice neighborhood for the correction.

4.2.1. Entropy spot

The linear properties of our approach are further investigated through the transport of an inviscid entropy spot. A Gaussian spot of density is superimposed to a uniform velocity mean flow at constant pressure:

$$\rho = [1 + 10^{-3} e^{-r^2}], \qquad p = 0, \qquad u_\alpha = (u_1, u_2) = (2, 2), \tag{26}$$

where r is the relative radius $r^2 = [(x - x_c)^2 + (y - y_c)^2]$, where $(x_c, y_c) = (5, 5)$ are the coordinates of the spot center. The computational domain is $\Omega = [-5 : 15] \times [-5 : 15]$ and covered with 200×200 cells. Periodic boundary conditions are considered and the final time is set to $t_{\text{final}} = 1$ so that the spot has traveled across the domain and is back to its original location, so that errors with respect to the exact solution can be computed. In Table 4 we report these errors and the corresponding rates of convergence. From these data it is clear that LBM0 and LBM produce 2nd order converging numerical solutions: LBM being two times more accurate than LBM0. Importantly, LBMOOD produces exactly the same results than LBM because no bad cell is detected. And this was expected for a smooth solution.

	LBM0		LBM		LBMOOD		
$N \times N$	L^2 error	order	L^2 error	order	L^2 error	order	% bad cells
50	2.24×10^{-5}	—	1.22×10^{-5}	—	1.22×10^{-5}	—	0
100	0.68×10^{-5}	1.7	0.33×10^{-5}	1.9	0.33×10^{-5}	1.9	0
150	0.33×10^{-5}	1.9	0.14×10^{-5}	2.0	0.14×10^{-5}	2.0	0
200	0.19×10^{-5}	1.8	0.09×10^{-5}	1.6	0.09×10^{-5}	1.6	0
Expected →		1		2		2	0%

Table 4: L^2 errors and experimental convergence rate for the entropy spot problem for LBM0, LBM and LBMOOD schemes — The percentage of bad cells is an average for all time steps.

In figure 7 we plot the numerical density for the LBM0, LBM, and LBMOOD schemes. The top panels present the colored density in 2D, then the bottom ones display the 1D density as a function of the radius r for all cells. The exact solution is under the the numerical solutions. The three schemes can capture this smooth entropy spot, and can not be discriminated with this test case.

4.2.2. Isentropic vortex in motion

Next a slightly more complex problem is simulated. The isentropic vortex problem was initially introduced in 2D space in [18] to test the accuracy of numerical methods since the exact analytical solution is smooth and is not trivial, i.e all variables do evolve simultaneously. We consider the computational domain $\Omega = [0, 10] \times [0, 10]$ and an ambient flow characterized by $\rho_\infty = 1.0$, $u_{1,\infty} = 1.0$, $u_{2,\infty} = 1.0$, $p_\infty = 1.0$, with a normalized ambient temperature $T^*_\infty = 1.0$ computed with the perfect gas equation of state and $\gamma = 7/5$. A vortex is centered at $(x_{\text{vortex}}, y_{\text{vortex}}) = (5, 5)$ and supplemented to the ambient gas at the initial time $t = 0$ with the following conditions $u_1 = u_{1,\infty} + \delta u_1$, $u_2 = u_{2,\infty} + \delta u_2$, $T^* = T^*_\infty + \delta T^*$ where

$$\delta u_1 = -y' \frac{\beta}{2\pi} \exp\left(\frac{1 - r^2}{2}\right), \quad \delta u_2 = x' \frac{\beta}{2\pi} \exp\left(\frac{1 - r^2}{2}\right), \quad \delta T^* = -\frac{(\gamma - 1)\beta}{8\gamma\pi^2} \exp\left(1 - r^2\right),$$

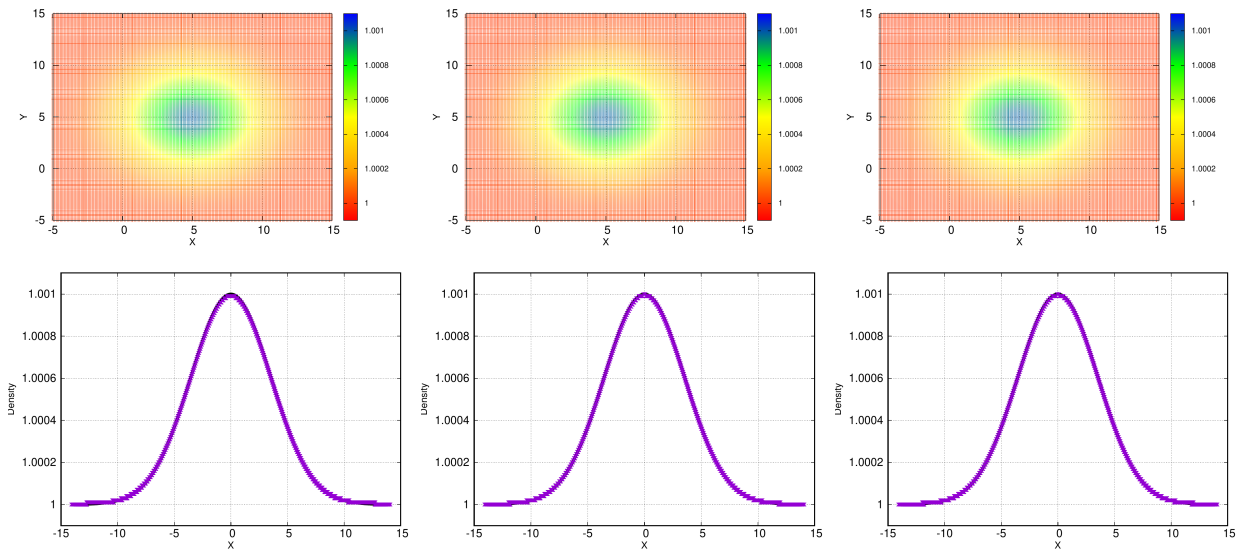

Figure 7: Entropy spot problem — 200×200 cells — LBM0 (left), LBM (middle) and LBMOOD (right) schemes — Top panels: density (color). Bottom panels: density as a function of radius r.

with $r = \sqrt{x'^2 + y'^2}$ and $x' = x - x_{\text{vortex}}, y' = y - y_{\text{vortex}}$. The vortex strength is given by $\beta = 5.0$ and the initial density follows the relation

$$\rho = \rho_\infty \left(\frac{T^*}{T^*_\infty}\right)^{\frac{1}{\gamma-1}} = \left(1 - \frac{(\gamma-1)\beta}{8\gamma\pi^2}\exp\left(1-r^2\right)\right)^{\frac{1}{\gamma-1}}. \tag{27}$$

Periodic boundary conditions are prescribed everywhere. At final time $t_{\text{final}} = 10$ the vortex is back to its original position. In Table 5 we report the errors and the corresponding rates of convergence.

	LBM0		LBM		LBMOOD		
$N \times N$	L^2 error	order	L^2 error	order	L^2 error	order	% bad cells
50	3.19×10^{-4}	-	1.53×10^{-4}		1.53×10^{-4}		0
100	3.23×10^{-4}	-	0.49×10^{-4}	1.6	0.49×10^{-4}	1.6	0
150	3.22×10^{-4}	-	0.24×10^{-4}	1.8	0.24×10^{-4}	1.8	0
200	3.17×10^{-4}	-	0.14×10^{-4}	1.9	0.14×10^{-4}	1.9	0
Expected $\rightarrow$		1		2		2	0%

Table 5: L^2 errors and convergence rate for the isentropic vortex problem for LBM0, LBM and LBMOOD schemes — The percentage of bad cells is an average for all time steps.

Next, in figure 8 we plot the density for the LBM0, LBM, and LBMOOD schemes in 2D (top panels) and 1D as a function of the radius r (bottom panels). From these results we can observe that the LBM0 scheme can not maintain the vortex shape while LBM can. Accordingly LBMOOD maintains the shape because the number of bad cells is

close to 0, consequently LBMOOD results are extremely close to LBM ones. This test verifies that a smooth solution is not spoiled by MOOD procedure.

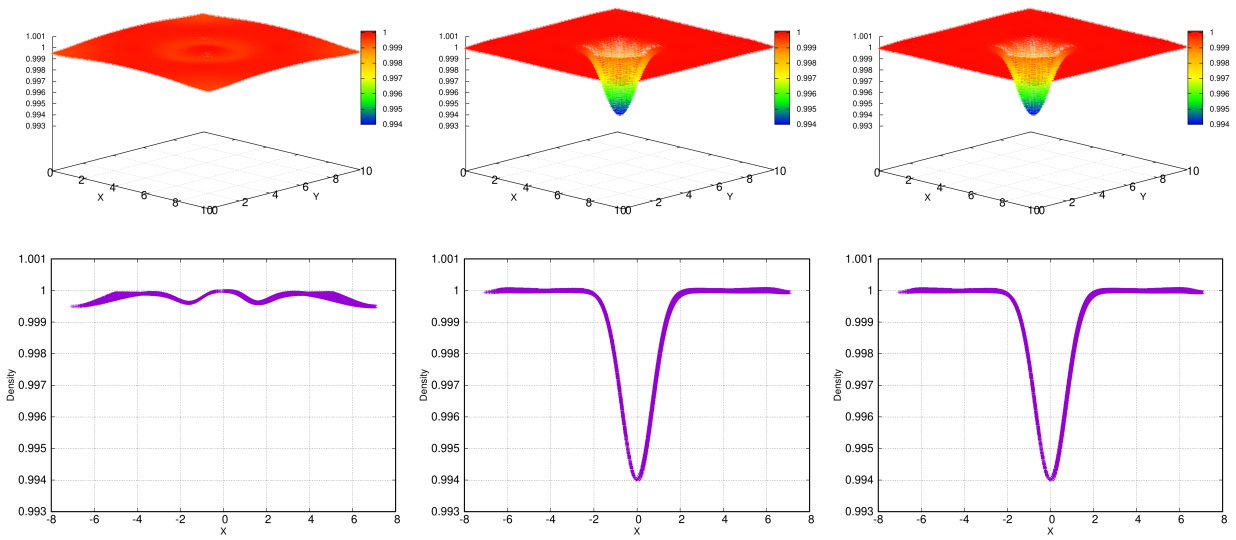

Figure 8: Vortex problem — 200 × 200 cells — LBM0 (left), LBM (middle) and LBMOOD (right) schemes — Top panels: density (color and azimuth). Bottom panels: density as a function of radius r.

4.2.3. 2D Sod problem

The 2D Sod problem, also called 'Explosion problem' consists in the same initial data of Sod 1D where the Left state is inside a disk of radius $r = 0.5$, and the Right one outside. The computational domain is $\Omega = [-1, 1] \times [-1, 1]$ and the final time is set to $t_{\text{final}} = 0.15$. The exact solution is radial, made of a converging rarefaction wave and diverging contact and shock waves. In-between these waves, because of the cylindrical geometry, the state is not constant anymore. In figure 9 we compare LBM0, LBM and LBMOOD scheme results. We plot the numerical density as a function of cell center radius for all cells when a 200×200 mesh is employed. As expected LBM0 scheme is overly diffusive especially at the rarefaction wave and some lack of symmetry is observed. Next, LBM scheme results in the middle panel presents some overshoots (close to the shock and the head of the rarefaction)/ but is visibly more accurate especially on the rarefaction wave. Then, the results of LBMOOD scheme are a rather good compromise between these two previous solutions: the dissipation is not too excessive and the oscillations have been reduced. In figure 10 the top panels display the 2D density in colors for meshes of size $N \times N$ with $N = 50$, 100 and 200 for LBMOOD scheme. As the mesh is refined the solution is more accurate as expected, the waves are sharply captured. More interestingly, on bottom panels of figure 10 we plot the troubled cells in red and the untouched ones in blue for the last time iteration. The troubled cells have been updated by LBM0 scheme and the good ones updated with LBM. We can observe that the troubled cells are mainly located at the cylindrical discontinuities (shock, contact,

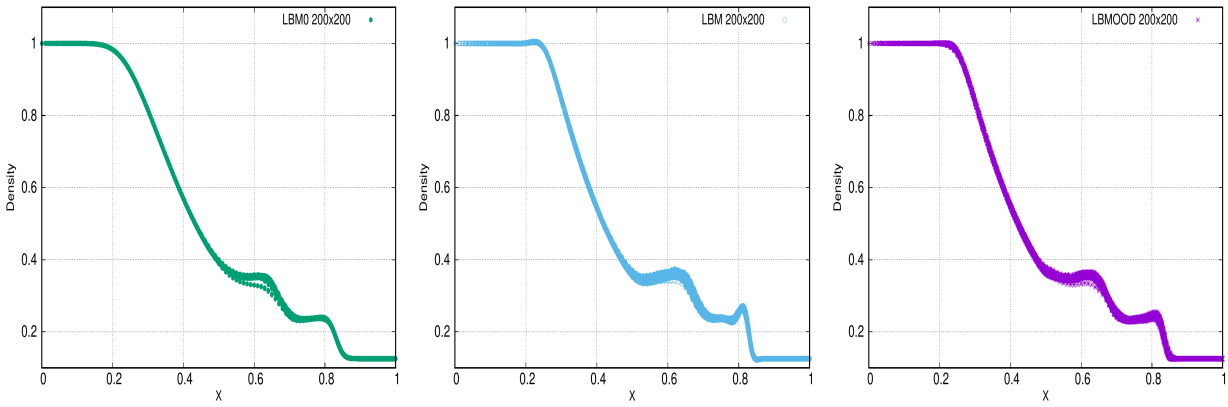

Figure 9: Sod problem in 2D — Numerical density for all 200×200 cells as a function of cell radius — LBM0 (left), LBM (middle) and LBMOOD (right).

head/tail of rarefaction), along with some spurious cells which requires more dissipation according to our detection criteria.

4.2.4. 2D Riemann problems

The 2D Riemann problems have been chosen in order to address complex 2D flow structures resulting from interactions between discontinuities. They demonstrate the ability of the LBMOOD method to handle such calculations without using additional stabilisation tools.

These 2D 4-state Riemann problems are described and simulated in [37] for instance. They consist of four quadrants with constant fluid states meeting at a quadruple point. The computational domain is $\Omega = [0,1]\times[0,1]$, the quadruple point is located at $(0.5, 0.5)$ and generally a mesh made of 400×400 cells is employed. The states $U_k = (\rho_k, u_k, v_k, p_k)$, $k = 1,2,3,4$ are located as $\begin{array}{|c|c|}\hline U_2 & U_1 \\ \hline U_3 & U_4 \\ \hline\end{array}$. We focus on three specific configurations labeled 3, 4 and 12 in [37] from which the reference solutions are borrowed.

These test cases are used to observe (i) the comparison of the solutions obtained by LBMOOD vs LBM0, (ii) the convergence of the results calculated by using LBMOOD, and, (iii) the effect of the number of corrected neighbours on the obtained solution by LBMOOD. Notice that LBM scheme fails for such test cases.

Configuration 3. This configuration is relatively complex due to the presence of small scaled vortices. It is initialised with $\rho_1 = 1.5$, $\rho_2 = \rho_4 = 0.5323$, $\rho_3 = 0.138$, $p_1 = 1.5$, $p_2 = p_4 = 0.3$, $p_3 = 0.029$ and $u_2 = u_3 = v_3 = v_4 = 1.206$ and $u_1 = v_1 = v_2 = u_3 = 0$. The time step is set as $dt/dx = 0.13$ and the final time is set to 0.3. For this test LBMOOD corrects 3 neighbors. LBM0 and LBMOOD scheme results are plotted in figure 11 where we see the complex flow patterns. The ability to sharply capture such small scale structures is an important feature of LB methods because they occur in more complex flows, e.g. turbulent flows. LBM0 scheme produces a diffused solution, while LBMOOD is able to visibly reduce the dissipation, leading to a more accurate numerical solution on the same mesh. Hence, the

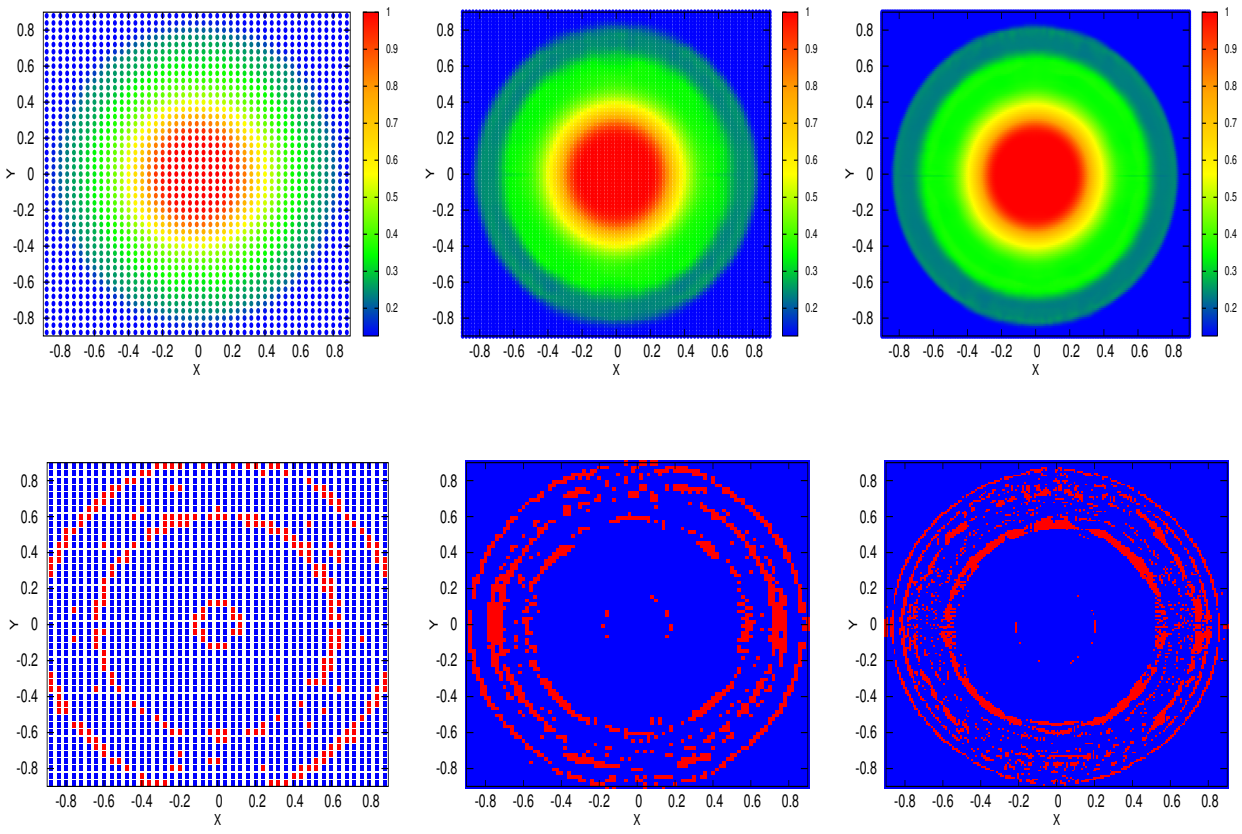

Figure 10: Sod problem in 2D — LBMOOD scheme — Top panels: Numerical density — Bottom panels: Bad cells in red — 50×50 (left), 100×100 (middle) and 200×200 (right) cells.

mix LBM0-LBM that MOOD is operating via the detection procedure makes the LBMOOD scheme robust, and, at the same time more accurate than LBM0.

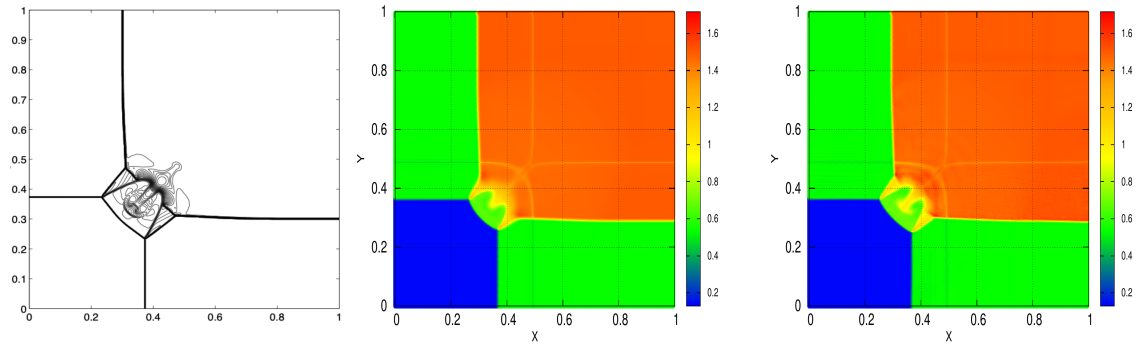

Figure 11: Riemann problem configuration 3 — Numerical density — 400×400 cells — LBM0 (middle) and LBMOOD (right) numerical density — Reference solution (left) from [37].

Next in figure 12 we present a mesh convergence study for LBMOOD scheme using $N \times N$ cells with $N = 100$ (left), 200 (middle) and 300 (right). The solution for 400×400 is displayed on figure 11-right panel. LBMOOD produces more accurate results as the mesh gets refined without spurious features as expected. We have observed

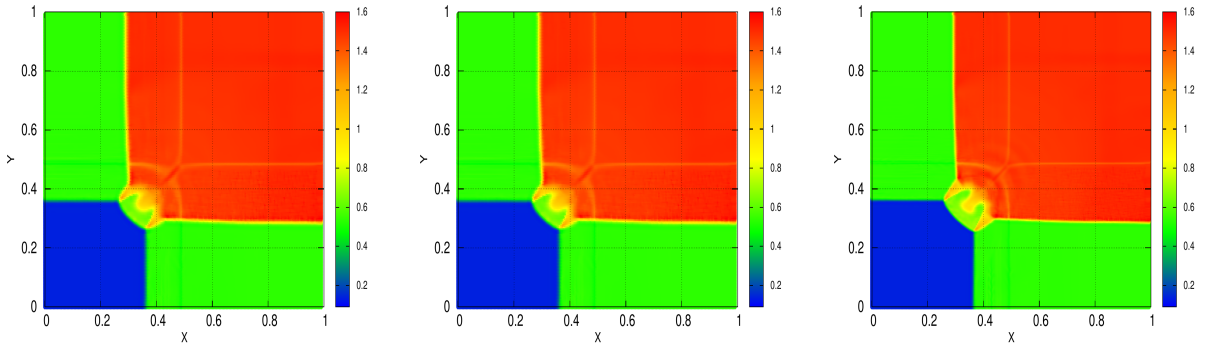

Figure 12: Riemann problem configuration 3 — Numerical density — LBMOOD scheme — $N \times N$ cells with $N = 100$ (left), 200 (middle) and 300 (right).

that the larger the neighborhood, the more diffused the solution becomes (we have simulated with 8 neighbors, not shown here). As already mentioned, this fact does not justify the use of too large a neighborhood for the correction in LBMOOD

Configuration 4. This configuration involves relatively weak discontinuities and is initialised as $\rho_1 = \rho_3 = 1.1$, $\rho_2 = \rho_4 = 0.5065$, $p_1 = p_3 = 1.1$, $p_2 = p_4 = 0.35$, and $u_1 = v_1 = v_2 = u_4 = 0$, $u_2 = u_3 = v_4 = 0.8939$ The time step is

set as $dt/dx = 0.19$ and the final time is set to 0.25.

The comparison of the final solutions is proposed in figure 13. Both schemes, LBM0 and LBMOOD, are capable of producing the solutions with, however, a more dissipative result by LBM0. This drawback is visibly improved on LBMOOD solution. Only lattice neighbors are corrected by LBMOOD for this configuration. We observe the convergence of the results (see figure 14 by varying the number of points in computational domain from 100×100 to 300×300. The numerical solution gets more accurate with finer mesh, and, seems to converge towards the reference solution.

Furthermore, the solution computed by LBMOOD with only immediate corrected neighbours is sharper in comparison to 3, 5 or 8 neighbours, see figure 15. This is an expected result, and, such an increase of corrected neighbours can be used in order to dissipate more, for instance in presence of stronger discontinuities.

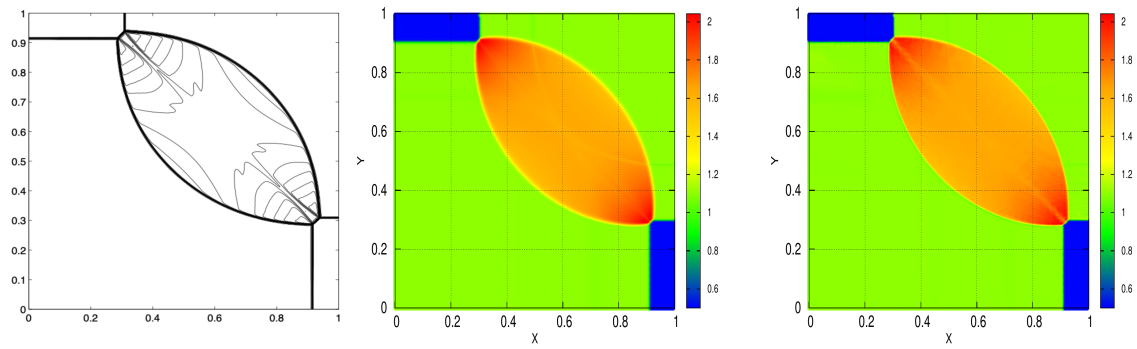

Figure 13: Riemann problem configuration 4 — Numerical density — 400×400 cells — LBM0 (middle) and LBMOOD (right) numerical density — Reference solution (left) from [37].

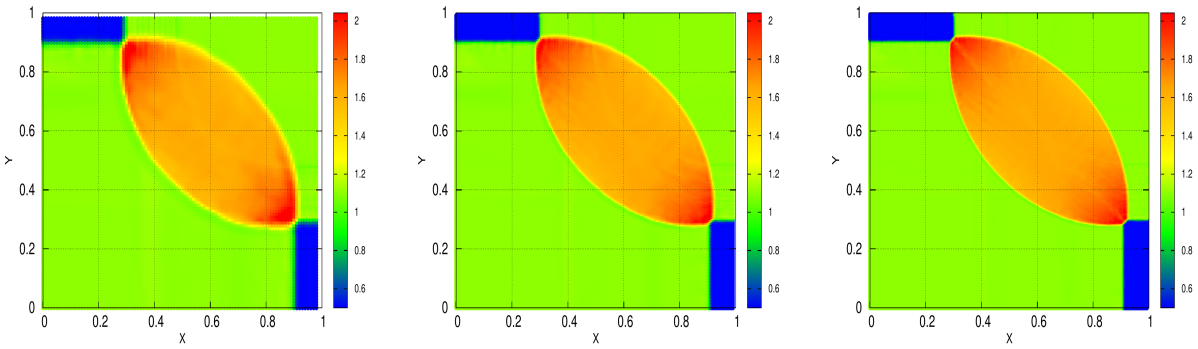

Figure 14: Riemann problem configuration 4 — Numerical density — LBMOOD scheme — $N \times N$ cells with $N = 100$ (left), 200 (middle) and 300 (right).

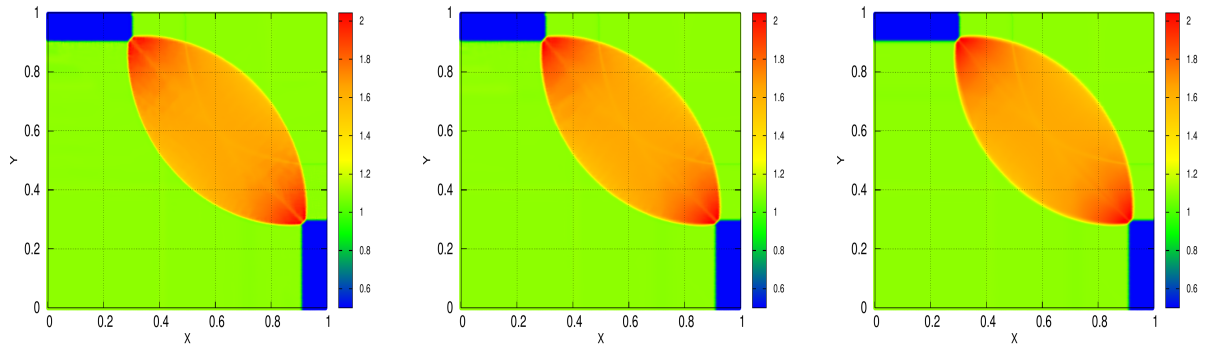

Figure 15: Riemann problem configuration 4 — Numerical density — LBMOOD scheme for 400×400 cells — Increase the number of neighbor cells corrected to 3 (left), 5 (middle) or 8 (right).

Configuration 12. This configuration involves smaller scaled vortices, and, thus, can be considered as a more challenging test case. Initial data are $\rho_2 = \rho_4 = 1$, $\rho_1 = 0.5313$ and $\rho_3 = 0.8$, $p_2 = p_3 = p_4 = 1$, $p_1 = 0.4$, and $u_1 = v_1 = v_2 = u_3 = v_3 = u_4 = 0$, $v_4 = u_2 = 0.7276$ The time step is set as $dt/dx = 0.19$ and the final time is set to 0.25.

The comparison of the final solutions is proposed in figure 16. Both schemes, LBM0 and LBMOOD, are capable of producing the solutions with, however, a dissipative result by LBM0. This drawback of the solution is improved by LBMOOD when only one neighbor is corrected. However, some instabilities are present in the solution of LBMOOD. Fortunately this situation can be improved by increasing number of corrected cells as can be seen in figure 17. Choosing the number of neighbors to be corrected in LBMOOD is a parameter to be fixed by the user. At last we present in figure 18 the bad cells at last time iteration for different sizes of corrected neighborhood, from 1 (lattice neighbors) to 8. As expected the smallest neighborhood has less damped the solution during the simulation, hence more bad cells remain. Conversely, using larger and larger neighborhood forces LBMOOD to damp the spurious oscillations, producing a smoother solution, and, consequently reducing the number of remaining bad cells at the end of the simulation.

5. Discussion and food for thought

Although the previous numerical section has validated our proof of concept of coupling MOOD with LB method for compressible inviscid flows, more benefits can already be anticipated.

More advanced models of PDEs. LBMOOD scheme is directly usable with more complex or different systems of PDEs involving source terms, non-conservative products, involutions, etc. For instance, compressible or incompressible Navier-Stokes, multi-phase flows, detonation models, for which LB methods are already employed. Obviously with such different models, different physical processes are to be expected. Therefore the PAD/NAD criteria from

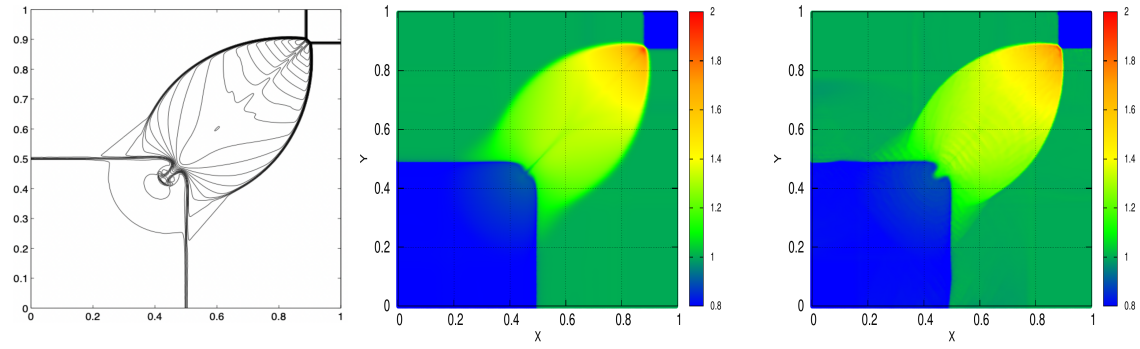

Figure 16: Riemann problem configuration 12 — Numerical density — 400×400 cells — LBM0 (middle) and LBMOOD (right) numerical density — Reference solution (left) from [37].

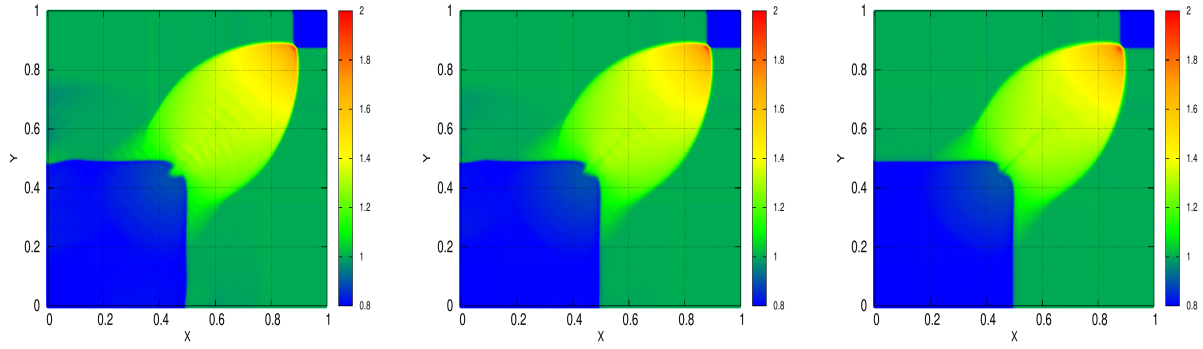

Figure 17: Riemann problem configuration 12 — Numerical density — LBMOOD scheme for 400 × 400 cells — Increase the number of neighbor cells corrected to 3 (left), 5 (middle) or 8 (right).

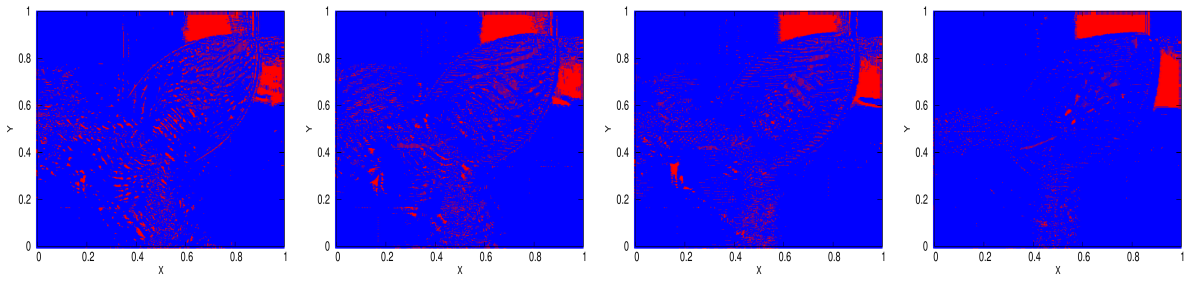

Figure 18: Riemann problem configuration 12 — Bad cells — LBMOOD scheme for 400 × 400 cells and, from left to right, 1, 3, 5 and 8 corrected neighbors.

MOOD (see section 3.2) must be appropriately adapted, see [22] for examples of detection criteria for complex models of PDEs.

More advanced 'parachute' LB method. The parachute scheme being the last scheme to be tried on troubled cells, it must fulfill the detection criteria by construction. In this paper the toy model of compressible fluid flows is relatively simple, making LBM0 a valid parachute scheme. However LBM0 may not always be such a parachute if more complex systems of PDEs are to be solved. In this case, especially if the PAD/NAD criteria have been modified, the parachute scheme may require some adaptation. Some work is indeed mandatory to design provably robust LB methods for each model of PDEs.

More advanced cascade of LB methods. Yet another possible evolution would be to test a few LB schemes in the so-called MOOD cascade [21, 22], see figure 19. In this article our simple cascade was composed by LBM and LBM0 schemes. Let us define the function $\theta(x) = \left(1 - \frac{\Delta t}{x}\right)$, which is the coefficient in front of f^{neq} in (13). LBM corresponds to setting $\theta(\tau) = \left(1 - \frac{\Delta t}{\tau}\right) \equiv \bar{\theta}$ while LBM0 to $\theta(\Delta t) = 0$. However one could add few intermediate values of θ in between these extremes values. For instance by reducing θ by the same factor $\delta > 1$, say 2, or 5. Hence $\theta^{\nu+1} = \frac{\theta^{\nu}}{\delta}$, where $0 \leq \nu$ is the index numbering the LB schemes. Obviously, $\lim_{\nu \to +\infty} \theta^{\nu} \longrightarrow 0$, and, numerically with only few iterations, θ^{ν} rapidly drops to 0, leading to employing LBM0 scheme. Doing so, a troubled cell could be re-computed with an intermediate scheme between LBM and LBM0. Because only few cells require to be recomputed, the extra CPU time spent dropping in the cascade should not be extremely large[2]. With more advanced 3D models, when refining the mesh is no more feasible, this slight gain in terms of accuracy may become important.

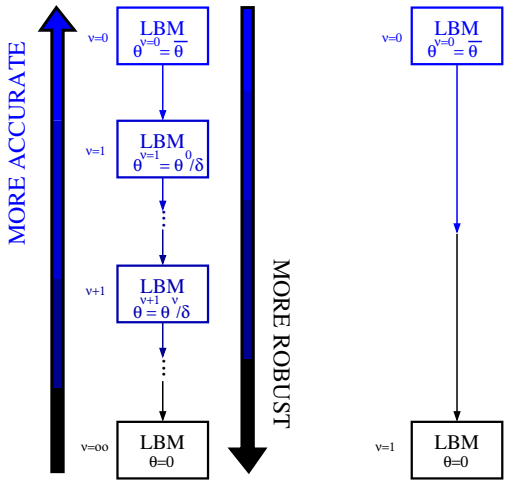

Figure 19: Left: Advanced cascade of LB methods that could be employed within a MOOD approach. Right: the cascade used in this paper to produce the numerical results.

[2]This approach mimics the use of MOOD within a Finite Volume framework [20–22].

More advanced implementation and codes. LBMOOD has no limitation when it comes to multiple dimensions, meaning that the extension to 3D is theoretically trivial. One difficulty is however the efficient massive parallelisation of LBMOOD. Because we can not anticipate how many troubled cells are detected and recomputed, then the computing load between processors is difficult to be maintained balanced. However, as already mentioned and seen in the numerical results, few troubled situations are generally detected. Therefore the expensive stage is the computation of LBM solution (and the first detection). This stage can be parallelised in the classical way because the same operations are made for each and every cell. Next, if troubled cells are detected, it becomes more complex to produce a balanced parallel code. One expects however that the recomputation involves so few cells that the parallelisation efficiency is maintained. At last, as already mentioned in section 2.4, tackling total energy equation by the extension of the lattice quadrature rules is not an issue for MOOD.

6. Conclusions and perspectives

This paper has presented a proof of concept of the hydridization of two Lattice-Boltzmann schemes by means of the *a posteriori* MOOD paradigm to solve the compressible fluid flows in its inviscid limit. A second-order LB method prone to instability is coupled with a more robust one within MOOD to produce a robust yet-accurate hybrid LB scheme in 1D and 2D. The MOOD paradigm detects a candidate solution produced by the accurate LBM scheme. If troubled cells are detected, they are further recomputed with LBM0 in a robust way. The detection criteria are physically and numerically based.

This hybrid LB scheme has been validated on a set of classical test cases involving smooth and discontinuous solutions. On these test cases the hybrid LB scheme is as accurate than the LBM scheme when this later is usable, but, more important is as robust than LBM0 on difficult situations involving shock waves for instance.

In the latest section we have discussed some lines of evolution for this LBMOOD method. Our future studies will focus on applying it to detonation models and to implement/validate the 3D version which does not present any theoretical difficulty.

Acknowledgments

We would like to acknowledge the partial financial support of the LRC Anabase. Part of this research was supported by the French project BALBUZARD funded by DGAC and supported by Next generation EU in the frame of "Plan national de Relance et de Résilience français (PNRR)". Part of this research has been performed during the SHARK-FV workshops `https://shark-fv.eu/`.

References

[1] S. Succi, The Lattice Boltzmann Equation for Fluid Dynamics and Beyond, Clarendon Press, Oxford, 2001.

[2] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E. M. Viggen, The Lattice Boltzmann Method, 1st Edition, Graduate Texts in Physics, Springer, Cham, 2017. doi:10.1007/978-3-319-44649-3.
URL https://link.springer.com/book/10.1007/978-3-319-44649-3

[3] J. Latt, C. Coreixas, J. Beny, Cross-platform programming model for many-core lattice Boltzmann simulations, Plos one 16 (4) (2021) e0250306.

[4] R. Nourgaliev, T. Dinh, T. Theofanous, D. Joseph, The lattice Boltzmann equation method: theoretical interpretation, numerics and implications, International Journal of Multiphase Flow 29 (1) (2003) 117–169. doi:https://doi.org/10.1016/S0301-9322(02)00108-8.
URL https://www.sciencedirect.com/science/article/pii/S0301932202001088

[5] S. A. Hosseini, P. Boivin, D. Thévenin, I. Karlin, Lattice Boltzmann methods for combustion applications, Progress in Energy and Combustion Science 102 (2024) 101140.

[6] J. Meng, Y. Zhang, Diffuse reflection boundary condition for high-order lattice Boltzmann models with streaming–collision mechanism, Journal of Computational Physics 258 (2014) 601–612.

[7] M. Saadat, S. Hosseini, B. Dorschner, I. Karlin, Extended lattice Boltzmann model for gas dynamics, Physics of Fluids 33 (4) (2021) 046104.

[8] Y. Feng, P. Boivin, J. Jacob, P. Sagaut, Hybrid recursive regularized thermal lattice Boltzmann model for high subsonic compressible flows, Journal of Computational Physics 394 (2019) 82 – 99.

[9] G. Farag, S. Zhao, T. Coratger, P. Boivin, G. Chiavassa, P. Sagaut, A pressure-based regularized lattice-Boltzmann method for the simulation of compressible flows, Physics of Fluids 32 (6) (2020) 066106.

[10] G. Farag, T. Coratger, G. Wissocq, S. Zhao, P. Boivin, P. Sagaut, A unified hybrid lattice-Boltzmann method for compressible flows: bridging between pressure-based and density-based methods, Physics of Fluids 33 (8) (2021) 086101.

[11] S. Guo, Y. Feng, P. Sagaut, On the use of conservative formulation of energy equation in hybrid compressible lattice Boltzmann method, Computers & Fluids 219 (2021) 104866.

[12] F. Renard, G. Wissocq, J.-F. Boussuge, P. Sagaut, A linear stability analysis of compressible hybrid lattice Boltzmann methods, Journal of Computational Physics 446 (2021) 110649.

[13] G. Wissocq, T. Coratger, G. Farag, S. Zhao, P. Boivin, P. Sagaut, Restoring the conservativity of characteristic-based segregated models: application to the hybrid lattice Boltzmann method, Physics of Fluids 34 (4) (2022) 046102.

[14] G. Wissocq, S. Taileb, S. Zhao, P. Boivin, A hybrid lattice Boltzmann method for gaseous detonations, Journal of Computational Physics 494 (2023) 112525.

[15] J. von Neumann, R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, Journal of Applied Physics 21 (1950) 232–237.
URL https://api.semanticscholar.org/CorpusID:119579536

[16] B. van Leer, Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme, Journal of Computational Physics 14 (4) (1974) 361–370. doi:https://doi.org/10.1016/0021-9991(74)90019-9.
URL https://www.sciencedirect.com/science/article/pii/0021999174900199

[17] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics 77 (2) (1988) 439–471. doi:https://doi.org/10.1016/0021-9991(88)90177-5.
URL https://www.sciencedirect.com/science/article/pii/0021999188901775

[18] C. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, NASA/CR-97-206253 ICASE Report No.97-65 (November 1997).

[19] S. Clain, S. Diot, R. Loubère, A high-order finite volume method for systems of conservation laws – Multi-dimensional Optimal Order Detection (MOOD), J. Comput. Phys. 230 (10) (2011) 4028 – 4050. doi:10.1016/j.jcp.2011.02.026.
URL http://www.sciencedirect.com/science/article/pii/S002199911100115X

[20] S. Diot, S. Clain, R. Loubère, Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials, Computers and Fluids 64 (2012) 43 – 63. doi:10.1016/j.compfluid.2012.05.004.

URL http://www.sciencedirect.com/science/article/pii/S0045793012001909

[21] S. Diot, R. Loubère, S. Clain, The MOOD method in the three-dimensional case: Very-high-order finite volume method for hyperbolic systems, International Journal of Numerical Methods in Fluids 73 (2013) 362–392.

[22] R. Loubère, M. Dumbser, S. Diot, A new family of high order unstructured MOOD and ADER finite volume schemes for multidimensional systems of hyperbolic conservation laws, Communication in Computational Physics 16 (2014) 718–763.

[23] O. Zanotti, M. Dumbser, R. Loubère, S.Diot, A posteriori subcell limiting for Discontinuous Galerkin finite element method for hyperbolic system of conservation laws, J. Comput. Phys. 278 (2014) 47–75.

[24] X. Nogueira, L. Ramírez, S. Clain, R. Loubère, L. Cueto-Felgueroso, I. Colominas, High-accurate sph method with multi-dimensional optimal order detection limiting, Computer Methods in Applied Mechanics and Engineering 310 (2016) 134–155. doi:https://doi.org/10.1016/j.cma.2016.06.032.
URL https://www.sciencedirect.com/science/article/pii/S0045782516306697

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

[26] W. G. Vincenti, C. H. Kruger, Introduction to physical gas dynamics, Vol. 246, Wiley New York, 1965.

[27] S. Marié, D. Ricot, P. Sagaut, Comparison between lattice Boltzmann method and Navier–Stokes high order schemes for computational aeroacoustics, Journal of Computational Physics 228 (4) (2009) 1056–1070.

[28] G. Farag, S. Zhao, G. Chiavassa, P. Boivin, Consistency study of lattice-Boltzmann schemes macroscopic limit, Physics of Fluids 33 (3) (2021) 031701.

[29] D. d. Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, L.-S. Luo, Multiple–relaxation–time lattice Boltzmann models in three dimensions, Phil. Trans. R. Soc. Lond. A 360 (2002) 437–451.

[30] O. Malaspinas, B. Chopard, J. Latt, General regularized boundary condition for multi-speed lattice Boltzmann models, Computers & Fluids 49 (1) (2011) 29–35.

[31] J. Jacob, O. Malaspinas, P. Sagaut, A new hybrid recursive regularised bhatnagar–gross–krook collision model for lattice Boltzmann method-based large eddy simulation, Journal of Turbulence (2018) 1–26.

[32] Y. Feng, P. Sagaut, W. Tao, A three dimensional lattice model for thermal compressible flow on standard lattices, Journal of Computational Physics 303 (2015) 514–529.

[33] G. Wissocq, C. Coreixas, J.-F. Boussuge, Linear stability and isotropy properties of athermal regularized lattice Boltzmann methods, Phys. Rev. E 102 (2020) 053305. doi:10.1103/PhysRevE.102.053305.
URL https://link.aps.org/doi/10.1103/PhysRevE.102.053305

[34] O. Malaspinas, Increasing stability and accuracy of the lattice Boltzmann scheme: recursivity and regularization, arXiv preprint arXiv:1505.06900 (2015).

[35] G. Wissocq, S. Taileb, S. Zhao, P. Boivin, A hybrid lattice Boltzmann method for gaseous detonations, Journal of Computational Physics 494 (2023) 112525. doi:https://doi.org/10.1016/j.jcp.2023.112525.
URL https://www.sciencedirect.com/science/article/pii/S0021999123006204

[36] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, 2nd Edition, Springer, 1999.

[37] A. Kurganov, E. Tadmor, Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers, Numerical Methods for Partial Differential Equations 18 (5) (2002) 584–608. doi:10.1002/num.10025.
URL http://dx.doi.org/10.1002/num.10025

Appendix A. Unified density-based model with recursive regularised collision operator

The LBM part of this work is based on the unified density-based model of [10] with a recursive regularised collision operator. This model, which can be considered as an extension of previous pressure-based and density-based

models, is presented below. The equilibrium distribution for D2Q9 reads,

$$f_i^{eq} = \omega_i \rho \left\{ 1 + \frac{\omega_i - \delta_{0i}}{\omega_i}(\Theta - 1) + \frac{c_{i,\alpha} u_\alpha}{c_s^2} + \frac{(c_{i,\alpha} c_{i,\beta} - \delta_{\alpha\beta} c_s^2) u_\alpha u_\beta}{2c_s^4} + \frac{c_{i,y}(c_{i,x}^2 - c_s^2) u_x^2 u_y}{2c_s^6} + \frac{c_{i,x}(c_{i,y}^2 - c_s^2) u_x u_y^2}{2c_s^6} \right\}, \quad \text{(A.1)}$$

where $\Theta = T/T_{ref}$, $c_s = \sqrt{RT_{ref}}$ and ω_i are Gaussian weights of the lattice. T_{ref} is an arbitrary reference temperature linked to the mesh size Δx and time Δt through the following equation,

$$\sqrt{RT_{ref}} = \frac{1}{\sqrt{3}} \frac{\Delta x}{\Delta t}. \quad \text{(A.2)}$$

The recursive regularisation allows to compute the off-equilibrium distribution functions,

$$f_i^{neq} = \omega_i \left\{ \frac{(c_{i,\alpha} c_{i,\beta} - \delta_{\alpha\beta} c_s^2)}{2c_s^4} a_{\alpha\beta}^{neq,(2)} + \frac{c_{i,y}(c_{i,x}^2 - c_s^2)}{2c_s^6} a_{xxy}^{neq,(3)} + \frac{c_{i,x}(c_{i,y}^2 - c_s^2)}{2c_s^6} a_{xyy}^{neq,(3)} \right\}, \quad \text{(A.3)}$$

where

$$a_{\alpha\beta}^{neq,(2)} = \bar{\alpha}_{\alpha\beta}^{neq,(2)} - \frac{\delta_{\alpha\beta}}{D} \bar{a}_{\gamma\gamma}, \;\; \bar{a}_{\alpha\beta}^{neq,(2)} = \sum_i (c_{i,\alpha} c_{i,\beta} - c_s^2 \delta_{\alpha\beta})(f_i - f_i^{eq} + \frac{\Delta t}{2} F_i) \quad \text{(A.4)}$$

$$a_{xxy}^{neq,(3)} = 2u_x a_{xy}^{neq,(2)} + u_y a_{xx}^{neq,(2)}, \quad \text{(A.5)}$$

$$a_{xyy}^{neq,(3)} = u_x a_{yy}^{neq,(2)} + 2u_y a_{xy}^{neq,(2)}. \quad \text{(A.6)}$$

The correction term is

$$F_i^E = \frac{\omega_i}{2c_s^4} (c_{i,\alpha} c_{i,\beta} - c_s^2 \delta_{\alpha\beta}) a_{\alpha\beta}^{F,(2)}, \quad \text{(A.7)}$$

where for D2Q9 lattice we have

$$a_{\alpha\beta}^{F,(2)} = c_s^2 \delta_{\alpha\beta} (\rho \frac{\partial u_\gamma}{\partial \gamma} - \frac{\partial(\rho(1-\Theta))}{\partial t}) + c_s^2 (u_\alpha \frac{\partial(\rho(1-\Theta))}{\partial \beta} + u_\beta \frac{\partial(\rho(1-\Theta))}{\partial \alpha}) - \frac{\partial(\rho u_\alpha^3)}{\partial \alpha^3} \delta_{\alpha\beta}. \quad \text{(A.8)}$$

The first-order upwind scheme is adopted to discretise the spatial derivatives in (A.8), except for the velocity divergence where the second-order centered finite difference scheme is used, and, the time derivative where a temporal upwind scheme is employed.

Appendix B. Lattice Boltzmann fluxes

The expressions for the LB mass and momentum fluxes are proposed below with the ± notations standing for $x^{\pm} = x \pm \Delta x$ and $y^{\pm} = y \pm \Delta y$.

$$F^{\rho}_{\Delta x/2} = \frac{\Delta x}{\Delta t}[f_1^{coll}(x,y) - f_5^{coll}(x^+,y) + \frac{1}{2}f_2^{coll}(x,y^-) - \frac{1}{2}f_6^{coll}(x^+,y) + \frac{1}{2}f_2^{coll}(x,y) - \frac{1}{2}f_6^{coll}(x^+,y^+) - \frac{1}{2}f_4^{coll}(x^+,y^-) + \frac{1}{2}f_8^{coll}(x,y) - \frac{1}{2}f_4^{coll}(x^+,y) + \frac{1}{2}f_8^{coll}(x,y^+)] \tag{B.1}$$

$$F^{\rho}_{\Delta y/2} = \frac{\Delta x}{\Delta t}[f_3^{coll}(x,y) - f_7^{coll}(x,y^+) + \frac{1}{2}f_2^{coll}(x^-,y) - \frac{1}{2}f_6^{coll}(x,y^+) + \frac{1}{2}f_2^{coll}(x,y) - \frac{1}{2}f_6^{coll}(x^+,y^+) + \frac{1}{2}f_4^{coll}(x,y) - \frac{1}{2}f_8^{coll}(x^-,y^+) + \frac{1}{2}f_4^{coll}(x^+,y) - \frac{1}{2}f_8^{coll}(x,y^+)] \tag{B.2}$$

$$F^{\rho u_x}_{\Delta x/2} = \frac{\Delta x^2}{\Delta t^2}[f_1^{coll}(x,y) + f_5^{coll}(x^+,y) + \frac{1}{2}f_2^{coll}(x,y^-) + \frac{1}{2}f_6^{coll}(x^+,y) + \frac{1}{2}f_2^{coll}(x,y) + \frac{1}{2}f_6^{coll}(x^+,y^+) + \frac{1}{2}f_4^{coll}(x^+,y^-) + \frac{1}{2}f_8^{coll}(x,y) + \frac{1}{2}f_4^{coll}(x^+,y) + \frac{1}{2}f_8^{coll}(x,y^+)] \tag{B.3}$$

$$F^{\rho u_x}_{\Delta y/2} = \frac{\Delta x^2}{\Delta t^2}[\frac{1}{2}f_2^{coll}(x^-,y) + \frac{1}{2}f_6^{coll}(x,y^+) + \frac{1}{2}f_2^{coll}(x,y) + \frac{1}{2}f_6^{coll}(x^+,y^+) - \frac{1}{2}f_4^{coll}(x,y) - \frac{1}{2}f_8^{coll}(x^-,y^+) - \frac{1}{2}f_4^{coll}(x^+,y) - \frac{1}{2}f_8^{coll}(x,y^+)] \tag{B.4}$$

$$F^{\rho u_y}_{\Delta x/2} = \frac{\Delta x^2}{\Delta t^2}[f_2^{coll}(x,y^-) + f_6^{coll}(x^+,y) + \frac{1}{2}f_2^{coll}(x,y) + \frac{1}{2}f_6^{coll}(x^+,y^+) - \frac{1}{2}f_4^{coll}(x^+,y^-) - \frac{1}{2}f_8^{coll}(x,y) - \frac{1}{2}f_4^{coll}(x^+,y) - \frac{1}{2}f_8^{coll}(x,y^+)] \tag{B.5}$$

$$F^{\rho u_y}_{\Delta y/2} = \frac{\Delta x^2}{\Delta t^2}[f_3^{coll}(x,y) + f_7^{coll}(x,y^+) + \frac{1}{2}f_2^{coll}(x^-,y) + \frac{1}{2}f_6^{coll}(x,y^+) + \frac{1}{2}f_2^{coll}(x,y) + \frac{1}{2}f_6^{coll}(x^+,y^+) + \frac{1}{2}f_4^{coll}(x,y) + \frac{1}{2}f_8^{coll}(x^-,y^+) + \frac{1}{2}f_4^{coll}(x^+,y) + \frac{1}{2}f_8^{coll}(x,y^+)] \tag{B.6}$$