



HAL
open science

Comprehensive comparison between the lattice Boltzmann and Navier–Stokes methods for aerodynamic and aeroacoustic applications

Alexandre Suss, Ivan Mary, Thomas Le Garrec, Simon Marié

► To cite this version:

Alexandre Suss, Ivan Mary, Thomas Le Garrec, Simon Marié. Comprehensive comparison between the lattice Boltzmann and Navier–Stokes methods for aerodynamic and aeroacoustic applications. *Computers and Fluids*, 2023, 257, pp.1-22. 10.1016/j.compfluid.2023.105881 . hal-04091897

HAL Id: hal-04091897

<https://cnam.hal.science/hal-04091897v1>

Submitted on 21 Jun 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.

Copyright

Comprehensive comparison between the lattice Boltzmann and Navier-Stokes methods for aerodynamic and aeroacoustic applications

Alexandre Suss^{1,*}, Ivan Mary¹, Thomas Le Garrec¹, Simon Marié^{2,3}

¹*DAAA, ONERA, Université Paris Saclay, F-92322 Châtillon - France*

²*Laboratoire DymFluid, F-75013 Paris - France*

³*Conservatoire National des Arts et Métiers, F-75003 Paris - France*

Abstract

In an effort to determine which Computational Fluid Dynamics method offers the best trade-off between accuracy and computational cost for aerodynamic and aeroacoustic applications, the lattice Boltzmann and finite-volume Navier-Stokes methods are compared. Unlike previous studies, the present framework enables a fair and unbiased comparison of the core capabilities of each numerical approach and focuses on schemes of practical relevance. With the aim of providing a comprehensive comparison of the methods, an extended von Neumann analysis is performed and the High Performance Computing capacities of both methods are thoroughly discussed. In addition, it is also shown through computations on canonical test cases that a “time to solution” metric has to be considered in order to objectively assess the suitability of one particular numerical method.

Three main conclusions are drawn: (1) both the lattice Boltzmann and Navier-Stokes schemes exhibit an anisotropic dissipative behavior, (2) a cell update using the lattice Boltzmann method is 2 to 3 times faster than with the Navier-Stokes method dedicated to cartesian grids, and (3) the use of the “time to solution” metric demonstrates that the relevance of one method over the other is closely linked to the underlying physics and the intended error target. In the light of these results, decision aids are provided to assist in selecting the most efficient method for a given application.

Keywords: lattice Boltzmann method, Navier-Stokes, finite-volume method, von Neumann analysis, High Performance Computing, Comparison

1. Introduction

The ability to simulate aerodynamic flows using Computational Fluid Dynamics (CFD) has progressed rapidly during the last decades owing to the growth of the computational power and the increasing accuracy and robustness of CFD solvers [1]. While Reynolds Averaged Navier-
5 Stokes (RANS) simulations are the current workhorse of the aerospace industry, the understanding of unsteady turbulent phenomena is becoming a crucial issue. As such, the use of high-fidelity

*Corresponding author.

Email address: `alexandre.suss@onera.fr` (Alexandre Suss)

methods, such as Large Eddy Simulations (LES) is getting increasingly compulsory. Hence, the development of efficient and accurate flow solvers that can produce LES-level results in reasonable amount of computational time is a major and active field of research.

10 Among all the available numerical methods, the direct discretisation of the Navier-Stokes (NS) equations through the finite-volume method (hereafter referred to as Navier-Stokes methods) is the most common approach in the CFD community since it offers a good trade-off between flexibility and robustness. However, in the last two decades, the lattice Boltzmann method (LBM) has become increasingly popular thanks to its simplicity of use and its efficiency [2, 3]. In this context,
15 Löhner [4] suggests that these two different numerical approaches might be the first ones to achieve industrial-level LES in the near future. Consequently, one question which naturally arises is: Which method is the most competitive, in terms of accuracy and computational cost, on canonical unsteady aerodynamic and aeroacoustic applications?

While the comparison between the Navier-Stokes and lattice Boltzmann methods has received a
20 growing attention in the recent years, it is still unclear how to answer the question raised above. Indeed, the comparison between numerical methods is not as straightforward since several parameters might drastically influence their accuracy and performance and thereby have to be controlled [5]. As such, most of the comparative studies available to date are either too theoretical or too specific (by focusing on a single industrial configuration) which makes it difficult to assess the relative merits of
25 the LBM when compared to the Navier-Stokes methods and conversely. From a purely theoretical point of view, Marié *et al.* [3] were the first to compare the intrinsic capabilities of both approaches through a von Neumann analysis. In particular, they demonstrate the low dissipative property and the low computational intensity of the LBM. Nevertheless, their results are only valid for the LBM-BGK model which has a severely limited range of applications because of its poor stability
30 [6]. While the stability of the LBM has been greatly enhanced by the introduction of advanced collision models, a comprehensive study assessing their numerical properties with respect to classical NS schemes is still lacking. Considering only runtimes, most of the published results introduce a skew in their conclusions. Indeed, in many cases the LB and NS methods are implemented within independent solvers with different standards in terms of High Performance Computing (HPC) opti-
35 misation and are thereby not really competing. Consequently, there are many covert ways in which some performance demonstrations fall short of practical interest. However, it is worth mentioning the study of Wichmann *et al.* [7] which appears to be the first to pave the way towards a fair runtime comparison between the LB and the finite-difference Navier-Stokes method. They show that the LBM is particularly efficient for complex flow problems and coarse tolerances. Yet, their
40 conclusion, even though being valuable and based on a variety of tuning parameters, still lacks of

concrete relevance. Indeed, the two solvers used in their study (which are highly simplified and unreflective of LES solvers) were developed by different teams and do not rely on the same HPC core thereby biasing the outcome. Besides, comparative studies on industrial-level LES applications can also hardly be used to draw informed conclusions. In the literature, a large variety of configurations were studied such as landing gears [8, 9, 10], automotive models [11], swirled flows typical of aeronautical combustion chambers [12], linear cascade configuration [13], and reactive flows [14]. One the whole, these surveys are in favor of the LBM indicating speedups up to 10 with respect to Navier-Stokes solvers while providing accurate results. Albeit interesting, these conclusions should be taken with caution as they do not compare competing approaches. Indeed, most of these studies are based on comparisons among results obtained by different teams around the globe using different solvers and running options. For instance, it makes little sense to compare the performance of structured and unstructured solvers or the accuracy of wall-modeled and wall-resolved computations as each of these choices leads to major differences in throughput which are expected to dramatically change the outcome of the aforementioned comparisons.

In this context, the aim of the present study is to perform a comprehensive and fair one-to-one comparison of the lattice Boltzmann and Navier-Stokes methods for aerodynamic and aeroacoustic applications. To this end, schemes of practical relevance are investigated and great care is taken such as to avoid all possible sources of bias when comparing both numerical methods. Moreover, rather than targeting complex industrial-like configurations, the focus is made on low Mach number canonical test cases representative of LES requirements for which an analytical solution is known and therefore for which the error levels can be precisely quantified. For all these reasons, only the core capabilities of each method (i.e. their ability to propagate acoustic waves, wakes or to simulate turbulent flows) are of interest here, which means that the present comparison does not cover configurations involving walls. Indeed, such computations introduce numerous skews which are found to be independent of the numerical methods, especially when discussing the way the wall is numerically handled since most of practical flows around obstacles require the use of wall models whose error outweighs the one of the scheme in the near wall region [15, 16]. In addition, it is very cumbersome to establish precise error metrics for such flows given the different levels of validation of unsteady data issued from CFD (see the hierarchy proposed in [17]).

The contribution of the present study is threefold and covers all the aspects of the numerical methods by: (1) investigating their intrinsic dispersive and dissipative properties, (2) thoroughly discussing their computational cost and parallel scalability, and (3) expressing their efficiency in carrying out practical simulations through a “time to solution” metric. Thus, notwithstanding the conscious choice not to deal with wall bounded flows, general and rigorous decision support on the

75 suitability of one particular CFD method over the other are provided for canonical aeroacoustic and free shear flow problems commonly encountered in Large Eddy Simulations.

The paper is organised as follows. Section 2 is devoted to the presentation of the lattice Boltzmann and Navier-Stokes methods. Then, the theoretical dispersion and dissipation errors of both methods are discussed in Section 3 through an extended von Neumann analysis. Section 4 compares both methods in terms of their intrinsic HPC capabilities. Finally, the methods are further
80 compared thanks to representative aerodynamic and aeroacoustic test cases in Section 5.

2. Numerical framework

The comparison between the lattice Boltzmann and Navier-Stokes methods is performed in the framework of ONERA’s Cassiopée/FAST CFD environment [18, 19, 20]. This consists in a
85 set of efficient and interoperable CFD modules sharing the same code architecture, data structure and parallel processing functions. This framework is a perfect candidate for method-to-method comparisons since it avoids the heterogeneities and the computational overhead of independent CFD codes implementing one single numerical method and optimized with different standards.

2.1. Finite-volume Navier-Stokes method

90 The three-dimensional compressible Navier-Stokes (NS) equations are solved using ONERA’s FastS solver dedicated to multi-block structured grids. Starting from the conservative form of the Navier-Stokes equations:

$$\frac{\partial}{\partial t} \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) - \nabla \cdot \mathbf{F}^\nu(\mathbf{U}) = \mathbf{0}, \quad (1)$$

where $\mathbf{U} = (\rho, \rho u_x, \rho u_y, \rho u_z, \rho E)^t$, $\mathbf{F}(\mathbf{U})$ and $\mathbf{F}^\nu(\mathbf{U})$ are the flow variable vector, the inviscid, and the viscous fluxes, respectively. The cell-centered finite volume method is obtained by splitting the
95 computational domain Ω into N non-overlapping cells Ω_{ijk} . The integration of Equation (1) over every cell of the mesh leads to a semi-discrete form as:

$$\frac{d}{dt} \mathbf{U}_{ijk} + \frac{1}{|\Omega_{ijk}|} \mathbf{R}_{ijk}(\mathbf{U}) = \mathbf{0}, \quad (2)$$

where \mathbf{U}_{ijk} is now the mean flow variable vector evaluated at the center of Ω_{ijk} , $|\Omega_{ijk}|$ the volume of Ω_{ijk} , and \mathbf{R}_{ijk} the residual of the discretised convective and viscous terms.

The convective fluxes are approximated by a second-order accurate spatial scheme proposed by
100 Mary & Sagaut [21] which relies on a modification of the AUSM+(P) scheme (Advection Upstream Splitting Method where “+(P)” denotes the addition of a modified pressure contribution to the interface mass flux for low-speed flows, see Edwards & Liou [22]). The viscous fluxes are discretised using a second-order accurate centered scheme. Two different time-stepping schemes are implemented in FastS and discussed hereafter: an explicit 3^{rd} -order accurate low-storage Runge-Kutta

105 scheme [23] and an implicit 2^{nd} -order accurate backward scheme of Gear with local Newton sub-iterations [24]. In the latter case, the Jacobians are approximated following the procedure presented in [25, 26] and the linear system is solved by the LU-SGS factorisation [25].

Modified AUSM scheme. Several modifications have been introduced to the standard AUSM+(P) scheme of Edwards & Liou [22] to enhance its accuracy and lower its computational cost in the low
110 Mach number range. By discarding the shock-capturing part, the convective fluxes on a face l of Ω_{ijk} are expressed as:

$$\mathbf{F}_l = U_l \frac{\mathbf{U}_L + \mathbf{U}_R}{2} - |U_{dis}| \frac{\mathbf{U}_L - \mathbf{U}_R}{2} + \mathbf{P}_l, \quad (3)$$

where L/R denotes the left and right third-order MUSCL (Monotone Upstream-centered Schemes for Conservation Laws [27]) interpolated states. The pressure term \mathbf{P}_l is given by $(p_L + p_R)/2 \cdot (0, \mathbf{n} \cdot \mathbf{e}_x, \mathbf{n} \cdot \mathbf{e}_y, \mathbf{n} \cdot \mathbf{e}_z, 0)^t$ where \mathbf{n} is the normal vector to the face l . U_l denotes the interface fluid
115 velocity and is defined as:

$$U_l = \mathbf{n} \cdot \frac{\mathbf{u}_L + \mathbf{u}_R}{2} - c_2(p_R - p_L). \quad (4)$$

U_{dis} , which has the dimension of a velocity, characterises the numerical dissipation acting on the velocity components through:

$$U_{dis} = \max(|U_l|, c_1). \quad (5)$$

Both c_1 and c_2 are constant parameters chosen as small as possible to minimize the numerical dissipation. In [28], an optimal value of 0.04 has been determined. Without a loss in comprehension
120 or generality this upwind scheme will henceforth be referred to as AUSM.

In order to further reduce the numerical dissipation of the AUSM scheme for LES-type computations, a hybrid centered/decentered modification of the AUSM scheme was proposed in [21] and extended by Laurent [29]. It relies on a binary sensor function Φ_l , which only depends on the smoothness of the primitive variables $\psi = (\rho, u_x, u_y, u_z, p)^t$, acting on the dissipative terms in
125 Equation (3):

$$\mathbf{F}_l = U_l \frac{\mathbf{U}_L + \mathbf{U}_R}{2} - |U_{dis}| \times \Phi_l \times \frac{\mathbf{U}_L - \mathbf{U}_R}{2} + \mathbf{P}_l, \quad (6)$$

where the dissipative term appearing in the definition of U_l becomes:

$$U_l = \mathbf{n} \cdot \frac{\mathbf{u}_L + \mathbf{u}_R}{2} - c_2(p_R - p_L) \times \Phi_l. \quad (7)$$

If no spurious oscillations are detected on ψ in the vicinity of cell Ω_{ijk} , then $\Phi_l = 0$ and the convective flux of Equation (6) degenerates to a fully centered approximation. The latter scheme will be denoted as ‘‘Sensor’’ in the following.

130 The FastS solver has been extensively used and validated for both academic and industrial unsteady flow simulations such as transitional separation bubble [30], airfoils in near-stall configurations [21, 31] and laminar transonic buffet [32].

2.2. Lattice Boltzmann method

The lattice Boltzmann method [2, 33] does not directly rely on the resolution of the Navier-Stokes equations. In fact, it originates from a very specific discretisation of the Boltzmann equation 135 describing the evolution of gases in terms of distribution functions $f_i(\mathbf{x}, t)$ which represent the probability density of finding fictitious particles at a location \mathbf{x} and time t being advected at a given discrete velocity $\boldsymbol{\xi}_i$. In the absence of a body-force term, the discrete velocity Boltzmann equation (DVBE) is given by:

$$\frac{\partial}{\partial t} f_i(\mathbf{x}, t) + \boldsymbol{\xi}_i \cdot \frac{\partial}{\partial \mathbf{x}} f_i(\mathbf{x}, t) = \Omega_i(\mathbf{x}, t) \quad i \in \llbracket 1 ; q \rrbracket, \quad (8)$$

140 where q is the number of discrete velocities. The right-hand side term $\Omega_i(\mathbf{x}, t)$ models the time evolution of the distribution functions due to collisions between particles. The latter can be approximated through the BGK collision operator [34] which describes the average collisions effect by a relaxation process towards a local equilibrium f_i^{eq} with a single relaxation time τ :

$$\Omega_i(\mathbf{x}, t) = -\frac{1}{\tau} \left(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t) \right). \quad (9)$$

The macroscopic quantities of interest such as the density ρ and the velocity field \mathbf{u} are computed 145 from the set of discrete distribution functions by taking their first two statistical moments:

$$\rho(\mathbf{x}, t) = \sum_{i=1}^q f_i(\mathbf{x}, t) \quad \text{and} \quad \rho \mathbf{u}(\mathbf{x}, t) = \sum_{i=1}^q \boldsymbol{\xi}_i f_i(\mathbf{x}, t). \quad (10)$$

The number, norm and orientation of the discrete velocities $\{\boldsymbol{\xi}_i\}_{i \in \llbracket 1 ; q \rrbracket}$ must follow particular rules that depend on the macroscopic behavior of interest. It can be shown through a Chapman-Enskog expansion [35] that the underlying physics at a macroscopic level are linked to the statistical moments of the equilibrium distribution functions f_i^{eq} defined as:

$$\mathbf{m}_{eq}^{(n)} = \sum_{i=1}^q \boldsymbol{\xi}_i^n f_i^{eq}, \quad (11)$$

150 where $\boldsymbol{\xi}_i^n$ is the n -rank tensor built by n tensor products of $\boldsymbol{\xi}_i$. Therefore, in order to retrieve the Navier-Stokes dynamics, the set of discrete velocities $\{\boldsymbol{\xi}_i\}_{i \in \llbracket 1 ; q \rrbracket}$ has to ensure the equality between the discrete moments of the discrete equilibria $\{f_i^{eq}\}_{i \in \llbracket 1 ; q \rrbracket}$ defined by Equation (11) and the continuous ones, at least up to a given order N [36]:

$$\sum_{i=1}^q \boldsymbol{\xi}_i^n f_i^{eq} = \int \boldsymbol{\xi}^n f^{eq} \quad \text{for } n \in \llbracket 0 ; N - 1 \rrbracket, \quad (12)$$

where f^{eq} is the continuous Maxwell-Boltzmann distribution:

$$f^{eq}(\mathbf{x}, \boldsymbol{\xi}, t) = \frac{\rho(\mathbf{x}, t)}{(2\pi RT)^{d/2}} \exp\left(-\frac{\|\boldsymbol{\xi} - \mathbf{u}(\mathbf{x}, t)\|^2}{2RT}\right). \quad (13)$$

155 This is usually ensured thanks to a Gauss-Hermite quadrature associated with a Hermite polynomial expansion of the equilibrium distribution function [36]. The nodes of the corresponding Gauss-Hermite quadrature then provide the discrete velocities as well as their associated weights. Naturally, the higher the order of the quadrature (i.e. the more discrete velocities), the more faithfully the physics of the Navier-Stokes equations will be reproduced.

160 The most common sets of discrete velocities $\{\boldsymbol{\xi}_i\}_{i \in \llbracket 1; q \rrbracket}$ are often referred to as a $DdQq$ lattices, where d is the spatial dimension and q the number of discrete velocities. Needless to say, the choice of the velocity set is usually a compromise between efficiency and accuracy. To this end, the most notorious lattices (D3Q19 and D3Q27) have an order of quadrature $Q = 5$. In this case, some discrepancies with respect to the classical Navier-Stokes equations [36] arise. While 165 the mass conservation equation is recovered without any error, a $\mathcal{O}(\text{Ma}^3)$ error appears in the momentum equation. In addition, the lack of discrete velocities leads to an incorrect evaluation of the convective and diffusive parts of the energy equation. Therefore, an athermal assumption is made by setting $T = T_0$ where T_0 is a constant reference temperature and the equation of state reduces to $p = \rho(c_0^{ath})^2$ where $c_0^{ath} = \sqrt{RT_0}$ is the athermal speed of sound. Consequently, the 170 lattice Boltzmann method is said to be restricted to isothermal and weakly compressible flows.

For this study, and as it is usually the case for practical computations, the D3Q19 lattice depicted in Figure 1 is considered in the following. This lattice is often preferred to the D3Q27

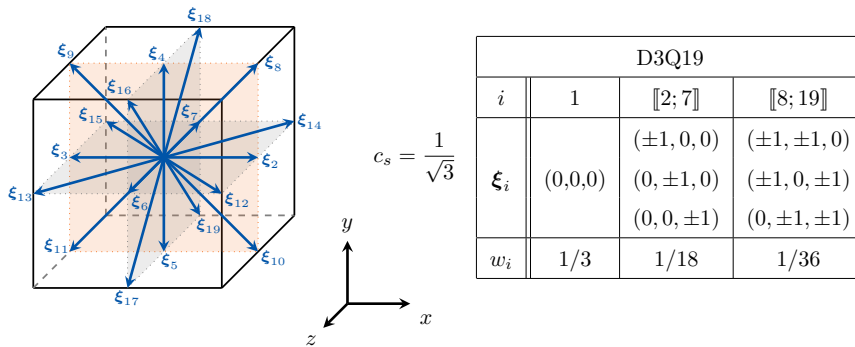


Figure 1: D3Q19 velocity set. The cube, drawn in solid lines, has an edge length of $2\Delta x$. For the sake of clarity, the rest velocity $\boldsymbol{\xi}_1 = \mathbf{0}$ is not represented as it lies at the center of the cube. Each discrete velocity $\boldsymbol{\xi}_i$ is expressed in its non-dimensional form. c_s is the lattice constant and corresponds to the lattice speed of sound.

since it requires about 40% less memory and computing power. Yet, in the light of recent studies [37, 38, 39, 40] it becomes clear that the D3Q27 velocity set has to be preferred when computing 175 high Reynolds number axisymmetric flows. However, for the flows studied in Section 5, the use of a

D3Q19 instead of a D3Q27 lattice is found to have very negligible effects on the numerical solution and thereby on the conclusions of the present study.

The lattice Boltzmann scheme is obtained by discretising the space and time variables of the DVBE (8). The left-hand side (LHS) linear convection term of Equation (8) is integrated along the ξ_i characteristic ensuring an exact advection step and a direct link between the grid and time step through $\Delta x = |\xi_i|\Delta t$. On the other hand, a trapezoidal integration rule (Crank-Nicolson scheme) is employed for the right-hand side (RHS) collision term [41]. This strategy, leads to:

$$g_i(\mathbf{x} + \xi_i\Delta t, t + \Delta t) = g_i(\mathbf{x}, t) + \Delta t\Omega_i(\mathbf{x}, t), \quad (14)$$

where $\{g_i\}_{i \in [1;q]}$ are the modified distribution functions so as to ensure an explicit formulation of the algorithm [42]. They are related to the original distribution functions $f_i(\mathbf{x}, t)$ through the relation $g_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) - \frac{\Delta t}{2}\Omega_i(\mathbf{x}, t)$ which implies $g_i^{eq}(\mathbf{x}, t) = f_i^{eq}(\mathbf{x}, t)$. In the case of the BGK collision model, the relaxation time also becomes $\bar{\tau} = \tau + \frac{\Delta t}{2}$.

The lattice Boltzmann method is classically applied in a non-dimensional form. Therefore, the time-step Δt (resp. the grid-step Δx) is chosen as the characteristic time (resp. characteristic length) for the non-dimensionalization. This leads to the lattice Boltzmann scheme:

$$g_i(\mathbf{x} + \xi_i, t + 1) = g_i(\mathbf{x}, t) + \Omega_i(\mathbf{x}, t), \quad (15)$$

An important point when focusing on acoustics is the physical speed of sound simulated by the LBM denoted hereafter by c_0^{LBM} . Indeed, the athermal sound speed $c_0^{\text{ath}} = \sqrt{RT_0}$ imposed by the D3Q19 lattice does not correspond to the expected one $c_0 = \sqrt{\gamma RT_0}$ where γ is the heat capacity ratio. However, in practice, one can enforce $c_0^{\text{LBM}} = c_0$ by computing the time-step following the acoustic scaling. The physical values of the speed of sound c_0^{LBM} and the viscosity ν are then given by:

$$c_0^{\text{LBM}} = c_0 = \frac{c_s\Delta x}{\Delta t} \quad \text{and} \quad \nu = (c_0^{\text{LBM}})^2 \left(\bar{\tau} - \frac{1}{2} \right), \quad (16)$$

where $c_s = 1/\sqrt{3}$ is the D3Q19 lattice constant. The acoustic scaling is equivalent to setting the LBM fictitious temperature to γT_0 where γ is the heat capacity ratio of the fluid to be simulated.

Based on these parameters, the LBM recovers the athermal and weakly-compressible Navier-Stokes dynamics with a second-order accuracy in both space and time [41].

The lattice Boltzmann method with the BGK collision model is known to suffer from stability issues especially at high Reynolds number [6] preventing it from being used in practical computations. These issues have been attributed to interactions between so called "ghost-modes" arising from the space and time discretisation of Equation (8) [43, 44]. To alleviate the lack of stability,

numerous advanced collision models have been proposed for the LBM in the recent years by increasing the number of free parameters (or relaxation rates) in the model [6, 40, 45]. In this context, the cumulant model introduced by Geier *et al.* [40] is noteworthy owing to its ability to resolve acoustics and turbulent flows [46, 47] with very little additional numerical dissipation w.r.t. the BGK model while remaining stable. For this reason, and for the sake of generality of the proposed study, the BGK model will still be studied in the following despite its low stability since all the new proposed collision models seek to approximate its low numerical dissipation while ensuring stability. Thus, the conclusions of this study obtained for the BGK model represent, by omitting for a while the question of stability, the optimal performance one can expect of the LBM.

In addition to the BGK model, regularised collision kernels [48, 49] are also considered hereafter. This choice is motivated by the fact that their implementation only requires very slight modifications with regard to the simple BGK collision operator. Moreover, regularised kernels drastically reduce the number of possible free parameters in the model which greatly favors their adoption on an industrial scale [9, 50] by simplifying user's intervention. The next two sections are therefore devoted to the introduction of regularised collision models.

2.2.1. Recursive regularised collision model

The regularised collision models are based on the fact that a Chapman-Enskog expansion up to the first order in Knudsen number is sufficient to recover the Navier-Stokes behavior at a macroscopic level. Hence, before each collision step, the distribution functions are reconstructed as:

$$g_i^{reg} = g_i^{eq} + g_i^{(1),reg}, \quad (17)$$

where $g_i^{eq} = f_i^{eq}$ is the equilibrium distribution function and $g_i^{(1),reg}$ the regularised contribution. On the basis of the BGK collision model, the collision step can be rewritten as:

$$\Omega_i = g_i^{eq} + \left(1 - \frac{1}{\tau}\right) g_i^{(1),reg}. \quad (18)$$

The first-order contribution $g_i^{(1),reg}$ is expanded on the basis of Hermite polynomials as it is the case for the equilibrium distribution function. Consequently, one has:

$$g_i^{(1),reg} = w_i \sum_{n=2}^{N_r} \frac{1}{c_s^{2n} n!} \mathbf{a}_1^{(n)} : \mathcal{H}_i^{(n)}, \quad (19)$$

where $\mathbf{a}_1^{(n)}$ is the n th-order off-equilibrium Hermite coefficient. For $n = 2$, the coefficient is computed by projecting the off-equilibrium distribution functions on the basis second-order tensor $\mathcal{H}_i^{(2)}$:

$$\mathbf{a}_1^{(2)} = \sum_{i=1}^q \mathcal{H}_i^{(2)} (g_i - g_i^{eq}). \quad (20)$$

Malaspinas [48] proposed a technique to reconstruct as many off-equilibrium Hermite coefficients $\mathbf{a}_1^{(n)}$ as possible, or at least up to a given order N_r . Thanks to a Chapman-Enskog expansion and some algebra, a recursive relation between the off-equilibrium coefficients is determined:

$$a_{1,\alpha_1\dots\alpha_n}^{(n)} = u_{\alpha_n} a_{1,\alpha_1\dots\alpha_{n-1}}^{(n-1)} + \sum_{i=1}^{n-1} u_{\alpha_1\dots\alpha_{n-2}} a_{1,\alpha_i\alpha_n}^{(2)} \quad \text{for } n \geq 3. \quad (21)$$

The recursive relation is initialised by Equation (20). In the following, this model will be referred to as the recursive regularised collision model at order N_r (RR N_r) where $N_r = 3$.

2.2.2. Hybrid recursive regularised collision model

The Hybrid Recursive Regularised (HRR) collision operator [49] has been proposed to further enhance the stability of the RR model. The very essence of the HRR collision operator lies in the way the second-order off-equilibrium coefficient $\mathbf{a}_1^{(2)}$ is computed [49]. In the HRR framework, this tensor is hybridised. It is decomposed into a linear combination of a projected part (see Equation (20)) and a finite difference part (FD). Indeed, the Chapman-Enskog expansion links the first-order off-equilibrium distribution to the deviatoric stress tensor $\mathbf{S} = \frac{1}{2} (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$. Thereby, the off-equilibrium coefficient $\mathbf{a}_1^{(2)}$ is given by:

$$\mathbf{a}_1^{(2)} = \sigma \left[\sum_{i=1}^q \mathcal{H}_i^{(2)} \left(g_i - g_i^{eq} + \frac{\psi_i}{2} \right) \right] + (1 - \sigma) \left[-\rho \bar{c}_s^2 (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \right], \quad (22)$$

where $0 \leq \sigma \leq 1$ is a user-tuned parameter to control the amount of hyper-viscosity added to the model [49]. ψ_i is a corrective term required to recover a Galilean invariant shear stress tensor and also to stabilise the HRR collision operator [51, 52]. The spatial derivatives of the velocity field in Equation (22) are evaluated with second-order centered finite differences. The higher-order off-equilibrium coefficients are then computed recursively using Malaspinas' formula (21).

3. Linear Stability Analyses (LSA)

In this section, it is proposed to extend the spectral analysis of the lattice Boltzmann and Navier-Stokes schemes of Marié *et al.* [3] by considering regularised lattice Boltzmann models and by investigating the isotropy properties of both the Navier-Stokes and lattice Boltzmann schemes. This constitutes a first step towards a fair comparison between the LB and NS methods. For the sake of simplicity, the von Neumann analysis is performed in only two dimensions of space.

3.1. General methodology and exact plane wave solutions

Starting from a set of non-linear differential equations written in the form of a dynamical system:

$$\frac{\partial \mathbf{q}}{\partial t} = \mathcal{L}(\mathbf{q}), \quad (23)$$

255 where \mathbf{q} is the state vector and \mathcal{L} is the non-linear differential operator, the state vector is perturbed around a base state $\bar{\mathbf{q}}$ as:

$$\mathbf{q} = \bar{\mathbf{q}} + \mathbf{q}'. \quad (24)$$

By construction, the base state is steady and homogeneous in all directions of space (i.e. $\partial_t \bar{\mathbf{q}} = \partial_{x_i} \bar{\mathbf{q}} = 0$). In Equation (24), \mathbf{q}' denotes a small perturbation of \mathbf{q} such that $\mathbf{q}' \ll \bar{\mathbf{q}}$. By substituting Equation (24) into Equation (23) and keeping only first-order terms, one gets:

$$\frac{\partial \mathbf{q}'}{\partial t} = \mathcal{J} \mathbf{q}', \quad (25)$$

260 where \mathcal{J} is the jacobian matrix of the non-linear differential operator \mathcal{L} evaluated at $\bar{\mathbf{q}}$.

A Fourier transform in space and time of the state vector is then performed, allowing the perturbations to be studied as plane monochromatic waves:

$$\mathbf{q}' = \hat{\mathbf{q}} \exp(i(\mathbf{k} \cdot \mathbf{x} - \omega t)), \quad (26)$$

where $i^2 = -1$, $\hat{\mathbf{q}}$ is the complex amplitude of the perturbations, $\mathbf{k} \in \mathbb{R}^n$ with $n \in \llbracket 1; 3 \rrbracket$ is the wave number, and $\omega \in \mathbb{C}$ is the pulsation of the wave. Finally, injecting Equation (26) into Equation 265 (25), the spatial and temporal derivatives can be simplified as $\partial_t = -i\omega$ and $\partial_{x_i} = ik_i$. This leads to the general eigenvalue problem:

$$\omega \hat{\mathbf{q}} = \mathcal{J} \hat{\mathbf{q}}. \quad (27)$$

In the case of a discrete numerical scheme, where the temporal derivative is often approximated through a difference scheme, Equation (27) then becomes:

$$e^{-i\omega} \hat{\mathbf{q}} = \tilde{\mathcal{J}} \hat{\mathbf{q}}. \quad (28)$$

The eigenvalue problems of Equations (27) and (28) are then solved for each value of \mathbf{k} . The 270 complex eigenvalues ω give access to the propagation speed $\text{Re}(\omega)$ and dissipation rate $\text{Im}(\omega)$ of each of the linear modes stemming from the set of equations defined by \mathcal{L} .

Exact plane wave solutions. In order to assess the dispersive and dissipative properties of the lattice Boltzmann and Navier-Stokes schemes, their spectral properties have to be compared with a reference. To this end, the 2D linearised Navier-Stokes equations are solved for plane wave solutions. 275 By applying the general LSA methodology to these equations (see [3] for further details), one gets the following eigenvalue problem:

$$\omega \hat{\mathbf{U}}' = \mathbf{M}^{\text{NS},\dagger} \hat{\mathbf{U}}', \quad (29)$$

where $\hat{\mathbf{U}}' = (\rho', \bar{\rho} u'_x, \bar{\rho} u'_y, p')^t$ and $\mathbf{M}^{\text{NS},\dagger}$ is the time-advance matrix. The solution of Equation (29) gives the linear modes of the isothermal Navier-Stokes equations: two acoustic modes (one

upstream ω_{Ac+} and one downstream ω_{Ac-}), one shear mode ω_{sh} in 2D (two in 3D), and one entropy
 280 mode ω_{entr} . Their general expressions are given by:

$$\begin{aligned}\omega_{Ac\pm} &= \mathbf{k} \cdot \bar{\mathbf{u}} \pm \|\mathbf{k}\|c_0 + i \left(\frac{2}{3}\nu + \frac{1}{2}\zeta \right) \|\mathbf{k}\|^2, \\ \omega_{sh} &= \mathbf{k} \cdot \bar{\mathbf{u}} + i\nu\|\mathbf{k}\|^2, \\ \omega_{entr} &= \mathbf{k} \cdot \bar{\mathbf{u}}.\end{aligned}\tag{30}$$

It can be seen from Equation (30) that the shear wave propagates at the mean flow velocity $\bar{\mathbf{u}}$,
 whereas the acoustic waves propagate at $\bar{\mathbf{u}} \pm c_0$. Moving to the dissipation rate, the attenua-
 tion of the shear wave is directly controlled by the viscosity ν while for the acoustic waves, the
 attenuation process is divided into two parts: dissipation induced through the viscosity ν , and
 285 compression/dilation effects through the bulk viscosity ζ . Due to the isothermal hypothesis of the
 LBM, the entropy mode is reduced to a non-dissipative wave propagating with the shear mode and
 is therefore currently ignored.

3.2. LSA of space and time discrete Navier-Stokes schemes

The combined effect of the space and time discretisations is studied here. This is necessary for
 290 the comparison with lattice Boltzmann schemes in which the space and time discretizations cannot
 be distinguished. The methodology, introduced by Marié *et al.* [3], is briefly recalled here.

Space discretisation. For the sake of simplicity, only centered schemes are studied in the present
 work. By definition, a $2N$ -order centered approximation of the spatial derivatives in direction i for
 a given quantity \mathbf{q} can be written as:

$$\frac{\partial \mathbf{q}}{\partial x_i} = \frac{1}{\Delta x_i} \sum_{j=-N}^N a_j \mathbf{q}(\mathbf{x} + j\Delta \mathbf{x}_i) + \mathcal{O}(\Delta x^{2N}),\tag{31}$$

295 where $a_j = -a_{-j}$ are the coefficients of the scheme and Δx_i is the mesh size in the i direction. In
 the following, three schemes are considered: a classic second-order one (denoted by ‘‘CenterO2’’),
 a fourth-order ‘‘dispersion relation preserving’’ scheme developed by Tam and Webb [53] (denoted
 by ‘‘DRPTWO4’’), and the optimized sixth-order Bogey scheme [54] (denoted by ‘‘FDo13’’). By
 applying the general von Neumann stability analysis, Equation (31) becomes:

$$\frac{\partial \mathbf{q}'}{\partial x_i} = \underbrace{\left[\frac{1}{\Delta x_i} \sum_{j=-N}^N a_j \exp(i j \mathbf{k} \cdot \Delta \mathbf{x}_i) \right]}_{\mathcal{K}_{\text{scheme}, x_i}} \hat{\mathbf{q}} + \mathcal{O}(\Delta x^{2N}),\tag{32}$$

300 where the so-called equivalent wave-number $\mathcal{K}_{\text{scheme}, x_i}$ has been introduced. Even though the
 ‘‘Sensor’’ scheme implemented in ONERA’s finite-volume Navier-Stokes solver isn’t strictly speaking

a second-order centered scheme due to the MUSCL interpolations, it behaves like a centered scheme in the absence of oscillations in the primitive variables. Thereby, centered schemes provide a lower-band estimate of the dissipative behavior of this finite-volume scheme.

305 *Time discretisation.* As in [3], this study is restricted to explicit Runge-Kutta time-stepping schemes. This is motivated by the fact that most computations in Section 5 use an explicit third order Runge-Kutta scheme. By definition, a p -step explicit Runge-Kutta (RK) method applied to Equation (23) can be expressed as:

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \sum_{j=1}^p \gamma_j \Delta t \mathcal{L}^j(\mathbf{q}^n), \quad (33)$$

310 where \mathbf{q}^{n+1} and \mathbf{q}^n are the values of \mathbf{q} at time $(n+1)\Delta t$ and $n\Delta t$ respectively. Δt is the time-step and \mathcal{L}^j denotes the j -th composition of the function \mathcal{L} . The coefficients γ_j are related to the specific Runge-Kutta scheme being used. In this paper, two Runge-Kutta schemes are studied: a third-order one denoted by RK3 and a sixth-order optimised one (RK6) proposed by Bogey and Bailly [54].

When applying the general LSA methodology to the space- and time-discrete Navier-Stokes schemes, one gets the following general eigenvalue problem:

$$e^{-i\omega} \hat{\mathbf{q}} = \left[\mathbf{I} + \sum_{j=1}^p \gamma_j \text{CFL}^j \mathbf{\Lambda}^j \right] \hat{\mathbf{q}} = \mathbf{M}^{\text{NS}} \hat{\mathbf{q}}, \quad (34)$$

315 where \mathbf{I} is the identity matrix and $\mathbf{\Lambda}$ is given by:

$$\mathbf{\Lambda} = -\frac{\Delta x}{c_0} [\mathcal{K}_{\text{scheme},x_1} \mathbf{M}_{x_1} + \mathcal{K}_{\text{scheme},x_2} \mathbf{M}_{x_2}]. \quad (35)$$

Note that $\mathcal{K}_{\text{scheme},x_1}$ and $\mathcal{K}_{\text{scheme},x_2}$ are also present in the definition of \mathbf{M}_{x_1} and \mathbf{M}_{x_2} since they involve derivatives in their generic expressions (see [3]).

3.3. LSA of lattice Boltzmann schemes

320 The main variables of the lattice Boltzmann method are the distribution functions. Therefore, the state vector \mathbf{q} is now defined as:

$$\mathbf{q}_f = (f_1, f_2, \dots, f_{q-1}, f_q), \quad (36)$$

where q is the number of discrete velocities. In order to distinguish the state vector of Equation (36) with the one used in the NS formalism, the former is written with the letter f as a subscript. This difference being noted, the general methodology introduced in Section 3.1 is applied in the same way. After some algebra (see [55] for the details), the following eigenvalue problem is obtained:

$$e^{-i\omega} \hat{\mathbf{q}}_f = \mathbf{M}^{\text{LB}} \hat{\mathbf{q}}_f, \quad (37)$$

325 where the time evolution matrix \mathbf{M}^{LB} depends on the collision operator. For the BGK, RR N_r , and HRR q collision models, its general expression is derived in [55, 56].

Equation (37) highlights one main difference between the LSA of NS and LB schemes. Indeed, for the Navier-Stokes schemes, the eigenvalue problem leads to 4 eigenmodes in 2D. However, in the lattice Boltzmann framework, the eigenvalue problem is of size q meaning that q eigenmodes are
 330 obtained. Owing to the fact that $q > 4$, the dynamics of the DVBE include more information than at the Navier-Stokes level.

Extended von Neumann analysis of LB schemes. In order to give a physical interpretation to the q modes resulting from the von Neumann analysis in the lattice Boltzmann framework, an extended von Neumann analysis methodology [44] is used. The latter relies on the study of the LB eigenvector
 335 $\hat{\mathbf{q}}_f = (\hat{f}_1, \hat{f}_2, \dots, \hat{f}_{q-1}, \hat{f}_q)$ and more particularly on its moments defined as:

$$\hat{\rho} = \sum_{I=1}^q \hat{f}_I \quad \text{and} \quad \hat{\rho}u = \sum_{I=1}^q \xi_I \hat{f}_I. \quad (38)$$

The resulting macroscopic vector $\mathbf{V} = [\hat{\rho}, \hat{\rho}u]^T$ is then projected onto the Navier-Stokes one. This analysis allows for a systematic identification of the modes carrying a macroscopic information at more than a prescribed ratio η . In the results presented below, this parameter will be set to $\eta = 0.99$. Such extended analysis has been extensively used in the recent years for a very large set
 340 of problems [51, 55, 57, 56].

In the following, three different collision models are studied, namely the BGK, RR3 and HRR for which the value of σ is set to $\sigma = 0.995$ which is commonly used for industrial applications [57].

3.4. Comparison between the Navier-Stokes and lattice Boltzmann schemes

In order to compare the dispersive and dissipative properties of both the lattice Boltzmann and
 345 Navier-Stokes schemes, a comparison metric has to be introduced. For this purpose, it is chosen to focus on the error committed on $\text{Re}(\omega)$ and $\text{Im}(\omega)$ as a function of the wavenumber [3]:

$$\begin{cases} \text{Err}^{\text{Re}}(\mathbf{k}) &= |\text{Re}(\omega^{th}) - \text{Re}(\omega)| \\ \text{Err}^{\text{Im}}(\mathbf{k}) &= |\text{Im}(\omega^{th}) - \text{Im}(\omega)| \end{cases}, \quad (39)$$

where ω^{th} refers to the solutions of the exact linearised Navier-Stokes equations (30), and ω refers to the solutions of the eigenvalue problems of Equations (34) and (37). The latter are solved numerically thanks to an in-house Python code, using the NumPy and SimPy libraries [58, 59].
 350 Since only the real part of the perturbations are of interest, and by virtue of Shannon's theorem it is sufficient to restrict the problem to $k_x \in [-\pi, \pi]$ and $k_y \in [0, \pi]$ in 2D. The wavenumber space is discretised with a uniform step $\Delta k = 0.01$ determined by a convergence study.

The values of $\text{Err}^{\text{Re}}(\mathbf{k})$ and $\text{Err}^{\text{Im}}(\mathbf{k})$ are computed for the same CFL number. Since the lattice Boltzmann method operates at a fixed CFL number of $1/\sqrt{3}$, the same value is chosen for the
 355 Navier–Stokes schemes. Other CFL numbers have been tested without changing the conclusions. The viscosity is set to $\nu = 10^{-5} \text{ m}^2/\text{s}$ which is representative of aeronautical applications.

Figure 2 displays the dispersion and dissipation errors of the lattice Boltzmann and Navier-Stokes schemes for plane monochromatic waves such as $k_y = 0$ superimposed to a horizontal mean flow at $\overline{\text{Ma}} = 0.2$. The dispersion curves indicate that switching from the BGK to regularised
 360 collision models has very little influence on the dispersive properties of the lattice Boltzmann method. Compared to centered NS schemes, the LBM dispersion error is between a second-order and an optimized third-order space scheme with a 3-step Runge-Kutta time-stepping. This result is quite interesting since the LBM is a second-order accurate method which has better spectral properties than centered second-order NS schemes. When it comes to the dissipation error, it
 365 can be seen that NS schemes have a higher dissipation on the acoustic modes than on the shear mode. Such observation is in accordance with previous results [3]. However, regardless of the collision model, the trend is reversed for the LBM. Indeed, the dissipation error on the shear mode is slightly higher than the one of the acoustic modes. Regarding the dissipation error curves, the dissipation error of the LBM highly depends on the collision model which is employed as
 370 well as the on the physical mode of interest. For instance, the LB regularised collision models are highly dissipative on the shear mode for which all the Navier-Stokes schemes tend to be less dissipative. The only LB model which competes with the NS schemes is the LBM-BGK for which the dissipation error is between an optimised third order and sixth-order Navier-Stokes centered scheme. This behavior of the regularised collision kernels was recently discussed in [51] and linked
 375 to a “hyperviscous degeneracy” phenomena. However, on acoustic modes, it can be seen that the BGK and RR collision models have the exact same low dissipation error when perturbations are aligned with the x -axis. In comparison, one has to consider a sixth-order NS scheme in order to get a lower dissipation error for low wavenumbers. Despite its higher dissipation, the HRR model is still better than traditional NS schemes lying in between an optimised third-order and sixth-order
 380 Navier-Stokes centered scheme. All in all, regularised collision kernels preserve the low dissipative capabilities of the lattice Boltzmann models on acoustic modes but exhibit an increased dissipation on shear modes with respect to the classical – but unstable – BGK collision model.

The tendencies outlined in the analysis of Figure 2 are now completed by exploring the whole spectral space (i.e. for $k_x \in [-\pi, \pi]$ and $k_y \in [0, \pi]$). Indeed, shear and acoustic fluctuations
 385 are very seldom aligned with the main axes of the mesh and numerical schemes are prone to anisotropic numerical properties. Therefore, planar plots of the dissipation errors are provided

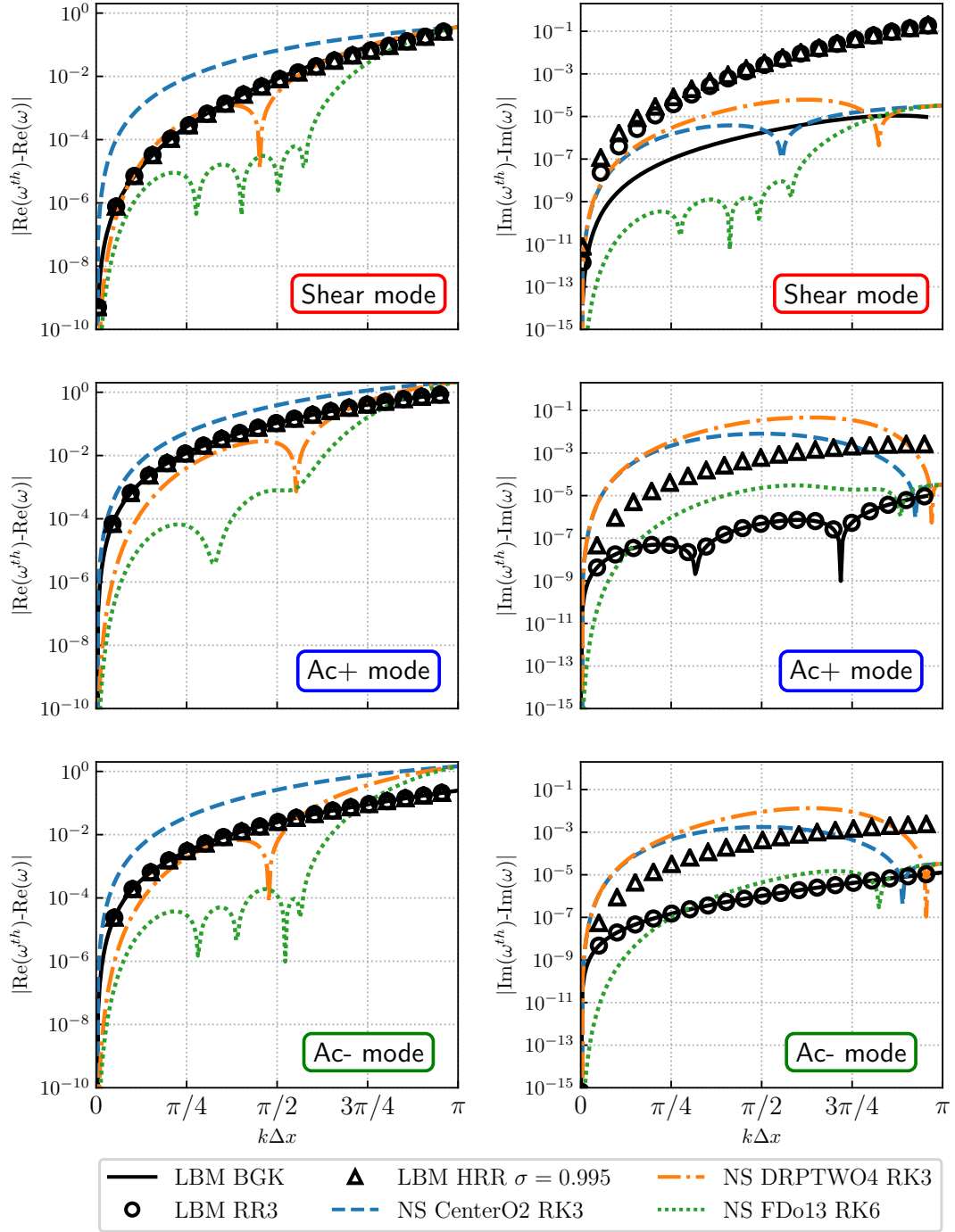


Figure 2: Dispersion (left), and dissipation (right) of the lattice Boltzmann and Navier–Stokes schemes for $\overline{\text{Ma}} = 0.2$ and $\nu = 10^{-5}$.

in Figure 3 for a horizontal mean flow at $\overline{\text{Ma}} = 0.2$. Note that similar tendencies are obtained by varying the angle of the mean flow. First of all, a rather anisotropic dissipative behavior is observed for all the numerical schemes on a broad range of wavenumbers. One can however notice
390 that increasing the order of the spatial NS scheme tends to increase the isotropic region in the limit of $k_x, k_y \rightarrow 0$. Inasmuch as centered schemes do not induce any numerical dissipation, the anisotropy observed in the dissipation error can be attributed to the coupling between the spatial and temporal discretisations. Regarding the lattice Boltzmann schemes, some grey regions are exhibited indicating that no physical information was identified by the extended von Neumann
395 analysis (the $\eta \geq 0.99$ condition is no more satisfied).

From a general point of view, the main conclusion which can be drawn from Figure 3 is that the spectral properties observed when considering a planar monochromatic wave with $k_y = 0$ hold over the whole wavenumber plane. The lattice Boltzmann method with BGK collision operator remains the least dissipative method. However, one can see that, all the regularised LB models have a much
400 higher dissipation when the perturbation is no longer fully aligned with the main mesh directions especially on acoustics. All in all, when considering shear-driven flows, the Navier-Stokes schemes seem to be better candidates to propagate such information over long distances and only the LBM-BGK presents an advantage in comparison with regularised LB schemes. However, for acoustic problems, even though regularised LBMs have a higher dissipation error w.r.t. the classical LBM-
405 BGK, they present a lower dissipation error than second and third-order Navier-Stokes schemes over the whole range of practical interest i.e. $\|\mathbf{k}\| \leq \pi/2$.

All in all, in this section, the study of Marié *et al.* [3] has been extended by taking into account advanced LB collision models of practical relevance and several directions of propagation for the perturbations. In addition, while in the NS community schemes are generally studied for one-
410 dimensional problems, the present study provides some insight into their anisotropic dissipative behavior when both the space and time discretisations have been simultaneously performed.

4. CPU Performance

It is not enough for a numerical method to have good dispersive and dissipative properties; it must also take advantage of modern computing resources in the context of massively parallel
415 systems. Therefore, this section focuses on the intrinsic CPU performance of the lattice Boltzmann and Navier-Stokes methods. The aim of this section is to optimise each code to the limits of the considered CPU architecture with the help of the Roofline performance model so as to avoid implementation bias. The algorithmic differences between the LB and NS methods are also discussed.

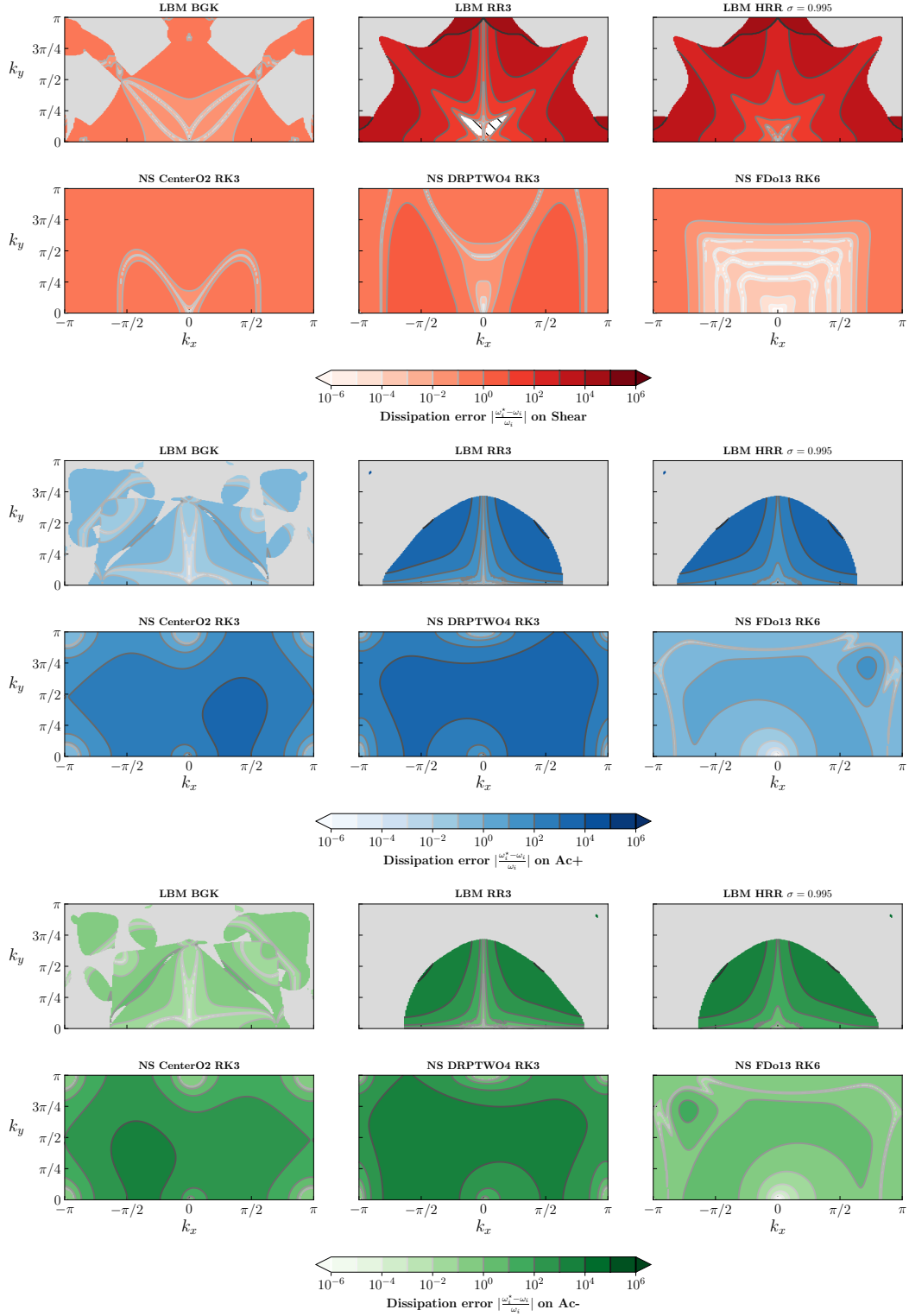


Figure 3: Spectral maps of the dissipation error of the lattice Boltzmann and Navier–Stokes schemes for $\overline{\text{Ma}} = 0.2$ and $\nu = 10^{-5}$. Top: shear mode; Middle: Acoustic+ mode; and Bottom: Acoustic- mode.

4.1. Implementation details

420 Before diving into the evaluation of the performance of the LB and NS methods, some implementation details are provided to clarify how the optimisation is achieved within ONERA's Cassiopee/Fast CFD environment. At a node-level, an OpenMP parallelisation strategy is implemented (see Alferez *et al.* [60]). Each thread computes its own local subset of indexes and the workload is distributed among the cores of the node (one thread per core). Note that only the HPC
425 layer dedicated to shared memory nodes is discussed here. Indeed, it is relatively easy to obtain a good scalability at a cluster level with the help of MPI asynchronous communications.

Data layout. Both the Navier-Stokes and lattice Boltzmann equations are solved using a domain decomposition technique in several structured blocks where ghost cells are used at the interfaces between them. The simulation domain is therefore represented by a multi-dimensional array featuring
430 a flag field for distinguishing between the computed cells centers and the not computed ones (e.g. ghost cells or obstacles). Consequently, all variables can be accessed by simple index arithmetic. The choice has also been made to store the variables in a Structure of Array mode for vectorisation purposes and to guarantee contiguous memory access in the NS and LBM schemes updates [60, 61].

By virtue of its dimensionless formulation, the LBM does not require any grid metrics to be
435 stored. However, when using finite-volume Navier-Stokes method, informations about each cell face normal and surface are needed for each topological direction of the mesh. Therefore, in an effort to maximise its HPC efficiency, 3 different versions of the NS method are implemented within FastS in order to take advantage of specific grid topologies: `3dcart` for cartesian grids where all the metrics are reduced to scalar values; `3dhomo` for curvilinear meshes in the (x, y) plane and cartesian in the
440 z direction and `3dfull` for fully curvilinear grids. The importance of distinguishing between these different implementations when comparing the performance of the LBM with the Navier-Stokes method will be discussed in the following.

Factorisation of the compute kernels. For the structured Navier-Stokes method, the number of functions has been reduced to 5 calls of subroutines [60] where the computation of fluxes balance is
445 responsible for approximately 80% of the overall computational time in the explicit case. Regarding the lattice Boltzmann method, a straightforward implementation would lead to separate functions, namely the streaming and the collision steps. However, the number of data transfers can be reduced by executing the collision and propagation step in the same loop [61]. Moreover, since three nested loops over the three spatial dimensions are involved, an additional level of optimisation can be
450 introduced by splitting the innermost loop into smaller ones and by storing common subexpression into buffer arrays allocated for each thread [62].

Vectorisation. Modern processors have few Single Instruction Multiple Data (SIMD) units per core that can perform operations by group of 4 or 8 for the price of one in the innermost loop. In order to ensure coalesced memory accesses, the threads are aligned along the x-axis while blocks
 455 of threads are aligned in the (y, z) plane. Consequently, the internal loop is instrumented with a SIMD directive to help the compiler to generate an efficient assembly code.

Cache Blocking. The memory traffic can be further reduced by using a cache-blocking technique. The main idea behind cache-blocking is to rearrange data access to pull subsets of data into cache and to operate on this block to avoid having to repeatedly fetch data from main memory. This
 460 optimisation technique is particularly useful for the Navier-Stokes schemes but does not play an important role when considering lattice Boltzmann models since no data reuse is present in the classic “Stream & Collide algorithm”.

4.2. Roofline Performance model

The Roofline model [63] is introduced in an effort to derive an upper performance limit of each
 465 method on the present hardware system. Throughout this paper, the performances are measured on ONERA’s supercomputer featuring Intel Xeon E5-2680v4 Broadwell dual-socket CPU nodes (denoted by BDW in the following) with 14 physical cores per socket operating at a base frequency of 2.4GHz. Table 1 provides the main specifications of the CPU system used in the present work.

Name		BDW		Name		BDW	
Processor		Intel Xeon		Processor		Intel Xeon	
		E5-2680v4				E5-2680v4	
micro-arch.		Broadwell		Memory bandwidth			
frequency	[GHz]	2.4		B_m copy	[GB/s]	117.0	
cores per sock.		14		B_m copy-19	[GB/s]	87.2	
sockets		2		Peak FLOPS			
L1 cache	[KB]	32		P_{peak} scalar	[GFLOPS]	57.6	
L2 cache	[KB]	32		P_{peak} vect.	[GFLOPS]	230.8	
L3 cache	[MB]	18					
ISA		AVX2		(b) Memory bandwidth and peak floating point performance for the BDW system.			

(a) Mains features of the BDW system.

Table 1: Specifications of the BDW system.

The Roofline model states that the performance of a given algorithm is either bounded by the
 470 floating point performance (expressed in floating point operations per second or FLOPS) or bounded

by the memory bandwidth (in bytes/s). Thereby, the achievable performance P is expressed by:

$$P = \min(P_{\text{peak}}, B_m \times \text{AI}), \quad (40)$$

where P_{peak} denotes the processor’s peak floating point performance, B_m is the memory bandwidth, and AI is the arithmetic intensity which depends on the studied algorithm. The latter corresponds to the number of floating point operation per bytes that must be transferred between the processor and memory in order to execute on step of the algorithm. The values of P_{peak} and B_m for the considered CPU system are provided in Table 1. The effective memory bandwidth is measured using the STREAM copy benchmark [64] as well as with an adjusted benchmark mimicking the LB streaming step on a D3Q19 lattice: `copy-19` which concurrently copies 19 arrays [65].

In its classical formulation, the Roofline model is expressed in FLOPS. However, for the LB and finite-volume Navier-Stokes methods it is better to think about performance in terms of Cell Updates Per Second (CUPS). In order to perform this change of units, one has to determine γ_{FP} which corresponds to the number of floating point operations required per cell update and $B_\ell = 1/\text{AI}$ which expresses the number of bytes which are transferred from the main memory to the CPU for one cell update. Table 2 provides the corresponding key figures for the lattice Boltzmann and the finite-volume Navier-Stokes methods.

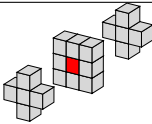
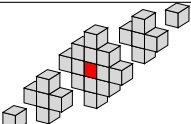
	lattice Boltzmann method			FV Navier-Stokes method		
	BGK	RR	HRR	3dcart	3dhomo	3dfull
<i>Variables</i>	2 sets of 19 distribution functions (f_1, f_2, \dots, f_{19})			3 sets of primitive variables (ρ, u_x, u_y, u_z, p)		
<i>Storage</i>	Double precision (8 bytes)			Double precision (8 bytes)		
<i>Stencil</i>						
γ_{FP}	204	325	490	1012	1295	1774
B_ℓ	456	456	520	1152	1430	2149

Table 2: Comparison between the lattice Boltzmann and Navier-Stokes methods implemented in ONERA’s Fast CFD environment in terms of memory and computational footprint. Note that only the case of an explicit time-stepping scheme for the NS solver is shown.

The γ_{FP} metric shows that, regardless of the collision models, the LBM performs less floating point operations per cell and time-step than the finite-volume NS method. Naturally, when implementing more robust collision model (such as the regularised ones), more operations have to be performed in comparison to the classic BGK model. However, on average, the LBM performs

490 between two to five times less operations than the NS method on Cartesian grids. Regarding the B_ℓ metric, while all the LB models have approximately the same memory traffic (the HRR collision model requires the storage of additional gradients), the topology of the grid used by the NS method strongly conditions the memory traffic.

Figure 4 displays the Roofline model’s curves on a full BDW node for the LBM-BGK, LBM-RR3
 495 and LBM-HRR as well as for the Navier-Stokes method on Cartesian grids. In these curves, the floating point performance bound is represented by the dashed and solid horizontal lines for scalar and vectorised (AVX2) double precision operations respectively. Conversely, the linear solid and dotted lines corresponds to the memory bandwidth performance bound. The solid vertical lines indicate the arithmetic intensity of each solver and thereby allow to determine their theoretical
 500 maximum achievable performance by reading the y-intercept of its point of intersection with the black curve. The maximum achievable performances of the methods are also reported in Table 3.

Solver	LBM BGK	LBM RR	LBM HRR	FV-NS <code>3dcart</code>
Achievable Performance [MCUPS]	285	283	234	101
Measured Performance [MCUPS]	277	275	228	98
Relative Performance	2.8	2.8	2.3	1.

Table 3: Maximum achievable performance and measured performance for each numerical method according to the results of the Roofline model (see Figure 4). The relative performance w.r.t. the NS method on cartesian grids is also provided. MCUPS corresponds to Million Cell Updates Per Second.

First, from the theoretical results of Figure 4, it can be seen that both the Navier-Stokes and lattice Boltzmann algorithms are memory bound owing to the use of vectorisation techniques. Regardless of the collision model, the LBM is expected to outperform the FV-NS method when
 505 only considering the CPU performance. It can be seen that the FV-NS method on Cartesian grids is expected to be about 2.3 to 3 times slower than the LBM. Note that this is only valid for the `3dcart` version. Indeed, when considering the FV-NS method on a fully curvilinear grid (`3dfull`), the LBM is approximately 10 times faster which is the value commonly reported in the literature.

Before drawing more general conclusions, it is crucial to ensure that the implementation of
 510 the LB and NS methods employed in this study does not introduce any additional bias, i.e. that the methods are well optimised at the processor limits. To this end, the Roofline models are confronted to the present implementations in ONERA’s FAST CFD environment by profiling the codes with the help of Intel’s Advisor software. To this end, colored vertical dashed lines indicate the arithmetic intensity and thus the global performance measured by the software (also reported in
 515 Table 3). As one can see, by comparing the theoretical and measured arithmetic intensities, which

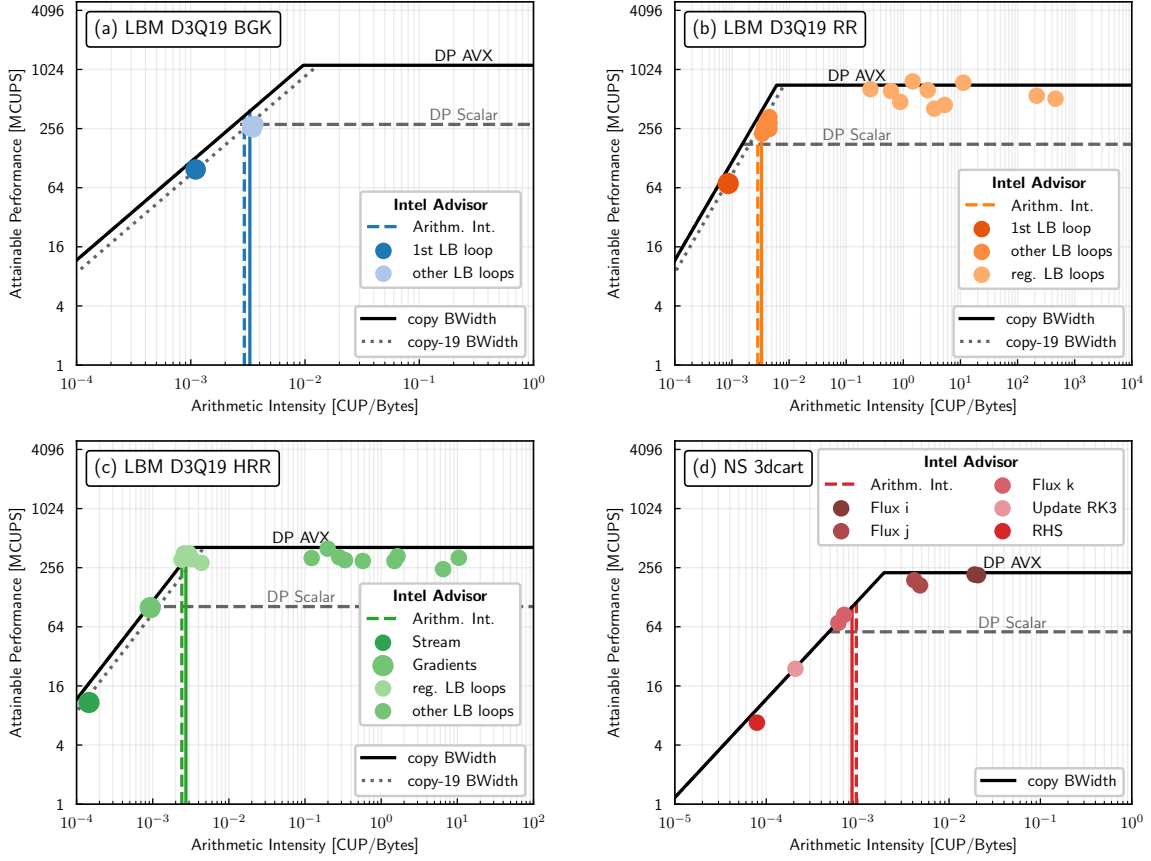


Figure 4: Roofline models of each of the numerical methods studied in this section for the BDW system expressed with the “cell update” metric. The vertical line on each plots indicates the corresponding arithmetic intensity and the maximum reachable performance is obtained by taking the intersection of this line with the Roofline model.

are almost superimposed, it can be confirmed that the methods are optimised to the hardware’s limit capabilities. Consequently, the LBM is shown to be about 2.3 to 2.8 times faster than a finite-volume Navier-Stokes method applied on Cartesian grids. In addition, the performance of each loop in the algorithms is provided in Figure 4 and depicted by circles. As one can observe in Figure 4, all the implemented loops are located on the rooflines meaning that the LB and NS methods are optimised up to the hardware limit. This further confirms that any bias from improper implementation preventing the methods from being fairly compared with each other is removed. The results of Figure 4 also confirm the fact that the LBM-BGK is completely memory bound since all the loops are on the copy-19 bandwidth limit. When it comes to the other LB models, one can see that once the streaming step is performed and all the buffer arrays initialised, the remaining computations are bound by the peak FLOP performance roofline. Figure 4 (d) also highlights the effect of cache-blocking on the Navier-Stokes Cartesian solver. Indeed, the fluxes are computed in the following order: first in the k -direction (or z coordinate), then in the j -direction (or y respectively) and in the i -direction (corresponding to the x coordinate). The values being loaded

530 into the cache for the fluxes computations in the k directions (the loop hits the bandwidth roofline) are reused by the other fluxes which, in turn, are bound by the peak floating point operations.

4.3. Parallel scaling

As already stated, only the performances on a single shared memory node are discussed here since it is relatively easy to obtain a good scalability on distributed memory architectures with the help of MPI asynchronous communications. Moreover, when it comes to non-uniform grids and explicit multi-rate time-stepping [66, 67], the issues related to load balancing between the nodes is essentially the same for the LB and NS methods.

Figure 5 shows the performance of the LB and NS methods on one single BDW node under strong scaling. To obtain this curve, computations of the three dimensional Taylor Green vortex on a grid consisting of 200^3 cells were performed. Such simulations are representative of the workload one would encounter at a node level for distributed memory clusters on large scale problems. Note that this test case will be further discussed in Section 5.3. The performance is then evaluated in MCUPS (Million Cell Updates Per Second) and is defined by:

$$P_{\text{MCUPS}} = \frac{n_{\text{substep}} \times 200^3}{t_{\text{CPU}} \times 10^6}, \quad (41)$$

where t_{CPU} is the CPU time required by the solver to perform one iteration expressed in μs , and n_{substep} corresponds to the number of sub-iteration per full time step ($n_{\text{substep}} = 3$ for the explicit RK3 time-stepping scheme while $n_{\text{substep}} = 1$ for all the LB models). In addition, in Figure 5, horizontal lines corresponding to the Roofline model predictions are plotted.

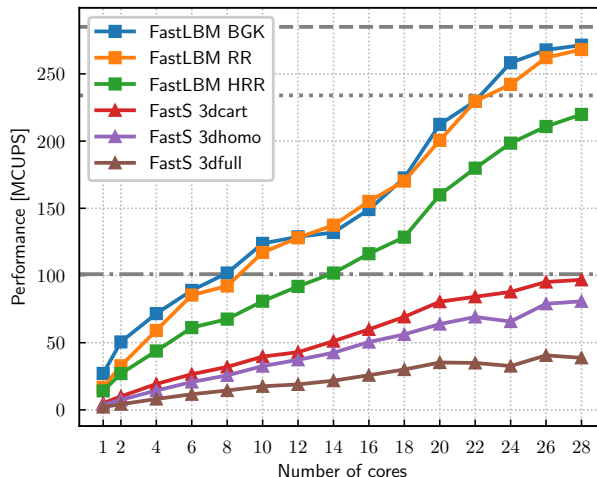


Figure 5: Comparison of the performance of each numerical scheme on one single BDW node. The horizontal lines correspond to the maximum achievable performances given in Table 3. (---) LBM BGK and RR, (.....) LBM HRR, and (-.-) FV-NS 3dcart.

It can be seen from Figure 5 that the full-node performance of the Navier-Stokes and lattice Boltzmann methods almost matches the results of the Roofline model. Indeed, the measured performances represent about 90% to 95% of the expected ones. For the Navier-Stokes method, the curves of Figure 5 highlight the substantial increase in performance achieved by implementing grid specific subroutines. The Cartesian version is naturally the fastest and is about 2.5 to 3 times faster than its full curvilinear counterpart. The `3dhomo` implementation of the Navier-Stokes method is only 1.5 times slower than the cartesian version making it a good candidate for LES or DNS in reasonable CPU time. Moving to the lattice Boltzmann models, they all are more than twice as fast as the Navier-Stokes method. As indicated by the Roofline model, when the full node is used, both the LBM-BGK and LBM-RR achieve the same performance. This property is quite interesting knowing the fact that the LBM-RR is much more stable than the LBM-BGK. On the other hand, due to the introduction of non-local operations in the algorithm of the LBM-HRR, the performance is decreased with respect to the other collision models. Yet, the LBM-HRR is still about 2.25 times faster than the Navier-Stokes method on Cartesian grids. All these results are in perfect agreement with the tendencies outlined by the Roofline model (see Table 3). Note that the performance figures of the NS method are only given in the explicit case. When considering an implicit time-stepping scheme, the value of n_{substep} depends on the number of iterations performed by the Newton approximation process. Therefore, implicit computations tend to be at least three times more expensive than the explicit ones.

In order to get more insight into the performance and scaling of each method, Figure 6 provides a plot of the performance per core (Figure (a)) as well as the effective time (Figure (b)) as functions of the number of cores. The effective time, expressed in μs , corresponds to the time required by the corresponding algorithm to perform one iteration on one cell of the mesh:

$$t_{\text{eff}} = \frac{t_{\text{CPU}} \times N_{\text{threads}}}{n_{\text{substep}} \times 200^3}. \quad (42)$$

The results of Figure 6 (a) indicate that the lattice Boltzmann models can iterate over 10 million cells per core when the node is fully loaded. In comparison, the Cartesian version of the Navier-Stokes method only performs one iteration over 3.3 million cells per core. One should keep in mind that in the explicit case 3 updates are performed within the Runge-Kutta algorithm indicating that 10 million cells can effectively be updated per core within the Navier-Stokes method but not one entire iteration as it is the case for lattice Boltzmann method.

Figure 6 (b) shows that a cell update by the lattice Boltzmann method is performed in approximately $0.1 \mu\text{s}$. Moreover, as already highlighted, the LBM-HRR is a bit slower than the other LB models. Its extra cost is found to be about 30%. Besides, Figure 6 (b) further demonstrates the need to compare the LBM with a NS method on Cartesian grids when seeking for fair one-to-one

comparisons since other formulations introduce a significant computational overhead. Consequently, in comparison to the Navier-Stokes method on Cartesian grids, the lattice Boltzmann method is about 2.2 to 3 times faster depending on the collision model. Now, with respect to the NS method on fully curvilinear grids, the lattice Boltzmann method is between 5 to 10 times faster.

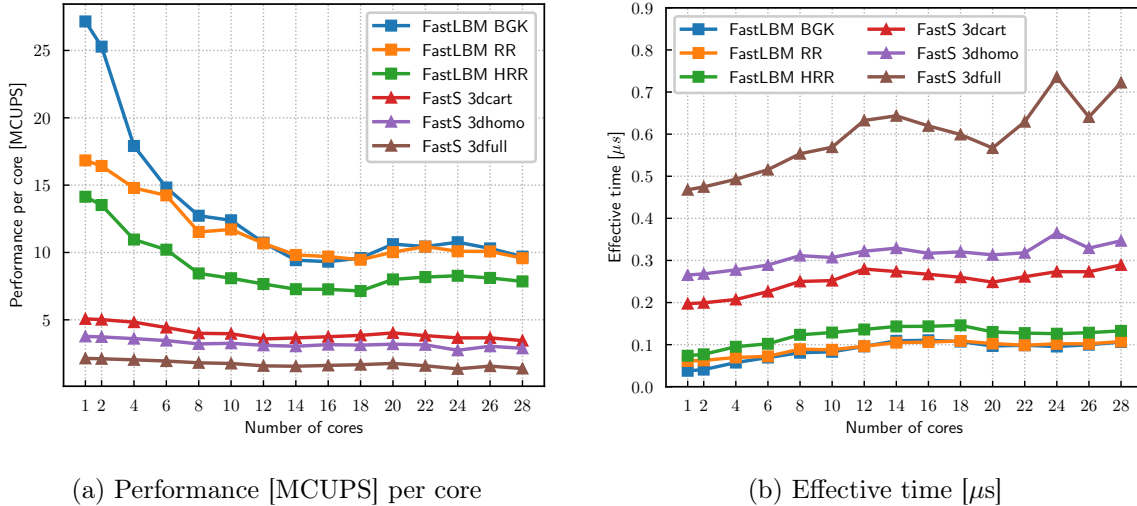


Figure 6: Performance comparison of the solver in terms of MCUPS per core and effective time.

585 In terms of strong scaling, Figure 6 (a) suggests that the Navier-Stokes and lattice Boltzmann methods do not behave in the same way. Indeed, a huge drop in the performance per core is observed for the LB schemes when increasing the number of cores whereas the performance of the NS method seems to be less affected. Such disparity might be explained by the fact that the LB “Stream & Collide” algorithm is essentially memory driven and that increasing the number of cores
590 does not lead in a significant increase in memory bandwidth. This might also explain why the LBM-RR and LBM-HRR show better scaling since they require more computations than the BGK collision model which in turn means a higher arithmetic intensity.

5. Numerical simulations

Having discussed the main differences between the lattice Boltzmann and Navier-Stokes methods
595 from a theoretical and computational point of view, the aim of this section is to bridge all the results in order to answer the question raised in the introduction which, it should be recalled, is to determine which method is the most competitive, in terms of accuracy and computational cost, for unsteady aerodynamic and aeroacoustic applications. To this end, the LB and NS methods are now compared through numerical simulations, admittedly canonical, but still representative of
600 industrial-like aeronautical Large Eddy simulation requirements in terms of simulation parameters.

In order to fairly assess the suitability of each method regarding both the accuracy and the computational cost, it is proposed to study the “time to solution” metric which corresponds to the time needed by each method to achieve a certain error target. The “time to solution” metric (denoted hereafter by T_{CPU}) is based on a variety of relevant factors and given by:

$$T_{\text{CPU}} = \frac{T c_0 t_{\text{eff}} N}{\Delta x \text{ CFL}}, \quad (43)$$

605 where T is the physical time intended to be simulated, c_0 is the speed of sound, Δx is the grid spacing, t_{eff} is the effective time introduced in Equation (42), and N is the number of grid points. Although the CFL number is also required, it is only relevant for NS schemes.

Throughout this paper, the time to solution specific to the NS method is only given for the Cartesian case. One can easily transpose the conclusions to the curvilinear NS method by using 610 the multiplicative factors given in Section 4. Moreover, all the simulations are performed on fully 3D computational domains using periodic boundary in all directions of space.

5.1. Plane monochromatic acoustic wave

The propagation of a downstream plane monochromatic acoustic wave is studied in order to assess the acoustic capabilities of the lattice Boltzmann and Navier-Stokes schemes introduced in 615 Section 2. For this purpose, the initial flow field is given by [55]:

$$\begin{aligned} \rho(x, y)|_{t=0} &= \bar{\rho} + \rho', \quad \text{where } \rho' = \epsilon \bar{\rho} \cos(k_x x + k_y y), \\ u_x(x, y)|_{t=0} &= \overline{\text{Ma}} c_s + \rho' c_s \cos(\theta_k) / \bar{\rho}, \\ u_y(x, y)|_{t=0} &= \rho' c_s \sin(\theta_k) / \bar{\rho}, \end{aligned} \quad (44)$$

where $\bar{\rho}$ and $\overline{\text{Ma}}$ denote the mean flow density and Mach number, respectively. The amplitude of the perturbation is set to $\epsilon = 10^{-3}$, which is sufficiently small to ensure linear acoustics. k_x and k_y correspond to the wavenumbers in the x - and y -direction respectively, and $\theta_k = \arctan(k_y/k_x)$ is the propagation angle. The latter are defined as $k_i = \frac{2\pi}{\Delta x N_{\text{ppw},i}}$ where $N_{\text{ppw},i}$ is the number of 620 points per wavelength in the i -direction. The speed of sound is given by $c_0 = 343.2 \text{ m.s}^{-1}$.

In the following, the grid size Δx is set to a constant value of $1 \times 10^{-2} \text{ m}$ and the computational domain extends over one wavelength in the direction of propagation with five cells in the other directions. The simulations are run for 50,000 time-steps (where $\Delta t \approx 1.68 \times 10^{-5} \text{ s}$ owing to the acoustic scaling) so as to observe significant effects of dispersion and dissipation. Note that, for this 625 study, the viscosity is set to $\nu = 1.5 \times 10^{-5} \text{ m}^2/\text{s}$ leading to $\bar{\tau} - 1/2 \approx 10^{-5}$ which is representative of air flows relevant to the aeronautical field. Consequently, the Reynolds number based on the the mean flow velocity is 10^5 . It can be shown that the density field at any given time t is given by:

$$\rho(x, y, t) = \bar{\rho} + \epsilon \bar{\rho} \exp[\text{Im}(\omega)t] \cos(\mathbf{k} \cdot \mathbf{x} - \text{Re}(\omega)t), \quad (45)$$

where the theoretical real and imaginary parts of the angular frequency ω are given by Equation (30). The numerical dispersion and the numerical dissipation induced by each method are estimated by least-square fitting Equation (45) to the density time signal. Consequently, the study of the acoustic properties of both approaches is made through the dispersion and dissipation ratios:

$$E_\omega = \frac{\text{Re}(\omega^{th})}{\text{Re}(\omega)} \quad \text{and} \quad E_\nu = \frac{\text{Im}(\omega^{th})}{\text{Im}(\omega)}. \quad (46)$$

Parametric study of the Navier-Stokes schemes. A short discussion regarding the tuning parameters of the Navier-Stokes schemes has to be made. Indeed, the latter offer a higher set of degrees of freedom than LBMs. For three fixed values of N_{ppw} , the following set of parameters is studied: the numerical scheme chosen to discretise the convective fluxes; the time-stepping scheme (which can be explicit or implicit); and the value of the CFL number. Figure 7 shows the results obtained in the case of a 1D plane monochromatic acoustic wave without mean flow (i.e. $k_y = 0$ and $\overline{\text{Ma}} = 0$).

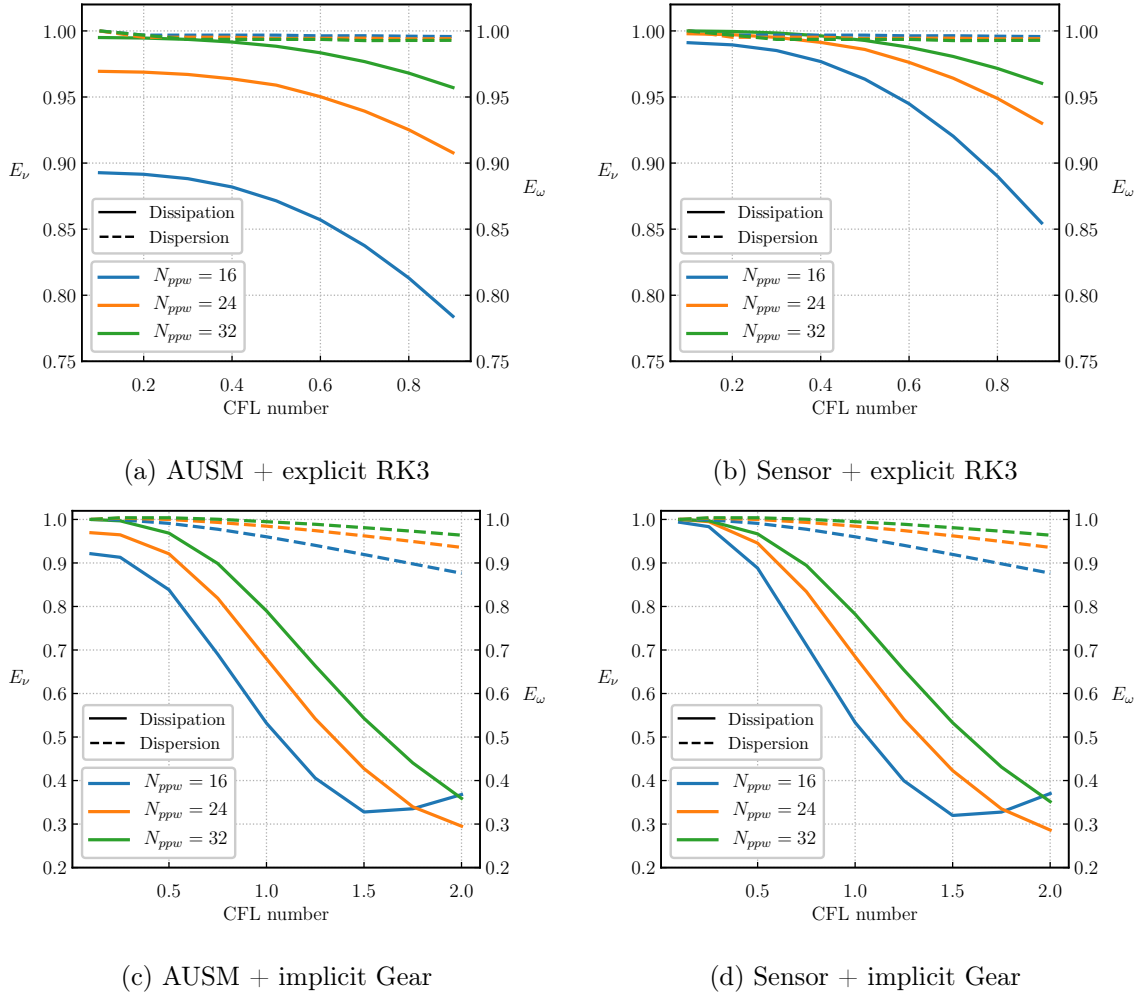
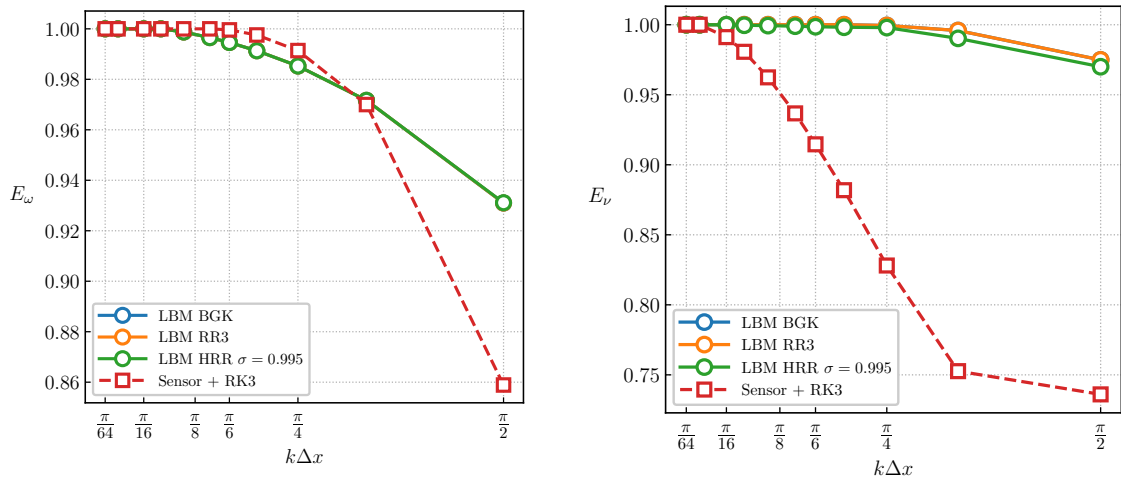


Figure 7: Dispersive (dashed lines) and dissipative (solid lines) behavior of the finite-volume Navier-Stokes schemes for different CFL numbers and points per wavelength.

It can be seen that, for all the combinations considered here, the numerical dispersion of finite-

volume Navier-Stokes schemes is lower than their numerical dissipation. Moreover, and as ex-
 640 pected, by increasing the number of points per wavelength, one approaches the theoretical behavior
 of acoustic waves. Regarding the CFL number, lowering its value tends to reduce the numerical
 dissipation of the schemes. Surprisingly, the number point per wavelength and the CFL number
 have a negligible influence on the dispersive capabilities of time-explicit schemes. Yet, despite its
 increased stability region, the implicit Gear time-stepping scheme is not suited for computational
 645 aeroacoustics (CAA) applications. Indeed, for typical CFL numbers encountered in industrial appli-
 cations, using the implicit Gear time-stepping scheme leads to an important numerical dissipation
 and dispersion. This effect can be attributed to the decentered nature of the scheme. Figure 7
 also evidences the low dissipation of the Sensor scheme in comparison to the AUSM while their
 dispersion curves are identical. This confirms the fact that the the binary function Φ only acts on
 650 the dissipative terms of the convective fluxes approximations. All in all, when considering CAA
 simulations the combination Sensor + explicit RK3 has to be favored since it offers a good tradeoff
 between dissipation and dispersion over a wide range of CFL numbers and points per wavelength.

Comparison. Figure 8 shows the dispersion and dissipation ratios as a function of the non-dimensional
 wavenumber $k\Delta x = 2\pi/N_{ppw}$ without mean flow. Since the values of E_ω and E_ν fall in the range
 655 $[0, 1]$, it can be concluded that the LB and NS methods tend to introduce some over-dissipation as
 well as a time delay (or frequency shift). Such behavior is in accordance with the results of the
 linear stability analyses of Section 3.



(a) Dispersion

(b) Dissipation

Figure 8: Comparison of the dispersive and dissipative behavior of the lattice Boltzmann (solid lines) and finite-
 volume Navier-Stokes (dashed lines) schemes for 1D plane monochromatic acoustic wave without mean flow (i.e.
 $k_y = 0$ and $\overline{Ma} = 0$).

Regarding the dispersive properties of the LB schemes, they all have the same values of E_ω ,

which also confirms the results of the LSA. However, up to 6 points per wavelength, the finite-volume
 660 Navier-Stokes scheme is slightly less dispersive than the LB ones. This result does not comply with
 Figure 2 where the dispersion error of LB schemes is expected to be less than the one made by
 second-order centered schemes. Yet, this discrepancy can be explained by the fact that both the
 AUSM and Sensor schemes are not completely equivalent to centered schemes thereby modifying
 their dispersion relations. When it comes to dissipation, the results of the linear stability analyses
 665 are retrieved. Indeed, all the LB schemes are far less dissipative than the finite-volume Navier-
 Stokes scheme. Consequently, even with more stable collision operators than the classical BGK
 model, the lattice Boltzmann method is especially suited for CAA simulations since it introduces
 very little numerical dissipation even at very low resolutions.

Figure 9 displays the minimal value of N_{ppw} for the LB and NS schemes required to achieve a
 670 given tolerated dispersion or dissipation error level. For the sake of clarity, the exact minimal value
 of N_{ppw} is reported on top of each bar. In terms of dispersion error, when considering high error
 levels (i.e. 10% or 1%), the minimal value of N_{ppw} is exactly the same for both the LB and NS
 schemes. However, with stricter requirements, the explicit Navier-Stokes Sensor scheme requires
 only 70% to 50% of the points needed by lattice Boltzmann schemes. When it comes to the minimal
 675 value of N_{ppw} required to achieve a given tolerated dissipation error level, Figure 9 clearly highlights
 the dissipation gap in favor of lattice Boltzmann schemes. Indeed, regardless of the error level and
 the chosen collision model, LB schemes require about 3 to 4 times less points per wavelength than
 the explicit finite-volume Navier-Stokes scheme.

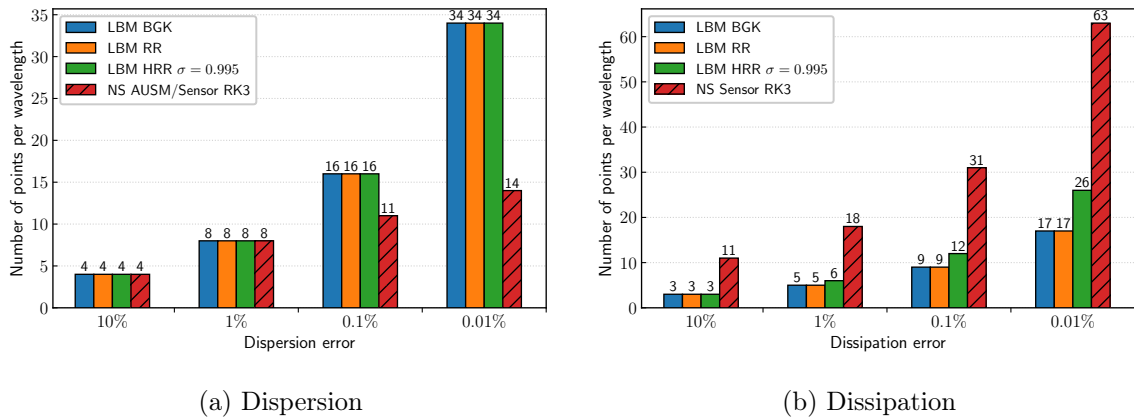


Figure 9: Minimal value of N_{ppw} for the LB and NS schemes required to achieve a given tolerated dispersion or
 dissipation error level on the acoustic mode.

Time to solution. From Equation (43), one can see that the value of T_{CPU} directly depends on the
 680 ratio between the number of points per wavelength and the CFL number. In the case of Navier-
 Stokes schemes, where the CFL number is a free parameter, the $\frac{N_{ppw}}{CFL}$ ratio has to be taken as small

as possible so as to minimise the total CPU time. According to the results of Figure 7, it can be shown that this ratio varies only by 10% around its value when $CFL_{NS} = CFL_{LBM} = 1/\sqrt{3}$. For this reason, the results are given with at $CFL_{NS} = 1/\sqrt{3}$ associated by a 10% margin to account for this slight variability. Figure 10 shows the time to solution for each method considered here when propagating the acoustic wave for 100 periods. To ease the comparison, the ratios of Navier-Stokes time to solution over the LBM ones are reported next to the bars with their corresponding color.

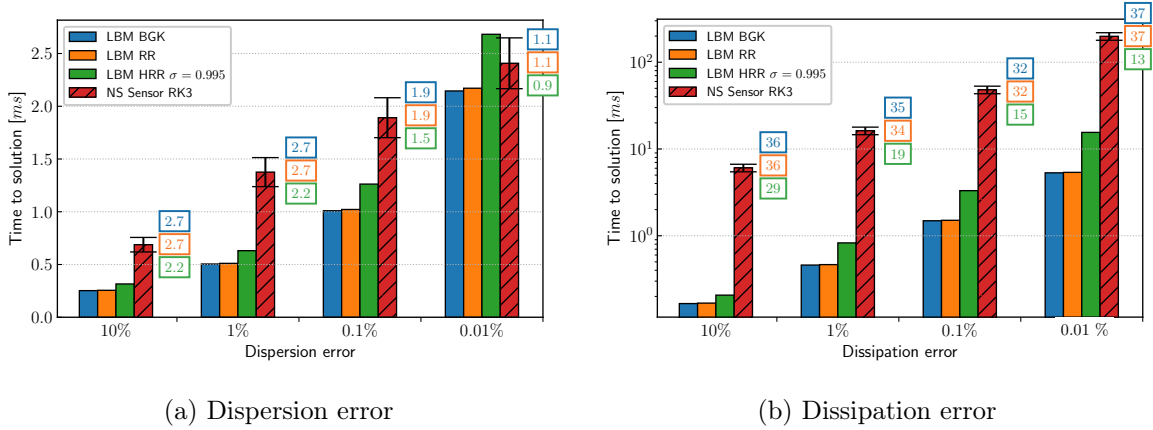


Figure 10: Time to solution for the plane monochromatic acoustic wave.

In terms of dispersion, regardless of the collision model, all the LB schemes are 2 times faster than the Navier-Stokes sensor scheme for error targets ranging from 10% to 0.1% even when taking into account the uncertainty margins. However, if one wants to keep the dispersion error below 0.01%, then all the methods seem to require the same computational time. This bar plot clearly shows that the information of the number of points per wavelength alone is not sufficient to determine whether one method is better than another. Indeed, from Figure 9 one would think that the LBM is at a disadvantage when it comes to dispersion. Moving to the dissipation results, here the advantage is clearly to the lattice Boltzmann method since it offers speedups between 15 and 30 over the whole range of error levels considered. These results correspond to a 1D planar wave, therefore, the speedup is actually between 2 to 3 times higher for full 3D computations. This demonstrates that the LBM contains intrinsic and serious acoustic capabilities thereby enabling the method to propagate acoustic fluctuations over long distances at very low computational cost.

5.2. Convected vortex

The convected vortex problem is ideal for comparing the lattice Boltzmann and Navier-Stokes approaches since they should be able to propagate the vortex without distortion for an indefinite amount of time. In addition, being capable of sustaining vortical flow structures with minimal numerical dissipation is crucial for industrial LES.

705 Most of the time, an isentropic formulation of the vortex is adopted. However, as discussed in dedicated publications [68, 69], the latter hypothesis is not consistent with the athermal approximation of standard LBMs since no energy conservation equation is solved. To alleviate this issue, a more suited "barotropic" version of the widely used Taylor vortex derived in [69] is chosen. It reads:

$$\begin{cases} \rho(x, y)|_{t=0} &= \bar{\rho} \exp \left[-\frac{\epsilon^2}{2c_s^2} \exp \left(\frac{-r^2}{R_c^2} \right) \right], \\ u_x(x, y)|_{t=0} &= \overline{\text{Mac}}_s - \epsilon \left(\frac{y - y_c}{R_c} \right) \exp \left[-\frac{(x - x_c)^2 + (y - y_c)^2}{2R_c^2} \right], \\ u_y(x, y)|_{t=0} &= \epsilon \left(\frac{x - x_c}{R_c} \right) \exp \left[-\frac{(x - x_c)^2 + (y - y_c)^2}{2R_c^2} \right], \end{cases} \quad (47)$$

710 where $\bar{\rho} = 1.1765 \text{ kg.m}^{-3}$ is the free-stream density, $\overline{\text{Ma}} = 0.1$ is the advection Mach number, $\epsilon = 0.07c_0$ is the vortex strength, and $R_c = 0.1 \text{ m}$ is the characteristic radius of the vortex. The center of the vortex is initially positioned at $(x_c, y_c) = (0.5, 0.5)$.

For this test case, the computational domain has a size of $[L, L, 10\Delta x]$, with L being the reference length equal to 1 m. The uniform grid size is set to $\Delta x = L/N$ where N is the grid resolution. The 715 simulations are performed for a range of grid resolutions such as $N_{\text{vortex}} \in \{6; 12; 25; 50; 100; 200\}$ where N_{vortex} is the number of grid points within the vortical structure leading to $\Delta x \in [0.0025 - 0.08] \text{ m}$ and $\Delta t \in [10^{-6} - 10^{-4}] \text{ s}$. The computations are run over 5 advection cycles defined by the normalised time $5t^*$ where $t^* = t\overline{\text{Mac}}_s/L$. The viscosity is set to $\nu = 1.5 \times 10^{-5} \text{ m}^2/\text{s}$ (corresponding to $\bar{\tau} - 1/2 = [10^{-5} - 10^{-7}]$) such as to mimic a vortex convection in air. The Reynolds number is 720 given by $\text{Re} = \text{Mac}_s L/\nu = 2 \times 10^6$. Thereby, viscous effects are expected to be negligible and the intrinsic numerical dissipation of each methods can be compared.

Analysis of the numerical dissipation. The convected vortex test case can be seen as the advection of a vorticity spot. Therefore, in the light of the modal analysis of the exact Navier-Stokes equations, the decay of the vorticity field is expected to be proportional to $e^{-\nu k^2 t}$. The numerical dissipation 725 rate G is estimated by averaging the norm of the vorticity field over 2 advection cycles and by computing the ratio between the 2 last and the 2 first cycles:

$$G = \left(\frac{\int_{3t^*}^{5t^*} \|\omega_z(t)\| dt}{\int_0^{2t^*} \|\omega_z(t)\| dt} \right)^{1/4t^*}. \quad (48)$$

Figure 11 displays the evolution of the vorticity norm dissipation as a function of the vortex resolution N_{vortex} . In addition, to ease the comparison between the numerical solutions and the theoretical dissipation rate, a reference curve is added in Figure 11.

730 From a general point of view, when refining the mesh, all the numerical schemes tend to recover the theoretical dissipative behavior. If one focuses on the Navier-Stokes method, one main difference

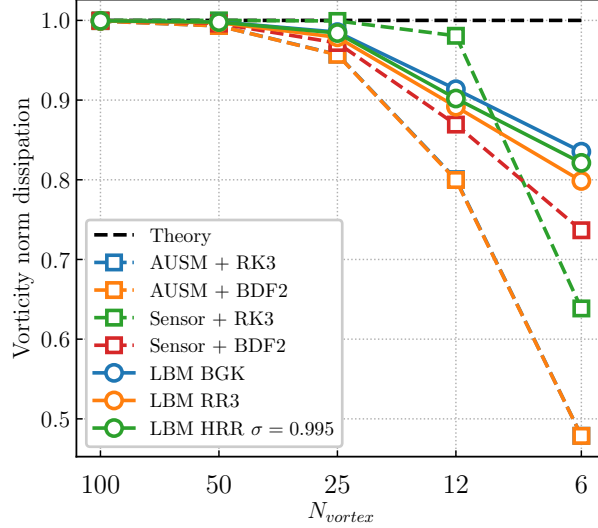


Figure 11: Numerical dissipation of the norm, with respect to the numerical wavenumber. The dashed line represents the theoretical dissipation rate of the vorticity mode.

between the AUSM and Sensor scheme can be highlighted. Indeed, regardless of the time-stepping scheme, the dissipation curves of the AUSM scheme are almost superimposed thereby indicating that the numerical dissipation is governed by the space discretisation error. Conversely, the behavior of the Sensor scheme highly depends on the time-stepping scheme thereby showing that the dissipation error is dominated by the latter. Regarding the lattice Boltzmann models, their numerical dissipation is slightly greater than the one of the Sensor scheme with explicit time-stepping scheme for $N_{vortex} > 6$. Moreover, as indicated by the linear stability analyses, switching to regularised collision models further increases the numerical dissipation especially at low resolutions. Yet, the numerical dissipation of the LB schemes is still less important than the one of the AUSM scheme. Therefore, when omitting well-designed schemes such as the Sensor scheme, the lattice Boltzmann method is a very good candidate for vortex advection. Now, the Sensor scheme, which was especially designed to achieve very little numerical dissipation, seems to outperform the LBM over a broad range of vortex resolutions. However, for typical LES-like resolutions ($6 \leq N_{vortex} \leq 12$) the advantage between both approaches is not clear and needs to be further studied.

Quality of the solution. Figure 12 compares the vortex shapes after 5 advection cycles. As one can see, for highly resolved cases ($N_{vortex} = 50$), all the numerical schemes converge to the same solution and match the analytical profile. When decreasing the vortex resolution, spurious dispersive and dissipative effects are evidenced. On the whole, the trends of Figure 11 are recovered here. For LES-like resolutions, i.e. for $N_{vortex} = 6$ or $N_{vortex} = 12$ a strong deformation of the vortex is observed thereby confirming the anisotropy of the schemes for coarse grids already pointed out by the LSA. As in the case of the acoustic wave, the numerical dissipation is the most critical flaw in

the numerical schemes studied. When $N_{\text{vortex}} = 6$, the lattice Boltzmann solvers tend to preserve in a better way the vorticity peak at the center of the vortex. The oscillations observed in the BGK vorticity profile indicate the onset of an instability that causes the calculation to diverge if it is continued. The regularised collision models however help to stabilise the computation. For $N_{\text{vortex}} \geq 12$, the NS explicit Sensor scheme has a very good behavior which converges rapidly towards the analytical solution. Therefore, the convergence of the schemes is now discussed.

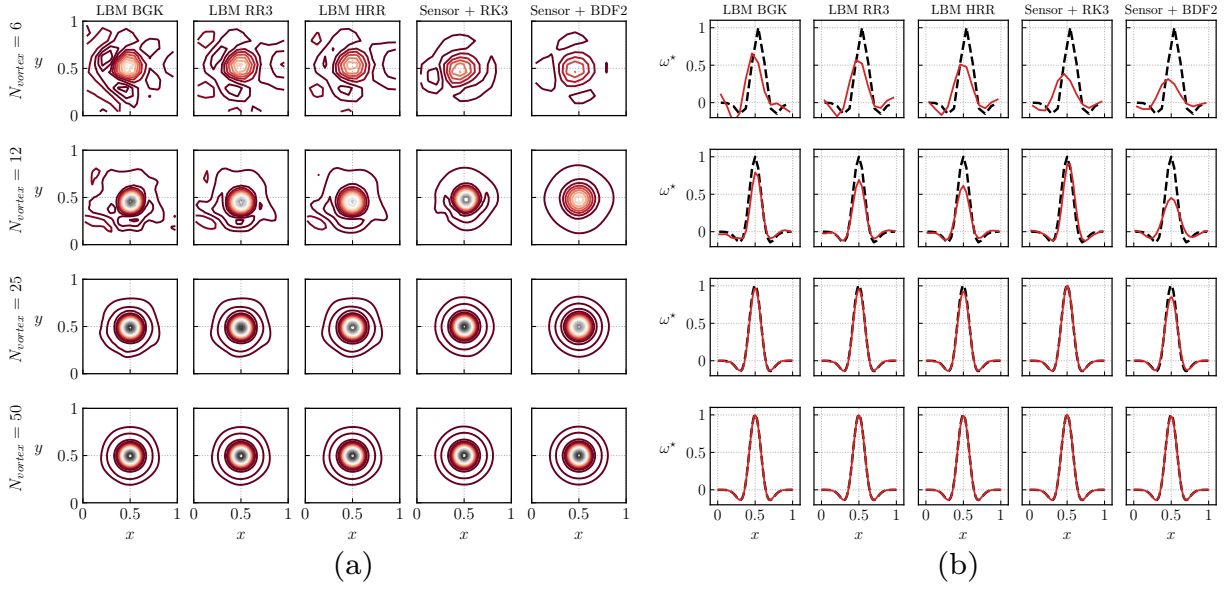


Figure 12: Comparison of the vortex shapes in terms of vorticity after 5 advection cycles. (a) Vorticity isocontours. 30 levels are displayed ranging from -1 (red) to 1 (grey). (b) Vorticity profiles for $y = 0$. (---) : Reference solution; and (—) numerical solution after 5 advection cycles.

Convergence. The convergence of both approaches is discussed through the analysis of their L^2 -error norm on the velocity field (the exact same conclusion can be obtained when reasoning with the density field). Figure 13 displays the evolution of the L^2 -error norm as a function of the vortex resolution. Regarding the lattice Boltzmann schemes, all the curves are almost superimposed and follow a second-order slope which is in accordance with the spatial order of the scheme. On the other hand, the Navier-Stokes schemes seem to follow a third-order slope even though they are designed as being second-order schemes. This discrepancy can be explained by the fact that all the computations were performed on cartesian grids with a third-order MUSCL reconstruction thereby biasing the results. Notwithstanding this unexpected behavior, the conclusions of Figures 11 and 12 are recovered. Indeed, for low resolutions, the lattice Boltzmann schemes provide a more accurate solution than classical finite-volume schemes.

Time to solution. Figure 14 reports the CPU time required by each method to achieve a certain error target on the L^2 -error norm of the velocity field after 5 advection cycles. As a matter of fact, the

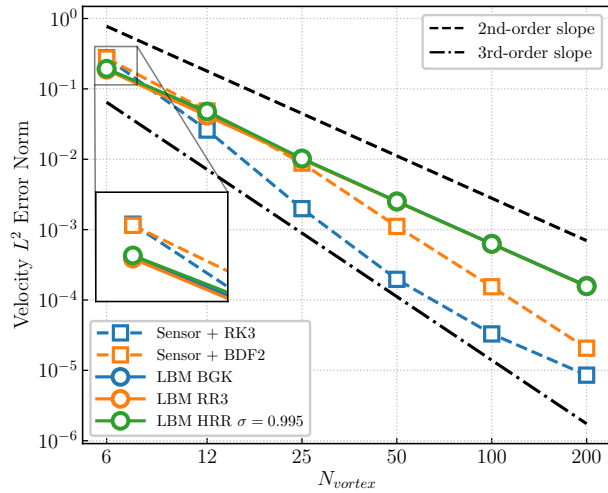


Figure 13: Convergence plot of the lattice Boltzmann (solid lines) and NS (dashed lines) schemes for the convected vortex test case. Two reference slopes are also represented.

greater the constraint, the greater the vortex resolution and therefore the longer the computational time. Compared to the monochromatic plane acoustic wave, no solver prevails over the whole range of error levels considered. Indeed, it is necessary to distinguish two cases. If large error levels are allowed (or equivalently if meshes consisting in less than 12 grid points by vortical structures are employed as it is commonly the case in LES), the lattice Boltzmann method is slightly faster than the NS Sensor scheme by a factor 2 or 3. However, when it comes to convergence down to several orders of magnitude, it is clear that Navier-Stokes methods implemented in ONERA's FastS solver are more appropriate. This tendency is quantified in Figure 14 by providing the ratio between the NS time to solution over the LB time to solution in black boxes. Thus, for $L^2(\mathbf{u}) \leq 0.1\%$, a solution is obtained 3 times faster with the NS solver than with the LBM regardless of its collision model.

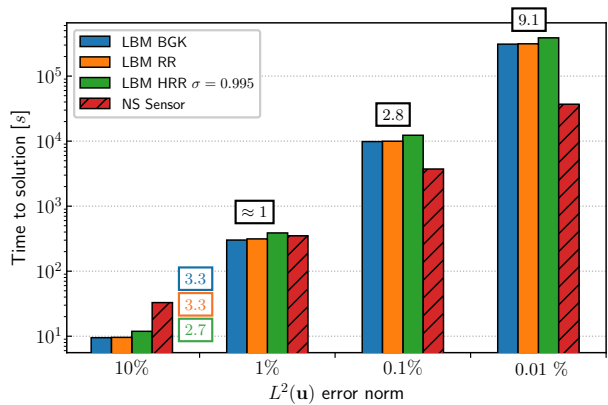


Figure 14: Time to solution for the convected vortex test case for different error levels.

It is essential to note that the conclusions drawn on this test case are not a general truth since

the order of convergence of the LB and NS methods do not match. Indeed, a fully second-order NS scheme might be less suited to $L^2(\mathbf{u})$ error minimisation. Yet, this little discussion has the merit of underlining the fact that by taking advantage of the combinations of spatial and temporal schemes and MUSCL-type reconstructions in the framework of finite-volume Navier-Stokes schemes, one can tailor more efficient numerical schemes than those offered by the LBM.

5.3. Taylor Green Vortex

In order to compare the lattice Boltzmann and Navier-Stokes methods on a 3D turbulent configuration, the Taylor-Green vortex (TGV) is considered. It is a fundamental prototype flow for vortex stretching and production of small-scale eddies which therefore allows to study the dynamics of transition to turbulence. In the following, no subgrid scale models are used so as to assess the implicit LES capabilities of each method. This test case has been widely employed to evaluate numerical methods by both the lattice Boltzmann and Navier-Stokes communities [70, 71, 72].

The flow is solved in a fully periodic cube of size $\Omega = (2\pi L)^3$, where L is a reference length. According to Brachet *et al.* [73], the initial velocity and pressure fields are given by:

$$\begin{aligned} u_x(x, y, z)|_{t=0} &= U_\infty \sin\left(\frac{x}{L}\right) \cos\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right), \\ u_y(x, y, z)|_{t=0} &= -U_\infty \cos\left(\frac{x}{L}\right) \sin\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right), \\ u_z(x, y, z)|_{t=0} &= 0, \\ p(x, y, z)|_{t=0} &= p_\infty + \frac{\rho_\infty U_\infty^2}{16} \left[\cos\left(\frac{2z}{L}\right) + 2 \right] \left[\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right) \right], \end{aligned} \quad (49)$$

where U_∞ , p_∞ , and ρ_∞ denote the reference velocity, pressure, and density respectively. All these parameters are chosen such as to impose a Reynolds number $\text{Re} = 1600$ and a Mach number of 0.1. Also note that for the LB computations, the initial distribution field is computed by calculating the velocity gradient [74].

Time dependent global quantities. As a first step to analyse the simulation results, time dependent global flow quantities are evaluated. These consist of the non-dimensional kinetic energy evolution, defined as:

$$E_k(t) = \frac{1}{2|\Omega|U_\infty^2} \int_\Omega \|\mathbf{u}\|^2 d\Omega, \quad (50)$$

and the non-dimensional enstrophy evolution :

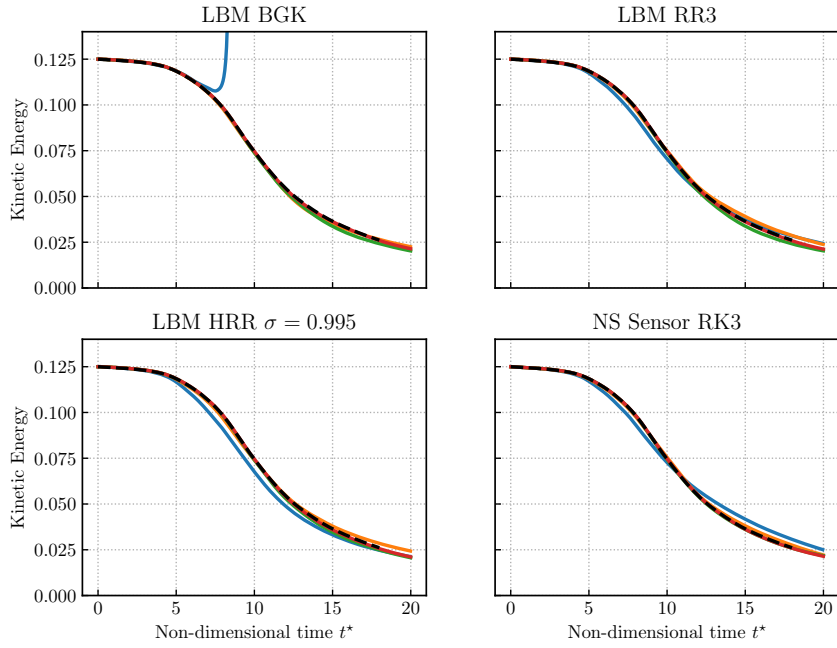
$$\mathcal{E}(t) = \frac{L^2}{2|\Omega|U_\infty^2} \int_\Omega \|\boldsymbol{\omega}\|^2 d\Omega. \quad (51)$$

The enstrophy (51) is computed from the vorticity field $\boldsymbol{\omega}$ for which a fourth-order reconstruction is used by both numerical methods for the sole purpose of post-processing. The temporal evolution of E_k and \mathcal{E} is expressed by means of the non-dimensional time scale defined by: $t^* = L/U_\infty$.

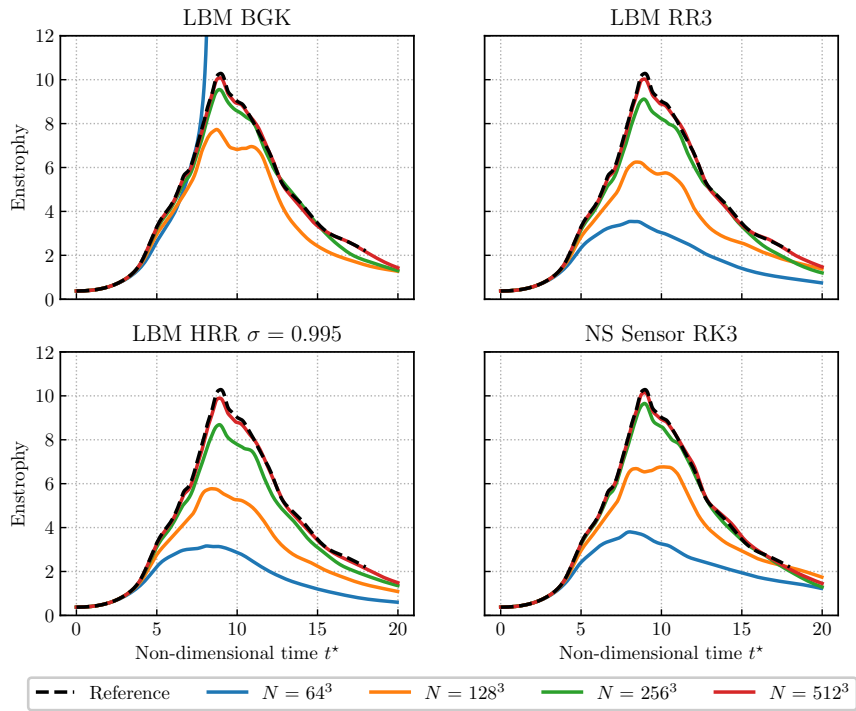
Figure 15 displays the temporal evolution of the kinetic energy and the enstrophy for three LBM computations (with BGK, RR3, and HRR collision models) and one explicit NS computation with the Sensor scheme. Indeed, the latter is especially designed such as to mitigate dissipative effect for turbulent flow computations. Different grid resolutions are investigated namely 64^3 , 128^3 , 256^3 , and 512^3 . This leads to $\Delta x \in [10^{-6} - 10^{-5}]$ and accordingly $\Delta t \in [10^{-8} - 10^{-7}]$. In addition, the viscosity is set to $\nu = 1.5 \times 10^{-5} \text{ m}^2/\text{s}$ which corresponds to $\bar{\tau} - 1/2 \approx 10^{-3}$. The results are compared to the spectral solution from [70] for a resolution of 512^3 .

The first thing to notice from Figure 15 is that the computation carried out by the lattice Boltzmann method with BGK collision operator for the highly under-resolved case (i.e. on the 64^3 grid) is unstable. This justifies the fact that regularised schemes are preferred for industrial computations. Consequently, one should keep in mind that even if stability issues are not studied here, the ability of numerical schemes to stay stable is fundamental in CFD. Both the LBM and the finite-volume NS schemes converge towards the reference solution. In particular, the enstrophy peak is being captured at $t^* = 9$ for sufficiently resolved cases (i.e. when $N \geq 128^3$). The temporal evolution of the enstrophy helps to discriminate between the numerical methods. As it has already been observed in the previous test cases, the LBM BGK method has very little numerical dissipation and therefore tends to better capture the enstrophy peak for $N = 128^3$. On the other hand, more stable collision models also come with an increased numerical dissipation since this test case is mainly vorticity driven. When $N = 512^3$, it can be shown that the grid size Δx is almost equal to the Kolmogorov length [47]. Therefore, comparing both numerical methods for this particular grid resolution gives some insight into their quasi-DNS capabilities. As can be seen on Figure 15, all the numerical schemes reach the spectral solution with very little error.

Accuracy. Figure 16 shows the convergence of the L^2 -error norm on the temporal evolution of the kinetic energy and enstrophy for the LB and NS schemes. It can be seen that both methods converge towards the reference solution at second-order in space. Since the vorticity field is reconstructed at fourth-order, no bias is attributable to the post-processing and the direct numerical behavior of each method is shown. Moreover, the curves of Figure 16 confirm the tendencies of Figure 15. Indeed, while all the numerical schemes capture well the kinetic energy evolution with an error between 10^{-4} and 10^{-6} , the enstrophy evolution is much more discriminating in terms of accuracy. Yet, all the LB schemes reach the spectral solution with the same ranges of error than the finite-volume sensor scheme, except when $N = 512^3$ where the error of the sensor scheme drops to 10^{-5} .



(a) Kinetic energy evolution



(b) Enstrophy evolution

Figure 15: Time evolution of the kinetic energy (a) and enstrophy (b) for the 3D Taylor-Green Vortex at $Re = 1600$. The reference solution is the spectral solution from [70].

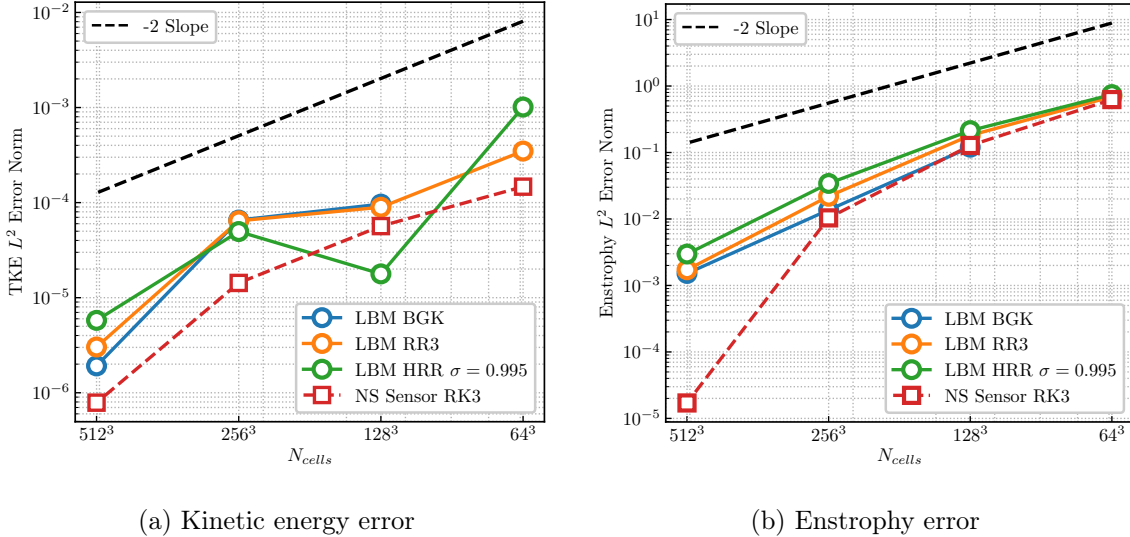


Figure 16: Accuracy and convergence study for the 3D Taylor-Green vortex test case at $\text{Re} = 1600$. All the L^2 -norms are computed with respect to the spectral solution of [70].

Energy spectrum. The last quantity investigated for the comparison of the lattice Boltzmann and
840 Navier-Stokes methods on the TGV is the turbulent energy spectrum which is defined as:

$$E(k, t) = \frac{1}{2} \sum_k |\hat{\mathbf{u}}(\mathbf{k}, t)|^2, \quad (52)$$

where $\hat{\mathbf{u}}$ is the complex Fourier transform of the velocity field, \mathbf{k} the wavenumber vector, and
 $k = |\mathbf{k}| = \sqrt{k_x^2 + k_y^2 + k_z^2}$ its norm. Equation (52) corresponds to the integration over shells of
equal wave number. Due to the sampling theorem, only wave numbers up to $k = 2/\Delta x$ in each
directions are considered. In the following, the results will be compared to those of Foti and
845 Duraisamy [75] at $t = 10t^*$.

Figure 17 shows the results for the LB and NS methods for the 64^3 , 128^3 , 256^3 , and 512^3
grids. From a general point of view, all the schemes tend to converge to the same turbulent kinetic
energy spectrum as Foti and Duraisamy [75] even though they tend to slightly over-estimate the
kinetic energy associated to very low wavenumbers. Such discrepancies with the reference solution
850 can also originate from the digitisation process of the low-resolution figure in [75]. For all the
schemes considered here, the energy cascade in the inertial range is well recovered. However, in
the dissipation range, it can be seen that regularised LB models tend to over-dissipate when going
toward the cutoff wavenumber. This confirms the previous observations indicating that regularised
models are more dissipative than NS and LBM-BGK schemes in shear-driven flows. Despite its
855 limited stability preventing it from being used in an industrial context, the LBM-BGK model
shows better convergence than the Navier-Stokes computation.

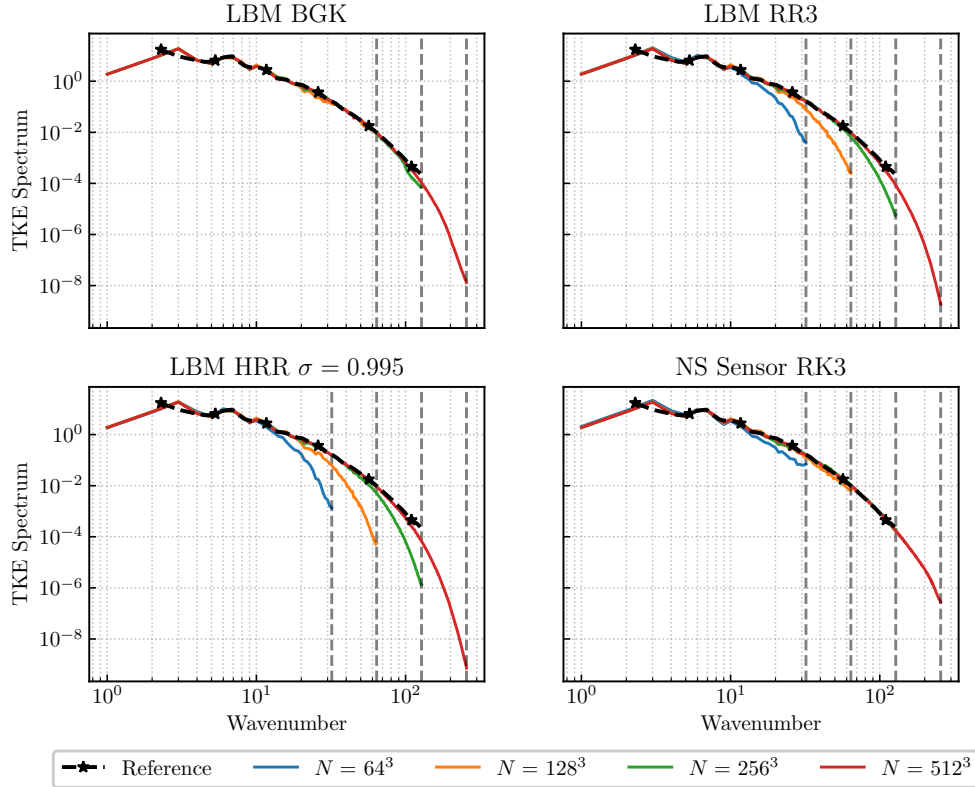


Figure 17: Turbulent kinetic energy spectrum for the 3D Taylor-Green vortex test case benchmark at $Re = 1600$. The vertical dashed lines indicate the cut-off wavenumbers associated to the smallest resolved scales.

Time to solution. Figure 18 (a) displays the computational time (expressed in hours) required by each method to compute the Taylor Green Vortex on a 512^3 grid with 8 BDW nodes thereby taking into account MPI communications. As in the single-node case, all the LB models, regardless of
 860 their collision operator, are at least two times faster than the Navier-Stokes solver.

In Figure 18 (b), the time to solution for each scheme are compared with respect to different error levels on the enstrophy curve. Once again, for high error tolerances, the lattice Boltzmann method achieves competitive runtimes. However, in the present case, due to hyperviscous effects, the HRR collision model is quickly overtaken by the performances of the Navier-Stokes method.
 865 Surprisingly, for an error target of 0.1%, the Navier-Stokes computation is about two time faster than all the LB models. This might be explained by the huge error drop depicted in Figure 16.

6. Conclusion

In this paper, a comprehensive comparison between the lattice Boltzmann and Navier-Stokes methods is performed in order to provide rigorous decision support on the suitability of one partic-
 870 ular CFD method over the other for canonical aerodynamic and aeroacoustic applications. While most of the existing approaches are subject to bias and therefore of very limited practical relevance

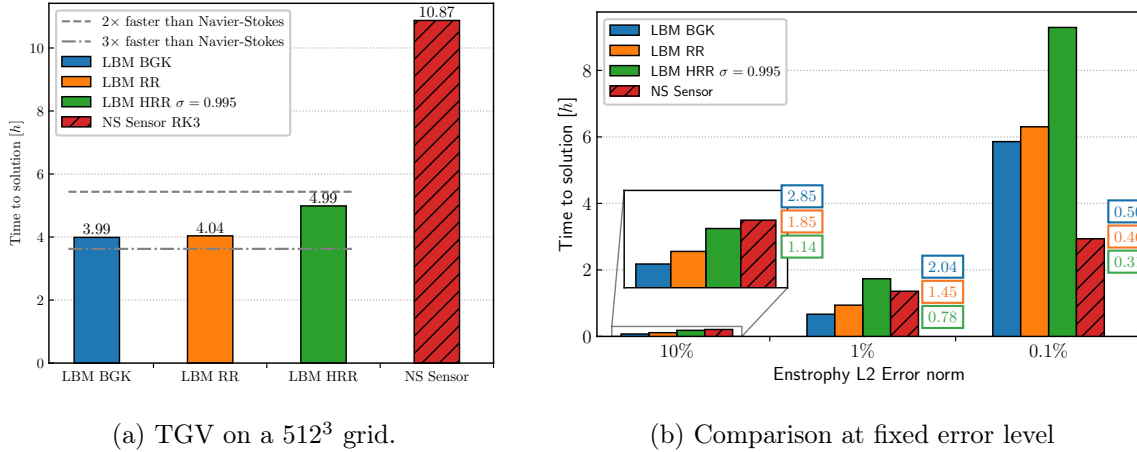


Figure 18: Time to solution for the 3D Taylor Green vortex test case.

when it comes to assess the core capabilities of the LB and NS methods, an interest of the present contribution is to thoroughly discuss each aspect of the numerical methods by taking great care so as to avoid all possible sources of bias. This leads to the following conclusions:

- 875 • The low dissipation of the LBM with respect to conventional Navier-Stokes schemes has to be nuanced especially when considering advanced LB collision models of practical relevance such as the regularised ones. While both the linear stability analyses and numerical simulations indicate that the LBM has serious acoustic capabilities, regularised collision models exhibit a higher numerical dissipation on shear modes than standard NS schemes.
- 880 • While in the NS community schemes are generally studied and optimised for one-dimensional problems, their anisotropic dissipative behavior when both the space and time discretisations are simultaneously performed is highlighted and discussed.
- 885 • When optimised to the limits of the considered CPU architecture, a cell update using the lattice Boltzmann method is found to be 2 to 3 times faster than for the Navier-Stokes method on Cartesian grids. The speedup of 10 in favor of the LBM classically reported is recovered but only for the LBM-BGK when compared to NS methods on fully curvilinear grids. Moreover, it is shown that the intrinsic HPC capabilities of the LBM have to be tempered since, as NS methods, the core algorithm is memory bound and therefore an increased performance can only be achieved by increasing the memory bandwidth.
- 890 • Finally, it is shown that, in order to fairly compare the numerical methods in terms of accuracy and computational cost, a “time to solution” metric has to be considered. Through the computation of test cases representative of LES requirements for which an analytical solution is known and therefore for which the error levels for each method can be precisely quantified,

it is shown that the efficiency of one method with respect to the other is closely related to
895 underlying physics and the chosen error threshold levels.

In the light of these conclusions, general decision aids can be provided to assist in selecting
the most efficient method for a given application. For aeroacoustic computations, regardless of
the collision model, the LBM is the most efficient method since it leads to speedups up to 30
(per direction) w.r.t. NS computations for an acoustic plane wave. In the case of shear-driven
900 flows, the most appropriate method varies according to the fidelity to be achieved. For the Direct
Numerical Simulations (DNS) of such flows, a finite-volume Navier-Stokes solver appears to be the
best candidate to achieve highly accurate results with a reduced time in comparison to the LBM.
When it comes to Large Eddy Simulations (or Quasi-DNS) the LB and finite-volume NS methods
seem slightly equivalent. However, for Very Large Eddy Simulation (VLES, where vortical structures
905 are resolved by 4 to 6 mesh points), the lattice Boltzmann method is particularly efficient. This
confirms the trend that, if the flow physics are not driven by the boundary layer (e.g. for massive
separation induced by the geometry), the LBM-VLES is very interesting in terms of runtimes.

Even if this comparison has only been performed on CPUs, the main conclusions are very
likely to be transposable to the GPU framework. Indeed, several recent studies of Navier-Stokes
910 solvers running on GPUs [76, 77] indicate a similar performance increase than the one observed
for the lattice Boltzmann method in [78]. Therefore, one could expect an acceleration of the same
order of magnitude for both the NS and LB methods. Further investigations of the performance
of these two methods on GPU by porting ONERA's research codes on such architectures may be
the purpose of future work. Also, a comprehensive comparison of the numerical treatment of walls
915 within the context of lattice Boltzmann and finite-volume Navier-Stokes methods is planned for
future work and will be the subject of a dedicated study. In addition, while the issue of stability
has received very little attention in the present work, the ability of numerical schemes to stay
stable is fundamental for real-world CFD applications especially if boundary conditions and mesh
refinements are considered. The present methodology can therefore be used as a basis for further
920 research aiming to compare the lattice Boltzmann and Navier-Stokes methods.

7. Acknowledgements

This work was performed using ONERA's HPC resources on the SATOR supercomputer. The
authors are grateful to N. Alferéz for his support on Intel Advisor and to G. Wissocq for the fruitful
discussions regarding linear stability analyses and LBM during ICMMES-22 conference. Antoine
925 M. D. Jost is also warmly thanked for the final proofreading of the manuscript.

References

- [1] P. G. Tucker, J. R. Debonis, Aerodynamics, computers and the environment, *Philosophical Transactions of the Royal Society A* 372 (2022) (2014) 20130331. doi:10.1098/rsta.2013.0331.
- [2] T. Krueger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E. M. Vigen, *The Lattice Boltzmann Method: Principles and Practice*, Graduate Texts in Physics, Springer, 2016.
- 930 [3] S. Marié, D. Ricot, P. Sagaut, Comparison between lattice Boltzmann method and Navier-Stokes high order schemes for computational aeroacoustics, *Journal of Computational Physics* 228 (4) (2009) 1056–1070. doi:10.1016/j.jcp.2008.10.021.
- [4] R. Löhner, Towards overcoming the LES crisis, *International Journal of Computational Fluid Dynamics* 33 (3) 935 (2019) 87–97. doi:10.1080/10618562.2019.1612052.
- [5] S. Geller, M. Krafczyk, J. Tölke, S. Turek, J. Hron, Benchmark computations based on lattice-Boltzmann, finite element and finite volume methods for laminar flows, *Computers and Fluids* 35 (8-9) (2006) 888–897. doi:10.1016/j.compfluid.2005.08.009.
- [6] P. Lallemand, L. S. Luo, Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean 940 invariance, and stability, *Physical Review E* 61 (6) (2000) 6546–6562. doi:10.1103/PhysRevE.61.6546.
- [7] K.-R. Wichmann, M. Kronbichler, R. Löhner, W. A. Wall, A runtime based comparison of highly tuned lattice Boltzmann and finite difference solvers, *The International Journal of High Performance Computing Applications* (2021) 370–390doi:10.1177/109434202111006169.
- [8] E. Manoha, B. Caruelle, Summary of the LAGOON solutions from the benchmark problems for airframe noise 945 computations-III workshop, in: 21st AIAA/CEAS Aeroacoustics Conference, 2015. doi:10.2514/6.2015-2846.
- [9] Y. Hou, D. Angland, A. Sengissen, A. Scotto, Lattice-boltzmann and navier-stokes simulations of the partially dressed, cavity-closed nose landing gear benchmark case, in: 25th AIAA/CEAS Aeroacoustics Conference, 2019, 2019. doi:10.2514/6.2019-2555.
- [10] M. F. Barad, J. G. Kocheemoolayil, C. C. Kiris, Lattice boltzmann and navier-stokes cartesian CFD approaches 950 for airframe noise predictions, 23rd AIAA Computational Fluid Dynamics Conference, 2017 (2017). doi:10.2514/6.2017-4404.
- [11] M. Aultman, Z. Wang, R. Auza-Gutierrez, L. Duan, Evaluation of CFD methodologies for prediction of flows around simplified and complex automotive models, *Computers & Fluids* 236 (2022) 105297. doi:10.1016/J.COMPFLUID.2021.105297.
- 955 [12] A. Aniello, D. Schuster, P. Werner, J. Boussuge, M. Gatti, C. Mirat, L. Selle, T. Schuller, T. Poinso, U. Rüde, Comparison of a finite volume and two Lattice Boltzmann solvers for swirled confined flows, *Computers & Fluids* 241 (2022) 105463. doi:10.1016/J.COMPFLUID.2022.105463.
- [13] M. Fiore, Influence of cavity flow on turbine aerodynamics, Ph.D. thesis, Institut Supérieur de l’Aéronautique et de l’Espace (2019).
- 960 [14] P. Boivin, M. Tayyab, S. Zhao, Benchmarking a lattice-Boltzmann solver for reactive flows: Is the method worth the effort for combustion?, *Physics of Fluids* 33 (7) (2021) 071703. doi:10.1063/5.0057352.
- [15] U. Piomelli, E. Balaras, Wall-Layer models for Large-Eddy Simulations, *Annual Review of Fluid Mechanics* 34 (1) (2002) 349–374. doi:10.1146/annurev.fluid.34.082901.144919.
- [16] U. Piomelli, Wall-layer models for large-eddy simulations, *Progress in Aerospace Sciences* 44 (6) (2008) 437–446. 965 doi:10.1016/j.paerosci.2008.06.001.
- [17] P. Sagaut, S. Deck, Large eddy simulation for aerodynamics: status and perspectives, *Philosophical Transactions of the Royal Society A* 367 (1899) (2009) 2849–2860. doi:10.1098/RSTA.2008.0269.

- [18] C. Benoit, S. Péron, S. Landier, Cassiopee: A CFD pre- and post-processing tool, *Aerospace Science and Technology* 45 (2015) 272–283. doi:10.1016/j.ast.2015.05.023.
- 970 [19] <http://elsa.onera.fr/Cassiopee/>.
- [20] I. Mary, J.-M. Le Gouez, C. Benoit, S. Peron, S. Landier, T. Renaud, D. Blaise, A. Jost, A. Suss, <https://w3.onera.fr/FAST/> (2022).
- [21] I. Mary, P. Sagaut, Large eddy simulation of flow around an airfoil near stall, *AIAA Journal* 40 (6) (2002) 1139–1145. doi:10.2514/2.1763.
- 975 [22] J. R. Edwards, M. S. Liou, Low-diffusion flux-splitting methods for flows at all speeds, *AIAA Journal* 36 (9) (1998) 1610–1617. doi:10.2514/2.587.
- [23] P. S. Lowery, W. C. Reynolds, Numerical simulation of a spatially-developping, forced, plane mixing layer, Tech. Rep. TF26, Stanford University (1986).
- [24] F. Daude, I. Mary, P. Comte, Self-Adaptive Newton-based iteration strategy for the LES of turbulent multi-scale
980 flows, *Computers & Fluids* 100 (2014) 278–290. doi:10.1016/J.COMPFLUID.2014.04.028.
- [25] A. Jameson, S. Yoon, Lower-upper implicit schemes with multiple grids for the Euler equations, *AIAA Journal* 25 (7) (1987) 929–935. doi:10.2514/3.9724.
- [26] T. J. Coakley, Implicit upwind methods for the compressible Navier-Stokes equations, *AIAA Journal* 23 (3) (1985) 374–380. doi:10.2514/3.8923.
- 985 [27] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method, *Journal of Computational Physics* 32 (1) (1979) 101–136. doi:10.1016/0021-9991(79)90145-1.
- [28] I. Mary, Méthode de Newton approchée pour le calcul d’écoulements instationnaires comportant des zones à très faible nombre de Mach, Ph.D. thesis, Université Paris-XI (1999).
- [29] C. Laurent, Étude d’écoulements transitionnels et hors-équilibre par des approches DNS et RANS., Ph.D. thesis, Ecole Nationale Supérieure d’Arts et Métiers (2012).
990
- [30] C. Laurent, I. Mary, V. Gleize, A. Lerat, D. Arnal, DNS database of a transitional separation bubble on a flat plate and application to RANS modeling validation, *Computers & Fluids* 61 (2012) 21–30. doi:10.1016/J.COMPFLUID.2011.07.011.
- [31] N. Alferez, I. Mary, E. Lamballais, Study of Stall Development Around an Airfoil by Means of High Fidelity
995 Large Eddy Simulation, *Flow, Turbulence and Combustion* 2013 91:3 91 (3) (2013) 623–641. doi:10.1007/S10494-013-9483-7.
- [32] J. Dandois, I. Mary, V. Brion, Large-eddy simulation of laminar transonic buffet, *Journal of Fluid Mechanics* 850 (2018) 156–178. doi:10.1017/JFM.2018.470.
- [33] Y. H. Qian, D. D’Humières, P. Lallemand, Lattice BGK Models for Navier-Stokes Equation, *Europhysics Letters* (EPL) 17 (6) (1992) 479–484. doi:10.1209/0295-5075/17/6/001.
1000
- [34] P. L. Bhatnagar, E. P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, *Physical Review* 94 (3) (1954) 511–525. doi:10.1103/PhysRev.94.511.
- [35] S. Chapman, T. G. Cowling, *The Mathematical Theory of Non-uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*, Cambridge Mathematical Library, Cambridge University Press, 1990.
1005
- [36] X. Shan, X. F. Yuan, H. Chen, Kinetic theory representation of hydrodynamics: A way beyond the Navier-Stokes equation, *Journal of Fluid Mechanics* 550 (2006) 413–441. doi:10.1017/S0022112005008153.
- [37] A. T. White, C. K. Chong, Rotational invariance in the three-dimensional lattice Boltzmann method is

- 1010 dependent on the choice of lattice, *Journal of Computational Physics* 230 (16) (2011) 6367–6378. doi:10.1016/J.JCP.2011.04.031.
- [38] S. K. Kang, Y. A. Hassan, The effect of lattice models within the lattice Boltzmann method in the simulation of wall-bounded turbulent flows, *Journal of Computational Physics* 232 (1) (2013) 100–117. doi:10.1016/J.JCP.2012.07.023.
- 1015 [39] G. Silva, V. Semiao, Truncation errors and the rotational invariance of three-dimensional lattice models in the lattice Boltzmann method, *Journal of Computational Physics* 269 (2014) 259–279. doi:10.1016/J.JCP.2014.03.027.
- [40] M. Geier, M. Schönherr, A. Pasquali, M. Krafczyk, The cumulant lattice Boltzmann equation in three dimensions: Theory and validation, *Computers & Mathematics with Applications* 70 (4) (2015) 507–547. doi:https://doi.org/10.1016/j.camwa.2015.05.001.
- 1020 [41] P. J. Dellar, An interpretation and derivation of the lattice Boltzmann method using Strang splitting, *Computers & Mathematics with Applications* 65 (2) (2013) 129–141. doi:10.1016/j.camwa.2011.08.047.
- [42] X. He, S. Chen, G. D. Doolen, A Novel Thermal Model for the Lattice Boltzmann Method in Incompressible Limit, *Journal of Computational Physics* 146 (1) (1998) 282–300. doi:10.1006/jcph.1998.6057.
- 1025 [43] P. J. Dellar, Bulk and shear viscosities in lattice Boltzmann equations, *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* 64 (3) (2001) 11. doi:10.1103/PhysRevE.64.031203.
- [44] G. Wissocq, P. Sagaut, J.-F. Boussuge, An extended spectral analysis of the lattice Boltzmann method: modal interactions and stability issues, *Journal of Computational Physics* 380 (2019) 311–333. doi:10.1016/j.jcp.2018.12.015.
- 1030 [45] D. D’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, L. S. Luo, Multiple-relaxation-time lattice Boltzmann models in three dimensions, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 360 (1792) (2002) 437–451. doi:10.1098/rsta.2001.0955.
- [46] C. Feuchter, Direct aeroacoustic simulation with a cumulant Lattice-Boltzmann model, *Computers & Fluids* (2021) 104970doi:10.1016/j.compfluid.2021.104970.
- 1035 [47] M. Geier, S. Lenz, M. Schönherr, M. Krafczyk, Under-resolved and large eddy simulations of a decaying Taylor–Green vortex with the cumulant lattice Boltzmann method, *Theor. Comput. Fluid Dyn.* (2020) 1–40doi:10.1007/s00162-020-00555-7.
- [48] O. Malaspinas, Increasing stability and accuracy of the lattice Boltzmann scheme: recursivity and regularization (may 2015). arXiv:1505.06900.
- 1040 [49] J. Jacob, O. Malaspinas, P. Sagaut, A new hybrid recursive regularised Bhatnagar–Gross–Krook collision model for lattice Boltzmann method-based large eddy simulation, *Journal of Turbulence* 19 (11) (2019) 1051–1076. doi:10.1080/14685248.2018.1540879.
- [50] J. Degriigny, Towards the Computational Prediction of Low-Speed Buffet: Improved Wall Modeling for the Lattice Boltzmann Method, Ph.D. thesis, Université Aix-Marseille (2021).
- 1045 [51] G. Wissocq, P. Sagaut, Hydrodynamic limits and numerical errors of isothermal lattice Boltzmann schemes, *Journal of Computational Physics* 450 (2022) 110858. doi:10.1016/J.JCP.2021.110858.
- [52] Y. Feng, P. Boivin, J. Jacob, P. Sagaut, Hybrid recursive regularized thermal lattice Boltzmann model for high subsonic compressible flows, *J. of Comp. Phys.* 394 (2019) 82–99. doi:10.1016/j.jcp.2019.05.031.
- [53] C. K. Tam, J. C. Webb, Dispersion-Relation-Preserving Finite Difference Schemes for Computational Acoustics, *Journal of Computational Physics* 107 (2) (1993) 262–281. doi:10.1006/JCPH.1993.1142.
- 1050 [54] C. Bogey, C. Bailly, A family of low dispersive and low dissipative explicit schemes for flow and noise compu-

- tations, *Journal of Computational Physics* 194 (1) (2004) 194–214. doi:10.1016/j.jcp.2003.09.003.
- [55] G. Wissocq, C. Coreixas, J.-F. Boussuge, Linear stability and isotropy properties of athermal regularized lattice Boltzmann methods, *Physical Review E* 102 (5) (2020) 053305. doi:10.1103/PhysRevE.102.053305.
- 1055 [56] F. Renard, G. Wissocq, J. F. Boussuge, P. Sagaut, A linear stability analysis of compressible hybrid lattice Boltzmann methods, *Journal of Computational Physics* 446 (2021) 110649. doi:10.1016/J.JCP.2021.110649.
- [57] T. Astoul, G. Wissocq, J.-F. Boussuge, A. Sengissen, P. Sagaut, Analysis and reduction of spurious noise generated at grid refinement interfaces with the lattice Boltzmann method, *Journal of Computational Physics* 418 (2020) 109645. arXiv:2004.11863, doi:10.1016/j.jcp.2020.109645.
- 1060 [58] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, Array programming with NumPy, *Nature* 585 (7825) (2020) 357–362. doi:10.1038/s41586-020-2649-2.
- 1065 [59] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz, Sympy: symbolic computing in python, *PeerJ Computer Science* 3 (2017) e103. doi:10.7717/peerj-cs.103.
- [60] N. Alferéz, I. Mary, J.-M. Le Gouez, A. Farjallah, C. Andreolli, Intel Xeon and Xeon Phi optimizations of an industry-oriented Computational Fluid Dynamics solver, in: Intel HPC Developer Conference, 2017.
- 1070 [61] G. Wellein, T. Zeiser, G. Hager, S. Donath, On the single processor performance of simple lattice Boltzmann kernels, *Computers & Fluids* 35 (8-9) (2006) 910–919. doi:10.1016/J.COMPFLUID.2005.02.008.
- [62] M. Bauer, H. Köstler, U. Rüde, lbmpy: Automatic code generation for efficient parallel lattice Boltzmann methods, *Journal of Computational Science* 49 (2021) 101269. doi:10.1016/j.jocs.2020.101269.
- 1075 [63] S. Williams, A. Waterman, D. Patterson, Roofline: an insightful visual performance model for multicore architectures, *Communications of the ACM* 52 (4) (2009) 65–76. doi:10.1145/1498765.1498785.
- [64] J. D. McCalpin, Memory bandwidth and machine balance in current high performance computers, *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter* (1995) 19–25.
- [65] J. Habich, A performance engineering process for developing high performance lattice Boltzmann implementations, Ph.D. thesis, Technischen Fakultät der Universität Erlangen-Nürnberg (2015).
- 1080 [66] G. Jeanmasson, I. Mary, L. Mieussens, On some explicit local time stepping finite volume schemes for CFD, *Journal of Computational Physics* 397 (2019) 108818. doi:10.1016/j.jcp.2019.07.017.
- [67] F. Schornbaum, U. Rüde, Massively parallel algorithms for the lattice boltzmann method on nonuniform grids, *SIAM Journal on Scientific Computing* 38 (2) (2016) C96–C126. arXiv:1508.07982, doi:10.1137/15M1035240.
- 1085 [68] F. Gendre, D. Ricot, G. Fritz, P. Sagaut, Grid refinement for aeroacoustics in the lattice Boltzmann method: A directional splitting approach, *Physical Review E* 96 (2) (2017) 023311. doi:10.1103/PhysRevE.96.023311.
- [69] G. Wissocq, J.-F. Boussuge, P. Sagaut, Consistent vortex initialization for the athermal lattice Boltzmann method, *Physical Review E* 101 (4) (2020) 043306. doi:10.1103/PhysRevE.101.043306.
- [70] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P. O. Persson, B. van Leer, M. Visbal, High-order CFD methods: Current status and perspective, *Int. J. Numer. Meth. Fluids* 72 (8) (2013) 811–845. doi:10.1002/fld.3767.
- 1090 [71] P. Nathen, D. Gaudlitz, M. J. Krause, N. A. Adams, On the Stability and Accuracy of the BGK, MRT and RLB Boltzmann Schemes for the Simulation of Turbulent Flows, *Communications in Computational Physics*

23 (3) (2018). doi:10.4208/cicp.oa-2016-0229.

- 1095 [72] M. Haussmann, S. Simonis, H. Nirschl, M. J. Krause, Direct numerical simulation of decaying homogeneous isotropic turbulence - Numerical experiments on stability, consistency and accuracy of distinct lattice Boltzmann methods, *International Journal of Modern Physics C* 30 (9) (sep 2019). doi:10.1142/S0129183119500748.
- [73] M. E. Brachet, D. I. Meiron, B. G. Nickel, R. H. Morf, U. Frisch, S. A. Orszag, Small-scale structure of the Taylor-Green vortex, *Journal of Fluid Mechanics* 130 (1983) 411–452. doi:10.1017/S0022112083001159.
- 1100 [74] P. A. Skordos, Initial and boundary conditions for the lattice Boltzmann method, *Physical Review E* 48 (6) (1993) 4823–4842. doi:10.1103/PhysRevE.48.4823.
- [75] D. Foti, K. Duraisamy, An investigation of an implicit large-eddy simulation framework for the vorticity transport equations, in: *2018 AIAA Fluid Dynamics Conference*, 2018. doi:10.2514/6.2018-3407.
- [76] M. Bernardini, D. Modesti, F. Salvadore, S. Pirozzoli, STREAMS: A high-fidelity accelerated solver for direct numerical simulation of compressible turbulent flows, *Computer Physics Communications* 263 (2021) 107906. doi:10.1016/j.cpc.2021.107906.
- 1105 [77] D. Jude, J. Sitaraman, A. Wissink, An octree-based, cartesian Navier–Stokes solver for modern cluster architectures, *Journal of Supercomputing* 78 (9) (2022) 11409–11440. doi:10.1007/S11227-022-04324-7/FIGURES/17.
- [78] J. Latt, C. Coreixas, J. Beny, Cross-platform programming model for many-core lattice Boltzmann simulations, *PLOS ONE* 16 (4) (2021) 1–29. doi:10.1371/journal.pone.0250306.
- 1110