



**HAL**  
open science

# Grid refinement for aeroacoustics in the lattice Boltzmann method: A directional splitting approach

Félix Gendre, Denis Ricot, Guillaume Fritz, Pierre Sagaut

► **To cite this version:**

Félix Gendre, Denis Ricot, Guillaume Fritz, Pierre Sagaut. Grid refinement for aeroacoustics in the lattice Boltzmann method: A directional splitting approach. *Physical Review E*, 2017, 96 (2), pp.023311. 10.1103/PhysRevE.96.023311 . hal-04348563

**HAL Id: hal-04348563**

**<https://hal.science/hal-04348563>**

Submitted on 16 Dec 2023

**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.

**Grid refinement for aeroacoustics in the lattice Boltzmann method: A directional splitting approach**Félix Gendre,<sup>1,2,\*</sup> Denis Ricot,<sup>1,†</sup> Guillaume Fritz,<sup>1,‡</sup> and Pierre Sagaut<sup>2,§</sup><sup>1</sup>Renault, Technocentre, 1 av. du Golf, 78280 Guyancourt, France<sup>2</sup>Aix Marseille Université, CNRS, Centrale Marseille, M2P2 UMR 7340, 13451 Marseille, France

(Received 9 June 2017; published 29 August 2017)

This study focuses on grid refinement techniques for the direct simulation of aeroacoustics, when using weakly compressible lattice Boltzmann models, such as the D3Q19 athermal velocity set. When it comes to direct noise computation, very small errors on the density or pressure field may have great negative consequences. Even strong acoustic density fluctuations have indeed a clearly lower amplitude than the hydrodynamic ones. This work deals with such very weak spurious fluctuations that emerge when a vortical structure crosses a refinement interface, which may contaminate the resulting aeroacoustic field. We show through an extensive literature review that, within the framework described above, this issue has never been addressed before. To tackle this problem, we develop an alternative algorithm and compare its behavior to a classical one, which fits our in-house vertex-centered data structure. Our main idea relies on a directional splitting of the continuous discrete velocity Boltzmann equation, followed by an integration over specific characteristics. This method can be seen as a specific coupling between finite difference and lattice Boltzmann, locally on the interface between the two grids. The method is assessed considering two cases: an acoustic pulse and a convected vortex. We show how very small errors on the density field arise and propagate throughout the domain when a vortical flow crosses the refinement interface. We also show that an increased free stream Mach number (but still within the weakly compressible regime) strongly deteriorates the situation, although the magnitude of the errors may remain negligible for purely aerodynamic studies. A drastically reduced level of error for the near-field spurious noise is obtained with our approach, especially for under-resolved simulations, a situation that is crucial for industrial applications. Thus, the vortex case is proved useful for aeroacoustic validations of any grid refinement algorithm.

DOI: [10.1103/PhysRevE.96.023311](https://doi.org/10.1103/PhysRevE.96.023311)**I. INTRODUCTION**

Due to its high computational efficiency and to its low dissipative behavior, the lattice Boltzmann method (LBM) has been successfully applied in many turbulent flow simulations for thermal and nonthermal flows during the past two decades [1–5]. It has also been shown that this method had interesting properties for computational aeroacoustics at low Mach number [6,7]. Its *stream and collide* algorithm relies on a Cartesian grid, which makes the meshing of complex geometries straightforward. Since the computational cost of three dimensional industrial simulations with a uniform mesh would be prohibitive, grid refinement is a crucial issue in lattice Boltzmann simulations. However, it frequently causes numerical instabilities and nonphysical noise. We will show that this issue is crucial for direct simulation of aeroacoustics (which is the subject of this work, as opposed to hybrid methods with Ffowcs Williams and Hawkings integration, etc.), especially when vortical structures cross refinement interfaces.

In aeroacoustics, the physical key point is the distinction between two types of density or pressure fluctuations: the so-called *acoustic* and *hydrodynamic* (or *vortical*) fluctuations. They can be distinguished by their propagation speed (or equivalently by their wave number domains) and by their amplitude. The main problem is that the acoustic pressure fluctuations in a realistic flow are several orders of magnitude

weaker than the hydrodynamic fluctuations, a difference that depends on the nature of the flow and on the Mach number [8–10]. These considerations have two consequences: first, small spurious currents or errors on the density field can have great negative consequences on the resulting aeroacoustic field and, second, these spurious fluctuations cannot be seen on the instantaneous density field unless a precise closeup is performed.

In order to study the impact on aeroacoustics of a grid refinement algorithm with LBM, we choose to restrain ourselves to weakly compressible models in order to keep a low computational cost. In the literature review that follows, we will not consider numerical methods that are not based on the standard *stream and collide* lattice Boltzmann algorithm: this means we will not study the finite difference lattice Boltzmann method (FDLBM) [11–13], the finite volume lattice Boltzmann method (FVLBM) [12,14–16], nor the volumetric formulation of Chen *et al.* [17,18]. We also only consider papers in which nonuniform meshes are used. As far as the grid refinement algorithm itself is concerned, we will obviously not consider methods in which the speed of sound is not kept constant during the refinement process: this is the case when the so-called *diffusive* scaling [19] is used, such as in the work of Rheinländer [20].

The main characteristics that have to be taken into account to determine whether an existing study corresponds to our goals and constraints are the following:

(i) *The data structure: cell-centered or vertex-centered (also called cell vertex).* This feature has a major influence on the geometry of the refinement area, as well as on the interpolation and filtering schemes. Since our solver is based on a vertex-centered data structure, we choose this approach for this study.

\*felix.gendre@renault.com

†denis.ricot@renault.com

‡guillaume.fritz@renault.com

§pierre.sagaut@univ-amu.fr

(ii) *The spatial interpolation method.* All the LBM grid refinement methods rely somehow (see Sec. IV B for more details) on spatial interpolations (and sometimes also on temporal interpolations). As far as our subject is concerned, they can be grouped in three categories: *linear* interpolations (or bilinear/trilinear), *compact* interpolations [21–23], *polynomial* interpolations (Lagrange), and cubic *splines*. However, Guzik *et al.* [24] also use a constrained least-square algorithm. As we will show later, the spatial interpolation method may have an impact on the density field. Regarding time interpolation, its effect on the density field is less investigated. Several authors do not use time interpolation at all thanks to specific cell-centered algorithms [21,25–27]. These issues will be addressed in Sec. IV B.

(iii) *The value of the bulk viscosity.* Many multiple relaxation time methods use relaxation parameters that highly increase the value of the bulk viscosity, which yields a much more stable scheme. Although it is very useful for purely aerodynamic studies, it prevents any accurate work on

aeroacoustics, as shown in the article of Marié *et al.* [6]. For more clarifications on that matter, the reader may refer to the end of Sec. III B.

(iv) *The presence of a fine-to-coarse filtering method.* When transferring data from the fine mesh to the coarse mesh, a filtering method is sometimes applied. We will show in Sec. IV C why this item is important, particularly in the case of aeroacoustic studies.

(v) *The presence of acoustic wave propagation tests.* In such benchmarks, refinement areas are not crossed by vortices but only by acoustic waves.

(vi) *The presence of an aeroacoustic study in a vortical regime.* We mean by vortical regimes that vortices have to cross refinement interfaces during the simulation. In these cases, the Mach number ( $M$ ) of the vorticity mode is of paramount importance in the context of aeroacoustics on nonuniform meshes, as we will show later.

In the light of this criteria list, we summarize our literature review in Table I.

TABLE I. This table summarizes our literature review on numerical studies with LBM on nonuniform meshes in the light of the above defined criteria. CC: cell centered; CV: cell vertex; ✓: the criterion is fulfilled; ~: the criterion is fulfilled in several sections of the study; -: the criterion is not fulfilled; ?: no information available.

Article	Data structure	Spatial interpolation method	Unmodified bulk viscosity	Fine-to-coarse filtering	Acoustic validation	Aeroacoustics with vortical flow
Stiebler <i>et al.</i> [28]	CV	Polynomial	–	–	–	–
Geller <i>et al.</i> [12,26]	CV	Polynomial	–	–	–	–
Dorschner <i>et al.</i> [29]	CV	Polynomial	–	–	–	–
Kuwata <i>et al.</i> [30]	CV	Polynomial	–	–	–	–
Peng <i>et al.</i> [31]	CV	Polynomial	–	–	–	–
Pellerin <i>et al.</i> [32]	CV	Polynomial	–	✓	–	–
Eitel-Amor <i>et al.</i> [33]	CC	Linear	–	✓	–	–
Geier [21]	CC	Compact	–	✓	–	–
Geier <i>et al.</i> [27]	CC	Compact	–	✓	–	–
Chen <i>et al.</i> [34]	CV	Spline	~	–	–	–
Yu <i>et al.</i> [35,36]	CV	Polynomial	✓	–	–	–
Liu <i>et al.</i> [37]	CV	Spline	✓	–	–	–
Crouse [38] (Ph.D. thesis)	CV	Polynomial	✓	–	–	–
Filipova <i>et al.</i> [39,40]	CV	Polynomial	✓	–	–	–
Dupuis <i>et al.</i> [41]	CV	Polynomial	✓	–	–	–
Qi <i>et al.</i> [22]	CC	Compact	✓	✓	–	–
Touil <i>et al.</i> [42]	CV	Polynomial	✓	✓	–	–
Rohde <i>et al.</i> [43]	CC	Linear	✓	✓	–	–
Yu <i>et al.</i> [44]	CC	Linear	✓	✓	–	–
Lagrava <i>et al.</i> [45]	CV	Polynomial	✓	✓	–	–
Guzik <i>et al.</i> [24]	CC	Linear & constrained least-squares	~ <sup>a</sup>	✓	✓ <sup>a</sup>	–
Marié [7] (Ph.D. thesis)	CV	Linear	✓ <sup>b</sup>	–	✓	–
Pasquali [46] (Ph.D. thesis)	CC	Compact	?	✓	–	✓ (Very low $M$ , very highly resolved case <sup>c</sup> )
Hasert [47] (Ph.D. thesis)	CC	Linear	?	✓	✓	✓ ( $M_{\max} = 0.15$ , high levels of spurious acoustics <sup>d</sup> )

<sup>a</sup>For the acoustic pulse case, Sec. 4.3, the collision model used is not specified. Hence, the value of the bulk viscosity is unknown. Comparisons to low-viscosity or inviscid analytical solutions are not given.

<sup>b</sup>The viscosity is modified only for high wave numbers, due to the use of *selective viscosity* filters [48].

<sup>c</sup>Section 3.3.4. Besides, the value of the bulk viscosity is not specified.

<sup>d</sup>Section 9.5. Besides, the value of the bulk viscosity is not specified.

This table reveals that only two studies focus on our present subject: the theses of Pasquali [46] and Hasert [47]. In the case of Pasquali (Sec. 3.3.4 of [46]), very few details are given on the simulation and on the results: we can only say that the Mach number based on the inlet velocity in the pipe is very low ( $M \simeq 0.01$ ) and that the mesh is very highly resolved, which is quite close to an ideal situation. However, it is indeed one of the rare studies in which a near-field acoustic spectrum is given with a vortical flow in a multidomain LBM simulation. As far as the work of Hasert is concerned, there are many more details and validations (although the crucial value of the bulk viscosity is never mentioned for the MRT simulations). Very interestingly, he shows that despite good standard aeroacoustic validations (i.e., without vortices crossing a refinement area), a highly overestimated far-field acoustic noise is obtained in his more realistic benchmark, compared to experimental data (part 9.5 of [47]). He concludes that “a possible source might be the errors introduced near the grid level interfaces.” Considering the snapshots that are shown in this study, this hypothesis seems very plausible. We also point out that the presented snapshot (Fig 9.10.b) of the pressure is a highly precise closeup ( $\simeq 0.001\%$  of the reference pressure). His benchmark also shows that a vortical velocity field free from artifacts does not guarantee a correct underlying aeroacoustic field.

On the other hand, simulations that exactly correspond to our goal (higher Mach number, high Reynolds number vortical flows crossing multiple grid refinement areas with athermal approximations, ...) are carried out in several papers but performed with commercial lattice Boltzmann solvers. The reader can refer to [49–52] amongst others. However, using commercial softwares does not allow any detailed study of the grid coupling algorithm. That is why we chose to develop a different grid refinement algorithm and compare it to the one of [45] on that matter. Our method is based on a specific coupling between a finite difference approximation and the lattice Boltzmann method on the refinement interface: it allows to better take into account the gradients normal to the interface in vertex-centered algorithms.

The structure of this work is as follows. We first describe the lattice Boltzmann method used as a basis, regardless of any mesh refinement issue. After that, we derive formally the equations used in our model in Sec. III, with the simple Bhatnagar-Gross-Krook (BGK) collision operator. We also show that this work can be extended to multiple relaxation times models. In Sec. IV, we remind the reader of several basic relations implied by a change of grid size, before describing our grid-coupling algorithm. We also address the question of interpolations, initialization, and local filtering for fine-to-coarse transfers. Finally, we compare in Sec. V our approach to a slightly improved version of the one presented in [45], also used in [42] (see Sec. IV for more details). Two benchmark cases are studied: the acoustic pulse and a convected source of vorticity. In this section, the second order accuracy of the method is demonstrated, while the full theoretical proof is given in Appendix A. For the vortex case, the effect of grid refinement on aeroacoustics is shown by calculating near-field density spectra. The influence of the free stream Mach number is also highlighted.

## II. LATTICE BOLTZMANN METHOD

In this section, we briefly summarize the steps that lead from the Boltzmann equation to the lattice Boltzmann numerical scheme since it is useful for the derivation of our model. We start from the original Boltzmann equation

$$\frac{\partial f}{\partial t} + \mathbf{c} \cdot \nabla f = \Omega(f), \quad (1)$$

with  $\Omega$  the collision operator,  $\mathbf{c}$  the particle velocity, and  $f(\mathbf{x}, \mathbf{c}, t)$  the distribution function.

Following the choices of Shan *et al.* in [53], we can write a fully dimensionless system by using a characteristic time  $t_0$ , length  $l_0/c_s$ , density  $\rho_0$ , and speed  $c_0/c_s$  ( $c_0$  corresponds to the “isothermal” speed of sound  $c_0^2 = rT_0$ ), with the constant  $c_s = 1/\sqrt{3}$ . The distribution function is normalized by  $\rho_0$ . We use the standard D3Q19 velocity set (valid for athermal and weakly compressible simulations) to discretize the velocity space of the Boltzmann equation (see [53]). This set is defined by the following dimensionless expressions with respect to  $c_0/c_s$ :

$$\mathbf{c}_\alpha = \begin{cases} (0,0,0), & \alpha = 0 \\ (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1), & \alpha = 1 \dots 6 \\ (\pm 1, \pm 1, 0), (\pm 1, 0, \pm 1), (0, \pm 1, \pm 1), & \alpha = 7 \dots 18 \end{cases} \quad (2)$$

with the weights

$$\omega_\alpha = \begin{cases} 1/3, & \alpha = 0 \\ 1/18, & \alpha = 1 \dots 6 \\ 1/36, & \alpha = 7 \dots 18. \end{cases} \quad (3)$$

If the relation  $l_0/c_0 t_0 = 1$  is verified (kinetic Strouhal number equal to unity), we obtain the following scaling of the discrete velocity Boltzmann equation (DVBE), which we write here using the BGK collision operator

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{c}_\alpha \cdot \nabla f_\alpha = \frac{-1}{\tilde{\tau}} (f_\alpha - f_\alpha^{\text{eq}}), \quad (4)$$

where  $\tilde{\tau} = \tau/t_0$  is the dimensionless relaxation time. It can be related to the physical value of the viscosity by performing a Chapman-Enskog expansion, which yields

$$\mu = \tau \rho c_0^2. \quad (5)$$

In this paper, we use the BGK collision operator for all the formal derivations and numerical applications. However, we show in Secs. III B and IV A that this approach can be extended to multiple relaxation time models. After integrating this set of equations along characteristics of direction  $\mathbf{c}_\alpha$  (see [54] for the details), we can obtain the lattice BGK equation, where  $\tilde{\Delta}t$  is the dimensionless time step with respect to  $t_0$ :

$$\begin{aligned} g_\alpha(\mathbf{x} + \mathbf{c}_\alpha \tilde{\Delta}t, t + \tilde{\Delta}t) \\ = \hat{g}(\mathbf{x}, t) = g_\alpha(\mathbf{x}, t) - \frac{\tilde{\Delta}t}{\tilde{\tau}_g} [g_\alpha(\mathbf{x}, t) - g_\alpha^{\text{eq}}(\mathbf{x}, t)] + O(\tilde{\Delta}t^3), \end{aligned} \quad (6)$$

with

$$\tilde{\tau}_g = \tilde{\tau} + \frac{\tilde{\Delta}t}{2}, \quad (7)$$

$$g_\alpha = f_\alpha + \frac{\tilde{\Delta}t}{2(\tilde{\tau}_g - \frac{\tilde{\Delta}t}{2})}(f_\alpha - f_\alpha^{\text{eq}}), \quad (8)$$

$$g_\alpha^{\text{eq}} = f_\alpha^{\text{eq}},$$

$$g_\alpha^{\text{eq}} = \omega_\alpha \rho \left\{ 1 + \frac{\mathbf{c}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} [(\mathbf{c}_\alpha \cdot \mathbf{u})^2 - c_s^2 |\mathbf{u}|^2] \right\} \quad (9)$$

for the D3Q19 athermal model. We call  $\widehat{g}(\mathbf{x}, t)$  the *collision* function. In order for the lattice nodes to coincide with the mesh nodes, we have the constraint  $\tilde{\Delta}x = |\mathbf{c}_\alpha| \tilde{\Delta}t$  for  $\alpha = 1 \dots 6$ , which yields the necessary condition

$$c_0 \frac{\tilde{\Delta}t}{\tilde{\Delta}x} = c_s. \quad (10)$$

Choosing  $t_0 = \Delta t$  and  $l_0/c_s = \Delta x$  yields  $\tilde{\Delta}t = \tilde{\Delta}x = 1$ , so that Eq. (6) becomes

$$g_\alpha(\mathbf{x} + \mathbf{c}_\alpha, t + 1) - g_\alpha(\mathbf{x}, t) = \frac{-1}{\tilde{\tau}_g} [g_\alpha(\mathbf{x}, t) - g_\alpha^{\text{eq}}(\mathbf{x}, t)], \quad (11)$$

with

$$\tilde{\tau}_g = \frac{\tau_g}{\Delta t} = \bar{\tau} + \frac{1}{2} = \frac{\tau}{\Delta t} + \frac{1}{2}. \quad (12)$$

This particular dimensionless system of units is usually called *lattice units*.

The zero and first order macroscopic moments are given by

$$\rho = \sum_\alpha g_\alpha, \quad (13)$$

$$\rho \mathbf{u} = \sum_\alpha \mathbf{c}_\alpha g_\alpha. \quad (14)$$

A Chapman-Enskog procedure performed on this lattice Boltzmann equation (LBE) gives the following expression for the stress tensor  $S_{ij}$ :

$$S_{ij} = \frac{-1}{2\tilde{\tau}_g \rho c_s^2} \sum_\alpha c_{\alpha,i} c_{\alpha,j} (g_\alpha - g_\alpha^{\text{eq}}), \quad (15)$$

the relation between  $\tilde{\tau}_g$  and the dynamic shear viscosity  $\bar{\mu}$  (related to a time scale  $\Delta t$ )

$$\bar{\mu} = \rho c_s^2 (\tilde{\tau}_g - \frac{1}{2}), \quad (16)$$

the expression of the bulk viscosity  $\bar{\xi}$ , which determines, in association with  $\bar{\mu}$ , the dissipation rate of the sound waves [55]

$$\bar{\xi} = 2\rho c_s^4 (\tilde{\tau}_g - \frac{1}{2}) = \frac{2}{3} \bar{\mu}, \quad (17)$$

as well as the so-called athermal equation of state

$$p = \rho c_s^2. \quad (18)$$

To retrieve the dimensional quantities, one has to multiply the dimensionless velocities by  $c_0/c_s = \Delta x/\Delta t$  [see Eq. (10)] and the density by  $\rho_0$ , consistently with the nondimensionalization approach we used at the beginning. From now on, we will omit the tildes and overlines for simplicity.

### III. DIRECTIONAL SPLITTING OF THE LATTICE BOLTZMANN EQUATION

#### A. A numerical scheme on the refinement interface

In this section we derive the equations used for our grid-coupling algorithm. Let us suppose that the transition interface is locally a plane of normal  $\mathbf{e}^\perp$ . We start from the DVBE (4) but we write it the following way:

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{c}^\parallel_\alpha \cdot \nabla^\parallel f_\alpha = \frac{-1}{\tau} (f_\alpha - f_\alpha^{\text{eq}}) - c_\alpha^\perp \nabla^\perp f_\alpha \quad (19)$$

with  $\mathbf{c}_\alpha = \mathbf{c}^\parallel_\alpha + c_\alpha^\perp \mathbf{e}^\perp$  and  $\nabla = \nabla^\parallel + \nabla^\perp \mathbf{e}^\perp$ . Following the approach used, for example, in [54], we integrate this equation along characteristic lines. However, unlike the traditional approach, we use the characteristics directed by  $\mathbf{c}^\parallel_\alpha$  instead of  $\mathbf{c}_\alpha$ .

Hereafter, we take  $\mathbf{e}^\perp = \mathbf{e}_y$  for the sake of simplicity, without any loss of generality. This give us  $\mathbf{c}^\parallel_\alpha = c_{\alpha,1} \mathbf{e}_x + c_{\alpha,3} \mathbf{e}_z$ . We can therefore write for a dimensionless time step  $\Delta t$  and for  $\mathbf{v}^\parallel(s) = \mathbf{x} + \mathbf{c}^\parallel_\alpha s$  and  $w(s) = s$ :

$$\begin{aligned} & f_\alpha(\mathbf{x} + \mathbf{c}^\parallel_\alpha \Delta t, t + \Delta t) - f_\alpha(\mathbf{x}, t) \\ &= \int_0^{\Delta t} \frac{-1}{\tau} [f_\alpha(\mathbf{v}^\parallel(s), w(s)) - f_\alpha^{\text{eq}}(\mathbf{v}^\parallel(s), w(s)) + \tau F_\alpha] ds \end{aligned} \quad (20)$$

with

$$F_\alpha = (\mathbf{c}_\alpha \cdot \mathbf{e}_y) \nabla_y f_\alpha = c_{\alpha,2} \frac{\partial f_\alpha}{\partial y}. \quad (21)$$

We use the classical trapezium rule to approximate the right-hand side of Eq. (20) and obtain

$$\begin{aligned} & f_\alpha(\mathbf{x} + \mathbf{c}^\parallel_\alpha \Delta t, t + \Delta t) - f_\alpha(\mathbf{x}, t) \\ &= \frac{-\Delta t}{2\tau} \{ f_\alpha(\mathbf{x} + \mathbf{c}^\parallel_\alpha \Delta t, t + \Delta t) \\ &\quad - f_\alpha^{\text{eq}}(\mathbf{x} + \mathbf{c}^\parallel_\alpha \Delta t, t + \Delta t) + f_\alpha(\mathbf{x}, t) - f_\alpha^{\text{eq}}(\mathbf{x}, t) \\ &\quad + \tau [F_\alpha(\mathbf{x} + \mathbf{c}^\parallel_\alpha \Delta t, t + \Delta t) + F_\alpha(\mathbf{x}, t)] \} + O(\Delta t^3), \end{aligned} \quad (22)$$

which yields

$$\begin{aligned} & g_\alpha(\mathbf{x} + \mathbf{c}^\parallel_\alpha \Delta t, t + \Delta t) + \frac{\Delta t}{2} F_\alpha(\mathbf{x} + \mathbf{c}^\parallel_\alpha \Delta t, t + \Delta t) \\ &= g_\alpha(\mathbf{x}, t) - \frac{\Delta t}{\tau_g} [g_\alpha(\mathbf{x}, t) - g_\alpha^{\text{eq}}(\mathbf{x}, t)] - \frac{\Delta t}{2} F_\alpha + O(\Delta t^3). \end{aligned} \quad (23)$$

This scheme is implicit: in order to make it explicit, we use the following change of variable:

$$h_\alpha = g_\alpha + \frac{\Delta t}{2} F_\alpha, \quad (24)$$

which implies

$$h_\alpha^{\text{eq}} = g_\alpha^{\text{eq}} + \frac{\Delta t}{2} F_\alpha^{\text{eq}}, \quad (25)$$

$$F_\alpha^{\text{eq}} = c_{\alpha,2} \frac{\partial f_\alpha^{\text{eq}}}{\partial y}. \quad (26)$$



With these notations, we can write a *stream and collide* algorithm for the nodes of the transition interface: after some algebra, we obtain from Eq. (22), written in lattice units and without the error term,

$$\begin{aligned} h_\alpha(\mathbf{x} + \mathbf{c}_\alpha^\parallel, t + 1) &= h_\alpha(\mathbf{x}, t) - \frac{1}{\tau_g} [h_\alpha(\mathbf{x}, t) - h_\alpha^{\text{eq}}(\mathbf{x}, t)] \\ &\quad + \left[ \left( \frac{1}{2\tau_g} - 1 \right) F_\alpha(\mathbf{x}, t) - \frac{1}{2\tau_g} F_\alpha^{\text{eq}}(\mathbf{x}, t) \right] \\ &= \widehat{h}_\alpha(\mathbf{x}, t) \end{aligned} \quad (27)$$

with

$$h_\alpha = g_\alpha + \frac{1}{2} F_\alpha, \quad (28)$$

$$h_\alpha^{\text{eq}} = g_\alpha^{\text{eq}} + \frac{1}{2} F_\alpha^{\text{eq}}. \quad (29)$$

This scheme requires the computation of two spatial gradient terms,  $F_\alpha$  and  $F_\alpha^{\text{eq}}$ . Equations (23) and (10) show that a second order approximation is sufficient: we chose a simple second order centered finite difference stencil. This step requires interpolations, which are discussed in Secs. IV and IV B.

In order to save computational time, we can write a derived scheme for which only  $F_\alpha$  is needed. After some algebra, we obtain a relation between the collision functions

$$\widehat{h}_\alpha = \widehat{g}_\alpha - \frac{1}{2} F_\alpha. \quad (30)$$

Calculating  $\widehat{h}_\alpha(\mathbf{x}, t)$  is now equivalent to

- (i) converting  $h_\alpha(\mathbf{x}, t)$  into  $g_\alpha(\mathbf{x}, t)$  via Eq. (28),
- (ii) computing  $\widehat{g}_\alpha(\mathbf{x}, t)$  with Eq. (6) and the macroscopic variables,
- (iii) converting  $\widehat{g}_\alpha(\mathbf{x}, t)$  into  $\widehat{h}_\alpha(\mathbf{x}, t)$  with Eq. (30).

However, does this equation allow to simulate a fluid governed by the same macroscopic equations than with the lattice Boltzmann equation (6)? A Chapman-Enskog analysis of Eq. (23) [or equivalently of Eq. (27)] confirms that we obtain at leading order the same macroscopic equations compared to the classical LBE, with second order accuracy in time, space, and Knudsen number, provided that the  $F_\alpha$  gradient term is evaluated with, at least, a second order approximation (see Appendix A for the full proof).

We remind the reader that the so-called *acoustic scaling* is used in this work ( $\Delta x \propto \Delta t$ ): this means that the Knudsen number error  $O(\epsilon^2) = O[(M/\text{Re})^2]$  does not vanish with decreasing mesh size and time step. As said in the Introduction to this work, several authors use the *diffusive scaling* ( $\Delta x^2 \propto \Delta t$ ) [19], which yields  $M \propto \Delta x$  and  $\epsilon^2 \propto \Delta x^2 \propto \Delta t$ . This makes the Knudsen number error vanish at second order rate in space (so first order rate in time). It also makes the  $O(M^3)$  error in the strain rate tensor vanish with increasing resolution. However, because of Eq. (10), this scaling yields a nonconstant physical speed of sound [ $c_0 = O(\Delta x^{-1})$ ] which cannot be used for aeroacoustics, as also explained in Ref. [25].

As far as external forces are concerned, their implementation is straightforward: one simply needs to incorporate the forcing term during the classical computation of  $\widehat{g}_\alpha$  and modify the value of the macroscopic variables accordingly, following for instance the method of Guo *et al.* [56].

## B. Extension to multiple relaxation time operators

All the previous derivations are extendable to multiple relaxation time operators. As can be seen in Ref. [54], it is indeed possible to start from a DVBE that uses a general collision matrix  $\Omega$ , so that

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{c}_\alpha \cdot \nabla f_\alpha = -\Omega_{\alpha\beta} (f_\beta - f_\beta^{\text{eq}}). \quad (31)$$

For a BGK equation, we have  $\Omega = (1/\tau)\mathbf{I}$ .

After integration along the classical characteristics, this gives in lattice units

$$\begin{aligned} g_\alpha(\mathbf{x} + \mathbf{c}_\alpha, t + 1) &= g_\alpha(\mathbf{x}, t) - \left[ \Omega(\mathbf{I} + \frac{1}{2}\Omega)^{-1} \right]_{\alpha\beta} \\ &\quad \times [g_\beta(\mathbf{x}, t) - g_\beta^{\text{eq}}(\mathbf{x}, t)] \end{aligned} \quad (32)$$

with

$$g_\alpha = f_\alpha + \frac{1}{2}\Omega_{\alpha\beta} (f_\beta - f_\beta^{\text{eq}}). \quad (33)$$

It is thus also possible to start from a splitted version of Eq. (31):

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{c}_\alpha^\parallel \cdot \nabla^\parallel f_\alpha = -\Omega_{\alpha\beta} (f_\beta - f_\beta^{\text{eq}}) - c_\alpha^\perp \nabla^\perp f_\alpha. \quad (34)$$

We can see that the splitted term on the right-hand side is still completely independent of the collision operator. This means that all the previous conversion formulas between distribution functions and between collision functions are still valid, provided the  $f_\alpha$  that appear in the gradient term  $F_\alpha$  are computed this time with Eq. (33). This implies that we need to find the value of  $\Omega$  as a function of the relaxation parameters. For example, if one wants to use the D3Q19 moment basis of D'Humières *et al.* [57], we immediately deduce from Eq. (32) that

$$\Omega(\mathbf{I} + \frac{1}{2}\Omega)^{-1} = \mathbf{M}^{-1}\widehat{\mathbf{S}}\mathbf{M} \quad (35)$$

with  $\mathbf{M}$  the orthogonal transformation matrix and  $\widehat{\mathbf{S}}$  the diagonal relaxation matrix.

This gives

$$\Omega = \mathbf{M}^{-1}\widehat{\mathbf{S}}\mathbf{M}(\mathbf{I} - \frac{1}{2}\mathbf{M}^{-1}\widehat{\mathbf{S}}\mathbf{M})^{-1}. \quad (36)$$

The only remaining question is the invertibility of the matrix  $\mathbf{A} = \mathbf{I} - \frac{1}{2}\mathbf{M}^{-1}\widehat{\mathbf{S}}\mathbf{M}$ . We first remark that

$$\mathbf{M}\mathbf{A}\mathbf{M}^{-1} = \mathbf{I} - \frac{1}{2}\widehat{\mathbf{S}} \quad (37)$$

which is a diagonal matrix. Provided that  $s_i \neq 2$  for all  $i$ , this matrix is invertible. This means that, under this condition,  $\mathbf{A}$  is invertible since

$$(\mathbf{M}\mathbf{A}\mathbf{M}^{-1})^{-1} = \mathbf{M}^{-1}\mathbf{A}^{-1}\mathbf{M}. \quad (38)$$

The rescaling formula for the distribution functions between a fine and a coarse grid are also affected by the collision model (see Sec. IV A).

However, it is well known that popular multiple relaxation time models, such as the one of D'Humières *et al.* [57], rely on the addition of a substantial amount of bulk viscosity that stabilizes the simulations but strongly dissipates the acoustic field. Although this technique is very useful in purely

aerodynamic cases, it prevents any correct aeroacoustic study as shown by Marié *et al.* [6]. A possible solution is to use the two relaxation times (TRT) model (see [47,58]), which does not modify the bulk viscosity compared to the BGK model.

The same effect appears in multiple relaxation times methods based on other collision operators such as the cascaded lattice Boltzmann method [59] or the cumulant lattice Boltzmann method [27], when used with their suggested default set of nonhydrodynamic relaxation frequencies (i.e., all set to unity). It also exists in variants of the entropic lattice Boltzmann method, such as in [60]. In order to better understand the effect of the relaxation times on the bulk viscosity for those models, the reader may refer to [46,61], to the appendices of [27] and to [29] [Eq. (2.10)].

#### IV. GRID-COUPLING ALGORITHM FOR THE DIRECTIONAL SPLITTING METHOD

We begin this section by a few useful definitions:

(1) *Transition interface*: set containing the nodes that do not have all the 19 required neighbors required by the standard LBM streaming algorithm. It is seen as the interface between the coarse and the fine grids (nodes pictured  $\blacksquare$ ).

(2) *Inner fine nodes*: set containing all the fine nodes that have their 19 required neighbors nodes (pictured  $\blacktriangle$ ).

(3) *Inner coarse nodes*: set containing all the coarse nodes that have their 19 required neighbors nodes (pictured  $\bullet$ ).

(4) *Fine transition nodes*: fine nodes that belong to the transition interface (pictured  $\blacksquare^f$ ).

(5) *Coarse transition nodes*: coarse nodes that belong to the transition interface (pictured  $\blacksquare^c$ ). These nodes always have a collocated fine node.

(6) *Ghost layer nodes (only for our approach)*: nodes that are computed by interpolation, outside the transition interface on the coarse grid side. These values are used for the calculation of the gradient term (with a two-point centered finite difference stencil). Two categories are distinguished,  $\Delta$  and  $\Delta'$ , that require different interpolation patterns (see Sec. IV B for the details).

We display on Fig. 1 the geometrical configuration of the grid refinement region with its different type of nodes defined above.

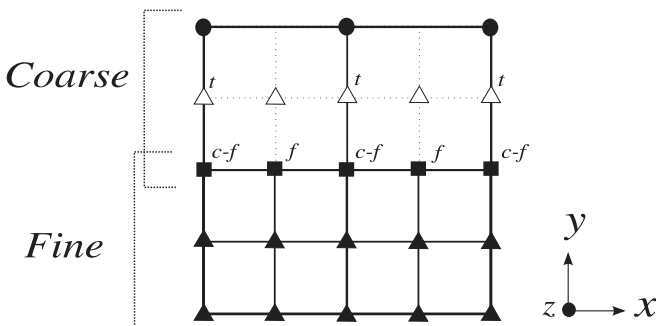


FIG. 1. Geometrical configuration of our grid refinement method.  $\bullet$ : inner coarse nodes;  $\blacktriangle$ : inner fine nodes;  $\Delta$  and  $\Delta'$ : fine ghost layer nodes;  $\blacksquare^f$ : fine transition nodes;  $\blacksquare^{c-f}$ : overlapping region where a  $\blacksquare^c$  and a  $\blacksquare^f$  are collocated.

We present here a two dimensional figure for clarity, but only one type of fine nodes is added when dealing with a three dimensional configuration. These nodes are located on the center of the face of the transition coarse cells (behind  $\blacksquare^f$  in the  $e_z$  direction) and are computed with our modified algorithm as any fine transition node.

##### A. Zonal grid refinement: Conversion relations

We summarize here very shortly the classical relations between the time steps, grid spacings, relaxation times, and nonequilibrium parts of the distribution functions when a change in grid size occurs. These relations are still valid with our approach and will be used in practice in our algorithm. More details can be found in Refs. [41,45]. We denote with a superscript  $f$  any quantity that relates to the *fine* grid, which grid size  $\Delta x^f$  relates to that of the so-called *coarse* grid,  $\Delta x^c$ , by the relation  $\Delta x^c = 2\Delta x^f$ . Since we choose to keep the speed of sound constant between the grids, the condition (10) gives  $\Delta t^c = 2\Delta t^f$ .

For the relaxation times, we thus have  $\tau_g^f = \frac{\tau}{\Delta t^f} + \frac{1}{2}$  and  $\tau_g^c = \frac{\tau}{2\Delta t^f} + \frac{1}{2}$ . Imposing the continuity of the viscosity across the refinement interface leads to

$$\tau_g^f = 2\tau_g^c - \frac{1}{2}. \quad (39)$$

We also remark, thanks to Eq. (8), that the function  $g$  is not continuous under a change of resolution whereas the original  $f$  from the DVBE is. Writing this relation on a coarse and on a fine grid and combining them gives the following conversion formula between fine and coarse nonequilibrium functions:

$$g_\alpha^c - g_\alpha^{\text{eq}} = \frac{2\tau_g^c}{\tau_g^f} (g_\alpha^f - g_\alpha^{\text{eq}}), \quad (40)$$

which is identical to the classical one of Dupuis and Chopard [41] and ensures the continuity of the stress tensor  $S_{ij}$ . The equilibrium distribution does not require any rescaling since it only depends on the macroscopic quantities.

The rescaling method in moment-based formulations (such as the D3Q19 MRT) is slightly different but well explained in [34].

##### B. Interpolation schemes

All the refinement algorithms rely somehow on interpolations on the interface, even if not explicitly said. For example, cell-centered algorithms rely on an *averaging* of the neighboring fine cells (also called *homogeneous redistribution* or *coalescence* procedure) during the fine-to-coarse step: this is equivalent to a linear interpolation. An interesting situation is the coarse-to-fine procedure of [43], called *homogeneous redistribution of particle densities from coarse to fine grid cells*: this step is in fact equivalent to an extrapolation. Such coarse-to-fine extrapolation techniques appear in other cell-centered algorithms, such as in the work of Yu and Fan [44] or that of Chen *et al.* [17] (which is not based on standard *stream and collide* LBM but worth to be mentioned [18]).

In our case, we need to fill a ghost layer of fine nodes on the coarse side at each fine time step by interpolations. This layer is represented by  $\Delta$  and  $\Delta'$  nodes on Fig. 1. Equation (6) shows that an interpolation of at least third order in space

and time on  $g_\alpha$  is necessary. Computational evidences of this necessity are given in [45], where this matter is also discussed. Numerical comparisons lead us to choose, when possible, a fourth order four-point symmetric stencil for spatial interpolation. To summarize, we used two different stencils for space interpolation:

$$\begin{aligned} f_\alpha(\mathbf{x}, t) = & \frac{2}{5}f_\alpha(\mathbf{x} + \mathbf{e}_y\Delta x, t) + 2f_\alpha(\mathbf{x} - \mathbf{e}_y\Delta x, t) \\ & - 2f_\alpha(\mathbf{x} - 2\mathbf{e}_y\Delta x, t) + \frac{3}{5}f_\alpha(\mathbf{x} - 3\mathbf{e}_y\Delta x, t) \\ & + O(\Delta x^4) \end{aligned} \quad (41)$$

for nodes  $\Delta^t$ ,

$$\begin{aligned} f_\alpha(\mathbf{x}, t) = & \frac{9}{16}[f_\alpha(\mathbf{x} + \mathbf{e}_x\Delta x, t) + f_\alpha(\mathbf{x} - \mathbf{e}_x\Delta x, t)] \\ & - \frac{1}{16}[f_\alpha(\mathbf{x} + 3\mathbf{e}_x\Delta x, t) + f_\alpha(\mathbf{x} - 3\mathbf{e}_x\Delta x, t)] \\ & + O(\Delta x^4) \end{aligned} \quad (42)$$

for nodes  $\Delta$ . For time interpolation, we used at odd time steps (see Sec. IV E)

$$\begin{aligned} f_\alpha(\mathbf{x}, t) = & \frac{-1}{8}f_\alpha(\mathbf{x}, t - 3\Delta t) + \frac{3}{4}f_\alpha(\mathbf{x}, t - \Delta t) \\ & + \frac{3}{8}f_\alpha(\mathbf{x}, t + \Delta t) + O(\Delta t^3). \end{aligned} \quad (43)$$

Interpolation can be either done on  $g$  or  $f$  distribution functions, depending on the chosen implementation. We note that with our method, there are no interpolated distribution functions that are directly used in any collision or streaming step. As we will see hereafter, the interpolated ghost layer of fine nodes is computed for the sole purpose of calculating the gradient  $F_\alpha$ .

For this study, we used Eqs. (42) and (43) for both our model and the algorithm of Lagrava [45], in order to make correct comparisons. The situation in which Eq. (41) has to be used does not exist in Ref. [45]. The stencil (42) is also used in the recent study of Dorschner *et al.* [60] and in the work of Stiebler *et al.* [28], which are also implemented in a vertex-centered data structure (all the details and the interpolation stencils of Ref. [28] can be found in the thesis of Crouse [38]). We also tested a linear time interpolation, as in Refs. [28,38], which showed no improvement in any of our simulations.

We also point out that studies in which a compact interpolation is employed, for example, in Refs. [21,26] (see also Ref. [23] for the mathematical concept), use a linear interpolation for the density and a higher order interpolation for the velocity field. Good results are obtained with compact interpolation by authors that only focus on aerodynamics and the method is promising. However, the authors are not aware of any evidence that a linear interpolation on the density is suitable for aeroacoustics, even if it is coupled to a higher order method for the velocity field (which obviously makes compact interpolation different than a linear interpolation on the whole distribution function). A linear interpolation on the whole distribution is equivalent to a linear interpolation on all its moments, which means that the density field is interpolated linearly as well. It has been shown in the work of Lagrava [45] that a linear interpolation on the whole distribution function, although having a quadratic  $O(\Delta x^2)$  error term, can lead to discontinuities on the density field, and our own experience leads to the same conclusion. Evidence of this phenomenon can be found in Refs. [13,62] using FDLBM.

Also, several authors who use compact interpolations do not use any time interpolation at all, thanks to specific cell-centered algorithms, such as in Refs. [21,25–27]. However, the above mentioned issue regarding the presence or absence of discontinuities on the density field is not investigated.

### C. Local filtering strategy

As emphasized by Lagrava *et al.*, a local filtering strategy is mandatory for grid refinements at the locations where information is transferred from the *fine* to the *coarse* grid. This is especially true when there is no artificial bulk viscosity added through the collision model, which, as we said earlier, must be the case when studying aeroacoustics.

For example, the stable results obtained in Refs. [28,60], in which this filtering step does not appear, can be explained by such an increase of the bulk viscosity, which prevents stability issues near the interface (as well as away from it). In the case of Stiebler *et al.* [28], this is simply made through the relaxation rates of the MRT model. It is less obvious in the work of Dorschner *et al.* [60]. However, the proof can be found in Ref. [29] [Eq. (2.10)], cited by the authors themselves, in which the entropic stabilizer  $\gamma$  is explicitly related to the value of the bulk viscosity for the model used in Ref. [60]. Thanks to the equation cited above, we can see that the figure showing the spatial evolution of  $\gamma$  in the turbulent channel of Ref. [60] indicates that the value of the bulk viscosity is particularly increased near the refinement interfaces on the coarse side, where instabilities are the most likely to appear. In addition to the fact that Lagrava *et al.* use only a single relaxation time, this may explain why the authors of Ref. [60] claim that “in contrast to the algorithm of Lagrava *et al.*,” they do not need a box filter to maintain stability with grid refinements. However, this feature also dismisses the use of this model for aeroacoustics.

On the other hand, for a low and constant value of the bulk viscosity, suitable for aeroacoustics, we confirm the observations of Lagrava *et al.*: the absence of a local fine-to-coarse filtering step leads to unstable results.

This filtering procedure is not only mandatory for numerical stability reasons: it also relies on purely physical grounds. Due to the factor 2 between the coarse and fine time steps and mesh sizes, reduced wave numbers and frequencies above  $\pi/2$  on the fine grid are not supported by the coarse grid. An ideal filter would be a time and space box filter in Fourier space, with an infinitely sharp cutoff at  $k^f = \pi/2$  and  $\omega^f = \pi/2$  (where  $k^f$  and  $\omega^f$  are the reduced wave numbers and frequencies, respectively, in fine lattice units), applied to the whole distribution function. This ideal case is of course unreachable in our situation. It is also very important to note that more or less sophisticated filtering steps appear in the majority of the existing grid refinement algorithms, although often not explicitly written: for example, the procedure that consists in computing the coarse cell by averaging the neighboring fine cells (which is very common in cell-centered algorithms, see Sec. IV B) acts of course indirectly as a basic spatial filter. More generally, any *fine-to-coarse* procedure that relies on the use of neighboring fine cells acts as a filter. In our case, we choose to filter the nonequilibrium part of the distribution



function because it yields a clearly less dissipative scheme than filtering the whole distribution function, as we will see after.

We therefore apply a *restriction* filter (denoted  $\mathcal{R}$ ) to the nonequilibrium part of the precollision distribution function, where information is transferred from the fine level to the coarse level. The collision Eq. (6) is filtered in the following way:

$$\widehat{g}^f(\mathbf{x}, t) = g_\alpha^f(\mathbf{x}, t) - \frac{\Delta t}{\tau_g} [\mathcal{R}(g_\alpha^{\text{neq},f})(\mathbf{x}, t)]. \quad (44)$$

The expression of our filter  $\mathcal{R}$  is derived from the one used in the work of Touil *et al.* [42]:

$$\begin{aligned} \mathcal{R}[g_\alpha^{\text{neq},f}(\mathbf{x}, t)] &= \frac{1}{7} g_\alpha^{\text{neq},f}(\mathbf{x}, t) + \frac{1}{14} \sum_{\beta=1}^6 g_\alpha^{\text{neq},f}(\mathbf{x} + \mathbf{c}_\beta, t) \\ &+ \frac{1}{28} \sum_{\beta=7}^{18} g_\alpha^{\text{neq},f}(\mathbf{x} + \mathbf{c}_\beta, t). \end{aligned} \quad (45)$$

This operation is mass and momentum conserving because

$$\sum_{\alpha=0}^{18} \mathcal{R}[g_\alpha^{\text{neq},f}(\mathbf{x}, t)] = 0, \quad (46)$$

$$\sum_{\alpha=0}^{18} \mathbf{c}_\alpha \mathcal{R}[g_\alpha^{\text{neq},f}(\mathbf{x}, t)] = \mathbf{0}. \quad (47)$$

This is obtained by noticing that, for any indices  $a$  and  $b$ , it follows by definition of the nonequilibrium part of the distribution function at a fixed location that

$$\begin{aligned} \sum_{\alpha=0}^{18} \sum_{\beta=a}^b g_\alpha^{\text{neq},f}(\mathbf{x} + \mathbf{c}_\beta, t) \\ = \sum_{\beta=a}^b \sum_{\alpha=0}^{18} g_\alpha^{\text{neq},f}(\mathbf{x} + \mathbf{c}_\beta, t) = 0, \end{aligned} \quad (48)$$

$$\begin{aligned} \sum_{\alpha=0}^{18} \mathbf{c}_\alpha \sum_{\beta=a}^b g_\alpha^{\text{neq},f}(\mathbf{x} + \mathbf{c}_\beta, t) \\ = \sum_{\beta=a}^b \sum_{\alpha=0}^{18} \mathbf{c}_\alpha g_\alpha^{\text{neq},f}(\mathbf{x} + \mathbf{c}_\beta, t) = \mathbf{0}. \end{aligned} \quad (49)$$

The construction of this restriction operator is based on a lattice discretization of the Laplacian that we note  $\Delta^\mathcal{L}$ : it is defined in dimensional units as [42]

$$\mathcal{R} = \mathbf{I} + \frac{(\sigma^* \Delta x^f)^2}{2} \Delta^\mathcal{L}. \quad (50)$$

Equation (45) corresponds to the special case  $\sigma^* = \sqrt{3/7}$  [42]. Equation (45) can also be interpreted as a volumetric average with weights 1, 0.5, and 0.25 for the center, face, and edge nodes, respectively, together with a normalization factor [42]. A full study on discretized Laplacians in 2D and 3D, including lattice discretizations  $\Delta^\mathcal{L}$  on the D2Q9, D3Q15, D3Q19, and D3Q27 lattices, is available in Ref. [63].

This operation acts as a filter on the collision operator, exactly as extensively studied in [48]. Therefore, the viscosity

in the zero-wave-number limit is not modified, but is wave number dependent for high  $\mathbf{k}$ . A linear analysis [Eq. (25) in Ref. [48]] shows that if  $\mathcal{R}$  is a filter of order  $n$  (in our case  $n = 2$ ), then the viscosity follows the same transfer function of order  $n$ . As far as its effect on the shear mode is concerned, the study of Ricot *et al.* [48] clearly shows that filtering the nonequilibrium part yields a higher order filter (which means less dissipative) than if it was applied to the whole distribution function: we can estimate that, since the present restriction is a second order filter on the viscosity, it yields a fourth order filter on the shear mode [the dissipation rate becomes  $-\nu + O(|\mathbf{k}|^2)|\mathbf{k}|^2 = -\nu|\mathbf{k}|^2 + O(|\mathbf{k}|^4)$ ]. However, as we said before, when transferring data from a fine site to a coarse site, only the additional dissipation in the range  $k^f \in [0; \pi/2]$  matters since the remaining wave numbers up to  $\pi$  do not exist on the coarse lattice.

Such a locally filtered collision operator preserves the second order accuracy of the scheme in the acoustic scaling: the proof is given in Appendix B. This 3D filter is used for the modified algorithm of Lagrava *et al.* in order to make fair comparisons.

However, as we will see in the next section, we need in the case of our method to perform this operation on nodes that belong to the transition interface, which do not possess all their 19 neighbors, unlike in Refs. [42,45]. On the interface, a 3D symmetric stencil is never available: a 2D symmetric equivalent of this stencil is used when a plane containing nine neighbors is available. This nine-point plane stencil corresponds to the D2Q9 stencil. By denoting  $\mathcal{F}$  the so called “face” links (corresponding to  $\alpha = 1 \dots 4$  in conventional D2Q9 LBM) and  $\mathcal{E}$  the “edge” links ( $\alpha = 5 \dots 8$  in D2Q9 LBM) [63], we obtain

$$\begin{aligned} \mathcal{R}^{2D}[g_\alpha^{\text{neq},f}(\mathbf{x}, t)] \\ = \frac{1}{4} g_\alpha^{\text{neq},f}(\mathbf{x}, t) + \frac{1}{8} \sum_{\beta \in \mathcal{F}} g_\alpha^{\text{neq},f}(\mathbf{x} + \mathbf{c}_\beta, t) \\ + \frac{1}{16} \sum_{\beta \in \mathcal{E}} g_\alpha^{\text{neq},f}(\mathbf{x} + \mathbf{c}_\beta, t). \end{aligned} \quad (51)$$

This can also be derived from Eq. (50) along with the D2Q9 discretization of the Laplacian [63] in the special case  $\sigma^* = \sqrt{9/20}$ .

On complex refinement geometries, there are a few configurations for which a plane containing nine neighbors cannot be found: one can then use a simple three-point average in 1D if needed. This 1D stencil corresponds to the one of the D1Q3: by denoting  $c_1 = 1$  and  $c_2 = -1$  its two nonzero discrete velocities, the average writes

$$\begin{aligned} \mathcal{R}^{1D}[g_\alpha^{\text{neq},f}(\mathbf{x}, t)] &= \frac{1}{2} g_\alpha^{\text{neq},f}(\mathbf{x}, t) + \frac{1}{4} [g_\alpha^{\text{neq},f}(\mathbf{x} + 1, t) \\ &+ g_\alpha^{\text{neq},f}(\mathbf{x} - 1, t)]. \end{aligned} \quad (52)$$

This is also a second order filter in 1D, as shown in the study of Falissard [64]. Higher order stencils in 1D, 2D, and 3D can be found in Ref. [64]. Both 2D and 1D restriction filters defined above are also mass and momentum conserving and preserve the second order accuracy of the scheme: the proof is straightforward and is obtained using the same arguments as in the 3D case.

A conversion from the fine scale to the coarse scale of this restricted distribution function is obtained by expression (40) when needed.

#### D. Initialization of the distribution functions

For a given initial macroscopic field, we need to initialize the value of the distribution functions on the whole grid. A Chapman-Enskog expansion at first order in Knudsen number yields the following expression for the distribution function, in lattice units:

$$g_\alpha = g_\alpha^{\text{eq}} - \omega_\alpha \rho \frac{\overline{\tau}_g}{c_s^2} \mathcal{H}_\alpha^{(2)} : \mathbf{S}, \quad (53)$$

where  $\mathbf{A} : \mathbf{B} = A_{ij} B_{ij}$ ,  $\mathcal{H}_\alpha^{(2)} = \mathbf{c}_\alpha \otimes \mathbf{c}_\alpha - c_s^2 \mathbf{I}$ , and  $\mathbf{S}$  is the strain rate tensor, given by

$$\mathbf{S} = \frac{\nabla \mathbf{u} + {}^T \nabla \mathbf{u}}{2}. \quad (54)$$

In our numerical tests, analytical expressions of the initial velocity fields are known: therefore, the strain tensor is computed analytically at initialization step. We used Eq. (53) (often called the *regularized* expression [65]) to compute the initial populations.

From these values, it is easy to compute the initial field for  $h_\alpha$ , using Eq. (28).

#### E. Proposition of a detailed algorithm

In this section, we use the equations previously derived in order to propose a grid-coupling algorithm, based on the notations of Fig. 1. We note that the transition interface between the fine and the coarse grid is the only zone where both regions overlap, through nodes  $\blacksquare^{c-f}$ .

We start from an initial state  $t = t_0$ : all the macroscopic variables and distribution functions  $g_\alpha$  are initialized by the user everywhere on the fine and on the coarse grid. Since  $\Delta t^c = 2\Delta t^f$ , we count time separately on the fine and on the coarse grid, in fine time scale units. With this convention, we describe here the algorithm in three parts: initialization, odd and even time steps.

(1) Fine grid:  $t = t_0$ , coarse grid:  $t = t_0$ , initialization step:

(a) Compute the ghost fine layer with Eq. (41) for nodes  $\Delta^t$  and then Eq. (42) for nodes  $\Delta$ . When using coarse values, use Eq. (40) for conversion into fine scale.

(b) Calculate  $F_\alpha$  by converting all the  $g_\alpha$  needed into  $f_\alpha$  with the help of Eq. (8).

(c) Collide  $g_\alpha$  everywhere using Eq. (6).

(d) Convert  $\widehat{g}_\alpha$  into  $\widehat{h}_\alpha$  with Eq. (30).

(2) Fine grid:  $t = t_0 + 1$ , coarse grid:  $t = t_0 + 2$ , odd time step:

(a) Stream  $\widehat{g}_\alpha$  on all the inner fine and coarse nodes and  $\widehat{h}_\alpha$  on the interface.

$g_\alpha(t_0 + 1)$  is now obtained on every inner fine node, and  $h_\alpha(t_0 + 1)$  are obtained on the fine transition nodes.  $g_\alpha(t_0 + 1)$  on the fine transition nodes is missing, and needed to compute macroscopic quantities. We also know  $g_\alpha(t_0 + 2)$  on every inner coarse node.

(b) Compute the ghost fine layer (step 1a). At odd time steps, a temporal interpolation on inner coarse nodes [Eq. (43)] is needed for nodes  $\Delta^t$ .

(c) Compute  $F_\alpha$ .

(d) Convert  $h_\alpha$  into  $g_\alpha$  by Eq. (28) and compute the macroscopic variables on the fine transition nodes.

(e) Collide  $g_\alpha$  using the filtered collision on the fine transition nodes (Sec. IV C), and using Eq. (6) on the inner fine nodes.

(f) Convert  $\widehat{g}_\alpha$  into  $\widehat{h}_\alpha$  with Eq. (30).

(3) Fine grid:  $t = t_0 + 2$ , coarse grid:  $t = t_0 + 2$ , even time step:

(a) Stream  $\widehat{g}_\alpha$  on all the inner fine nodes and  $\widehat{h}_\alpha$  on the interface.

$g_\alpha(t_0 + 2)$  is now obtained on every inner fine node and  $h_\alpha(t_0 + 2)$  is obtained on the fine transition nodes. We need to compute  $g_\alpha(t_0 + 2)$  on the fine transition nodes.

(b) Compute the ghost fine layer (repeat step 1a). At even time steps, no temporal interpolation is needed.

(c) Repeat steps 2c to 2f.

(d) Compute  $g_\alpha^c(\blacksquare^c, t_0 + 2)$  from its collocated fine value using Eq. (40).

(e) Collide  $g_\alpha^c(\blacksquare^c, t_0 + 2)$ .

(4) Loop over steps 2a to 3e until the end.

## V. NUMERICAL VALIDATIONS

In this section, we present numerical computations on two different cases: the acoustic pulse and a convected source of vorticity. Our computational domains are pseudo-two-dimensional, which means that their extension in the  $z$  direction is of only one coarse mesh size. Moreover, our domains are always chosen periodic in the  $z$  direction. Therefore, we neglect the influence of the  $z$  coordinate although our algorithm and our solver are in fact three dimensional. We use throughout the whole section a reference length  $L = 1$  m for the dimensions of the computational domains and the corresponding dimensionless variables  $x^* = x/L$ ,  $y^* = y/L$ , and  $\Delta x^* = \Delta x/L$ . Since we have  $L = 1$  m, we consider these three dimensionless values as equal to their dimensional

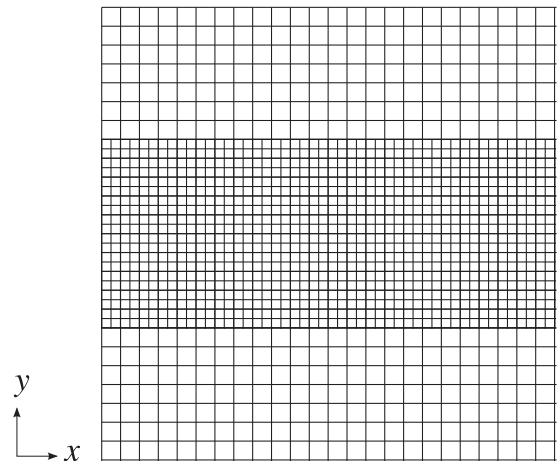


FIG. 2. Grid strategy for the acoustic pulse test case.

counterpart. We also note  $t^*$  the time step number, so that  $t = t^* \Delta t^f$ .

For the whole section, we set  $c_0$  the speed of sound to  $c_0 = 343.2$  m/s and the reference density to  $\rho_0 = 1$  kg/m<sup>3</sup> for simplicity.

### A. Acoustic pulse

We first consider a pseudo-2D acoustic pulse. This is the most standard and easy aeroacoustic test case. We work at very low viscosity,  $\nu = 1.5 \times 10^{-5}$  m<sup>2</sup>/s. The initial profile at  $t = t_0 = 0$  is given as follows:

$$\begin{aligned} \rho(t_0) &= \rho_0[1 + \rho'(t_0)], \\ \mathbf{u}(t_0) &= \mathbf{0} \end{aligned} \quad (55)$$

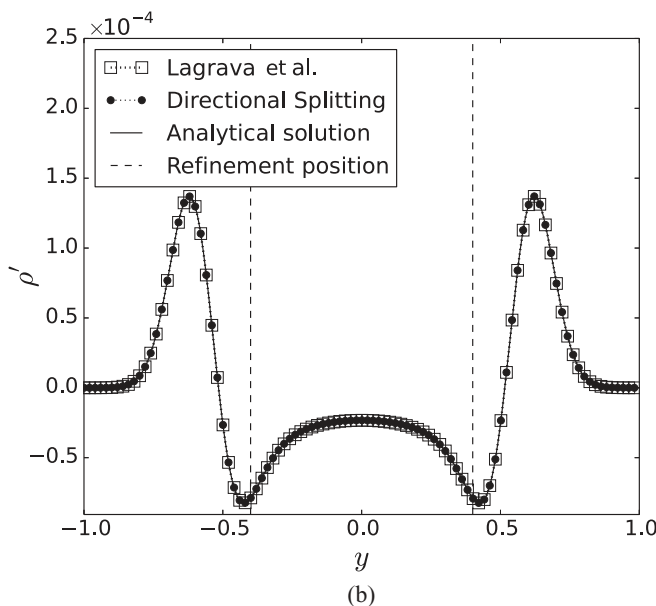
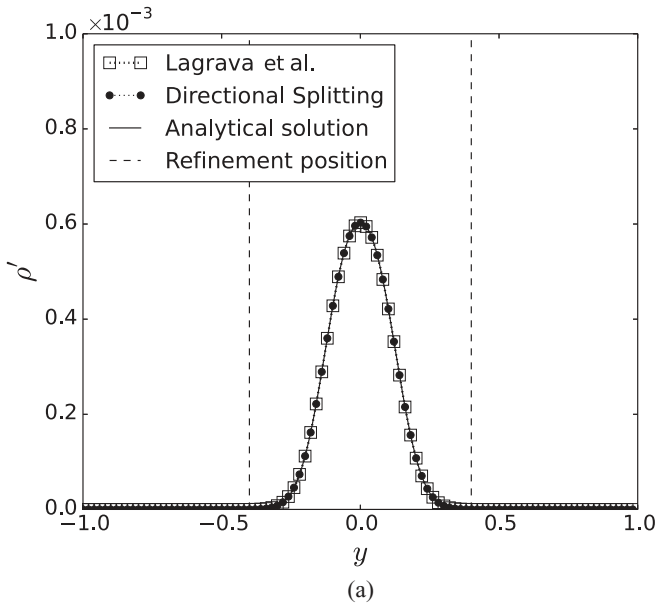


FIG. 3. Value of the perturbation  $\rho'$  at (a)  $t^* = 10$  and (b)  $t^* = 100$  compared to the analytical Euler fluctuation  $\rho'_{\text{th}}$ .

with

$$\begin{aligned} \rho'(x, y, t_0) &= \epsilon \exp(-\alpha r^2), \quad \epsilon = 10^{-3}, \quad \alpha = \frac{\ln(2)}{b^2}, \\ b &= 10^{-1}, \quad r = \sqrt{x^2 + y^2}. \end{aligned} \quad (56)$$

Assuming that the effect of a very low viscosity is negligible on acoustic waves (especially on short distances), and that the amplitude of the acoustic perturbation is small, the density fluctuation  $\rho'(x, y, t)$  can be compared to an analytical solution  $\rho'_{\text{th}}(x, y, t)$  that comes from the resolution of the linearized Euler equations [66]:

$$\rho'_{\text{th}}(x, y, t) = \frac{\epsilon}{2\alpha} \int_0^\infty \exp\left(\frac{-s^2}{4\alpha}\right) \cos(c_0 t s) J_0(r s) s ds, \quad (57)$$

where  $J_0$  is the Bessel zeroth order function of the first kind.

Our first computational domain is a pseudo-2D fully periodic box of size  $[2, 2, \Delta x^c]$  with  $\Delta x^c = 0.02 = 2\Delta x^f$ . The pulse is initialized at the center of the box. Refinement regions are located at  $y = -2/5$  and  $2/5$ . The geometrical configuration is summarized on Fig. 2, drawn with a coarser mesh size for clarity.

We first compare for two different time steps the density function between our model, the algorithm of Lagrava *et al.* [45], and the analytical solution on Fig. 3.

We see that there is a very good match between both models and the analytical solution. To have a better quantitative assessment of the magnitude of the error on the density, we computed its  $L_2$  norm for three different mesh resolutions (the size of the domain is kept fixed). We define the normalized  $L_2$  error as

$$\|\delta\rho\|_{L_2}(t) = \sqrt{\frac{\sum_{x,y} [\rho'(x, y, t) - \rho'_{\text{th}}(x, y, t)]^2}{\sum_{x,y} \rho'_{\text{th}}(x, y, t)^2}}. \quad (58)$$

We denote by  $N$  the equivalent number of fine cells in both  $x$  and  $y$  directions (so that  $L = 1 = N * \Delta x^f$ ). For three different resolutions, we compute the error norm at time  $t^* = 100$  in order to take into account the effect of refinement and plot the results on Fig. 4.

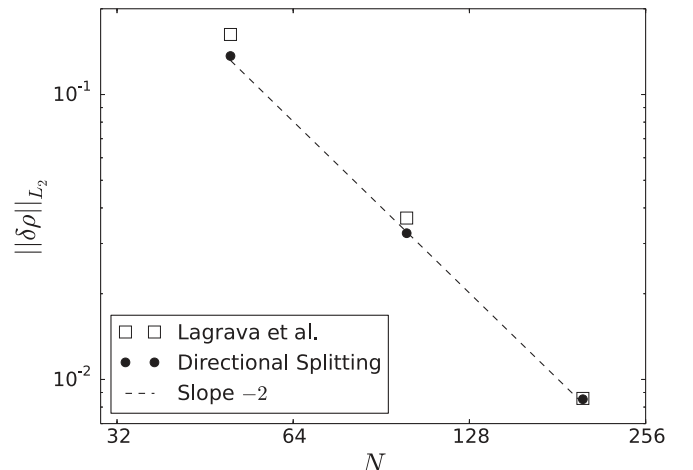


FIG. 4. Relative  $L_2$  norm of the error on the density fluctuations for the acoustic pulse: comparison at  $t^* = 100$ .

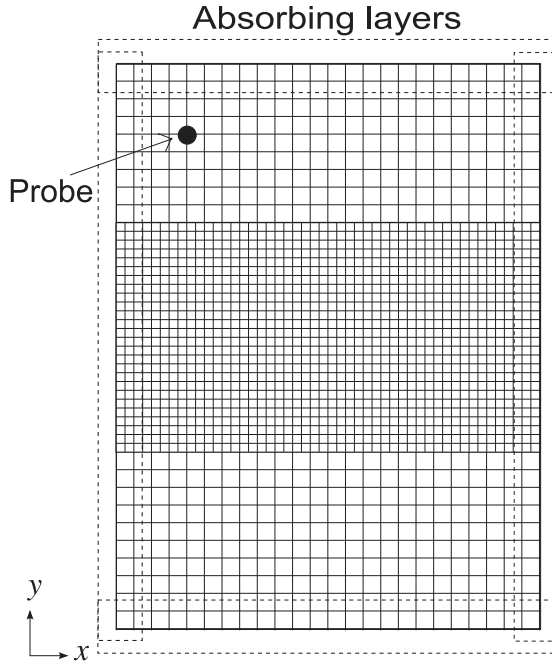


FIG. 5. Grid strategy for the vortex test case, with absorbing buffer zones on the  $x$  and  $y$  boundaries and a probe used later for PSD computation.

We can see that our approach also exhibits a second order global convergence in space and time for the density. In the next section, we demonstrate that such acoustic wave propagation cases are not sufficient to validate a grid refinement method for aeroacoustics.

**B. Convection of a vortex**

As a second benchmark, we study the convection of a Gaussian source of vorticity. The first problem is the computation of the initial states that would allow us to study precisely the evolution of the density. An incorrect initialization for this quantity would lead to the emission of

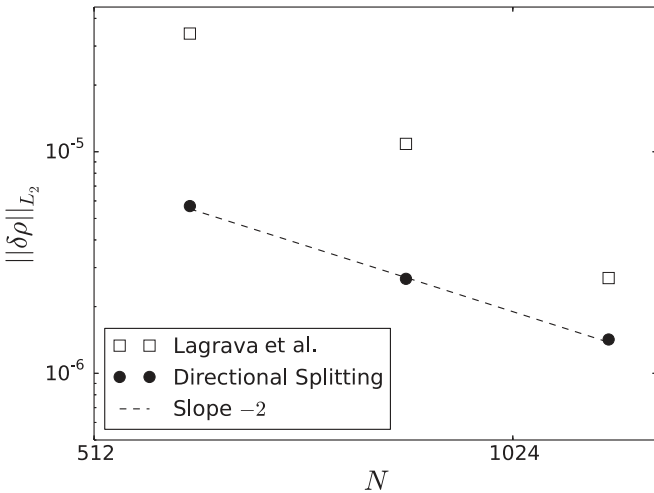


FIG. 6. Relative  $L_2$  norm of the error on the density fluctuations for the vortex case: comparison between the Lagrava *et al.* algorithm and the directional splitting approach at  $t^* = 21\,000$ .

a very strong spurious acoustic wave at  $t = 0$ . We take care of this issue in Appendix C. We choose a vortex of medium strength, which yields a more realistic simulation. In order to get stable conditions with the simple BGK model, we take a viscosity of  $\nu = 2 \times 10^{-4} \text{ m}^2/\text{s}$ .

We summarize the initial states considered here, with  $V_0$  the free stream velocity and  $\gamma = 1.4$ ; it writes (see Appendix C)

$$\begin{aligned}
 u(x, y, t_0) &= -\epsilon c_0 \frac{y}{R} \exp\left[\frac{1}{2}\left(1 - \frac{r^2}{R^2}\right)\right], \\
 v(x, y, t_0) &= V_0 + \epsilon c_0 \frac{x}{R} \exp\left[\frac{1}{2}\left(1 - \frac{r^2}{R^2}\right)\right], \\
 \rho_1(x, y, t_0) &= -\frac{\rho_0 \epsilon^2}{2} \exp\left(1 - \frac{r^2}{R^2}\right), \\
 \rho(x, y, t_0) &= \rho_0 + \rho_1 + \frac{1}{2\gamma\rho_0} \rho_1^2, \\
 \rho'(x, y, t_0) &= \rho(x, y, t_0) - \rho_0.
 \end{aligned}
 \tag{59}$$

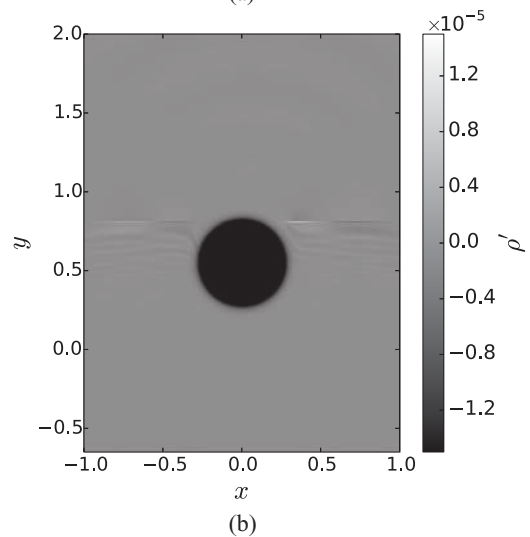
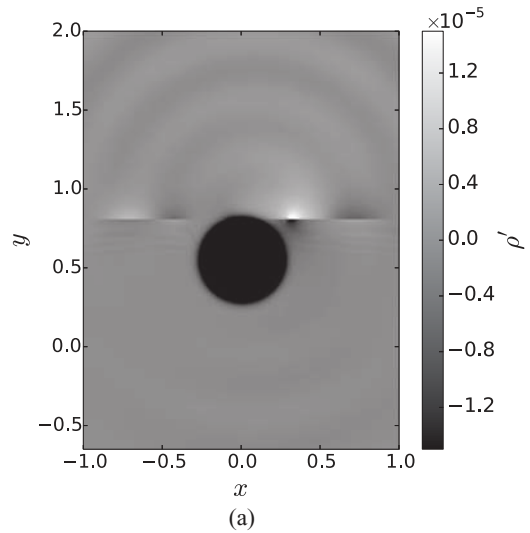


FIG. 7. Density field  $\rho'$  at time  $t^* = 9500$  (very precise closeup) for (a) the algorithm of Lagrava *et al.* and (b) the directional splitting method. The transition interface is located at  $y = 0.8$ , the coarse mesh being above the interface. The vortex is convected upwards.

For the simulations we chose, with  $M$  the free stream Mach number

$$\frac{V_0}{c_0} = M, \quad \epsilon = 0.14, \quad R = 0.1. \quad (60)$$

The vortex is thus convected upwards along the  $y$  axis at a speed  $V_0$ . We also define  $\Delta\rho_{\max} = \rho_0 - \rho(x=0, y=0, t=0) \simeq 2.7 \times 10^{-2}$ , which represents the initial density drop across the vortex. For the study of the  $L_2$  convergence, we take a highly refined ( $\Delta x = 0.001$ ) simulation on a uniform mesh as a reference. The density fluctuation in this reference simulation is denoted  $\rho'_{\text{ref}}(x, y, t) = \rho_{\text{ref}}(x, y, t) - \rho_0$ , so that the  $L_2$  error can be redefined here as

$$\|\delta\rho\|_{L_2}(t) = \sqrt{\frac{\sum_{x,y} [\rho'(x, y, t) - \rho'_{\text{ref}}(x, y, t)]^2}{\sum_{x,y} \rho'_{\text{ref}}(x, y, t)^2}}. \quad (61)$$

Our fluid domain is of size  $[3, 4, \Delta x^c]$ , bordered by nonreflecting zones (the simple *type II* model of Ref. [67] with thickness  $\delta = 0.2$  and absorbing parameter  $\chi = 1.999$ ) in order to minimize any reflection of spurious acoustics on the boundaries. In this benchmark, there is no interaction between the vortex and the absorbing layer. The domain is represented on Fig. 5 with a coarse mesh size for clarity. We denote by  $N$  the equivalent number of fine cells in the  $x$  direction and impose the velocity at  $\mathbf{u} = V_0 \mathbf{e}_y$  on all the boundaries in the  $x$  and  $y$  directions.

First of all, we run three simulations with  $M = 0.04$  and  $\Delta x^f \in \{0.0025, 0.0035, 0.005\}$  for both our model and the Lagrava algorithm, in order to study the grid convergence. The results are plotted on Fig. 6.

The second order convergence is rigorously recovered with our approach, whereas a fortuitous superconvergent behavior is observed for the Lagrava algorithm, as well as a higher level of error, especially for under-resolved simulations. The ability of a grid refinement algorithm to treat under-resolved vortices is crucial for any industrial applications. The shape of the curve for the Lagrava algorithm can be explained by third or fourth order interpolation error terms on the interface that radiate in the near field and dominate, in amplitude, the standard second

order error terms of the LBM. However, despite very low  $L_2$  errors (below 0.005% for all the simulations on Fig. 6), we show hereafter that this cannot guarantee an aeroacoustically correct simulation.

Let us now compare the corresponding instantaneous density field of the most refined case  $\Delta x^f = 0.0025$  ( $N = 1200$ ), displayed on Fig. 7.

The values are intentionally plotted with an amplitude of  $\Delta\rho = \pm 1.5 \times 10^{-5}$ , which is 18 000 times smaller than  $\Delta\rho_{\max}$ . It shows that very low amplitude spurious acoustic waves emerge from the refinement interface when crossed by the vortex and that this phenomenon is largely reduced with our directional splitting approach. Although these errors are often not visible on the pressure field because of their very low amplitude compared to hydrodynamic fluctuations, they are of paramount importance for aeroacoustics.

In order to have a better insight into these spurious fluctuations, we compute an estimation of the power spectral density (PSD) for the density at the probe location shown on Fig. 5. The PSD  $\hat{S}$  is computed using samples picked at each time step on the coarse grid between  $t^* = 8400$  and 21 000, in order to let the vortex entirely cross the transition interface: this gives us  $N_s = 6300$  samples. We use a simple periodogram estimator and a Hanning window (with energy correction), which yields

$$\hat{S}(\omega^*) = \frac{1}{N_s} |\mathcal{F}(\rho - \rho_0)(\omega^*)|^2, \quad (62)$$

where  $\mathcal{F}$  represents the window-corrected discrete Fourier transform and  $\omega^*$  the nondimensional angular frequency ( $0 \leq \omega^* \leq \pi$ ).

Theoretically, a convected vortex alone does not emit noise, which is another reason why this benchmark is interesting: the theoretical value of the density fluctuation at the location of our probe is simply  $\rho'_{\text{th}}(t) = 0$ . This was also numerically verified by studying the density field of our very fine reference simulation. Therefore, one does not need to compare the numerical results to any experimental data when using this benchmark.

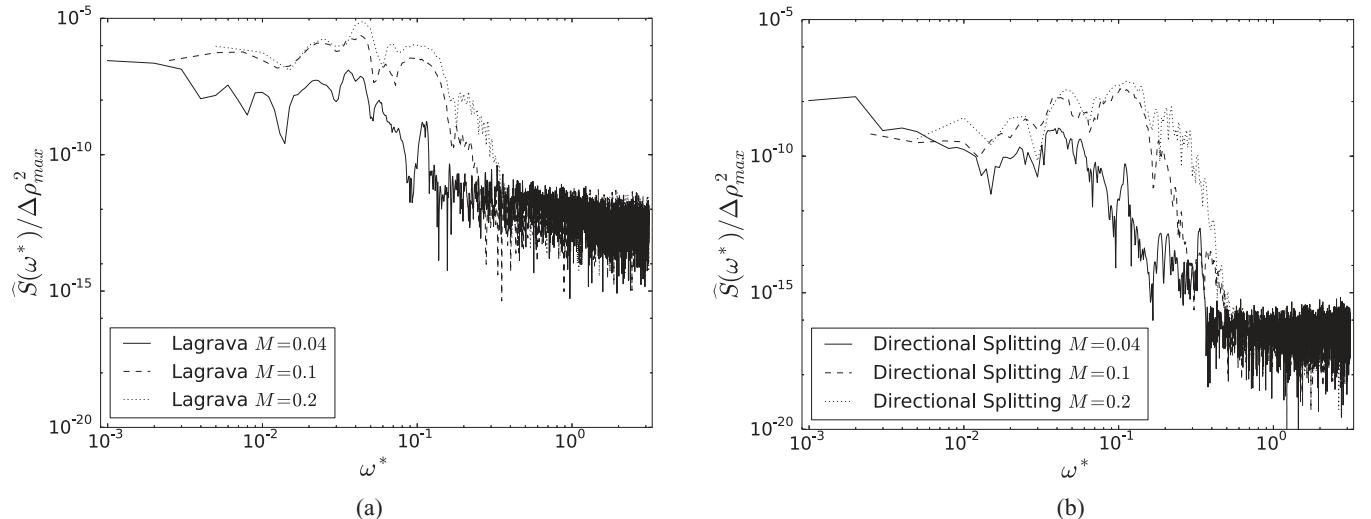


FIG. 8. Nondimensional power spectral densities of the spurious density fluctuations at a near-field location (see Fig. 5) for (a) the method of Lagrava *et al.* and (b) our directional splitting approach.



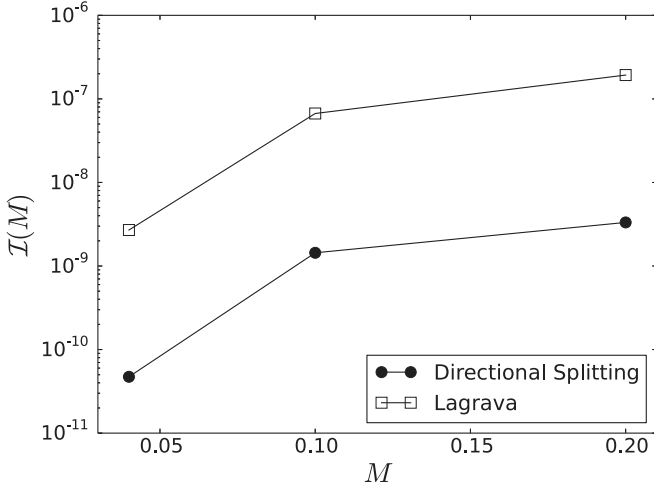


FIG. 9. Integrated power spectral densities of the spurious density fluctuations over the whole frequency spectrum, as a function of the convective Mach number.

Our results are displayed on Fig. 8 using a logarithmic scaling. In order to better compare the level of the fluctuations to the density drop across the vortex, the periodograms are normalized by  $\Delta\rho_{\max}^2$  (see Sec. VB). We also show on Fig. 9 the values of the corresponding integrated spectra (integration is made on the whole frequency domain, using a trapezium rule), as a function of the Mach number. These integrated spectra are computed as

$$\mathcal{I} = \int_0^\pi \widehat{S}(\omega^*) / \Delta\rho_{\max}^2 d\omega^*, \quad (63)$$

for each algorithm and at the same Mach numbers than in Fig. 8.

Even in the worst case (the Lagrava algorithm at  $M = 0.2$ ), these figures show that the spurious fluctuations have a very low power level compared to the vortex itself, which is why they often cannot be seen “at first” on the density or pressure field. Second, we see that our approach provides a clearly better spectral behavior on the whole spectrum. Figure 9 reveals that the power of the spurious near-field noise is reduced approximately by a factor 55 for free stream Mach numbers from 0.04 to 0.2 on this benchmark. It is also obvious that increased Mach numbers strongly deteriorate the spectra, even if the overall behavior remains better with our method.

## VI. CONCLUSION

In this paper, we developed a vertex-centered approach for treating grid refinements in the lattice Boltzmann method within the framework of aeroacoustics. This approach is based on a directional splitting of the discrete velocity Boltzmann equation. We used a simple BGK equation as a basis but showed that this work could be extended to multiple relaxation time operators. However, independently from grid refinement considerations, we emphasized on the fact that the relaxation parameters of multiple relaxation time models had to be chosen very carefully for aeroacoustic studies, so that nonphysical bulk viscosity is not added. We also recalled that the same issue appears in several variants of the entropic lattice Boltzmann

method. As far as mesh refinement is concerned, the crucial questions of interpolations and fine-to-coarse filtering were also addressed.

To assess the method, we considered two benchmark cases: the acoustic pulse propagation and the convected pseudoisotropic vortex. For these simulations, we compared the results to those of the Lagrava algorithm [45]. The six main conclusions that can be drawn thanks to this work are thus the following:

(i) Standard sound propagation benchmarks are not sufficient for an aeroacoustic validation of a grid refinement algorithm.

(ii) When a refinement interface is crossed by a vortical flow, nonphysical noise can emerge from it and propagate throughout the domain, especially in under-resolved areas. Considering the energy levels involved, these spurious fluctuations may be negligible for purely aerodynamic studies but must be taken care of for aeroacoustics.

(iii) The energy level of this spurious noise is even higher as the convective Mach number is increased.

(iv) Our approach exhibits a better behavior as far as the spectral content of the spurious interface noise is concerned: the power of the spurious acoustic noise for the vortex benchmark is reduced by a factor  $\simeq 55$  for convective Mach numbers from 0.04 up to 0.2.

(v) Our method is second order accurate in time, space, and Knudsen number, which was shown both numerically and analytically.

(vi) Most importantly, we believe that the procedure described in Sec. VB constitutes a useful benchmark for any validation of a grid refinement algorithm for aeroacoustics.

## APPENDIX A: CHAPMAN-ENSKOG EXPANSION FOR THE DIRECTIONAL SPLITTING MODEL

We start from the dimensionless splitted DVBE [Eq. (19)]:

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{c}_\alpha^\parallel \cdot \nabla^\parallel f_\alpha = \frac{-1}{\tilde{\tau}} (f_\alpha - f_\alpha^{\text{eq}}) - c_\alpha^\perp \nabla^\perp f_\alpha, \quad (\text{A1})$$

with  $\tilde{\tau} = \tau/t_0$ .

In this Appendix, we choose to write the same equation in a slightly different way: by introducing the mean free path  $\lambda = \tau c_0$ , a characteristic length  $l_0 = c_0 t_0$  (kinetic Strouhal number equal to 1), the Knudsen number  $\epsilon = \frac{\lambda}{l_0} = \frac{\tau}{t_0} \ll 1$  and the quantity  $\tilde{\tau} = 1$ , we have

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{c}_\alpha^\parallel \cdot \nabla^\parallel f_\alpha = \frac{-1}{\epsilon \tilde{\tau}} (f_\alpha - f_\alpha^{\text{eq}}) - c_\alpha^\perp \nabla^\perp f_\alpha. \quad (\text{A2})$$

The classical Chapman-Enskog expansion can be performed on this continuous system by taking

$$f[\rho(\mathbf{x}, t), \rho\mathbf{u}(\mathbf{x}, t)] = \sum_{k=0}^{\infty} \epsilon^k f^{(k)}[\rho(\mathbf{x}, t), \rho\mathbf{u}(\mathbf{x}, t)] \quad (\text{A3})$$

and

$$\frac{\partial}{\partial t} = \sum_{k=0}^{\infty} \epsilon^k \frac{\partial^{(k)}}{\partial t}. \quad (\text{A4})$$

From Eq. (A2), it is possible to make all the derivations of Sec. III that lead to the splitted lattice Boltzmann equation by

simply replacing  $\tilde{\tau}$  by  $\epsilon\tilde{\tau}$ . We obtain immediately

$$\begin{aligned} g_\alpha(\mathbf{x} + \mathbf{c}_\alpha^\parallel \tilde{\Delta}t, t + \tilde{\Delta}t) + \frac{\tilde{\Delta}t}{2} F_\alpha(\mathbf{x} + \mathbf{c}_\alpha^\parallel \tilde{\Delta}t, t + \tilde{\Delta}t) \\ = g_\alpha(\mathbf{x}, t) - \frac{\tilde{\Delta}t}{\frac{\tilde{\Delta}t}{2} + \epsilon\tilde{\tau}} [g_\alpha(\mathbf{x}, t) - g_\alpha^{\text{eq}}(\mathbf{x}, t)] \\ - \frac{\tilde{\Delta}t}{2} F_\alpha + O(\tilde{\Delta}t^3) \end{aligned} \quad (\text{A5})$$

with

$$F_\alpha = c_\alpha^\perp \nabla^\perp f_\alpha. \quad (\text{A6})$$

We now introduce the operators

$$D = \frac{\partial}{\partial t} + \nabla \cdot (\mathbf{c}_\alpha) = D^\parallel + D^\perp, \quad (\text{A7})$$

$$D^\parallel = \frac{\partial}{\partial t} + \nabla^\parallel \cdot (\mathbf{c}_\alpha^\parallel), \quad (\text{A8})$$

$$D^\perp = c_\alpha^\perp \nabla^\perp, \quad (\text{A9})$$

which gives

$$F_\alpha = D^\perp f_\alpha. \quad (\text{A10})$$

This means that for recovering the expansion of the original lattice Boltzmann equation, one simply has to take  $D = D^\parallel$  and  $D^\perp = 0$  in what follows.

A second order Taylor expansion of the left-hand side of Eq. (A5) yields

$$\begin{aligned} \tilde{\Delta}t D^\parallel g_\alpha + \frac{\tilde{\Delta}t^2}{2} D^{\parallel 2} g_\alpha \\ = \frac{-\tilde{\Delta}t}{\epsilon\tilde{\tau} + \frac{\tilde{\Delta}t}{2}} (g_\alpha - g_\alpha^{\text{eq}}) - \frac{\tilde{\Delta}t}{2} F_\alpha + O(\tilde{\Delta}t^3), \end{aligned} \quad (\text{A11})$$

with

$$g_\alpha - g_\alpha^{\text{eq}} = \left(1 + \frac{\tilde{\Delta}t}{2\epsilon\tilde{\tau}}\right) (f_\alpha - f_\alpha^{\text{eq}}). \quad (\text{A12})$$

Equation (A12) shows that there is a difference of one order of magnitude in  $\epsilon$  between  $f_\alpha^{\text{neq}}$  and  $g_\alpha^{\text{neq}}$ . Moreover, the term  $\tilde{\Delta}t/(\tilde{\Delta}t/2 + \epsilon\tilde{\tau})$  in Eq. (A11) is not of order  $1/\epsilon$ . The Chapman-Enskog expansion can therefore not be safely performed on the discrete system with  $g_\alpha$  the exact same way it is for the continuous case with  $f_\alpha$ . We note that this remark is also valid for a Chapman-Enskog expansion of the discrete velocity Boltzmann equation in its original form.

From now on, we omit the tildes for clarity. In order to recover the Navier-Stokes equations, a first order Chapman-Enskog expansion in  $\epsilon$  is sufficient. We therefore introduce other operators for simplicity:

$$D = D_0 + \epsilon D_1 + O(\epsilon^2), \quad (\text{A13})$$

$$D_0 = D_0^\parallel + D_0^\perp, \quad (\text{A14})$$

$$D_0^\parallel = \frac{\partial^{(0)}}{\partial t} + \nabla \cdot (\mathbf{c}_\alpha^\parallel), \quad (\text{A15})$$

$$D_0^\perp = c_\alpha^\perp \nabla^\perp = D^\perp, \quad (\text{A16})$$

$$D_1 = \frac{\partial^{(1)}}{\partial t}. \quad (\text{A17})$$

If we expand  $f_\alpha$  as  $f_\alpha = f_\alpha^{\text{eq}} + \epsilon f_\alpha^{(1)} + \epsilon^2 f_\alpha^{(2)} + O(\epsilon^3)$ , we obtain

$$g_\alpha = f_\alpha^{\text{eq}} + \frac{\Delta t}{2\tau} f_\alpha^{(1)} + \epsilon f_\alpha^{(1)} + \epsilon \frac{\Delta t}{2\tau} f_\alpha^{(2)} + O(\epsilon^2). \quad (\text{A18})$$

This gives us

$$\begin{aligned} (D_0^\parallel + \epsilon D_1) \left( f_\alpha^{\text{eq}} + \frac{\Delta t}{2\tau} f_\alpha^{(1)} + \epsilon f_\alpha^{(1)} + \epsilon \frac{\Delta t}{2\tau} f_\alpha^{(2)} \right) \\ + \frac{\Delta t}{2} (D_0^\parallel + \epsilon D_1)^2 \left( f_\alpha^{\text{eq}} + \frac{\Delta t}{2\tau} f_\alpha^{(1)} + \epsilon f_\alpha^{(1)} + \epsilon \frac{\Delta t}{2\tau} f_\alpha^{(2)} \right) \\ + \left[ D_0^\perp + \frac{\Delta t}{2} (D_0^\parallel + \epsilon D_1) D_0^\perp \right] (f_\alpha^{\text{eq}} + \epsilon f_\alpha^{(1)}) \\ = \frac{-1}{\tau} (f_\alpha^{(1)} + \epsilon f_\alpha^{(2)}) + O(\Delta t^2) + O(\epsilon^2). \end{aligned} \quad (\text{A19})$$

By separating the different contributions in powers of  $\epsilon$ , we derive for order  $\epsilon^0$

$$(D_0^\parallel + D_0^\perp) f_\alpha^{\text{eq}} + \frac{\Delta t}{2\tau} (A_\alpha^{(0)} + B_\alpha^{(0)}) = \frac{-1}{\tau} f_\alpha^{(1)} + O(\Delta t^2), \quad (\text{A20})$$

with

$$A_\alpha^{(0)} = D_0 f_\alpha^{(1)} + \tau D_0^2 f_\alpha^{\text{eq}}, \quad (\text{A21})$$

$$B_\alpha^{(0)} = -D_0^\perp f_\alpha^{(1)} - \tau (D_0^{\perp 2} + D_0^\parallel D_0^\perp) f_\alpha^{\text{eq}}, \quad (\text{A22})$$

and for order  $\epsilon^1$

$$\begin{aligned} D_1 f_\alpha^{\text{eq}} + (D_0^\parallel + D_0^\perp) f_\alpha^{(1)} + \frac{\Delta t}{2\tau} (A_\alpha^{(1)} + B_\alpha^{(1)}) \\ = \frac{-1}{\tau} f_\alpha^{(2)} + O(\Delta t^2), \end{aligned} \quad (\text{A23})$$

with

$$A_\alpha^{(1)} = (D_1 + \tau D_0^2) f_\alpha^{(1)} + D_0 f_\alpha^{(2)} + 2\tau D_0 D_1 f_\alpha^{\text{eq}}, \quad (\text{A24})$$

$$B_\alpha^{(1)} = -D_0^\perp f_\alpha^{(2)} - \tau (D_0^{\perp 2} + D_0^\parallel D_0^\perp) f_\alpha^{(1)} - \tau D_0^\perp D_1 f_\alpha^{\text{eq}}. \quad (\text{A25})$$

For the original lattice Boltzmann equation, the terms  $B_\alpha^{(i)}$  are not present.

As far the  $A_\alpha^{(i)}$  are concerned, they are both of order  $O(\Delta t)$ , which guarantees the second order accuracy in time and space of the original LBE. For  $A_\alpha^{(0)}$ , the result is obtained immediately by applying the operator  $D_0$  to Eq. (A20). For  $A_\alpha^{(1)}$ , we obtain by applying the operator  $D_0$  to Eq. (A23)

$$D_0^2 f_\alpha^{(1)} = -\frac{D_0}{\tau} f_\alpha^{(2)} - D_0 D_1 f_\alpha^{\text{eq}} + O(\Delta t). \quad (\text{A26})$$

According to Eq. (A20) we also have

$$D_0 D_1 f_\alpha^{\text{eq}} = -\frac{D_1}{\tau} f_\alpha^{(1)} + O(\Delta t). \quad (\text{A27})$$

Inserting Eqs. (A26) and (A27) into Eq. (A24) ends the proof for  $A_\alpha^{(1)}$ .

This conclusion also applies to the  $B_\alpha^{(i)}$ . Applying the operator  $D_0^\perp$  to Eq. (A20) and inserting the resulting expression  $D_0^\perp f_\alpha^{(1)}$  of into Eq. (A22) proves that  $B_\alpha^{(0)}$  is of order  $O(\Delta t)$ . For  $B_\alpha^{(1)}$ , applying the operator  $D_0^\perp$  to Eq. (A23) and inserting

the resulting expression  $D_0^+ f_\alpha^{(2)}$  of into Eq. (A25) ends the proof. In the end, this gives us for order  $\epsilon^0$

$$D_0 f_\alpha^{\text{eq}} = \frac{-1}{\tau} f_\alpha^{(1)} + O(\Delta t^2), \quad (\text{A28})$$

and for order  $\epsilon^1$

$$D_1 f_\alpha^{\text{eq}} + D_0 f_\alpha^{(1)} = \frac{-1}{\tau} f_\alpha^{(2)} + O(\Delta t^2), \quad (\text{A29})$$

which is exactly the result obtained with the original LBE.

The fluid equations are obtained by calculating the discrete moments of Eqs. (A20) and (A23). The conservation equation for mass at first order in Knudsen number corresponds to the sum of the zeroth order moment of Eqs. (A20) and (A23). The same principle is followed in order to obtain the momentum conservation, but with the first order moments.

For the conservation of mass at order  $\epsilon^0$ , we have

$$\frac{\partial^{(0)}\rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) + O(\Delta t^2) = 0. \quad (\text{A30})$$

For the conservation of mass at order  $\epsilon^1$ , we obtain

$$\frac{\partial^{(1)}\rho}{\partial t} + O(\Delta t^2) = 0. \quad (\text{A31})$$

For the conservation of momentum at order  $\epsilon^0$ , we have

$$\frac{\partial^{(0)}\rho \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{\Pi}^{\text{eq}} + O(\Delta t^2) = 0, \quad (\text{A32})$$

with  $\mathbf{\Pi}^{\text{eq}} = \rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I} = \rho \mathbf{u} \otimes \mathbf{u} + \rho c_s^2 \mathbf{I}$ .

For the conservation of momentum at order  $\epsilon^1$ , we have

$$\frac{\partial^{(1)}\rho \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{\Pi}^{(1)} + O(\Delta t^2) = 0, \quad (\text{A33})$$

with  $\mathbf{\Pi}^{(k)} = \sum_\alpha \mathbf{c}_\alpha \otimes \mathbf{c}_\alpha f_\alpha^{(k)}$ .

Finally, we obtain the full fluid equations by summing the contributions of order  $\epsilon^0$  and  $\epsilon^1$  {i.e., [(A30)+ $\epsilon$  (A31)] and [(A32)+ $\epsilon$  (A33)]}, which gives us a solution equivalent to a first order Chapman-Enskog expansion:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) + O(\Delta t^2) + O(\epsilon^2) = 0, \quad (\text{A34})$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{\Pi}^{\text{eq}} + \epsilon \mathbf{\Pi}^{(1)}) + O(\Delta t^2) + O(\epsilon^2) = 0, \quad (\text{A35})$$

with

$$\epsilon \mathbf{\Pi}^{(1)} = -2\epsilon \tau \rho c_s^2 [\mathbf{S} + O(\text{Ma}^3)] = -2\mu [\mathbf{S} + O(\text{Ma}^3)], \quad (\text{A36})$$

in which  $\mathbf{S}$  is the strain rate tensor, given by

$$\mathbf{S} = \frac{\nabla \mathbf{u} + {}^T \nabla \mathbf{u}}{2}. \quad (\text{A37})$$

This shows that the Navier-Stokes equations are, within the framework of an athermal or weakly compressible approxi-

mation, retrieved with a second order accuracy in time, space, and Knudsen number, provided that the gradient term  $F_\alpha$  is evaluated with, at least, a second order approximation.

Several convergence studies (such as Refs. [56,68]) show nondimensional equations with a viscous term written as

$$\mu = \Delta t \left( \bar{\tau}_g - \frac{1}{2} \right) \rho c_s^2, \quad (\text{A38})$$

which yields an apparently first order in time fluid equation. However, the nondimensional relaxation parameter  $\bar{\tau} = \bar{\tau}_g - \frac{1}{2}$  scales as  $O(\frac{M}{\text{Re}\Delta t}) = O(\epsilon/\Delta t)$  [54], so the viscosity of Eq. (A38) reduces to a  $O(\epsilon)$  effect, equivalently as in Eqs. (A35) and (A36). The difference between those convergence studies and ours is therefore purely formal and comes from the fact that it is not  $\bar{\tau} = \bar{\tau}_g - \frac{1}{2}$  that appears in Eq. (A36) but  $\tau = \tau_g - \frac{\Delta t}{2} = \Delta t \bar{\tau}$  instead.

## APPENDIX B: INFLUENCE OF THE RESTRICTION FILTER ON ACCURACY

In acoustic scaling, we can write the restricted collision by modifying Eq. (44) with its error term, according to Eq. (50), Ref. [63], Eq. (10), and by denoting  $\Delta$  the Laplacian operator:

$$\begin{aligned} \hat{g}^f(\mathbf{x}, t) &= g_\alpha^f(\mathbf{x}, t) - \frac{\tilde{\Delta} t}{\tilde{\tau}_g} [\mathcal{R}(g_\alpha^{\text{neq},f})(\mathbf{x}, t)] + O(\tilde{\Delta} t^3) \\ &= g_\alpha^f(\mathbf{x}, t) - \frac{\tilde{\Delta} t}{\tilde{\tau}_g} \left[ g_\alpha^{\text{neq},f}(\mathbf{x}, t) + \frac{(\sigma^* \tilde{\Delta} x)^2}{2} \Delta g_\alpha^{\text{neq},f} \right. \\ &\quad \left. + O(\tilde{\Delta} x^4) \right] + O(\tilde{\Delta} t^3) \\ &= g_\alpha^f(\mathbf{x}, t) - \frac{\tilde{\Delta} t}{\tilde{\tau}_g} \left[ g_\alpha^{\text{neq},f}(\mathbf{x}, t) + \frac{(c_0 \sigma^* \tilde{\Delta} t)^2}{2c_s^2} \Delta g_\alpha^{\text{neq},f} \right. \\ &\quad \left. + O(\tilde{\Delta} t^4) \right] + O(\tilde{\Delta} t^3). \end{aligned} \quad (\text{B1})$$

Since  $\tilde{\tau}_g$  depends on  $\tilde{\Delta} t$ , we cannot instantly prove that the accuracy of the scheme remains unaffected by the filter.

Let us define  $\eta = (c_0 \sigma^*)^2 / (2c_s^2)$  and drop the time and space variables for clarity. We obtain with a Taylor expansion and with the operator defined in Eq. (A7) (without the parallel and orthogonal decomposition)

$$\begin{aligned} \tilde{\Delta} t D g_\alpha^f + \frac{\tilde{\Delta} t^2}{2} D^2 g_\alpha^f &= \frac{\tilde{\Delta} t}{\epsilon \tilde{\tau} + \frac{\tilde{\Delta} t}{2}} [g_\alpha^{\text{neq},f} + \eta \Delta g_\alpha^{\text{neq},f} \tilde{\Delta} t^2 \\ &\quad + O(\tilde{\Delta} t^4)] + O(\tilde{\Delta} t^3). \end{aligned} \quad (\text{B2})$$

We do not expand the left-hand side of Eq. (B2) thanks to Eq. (A12) since it is already the one that appears in the Taylor expansion of the original lattice Boltzmann equation. We are only interested in the right-hand side, where the modifications due to the filter appear. Thanks to Eq. (A12) we have

$$\Delta g_\alpha^{\text{neq},f} = \left( 1 + \frac{\tilde{\Delta} t}{2\epsilon \tilde{\tau}} \right) \Delta f_\alpha^{\text{neq},f}. \quad (\text{B3})$$

Equations (B2), (B3), and (A12) yield

$$\begin{aligned} Dg_\alpha^f + \frac{\tilde{\Delta}t}{2} D^2 g_\alpha^f \\ = \frac{1}{\epsilon\tilde{\tau}} [f_\alpha^{\text{neq},f} + \eta\Delta f_\alpha^{\text{neq},f} \tilde{\Delta}t^2 + O(\tilde{\Delta}t^4)] + O(\tilde{\Delta}t^2), \end{aligned} \quad (\text{B4})$$

so that we finally have

$$Dg_\alpha^f + \frac{\tilde{\Delta}t}{2} D^2 g_\alpha^f = \frac{1}{\epsilon\tilde{\tau}} f_\alpha^{\text{neq},f} + O(\tilde{\Delta}t^2), \quad (\text{B5})$$

which is exactly what is obtained for the original lattice Boltzmann equation. This shows that the accuracy of the lattice Boltzmann scheme remains unaffected by the restriction filter on the collision operator.

### APPENDIX C: PSEUDOISENTROPIC VORTEX

There are several analytical solutions for vortex evolutions in fluid mechanics. However, our D3Q19 velocity discretization simulates a fluid that does not correspond to any of these situations: the equivalent macroscopic system of our LBE is an athermal and weakly compressible version of the Navier-Stokes equations. It implies that temperature does not exist in our approach since no equation for energy conservation is solved. However, we can try to find a way around and obtain an approximate solution in order to, at least, initialize the case properly.

We take as a starting point the isentropic vortex solution of the compressible Euler equations, studied by Yee *et al.* [69] or Shu *et al.* [70]. The initial states are given in terms of velocity and temperature fluctuations, which yields the pressure fluctuations. We again write the fluctuations with primed variables. The strength of the vortex is controlled by a parameter  $\epsilon$ . We also note  $\mathbf{u} = (u, v)$ . We pose  $u = u'$ ,  $v = V_0 + v'$ ,  $p = p_0 + p'$ ,  $\rho = \rho_0 + \rho'$ ,  $T = T_0 + T'$ ,  $r = \sqrt{x^2 + y^2}$ , and  $R$  a characteristic radius. We obtain the following system, equivalent to the one of [69] with their parameter  $\alpha$  taken as  $\alpha = 1/2$ :

$$\begin{aligned} u'(x, y, t_0) &= -\epsilon c_0 \frac{y}{R} \exp\left[\frac{1}{2}\left(1 - \frac{r^2}{R^2}\right)\right], \\ v'(x, y, t_0) &= \epsilon c_0 \frac{x}{R} \exp\left[\frac{1}{2}\left(1 - \frac{r^2}{R^2}\right)\right], \\ T'(x, y, t_0) &= -\epsilon^2 c_0^2 \frac{(\gamma - 1)}{2\gamma} \exp\left(1 - \frac{r^2}{R^2}\right). \end{aligned} \quad (\text{C1})$$

In accordance with [69], this system is associated to the equation  $p = \rho T$  for a perfect gas. This yields the expression of the reference pressure  $p_0 = \rho_0 T_0$ . The gas has an isentropic evolution, we thus have the relation  $pT^{\frac{\gamma}{1-\gamma}} = p_0 T_0^{\frac{\gamma}{1-\gamma}}$  with  $\gamma$  the isentropic constant, which yields

$$p(x, y, t_0) = p_0 \left(1 + \frac{T'}{T_0}\right)^{\frac{\gamma}{\gamma-1}}. \quad (\text{C2})$$

The exact solution with given initial states is a passive convection of the vortex with  $V_0$ .

We can seek for an approximation of this solution in the limit of weakly compressible and nearly inviscid flows. If we perform a first order Taylor expansion around one on the obtained pressure for very small  $T'$ , we have

$$\begin{aligned} p(x, y, t_0) &= p_0 \left(1 + \frac{\gamma}{\gamma - 1} \frac{T'}{T_0}\right) + O\left(\frac{T'^2}{T_0^2}\right) \\ &= p_0 - \frac{\rho_0 \epsilon^2 c_0^2}{2} \exp\left(1 - \frac{r^2}{R^2}\right) + O(\epsilon^4). \end{aligned} \quad (\text{C3})$$

We now need to adapt this expression to be used with a lattice Boltzmann D3Q19 solver. The dimensional athermal equation of state of our LBM gives

$$p = \rho c_0^2 \Rightarrow p_0 = \rho_0 c_0^2. \quad (\text{C4})$$

This is equivalent to setting the value of the nonphysical constant  $T_0$  to  $T_0 = c_0^2$ , which allows us to obtain the following expression for the density:

$$\rho(x, y, t_0) = \rho_0 \left[1 - \frac{\epsilon^2}{2} \exp\left(1 - \frac{r^2}{R^2}\right)\right] + O(\epsilon^4). \quad (\text{C5})$$

Finally, we get, with  $V_0$  the free stream velocity

$$\begin{aligned} u(x, y, t_0) &= -\epsilon c_0 \frac{y}{R} \exp\left[\frac{1}{2}\left(1 - \frac{r^2}{R^2}\right)\right], \\ v(x, y, t_0) &= V_0 + \epsilon c_0 \frac{x}{R} \exp\left[\frac{1}{2}\left(1 - \frac{r^2}{R^2}\right)\right], \\ \rho(x, y, t_0) &= \rho_0 - \frac{\rho_0 \epsilon^2}{2} \exp\left(1 - \frac{r^2}{R^2}\right). \end{aligned} \quad (\text{C6})$$

In order to reduce even more the amplitude of the initial acoustic wave, one should expand (C2) to higher orders: in our work, we chose to use a second order expansion, which clearly reduced the initial emission. We obtain

$$\begin{aligned} u(x, y, t_0) &= -\epsilon c_0 \frac{y}{R} \exp\left[\frac{1}{2}\left(1 - \frac{r^2}{R^2}\right)\right], \\ v(x, y, t_0) &= V_0 + \epsilon c_0 \frac{x}{R} \exp\left[\frac{1}{2}\left(1 - \frac{r^2}{R^2}\right)\right], \\ \rho_1(x, y, t_0) &= -\frac{\rho_0 \epsilon^2}{2} \exp\left(1 - \frac{r^2}{R^2}\right), \\ \rho(x, y, t_0) &= \rho_0 + \rho_1 + \frac{1}{2\gamma\rho_0} \rho_1^2 \end{aligned} \quad (\text{C7})$$

with  $\gamma = 1.4$ .

- [1] C. K. Aidun and J. R. Clausen, *Annu. Rev. Fluid Mech.* **42**, 439 (2010).
- [2] O. Malaspinas and P. Sagaut, *J. Fluid Mech.* **700**, 514 (2012).
- [3] C. Van Treeck, E. Rank, M. Krafczyk, J. Tölke, and B. Nachtwey, *Comput. Fluids* **35**, 863 (2006).
- [4] K. Suga, Y. Kuwata, K. Takashima, and R. Chikasue, *Comput. Math. Appl.* **69**, 518 (2015).
- [5] E. K. Far, M. Geier, K. Kutscher, and M. Krafczyk, *Comput. Fluids* **140**, 222 (2016).
- [6] S. Marié, D. Ricot, and P. Sagaut, *J. Comput. Phys.* **228**, 1056 (2009).
- [7] S. Marié, Ph.D. thesis, Université Pierre et Marie Curie-Paris VI, 2008.
- [8] B. Arguillat, D. Ricot, C. Bailly, and G. Robert, *J. Acoust. Soc. Am.* **128**, 1647 (2010).
- [9] X. Gloerfelt and J. Berland, *J. Fluid Mech.* **723**, 318 (2013).
- [10] X. Gloerfelt and P. Lafon, *Comput. Fluids* **37**, 388 (2008).
- [11] M. Tsutahara, *Fluid Dyn. Res.* **44**, 045507 (2012).
- [12] S. Geller, M. Krafczyk, J. Tölke, S. Turek, and J. Hron, *Comput. Fluids* **35**, 888 (2006).
- [13] A. Fakhari and T. Lee, *Comput. Fluids* **107**, 205 (2015).
- [14] G. Peng, H. Xi, C. Duncan, and S.-H. Chou, *Phys. Rev. E* **59**, 4675 (1999).
- [15] M. Tsutahara, T. Kondo, and K. Mochizuki, in *12th AIAA/CEAS Aeroacoustics Conference (27th AIAA Aeroacoustics Conference)* (AIAA, Reston, VA, 2006), p. 2570.
- [16] M. Stiebler, J. Tölke, and M. Krafczyk, *Comput. Fluids* **35**, 814 (2006).
- [17] H. Chen, O. Filippova, J. Hoch, K. Molvig, R. Shock, C. Teixeira, and R. Zhang, *Phys. A (Amsterdam)* **362**, 158 (2006).
- [18] H. Chen, *Phys. Rev. E* **58**, 3955 (1998).
- [19] M. Junk, A. Klar, and L.-S. Luo, *J. Comput. Phys.* **210**, 676 (2005).
- [20] M. Rheinländer, *J. Stat. Phys.* **121**, 49 (2005).
- [21] M. Geier, A. Greiner, and J. Korvink, *Eur. Phys. J.: Spec. Top.* **171**, 173 (2009).
- [22] J. Qi, H. Klimach, and S. Roller, *Comput. Math. Appl.* (2016).
- [23] J. Tölke and M. Krafczyk, *Comput. Math. Appl.* **58**, 898 (2009).
- [24] S. M. Guzik, T. H. Weisgraber, P. Colella, and B. J. Alder, *J. Comput. Phys.* **259**, 461 (2014).
- [25] M. Schönherr, K. Kucher, M. Geier, M. Stiebler, S. Freudiger, and M. Krafczyk, *Comput. Math. Appl.* **61**, 3730 (2011).
- [26] S. Geller, S. Uphoff, and M. Krafczyk, *Comput. Math. Appl.* **65**, 1956 (2013).
- [27] M. Geier, M. Schönherr, A. Pasquali, and M. Krafczyk, *Comput. Math. Appl.* **70**, 507 (2015).
- [28] M. Stiebler, M. Krafczyk, S. Freudiger, and M. Geier, *Comput. Math. Appl.* **61**, 3475 (2011).
- [29] B. Dorschner, F. Bösch, S. S. Chikatamarla, K. Boulouchos, and I. V. Karlin, *J. Fluid Mech.* **801**, 623 (2016).
- [30] Y. Kuwata and K. Suga, *J. Comput. Phys.* **311**, 348 (2016).
- [31] Y. Peng, C. Shu, Y.-T. Chew, X. Niu, and X.-Y. Lu, *J. Comput. Phys.* **218**, 460 (2006).
- [32] N. Pellerin, S. Leclaire, and M. Reggio, *Comput. Math. Appl.* **70**, 3001 (2015).
- [33] G. Eitel-Amor, M. Meinke, and W. Schröder, *Comput. Fluids* **75**, 127 (2013).
- [34] S. Chen, C. Peng, Y. Teng, L.-P. Wang, and K. Zhang, *Comput. Fluids* **136**, 228 (2016).
- [35] D. Yu, R. Mei, and W. Shyy, *Int. J. Numer. Methods Fluids* **39**, 99 (2002).
- [36] D. Yu and S. S. Girimaji, *Phys. A (Amsterdam)* **362**, 118 (2006).
- [37] H. Liu, J. G. Zhou, and R. Burrows, *Comput. Fluids* **38**, 1108 (2009).
- [38] B. Crouse, Ph.D. thesis, TU München, 2003.
- [39] O. Filippova, S. Succi, F. Mazzocco, C. Arrighetti, G. Bella, and D. Hänel, *J. Comput. Phys.* **170**, 812 (2001).
- [40] O. Filippova and D. Hänel, *J. Comput. Phys.* **147**, 219 (1998).
- [41] A. Dupuis and B. Chopard, *Phys. Rev. E* **67**, 066707 (2003).
- [42] H. Touil, D. Ricot, and E. Lévêque, *J. Comput. Phys.* **256**, 220 (2014).
- [43] M. Rohde, D. Kandhai, J. Derksen, and H. Van den Akker, *Int. J. Numer. Methods Fluids* **51**, 439 (2006).
- [44] Z. Yu and L.-S. Fan, *J. Comput. Phys.* **228**, 6456 (2009).
- [45] D. Lagrava, O. Malaspinas, J. Latt, and B. Chopard, *J. Comput. Phys.* **231**, 4808 (2012).
- [46] A. Pasquali, Ph.D. thesis, TU Braunschweig, 2016.
- [47] M. Hasert and S. P. Roller, Ph.D. thesis, Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen, 2014.
- [48] D. Ricot, S. Marié, P. Sagaut, and C. Bailly, *J. Comput. Phys.* **228**, 4478 (2009).
- [49] P.-T. Lew, L. Mongeau, and A. Lyrantzis, *J. Acoust. Soc. Am.* **128**, 1118 (2010).
- [50] D. Casalino, A. F. Ribeiro, E. Fares, and S. Nölting, *AIAA J.* **52**, 1232 (2014).
- [51] K. Habibi and L. Mongeau, *Appl. Acoust.* **87**, 153 (2015).
- [52] A. de Jong, H. Bijl, A. Hazir, and J. Wiedemann, *J. Sound Vib.* **332**, 1687 (2013).
- [53] X. Shan, X.-F. Yuan, and H. Chen, *J. Fluid Mech.* **550**, 413 (2006).
- [54] P. J. Dellar, *J. Comput. Phys.* **190**, 351 (2003).
- [55] P. J. Dellar, *Phys. Rev. E* **64**, 031203 (2001).
- [56] Z. Guo, C. Zheng, and B. Shi, *Phys. Rev. E* **65**, 046308 (2002).
- [57] D. D’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo, *Philos. Trans. R. Soc. London A* **360**, 437 (2002).
- [58] F. Dubois and P. Lallemand, *Comput. Math. Appl.* **61**, 3404 (2011).
- [59] M. Geier, A. Greiner, and J. G. Korvink, *Phys. Rev. E* **73**, 066705 (2006).
- [60] B. Dorschner, N. Frapolli, S. S. Chikatamarla, and I. V. Karlin, *Phys. Rev. E* **94**, 053311 (2016).
- [61] K. N. Premnath and S. Banerjee, *Phys. Rev. E* **80**, 036702 (2009).
- [62] A. Fakhari and T. Lee, *Phys. Rev. E* **89**, 033310 (2014).
- [63] S. P. Thampi, S. Ansumali, R. Adhikari, and S. Succi, *J. Comput. Phys.* **234**, 1 (2013).
- [64] F. Falissard, *J. Comput. Phys.* **253**, 344 (2013).
- [65] J. Latt and B. Chopard, *Math. Comput. Simul.* **72**, 165 (2006).
- [66] C. K. Tam and J. C. Webb, *J. Comput. Phys.* **107**, 262 (1993).
- [67] H. Xu and P. Sagaut, *J. Comput. Phys.* **245**, 14 (2013).
- [68] F. Dubois, *Comput. Math. Appl.* **55**, 1441 (2008).
- [69] H. C. Yee, N. D. Sandham, and M. Djomehri, *J. Comput. Phys.* **150**, 199 (1999).
- [70] C.-W. Shu, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations* (Springer, Berlin, 1998), pp. 325–432.