Extended comparison between lattice Boltzmann and Navier-Stokes solvers for unsteady aerodynamic and aeroacoustic computations

Alexandre Suss[1,*], Ivan Mary[1], Thomas Le Garrec[1], Simon Marié[2,3]

[1]DAAA, ONERA, Université Paris Saclay, F-92322 Châtillon - France
[2]Laboratoire DynFluid, F-75013 Paris - France
[3]Conservatoire National des Arts et Métiers, F-75003 Paris - France

Abstract

Among all the computational fluid dynamics methods, structured finite-type Navier-Stokes methods and lattice Boltzmann methods have emerged as excellent candidates to achieve high-fidelity computations of complex configurations in the near future. In an effort to understand the numerical capabilities of both methods, a lattice Boltzmann solver and an LES-type finite-volume Navier-Stokes solver are compared in terms of accuracy and computational cost on canonical aerodynamic and aeroacoustic applications.

An extended von Neumann analysis of both approaches gives some insight into the theoretical dispersion and dissipation errors of each numerical schemes. Using these results, numerical simulations are performed to asses the capabilities of the lattice Boltzmann and Navier-Stokes methods on linear and non-linear a test cases. The CPU performance is also discussed with the help of performance estimates and single-node scalings. In this study, both approaches are implemented within an unified CFD framework and rely on the optimisation layers thereby providing an unbiased comparison. It is shown that, a "time to solution" metric has to be considered in order to fairly compare both numerical methods when used to their optimal extent.

While a cell update is about two to three times faster with the lattice Boltzmann method w.r.t to the finite-volume Navier-Stokes solver on cartesian grids, the proposed test cases demonstrate that the relevance of one method over the other strongly depends on the underlying physics, the mesh resolution and the intended error target.

Keywords: Lattice Boltzmann Method, Navier-Stokes, Finite-volume method, von Neumann analysis, Performance

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 in combination with the increase of robustness and accuracy of CFD solvers. While Reynolds Averaged Navier-Stokes (RANS) simulations on body-fitted meshes are the current workhorse of the aeronautical industry, the understanding of unsteady phenomena is becoming a crucial issue, making the use of high fidelity methods such as Large Eddy Simulaitions (LES) compulsory. Consequently, the development of efficient, robust and accurate CFD methods that can produce LES level results is reasonable CPU time is an important field of research.

[*] Corresponding author.
Email address: alexandre.suss@onera.fr (Alexandre Suss)

As noted by Spalart & Venkatakrishnan [1], a large variety of numerical methods exist when it comes to solving the fluid mechanics equations. The direct discretisation of the Navier-Stokes (NS) equations through the finite volume method has become established approach since it offers a good compromise between flexibility and robustness. However, in the last two decades, a new method has become increasingly popular within the industry: the lattice Boltzmann Method (LBM). The latter methods, when applied to structured grids, are expected to be the first to achieve industrial-level LES [2]. 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 ?

The comparison between traditional Navier-Stokes and lattice Boltzmann methods is not straightforward since several criteria have to be taken into account (such as their their numerical properties and their computational cost) [3] and only few studies provide general tendencies. From a theoretical point of view, Marié *et al.* [4] were the first to compare the spectral properties of optimized Navier-Stokes schemes with LB models. They demonstrate the low dissipative capabilities the low computational intensity of the LBM. Nevertheless, their study is restricted to the LBM-BGK model which is of very little use in practice because of its poor stability [5]. In the light of recent studies focusing on the spectral properties of more stable and advanced LB models [6, 7], the results of [4] have to be revised by also focusing on the isotropy of the different schemes. From a computational point of view, the memory bandwidth, and thus the architecture, is expected to play an important role on the performances of a given application [8]. In this respect, even though the number of operations provides insightful information, the latter metric is not sufficient to conclude on the cost of one method over the other. When it comes to runtimes, Wichmann *et al.* [9] were the first to pave the way towards a fair and unbiased comparison between the LBM and a finite-difference Navier-Stokes solver. Indeed, most of the papers including timing information compared specialized CFD solvers typically optimized for a very limited set of problems and architecture types. Consequently, there are many covert ways in which some performance demonstrations fall short of practical applicability. Wichmann *et al.* [9] show that the LBM tends to exhibit a higher performance for complex flow problems and coarser tolerances. Yet, their conclusion, even though being valuable still lacks of relevance since the two solver being compared were developed by two independent teams and thereby do not rely on the same HPC layers. Moreover, there still is a lack of data when considering LES-type applications such as turbulent flows and acoustic phenomena. Comparisons focusing on industrial-levels LES have been applied to a large variety of configurations such as landing gears [10, 11, 12], automotive models [13], swirled flows typical of aeronautical combustion chambers [14], linear cascade configuration [15] and reactive flows [16]. All these studies are in favor of the LBM indicating speedups up to 10 with respect to Navier-Stokes solvers. Yet, the latter conclusion has to be tempered since most of the simulations being confronted were performed by different teams around the globe using different proprietary and open source solvers each having its own optimisation level. Even when the computations were set up by the same team, both the comparison of the aerodynamic fields and the runtimes do not distinguish between the use of structured and unstructured meshes, and between wall-modeled and wall-resolved. For this reason, in order to perform a fair and unbiased comparison between the lattice Boltzmann and Navier-Stokes methods, both methods have to rely on a common basis using the same data structure and running conditions.

The aim of this study is to fill this gap by introducing a unified theoretical and numerical framework enabling a one-to-one comparison of the lattice Boltzmann and Navier-Stokes methods for unsteady aerodynamic and aeroacoustic problems. Rather than comparing both approaches on complex industrial-like configurations, the focus is made on canonical test cases representative of LES requirements. Therefore, the influence of multiple resolution domains and boundary conditions is left aside and only the core capabilities of each method are studied.

The paper is organized as follows: Section 2 is devoted to the presentation of the numerical framework and the basic theoretical background of each method. Then, the comparison of the theoretical dispersion and dissipation errors of both methods through an extended von Neumann analysis is performed in Section 3. The main results are further numerically validated thanks to the simulation of linear waves and representative aerodynamic and aeroacoustic test cases in Section 4. Finally, Section 5 compares both solvers in terms of CPU performance. Therefore, a performance model is introduced and confronted to measurement on a selected architecture. A “time to solution" metric is also studied to conclude on the relevance of one method over the other on the test cases of Section 4.

2. Numerical methods

The comparison between the lattice Boltzmann and Navier-Stokes solvers is performed in the framework of ONERA’s Cassiopée/FAST CFD environment. The consists in efficient and interoperable CFD modules sharing the same code architecture and parallel processing functions. This framework is a perfect candidate for performance comparisons since it avoids the heterogeneities and the computational overhead of independent CFD codes typically implementing one single numerical method and optimized on a very limited set of configurations. The simplified infrastructure design of ONERA’s Cassiopée/FAST CFD environment is illustrated in Figure 1. Each block corresponds to a Python module or package designed to perform specific tasks. All blocks can be assembled to built very large user applications from defined sequences of modules.

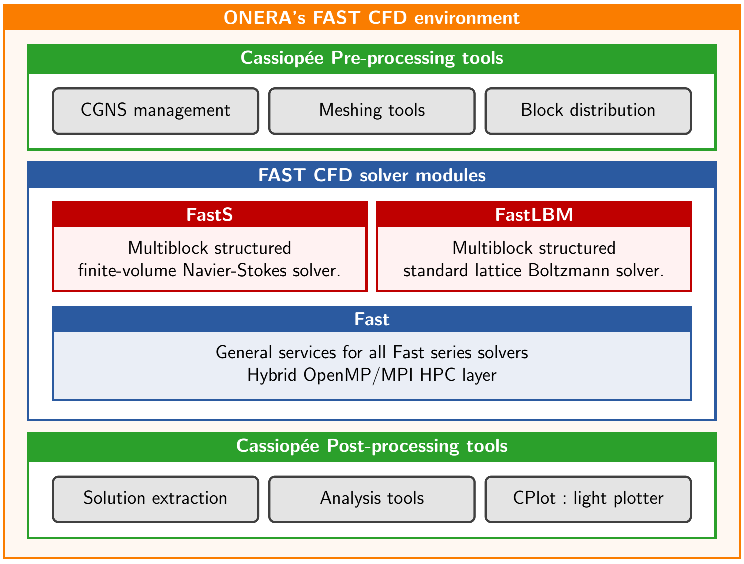

ONERA's FAST CFD environment
Cassiopée Pre-processing tools
CGNS management
Meshing tools
Block distribution
FAST CFD solver modules
FastS
Multiblock structured finite-volume Navier-Stokes solver.
FastLBM
Multiblock structured standard lattice Boltzmann solver.
Fast
General services for all Fast series solvers
Hybrid OpenMP/MPI HPC layer
Cassiopée Post-processing tools
Solution extraction
Analysis tools
CPlot : light plotter

Figure 1: Schematic diagram of the computational workflow in ONERA’s Cassiopée/FAST CFD environment. Other computational modules (such as unstructured Navier-Stokes solvers) are not represented here for the sake of clarity.

In Figure 1, one typical assembly example is shown in the case of a basic computation. The pre- and post-processing tasks are performed using Cassiopee [17, 18] which provides all the necessary tools to handle the computational grids and their associated data-structure. The computation in itself is done by the FAST (Flexible Aerodynamic Software Technology) CFD suite [19] which also consists in a set of Python modules implementing highly efficient solvers for unsteady fluid dynamics applications. The target being High Performance Computing, where the key point is the ability to handle very large data sets within an reasonable computational time, the solvers core code is implemented in appropriated low-level programming language like C, C++, Fortran wrapped to the Python layer thanks to dedicated APIs.

In the following, a more detailed description of the lattice Boltzmann and Navier-Stokes computational modules is given.

2.1. Finite-volume Navier-Stokes flow solver

The three-dimensionnal compressible unsteady Navier-Stokes (NS) equations are solved using ONERA's FastS solver dedicated to multi-block structered grids. Starting from the the conservative form of the Navier-Stokes equations:

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

where $\mathbf{U} = (\rho, \rho u_i, \rho E)^t$, $\mathbf{F}(\mathbf{U})$ and $\mathbf{F}^v(\mathbf{U})$ are the flow variable vectors, the inviscid and viscous fluxes, respectively; the cell-centered finite volume method is obtained by splitting the 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:

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

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. Actually, the residual is defined as the algebraic sum of the convective and viscous fluxes over the whole boundary of a cell.

The convective fluxes are approximated through second-order accurate spatial schemes depending on the flow regime: the Roe-MUSCL scheme [20] for transonic and supersonic flow simulations or a modified AUSM scheme proposed by Mary & Sagaut [21] for subsonic flow simulations, which is based on the AUSM+(P) scheme (see Edwards & Liou [22]). Due to the low Mach restriction of the standard lattice Boltzmann methods (as will be seen in Section 2.2), only the AUSM scheme is introduced in the following. Regarding the viscous fluxes, the latter are discretised using a second-order accurate centered scheme.

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

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

where L/R denotes the left and right third-order MUSCL interpolated states. The pressure term $\mathbf{P}_l$ is given by $(p_L + p_R)/2 \cdot (0, \mathbf{n}\cdot\mathbf{e}_1, \mathbf{n}\cdot\mathbf{e}_2, \mathbf{n}\cdot\mathbf{e}_3, 0)^t$. U_l denotes the interface fluid velocity which is defined as:

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

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

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

Both c_1 and c_2 are constant parameters chosen as small as possible to minimize the numerical dissipation. In [23], an optimal value of 0.04 has been determined. This decentered scheme will be denoted as AUSM.

In order to 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 [24].

It relies on a binary sensor function Φ_l, which only depends on the smoothness of the primitive variables $\psi = (\rho, u_1, u_2, u_3, p)^t$, acting on the dissipative terms in Equations (3):

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

where the dissipative term in U_l becomes:

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

If no spurious oscillations are detected on ψ around cell Ω_{ijk}, then the convective flux of Equation (6) degenerates to a fully centered approximation. The latter scheme will be denoted as "Sensor" in the following.

Two different time-stepping schemes are implemented in FastS: an explicit 3^{rd}-order accurate low-storage Runge-Kutta scheme [25] and an implicit 2^{nd}-order accurate backward scheme of Gear with local Newton sub-iterations [26]. In the latter case, the Jacobians are approximated following the procedure presented in [27, 28] and the linear system is solved by the LU-SGS factorisation [27].

The FastS solver has been extensively used and validated for both academic and industrial unsteady flow simulations such as transitional separation bubble [29], airfoils in near-stall configurations [21, 30] and laminar transonic buffet [31].

2.2. Lattice Boltzmann method

Unlike the finite-volume Navier-Stokes method described in Section 2.1, the lattice Boltzmann method [32, 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 describing the evolution of gases in terms of distribution functions $f_i(\boldsymbol{x}, t)$ which represent the probability density of finding fictitious particles at a location $\boldsymbol{x}$ 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:

$$\frac{\partial}{\partial t} f_i(\boldsymbol{x}, t) + \boldsymbol{\xi}_i \cdot \frac{\partial}{\partial \boldsymbol{x}} f_i(\boldsymbol{x}, t) = \Omega_i(\boldsymbol{x}, t) \qquad i \in [\![1 \; ; \; q]\!] \tag{8}$$

where q is the number of discrete velocities. The right-hand side term $\Omega_i(\boldsymbol{x}, t)$ 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:

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

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

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

The number, norm and orientation of the discrete velocities $\{\boldsymbol{\xi}_i\}_{i \in [\![1;q]\!]}$ 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:

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

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]:

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

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

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

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 Gaus-Hermite quadrature naturally provide the discrete velocities as well as their associated weights.

The expansion of the equilibrium distribution functions leads to:

$$f_i^{eq} = w_i \sum_{n=0}^{N} \frac{1}{c_s^{2n} n!} \boldsymbol{a}_{eq}^{(n)} : \boldsymbol{\mathcal{H}}_i^{(n)} \tag{14}$$

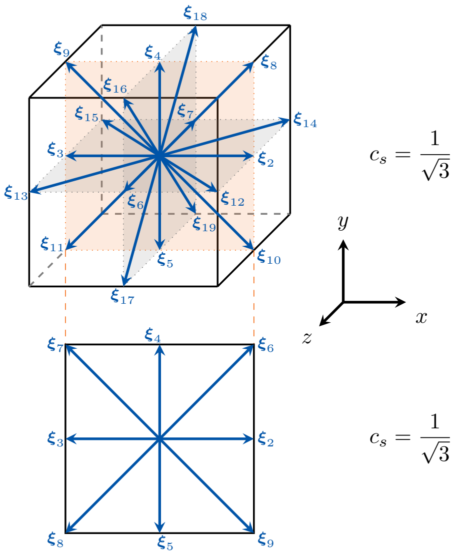
where ":" stands for the full contraction of indices between two tensors. The discrete Hermite polynomials $\boldsymbol{\mathcal{H}}_i^{(n)} = \boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi}_i)$ are defined as:

$$\boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi}) = \frac{(-c_s^2)^n}{w(\boldsymbol{\xi})} \frac{\partial^n w}{\partial \boldsymbol{\xi}^n} \ \text{ where } \ w(\boldsymbol{\xi}) = \frac{1}{(2\pi c_s^2)^{D/2}} \exp\left(-\frac{||\boldsymbol{\xi}||^2}{2c_s^2}\right), \tag{15}$$

and $\boldsymbol{a}_{eq}^{(n)}$ are the Hermite equilibrium moments which correspond to the projection of the Maxwell-Boltzmann distribution function onto the n-th Hermite polynomial. Consequently, Equation (12) is satisfied provided $Q \geq 2N$ where Q is the order of quadrature of the lattice.

The most common sets of discrete velocities $\{\boldsymbol{\xi}_i\}_{i\in\llbracket 1;q\rrbracket}$, often referred to as a DdQq lattices, where d is the spatial dimension and q the number of discrete velocities, have an order of quadrature $Q = 5$. In this case, it can be shown that $N \leq 2$ which leads in some discrepancies with respect to the macroscopic Navier-Stokes equations [36]. While the mass conservation equation is recovered without any error, a $\mathcal{O}(Ma^3)$ error appears is the momentum equations. In addition, the energy equation is wrong leading to an isothermal assumption by setting $T = T_0$ where T_0 is a constant reference temperature. Consequently, the equation of state reduces to $p = \rho c_s^2 = \rho R T_0$. For this reason, the lattice Boltzmann method is said to be restricted to isothermal and weakly compressible flows.

For this study, two lattices will be considered. The D2Q9 lattice is used for the von Neumann stability analysis while the usual D3Q19 lattice, implemented in ONERA's in-house LB solver FastLBM, is employed for all the numerical computations and for the cost analysis. Figure 2 represents both lattices along with their defining parameters.



$c_s = \dfrac{1}{\sqrt{3}}$

D3Q19			
i	1	$[\![2;7]\!]$	$[\![8;19]\!]$
$\boldsymbol{\xi}_i$	(0,0,0)	$(\pm1,0,0)$ $(0,\pm1,0)$ $(0,0,\pm1)$	$(\pm1,\pm1,0)$ $(\pm1,0,\pm1)$ $(0,\pm1,\pm1)$
w_i	1/3	1/18	1/36

$c_s = \dfrac{1}{\sqrt{3}}$

D2Q9			
i	1	$[\![2;5]\!]$	$[\![6;9]\!]$
$\boldsymbol{\xi}_i$	(0,0)	$(\pm1,0)$ $(0,\pm1)$	$(\pm1,\pm1)$
w_i	4/9	1/9	1/36

Figure 2: D2Q9 and D3Q19 velocity sets. The square (resp. 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 square (resp. cube). Each discrete velocity $\boldsymbol{\xi}_i$ is expressed in its non-dimensional form.

The lattice Boltzmann scheme [33] 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 $\boldsymbol{\xi}_i$ characteristic ensuring an exact advection step and a direct link between the grid and time step through $\Delta x = |\boldsymbol{\xi}_i|\Delta t$. On the other hand, a trapezoidal integration rule is employed for the right-hand side (RHS) collision term. This strategy, leads to:

$$g_i(\boldsymbol{x} + \boldsymbol{\xi}_i \Delta t, t + \Delta t) = g_i(\boldsymbol{x}, t) + \Delta t \Omega_i(\boldsymbol{x}, t) \tag{16}$$

where $\{g_i\}_{i\in[\![1;q]\!]}$ are the modified distribution functions so as to ensure an explicit formulation of the algorithm [37]. They are related to the original distribution functions $f_i(\boldsymbol{x}, t)$ through the relation $g_i(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) - \frac{\Delta t}{2}\Omega_i(\boldsymbol{x}, t)$ which implies $g_i^{eq}(\boldsymbol{x}, t) = f_i^{eq}(\boldsymbol{x}, t)$. In the case of the BGK collision model, the relaxation time also becomes $\overline{\tau} = \tau + \frac{\Delta t}{2}$.

The lattice Boltzmann method is classically applied in a non-dimensional form by setting Δt (resp. Δx) as the characteris time (resp. length) scale. Physical quantities of interest can then be retrieved with the physical values of Δt and Δx. For instance, the speed of sound c_0 and the the viscosity are given by:

$$\Delta t = \frac{c_s \Delta x}{c_0} \quad \text{and} \quad \nu = c_0^2 \left(\overline{\tau} - \frac{1}{2}\right). \tag{17}$$

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 [38, 39].

Despite its simplicity, the basic lattice Boltzmann method with the single relaxation time BGK collision model suffers from stability issues especially in the low viscosity regime (i.e. at high Reynolds number) [5]. These issues have been attributed to interactions between so called "ghost-modes" arising from the space and time discretisation of Equation (8) [4, 40, 6]. To alleviate this problem, a great number of collision models have been proposed such as Multiple Relaxation Times (MRT) operators [5, 41, 42], entropic LBMs [43], and regularised approaches [44, 45, 46]. Some authors also suggest to employ selective filters [47, 48] in order to remove the high wave number instabilities without affecting the large scale dynamics. In the following section, two regularised collision models which are implemented is FastLBM are presented, namely the recursive regularised collision model [46] and the hybrid recursive regularised collision model [49].

2.2.1. *Principle of regularised collision models*

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, the distribution functions are reconstructed before each collision step as:

$$g_i^{reg} = g_i^{eq} + g_i^{(1),reg}, \tag{18}$$

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:

$$\Omega_i = g_i^{eq} + \left(1 - \frac{1}{\tau}\right) g_i^{(1),reg} \tag{19}$$

Regularised models differ in the way they calculate the first-order contribution $g_i^{(1),reg}$. Most of the time, the latter is expanded on the basis of Hermite polynomials as it is the case for the equilibrium distribution functions. Consequently, one has:

$$g_i^{(1),reg} = w_i \sum_{n=2}^{N_r} \frac{1}{c_s^{2n} n!} \boldsymbol{a}_1^{(n)} : \boldsymbol{\mathcal{H}}_i^{(n)}, \tag{20}$$

where $\boldsymbol{a}_1^{(n)}$ is the nth-order off-equilibrium Hermite coefficient.

2.2.2. *Recursive regularised collision model*

The original model, introduced by Latt and Chopard [44] relies on the observation that only the second-order off-equilibrium Hermite coefficient is required to retrieve the Navier-Stokes dynamics. Therefore, Equation (20) reduces to:

$$g_i^{(1),reg} = w_i \frac{1}{2c_s^4} \boldsymbol{a}_1^{(2)} : \boldsymbol{\mathcal{H}}_i^{(2)}, \tag{21}$$

where $\boldsymbol{a}_1^{(2)}$ is computed by projecting the off-equilibrium distribution functions on the basis second-order tensor $\boldsymbol{\mathcal{H}}_i^{(2)}$:

$$\boldsymbol{a}_1^{(2)} = \sum_{i=1}^{q} \boldsymbol{\mathcal{H}}_i^{(2)} \left(g_i - g_i^{eq}\right). \tag{22}$$

Later, Malaspinas [46] proposed to extend the regularisation procedure of Equation (22) by reconstructing as many off-equilibrium Hermite coefficients as possible, at least up to an order N_r. Thanks to a Chapman-Enskog expansion and some algebra, a recursive relation between the off-equilibrium coefficients is determined:

$$a^{(n)}_{1,\alpha_1...\alpha_n} = u_{\alpha_n} a^{(n-1)}_{1,\alpha_1...\alpha_{n-1}} + \sum_{i=1}^{n-1} u_{\alpha_1}...u_{\alpha_{n-2}} a^{(2)}_{1,\alpha_i\alpha_n} \quad \text{for } n \geq 3. \tag{23}$$

The recursive relation is initialised by providing $\boldsymbol{a}_1^{(2)}$ through Equation (22). In the following, this model will be referred to as the recursive regularised collision model at order N_r (RRN_r) where $N_r = 3$ or $N_r = 4$ (only for the D2Q9 lattice).

2.2.3. 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 $\boldsymbol{a}_1^{(2)}$ is computed [49]. In the HRR framework, this tensor is hybridised. It is decomposed into a linear combination of a regularised part (see Equation (22)) and a finite difference part (FD). Indeed, the Chapman-Enskog expansion links the first-order off-equilibrium distribution to the deviatoric stress tensor $\boldsymbol{S} = \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right)$. Thereby, the off-equilibrium coefficient $\boldsymbol{a}_1^{(2)}$ is given by:

$$\boldsymbol{a}_1^{(2)} = \sigma \left[\sum_{i=1}^{q} \boldsymbol{\mathcal{H}}_i^{(n)} \left(g_i - g_i^{(0)}\right) \right] + (1-\sigma)\left[-\rho\bar{\tau}c_s^2 \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right)\right] \tag{24}$$

where $0 \leq \sigma \leq 1$ is a user-tuned parameter to control the amount of hyper-viscosity added to the model [49]. The spatial derivatives of the velocity field present in Equation (24) are evaluated with second-order centered finite differences. The higher-order off-equilibrium coefficients are then computed recursively using Malaspinas' recursive formula (23).

3. Linear Stability Analyses

As a first step towards the comparison between the lattice Boltzmann method and traditional Navier-Stokes schemes, linear stability analyses are conducted in the von Neumann formalism. Such methodology has already been introduced by Marié *et al.* [4] to compare the spectral properties of optimized Navier-Stokes schemes dedicated to computational aeroacoustics with two LB models. In this section, it is proposed to extend the spectral analysis by considering regularised collision operators for the lattice Boltzmann models and by investigating the isotropy properties of both the Navier-Stokes and lattice Boltzmann schemes.

For the sake of simplicity, the von Neumann analysis is performed in two dimensions of space. However, numerical simulations are introduced in Section 4 to validate the results in three dimensions of space.

3.1. Exact plane wave solutions

Before diving into the LSA of the Navier-Stokes and lattice Boltzmann schemes, the 2D linearised Navier-Stokes equations are solved for plane wave solutions in order to get their theoretical dispersion and dissipation relations. Thereby, the general principle of the von Neumann analysis is introduced.

To be consistent with the underlying physics of the lattice Boltzmann method, the analysis is restricted to isothermal flows. In this case, the Navier-Stokes equations can be written as:

$$\left\{ \begin{array}{lcll} \dfrac{\partial \rho}{\partial t} + \dfrac{\partial}{\partial x_j}(\rho u_j) & = & 0 & \\ \dfrac{\partial \rho u_i}{\partial t} + \dfrac{\partial}{\partial x_j}(\rho u_i u_j) & = & -\dfrac{\partial p}{\partial x_i} + \dfrac{\partial}{\partial x_j}(\tau_{ij}) & \quad i = 1,2 \end{array} \right. \tag{25}$$

where ρ, u_i, p denote the fluid density, the i-velocity component, and the pressure respectively. In its most general form, the shear stress tensor $\boldsymbol{\tau}$ involves a bulk viscosity coefficient ζ:

$$\tau_{ij} = \rho\nu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \right] + \rho\zeta \frac{\partial u_k}{\partial x_k}\delta_{ij}. \tag{26}$$

Even though the flow is considered as isothermal, the Navier-Stokes Equations (25) still have to be completed by an energy conservation equation which, when expressed in terms of internal energy, reads as:

$$\frac{\partial \rho e}{\partial t} + \frac{\partial}{\partial x_j}(\rho e u_j) = -p\frac{\partial u_i}{\partial x_i} + \tau_{ij}\frac{\partial u_i}{\partial x_j}. \tag{27}$$

The system composed by Equations (25) and (27) can be recast in the dynamical system form as:

$$\frac{\partial \mathbf{q}}{\partial t} = \mathcal{L}(\mathbf{q}). \tag{28}$$

where $\mathbf{q}$ is the state vector of the conservative variables and $\mathcal{L}$ is the non-linear differential operator of the isothermal Navier-Stokes equations.

In order to perform the linear stability analysis, the state vector is perturbed around a base flow as:

$$\mathbf{q} = \overline{\mathbf{q}} + \mathbf{q}' \tag{29}$$

where $\overline{\mathbf{q}}$ denotes the base flow which is steady and homogeneous in all direction of space (i.e. $\partial_t \overline{\mathbf{q}} = \partial_{x_1} \overline{\mathbf{q}} = \partial_{x_2} \overline{\mathbf{q}} = 0$), and $\mathbf{q}'$ denotes the state vector perturbations such that $\mathbf{q}' \ll \overline{\mathbf{q}}$. By substituting Equation (29) into the the dynamical system form of the NS equations and by neglecting high order perturbations, one gets the semi-discrete form of the linearised Navier-Stokes equations:

$$\frac{\partial \mathbf{q}}{\partial t} = \mathcal{J}\mathbf{q}'. \tag{30}$$

where $\mathcal{J}$ is the jacobian matrix of the Navier-Stokes equations.

Then a Fourier transform in both space and time is performed, allowing the perturbations to be studied as the following global modes:

$$\mathbf{q}' = \widehat{\mathbf{q}} \exp\left(\mathrm{i}\left(\boldsymbol{k} \cdot \boldsymbol{x} - \omega t\right)\right) \tag{31}$$

where $\mathrm{i}^2 = -1$, $\widehat{\mathbf{q}}$ is the complex amplitude of the perturbations, $\boldsymbol{k}$ is the wave number and ω is the pulsation of the mode. In the classical framework of temporal analysis, $\boldsymbol{k} \in \mathbb{R}^2$ and $\omega \in \mathbb{C}$.

Finally, injecting Equation (31) into Equation (30), 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:

$$\omega\widehat{\mathbf{q}'} = \mathcal{J}\widehat{\mathbf{q}'}. \tag{32}$$

In the common von Neumann approach, the eigenvalue problem of Equation (33) is solved for each value of $\boldsymbol{k}$. The complex eigenvalues ω give access to the propagation speed $\mathrm{Re}(\omega)$ and dissipation rate $\mathrm{Re}(\omega)$ of each of the linear modes stemming from the set of equations defined by $\mathcal{L}$.

By applying this general methodology to Equations (25) and (27) (see Appendix A for details), the eigenvalue problem of the exact plane wave solution of the isothermal NS equations is:

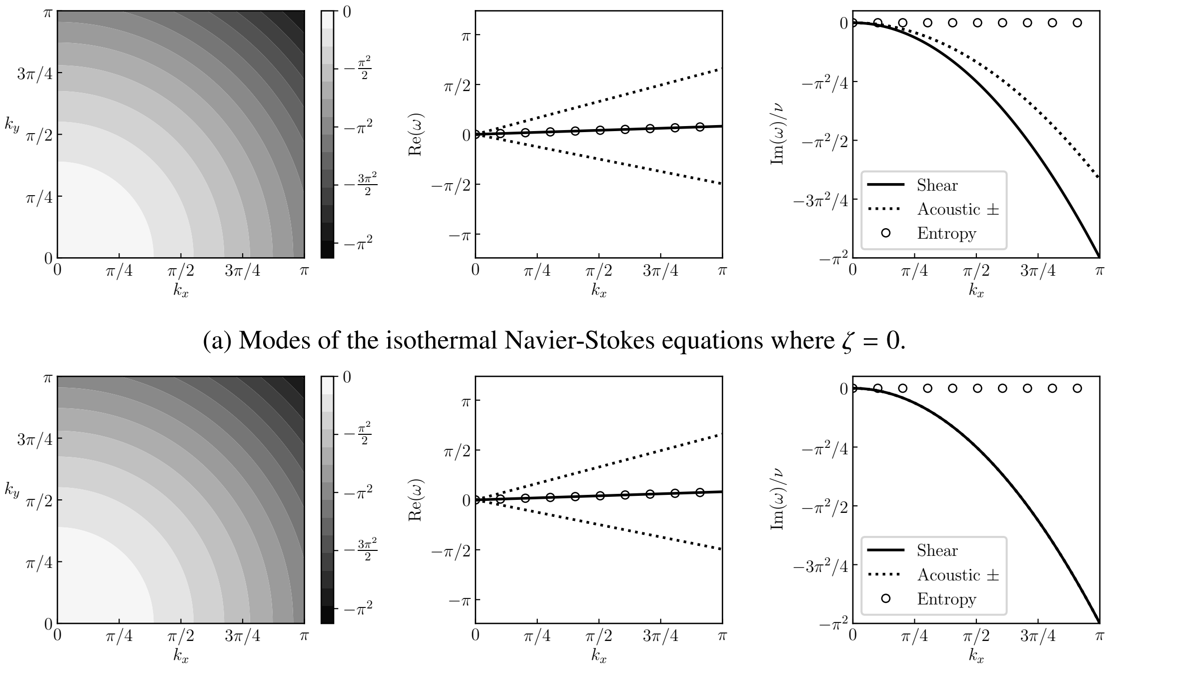
$$\omega\hat{\mathbf{U}}' = \mathbf{M}^{\mathrm{NS},\dagger}\hat{\mathbf{U}}' \tag{33}$$

where $\mathbf{M}^{\mathrm{NS},\dagger} = k_{x_1}\mathbf{M}_{x_1} + k_{x_2}\mathbf{M}_{x_2}$ in which the remaining spatial derivatives were replaced using: $\partial_{x_i} = \mathrm{i}k_i$. The solution of Equation (33) gives the linear modes of the isothermal Navier–Stokes equations: two acoustic modes (one upstream $\omega_{\mathrm{Ac}+}$ and one downstream $\omega_{\mathrm{Ac}-}$), one shear mode ω_{sh} in 2D (two in 3D) and one entropy mode ω_{entr}. Their general expressions are given by:

$$\begin{aligned}\omega_{\mathrm{Ac}\pm} &= \boldsymbol{k}\cdot\overline{\boldsymbol{u}} \pm \|\boldsymbol{k}\|c_0 + i\left(\frac{2}{3}\nu + \frac{1}{2}\zeta\right)\|\boldsymbol{k}\|^2 \\ \omega_{\mathrm{sh}} &= \boldsymbol{k}\cdot\overline{\boldsymbol{u}} + i\nu\|\boldsymbol{k}\|^2 \\ \omega_{\mathrm{entr}} &= \boldsymbol{k}\cdot\overline{\boldsymbol{u}}\end{aligned} \tag{34}$$

It can be seen from Equation (34) that he shear wave propagates at the mean flow velocity $\overline{\boldsymbol{u}}$, whereas the acoustic waves propagate at $\overline{\boldsymbol{u}} \pm c_0$. Moving to the dissipation rate, the attenuation 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 by the viscosity through ν, and compression/dilation effects through the bulk viscosity ζ. Because of the isothermal hypothesis, the entropy mode is reduced to a non-dissipative wave propagating with the shear mode. Therefore, in the following, the entropy mode will be ignored.

The solutions of Equation (34) will be used as references for the dispersion and dissipation analysis of the different schemes. In Figure 3, the four modes of Equation (34) are plotted in the case of a nil bulk viscosity (as it is the case for NS schemes) and in the case of a non-vanishing bulk-viscosity (for LB models, it can be shown that $\zeta = 2/3\nu$ [40]). As one can see on the dissipation map of the shear mode, the Navier-Stokes equations are isotropic. However, as will be seen in the following, this property is lost when discretising the equations in both space and time.



(a) Modes of the isothermal Navier-Stokes equations where $\zeta = 0$.

(b) Modes of the isothermal Navier-Stokes equations where $\zeta = 2/3\nu$.

Figure 3: Linear stability analysis of the exact Navier-Stokes equations for an uniform mean flow at $M = 0.2$ with a propagation angle of $\theta = 45°$ and $\nu = \nu_{air}$. Left: dissipation map of the shear mode (indicating the isotropy of the NS equations), Center: Dispersion curve for $k_y = 0$ and Right: Dissipation curve for $k_y = 0$.

3.2. *von Neumann analysis of space & time discrete Navier-Stokes schemes*

The general methodology of the von Neumann analysis is now applied to traditional Navier-Stokes schemes. While in classical approaches, dispersion and dissipation are studied separately for the space and time discretisations, their combined effect is investigated here. This is necessary for the comparison with lattice Boltzmann schemes since the space and time discretizations cannot be distinguished for the latter. The methodology, which was introduced by Marié *et al.* [4], is briefly recalled here.

Space discretisation. For the sake of simplicity, only centered schemes are studied in the present work. Even though the "Sensor" scheme implemented in ONERA's finite-volume Navier-Stokes solver isn't a centered scheme, it reduces to a second-order centered scheme in the absence of oscillations in the primitive variables. Thereby, centered schemes correspond to the "best case scenario" and provide a lower-band estimate of its dissipative behavior.

A general approximation of the spatial derivatives by a centered scheme relying on a $2N+1$ point stencil for a given quantity $\mathbf{q}$ can be written as:

$$\frac{\partial \mathbf{q}}{\partial x_i} = \frac{1}{\Delta x_i} \sum_{j=-N}^{N} a_j \mathbf{q}(\boldsymbol{x} + j\boldsymbol{\Delta x}_i) + O(\Delta x^{2N}) \tag{35}$$

where a_j are the coefficients related to the specific centered scheme and Δx_i is the grid step in the i direction. The standard coefficients of the scheme used in this paper age given in Table 1. Three schemes are considered: a classic second-order one (which will be confronted to the "Sensor" scheme), a fourth-order "dispersion relation preserving" scheme developed by Tam and Webb [50] and the optimized sixth-order Bogey scheme [51].

Scheme	2$^{\text{nd}}$-order	4$^{\text{th}}$-order o	6$^{\text{th}}$-order o
a_1	0.5	0.79926643	0.907646591371
a_2	-	-0.18941314	-0.337048393268
a_3	-	0.02651995	0.133442885327
a_4	-	-	-0.045246480208
a_5	-	-	0.011169294114
a_6	-	-	-0.001456501759

Table 1: Coefficients of centered schemes ($a_0 = 0$ and $a_j = -a_j$).

By applying the general von Neumann stability analysis Equation (35) becomes:

$$\frac{\partial \mathbf{q}'}{\partial x_i} = \widehat{\mathbf{q}} \underbrace{\frac{1}{\Delta x_i} \sum_{j=-N}^{N} a_j \exp\left(\mathrm{i}\ j\boldsymbol{k} \cdot \boldsymbol{\Delta x}_i\right)}_{\mathcal{K}_{\text{scheme},x_i}} + O(\Delta x^{2N}). \tag{36}$$

where the so-called equivalent wave-number $\mathcal{K}_{\text{scheme},x_i}$ corresponds to the right-hand side of Equation (36). By looking at the real and imaginary parts of $\mathcal{K}_{\text{scheme},x_i}$, one can access the dissipation and dispersion errors respectively, occasioned by the truncation error spatial scheme. The advantage of using centered scheme is that they induce zero dissipation.

Time discretisation. As in [4], this study is restricted to explicit Runge-Kutta time-stepping schemes. This is motivated by the fact that most computations in Section 4 use an explicit third-order Runge-Kutta scheme.

A p-step explicit Runge-Kutta (RK) method applied to Equation (28) can be expressed as:

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \sum_{j=1}^{p} \gamma_j \Delta t \mathcal{L}^j(\mathbf{q}^n), \tag{37}$$

where Δt is the time-step and $\mathcal{L}^j$ denotes the 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-oder one denoted by RK-3 and a sixth-order optimised one (RK-6 o) proposed by Bogey and Bailly [51]. Table 2 provides the coefficients γ_j for each scheme.

Scheme	RK-3	RK-6 o
γ_1	1	1
γ_2	0.5	0.5
γ_3	1/6	0.165919771368
γ_4	-	0.040919732041
γ_5	-	0.007555704391
γ_6	-	0.000891421261

Table 2: Coefficients of the Runge-Kutta schemes.

When applying the von Neumann analysis framework to the space- and time-discrete Navier-Stokes schemes, one gets the following general eigenvalue problem:

$$e^{-i\omega}\widehat{\mathbf{q}} = \left[\mathbf{I} + \sum_{j=1}^{p} \gamma_j \mathrm{CFL}^j \mathbf{\Lambda}^j\right] \widehat{\mathbf{q}}^n, \tag{38}$$

where

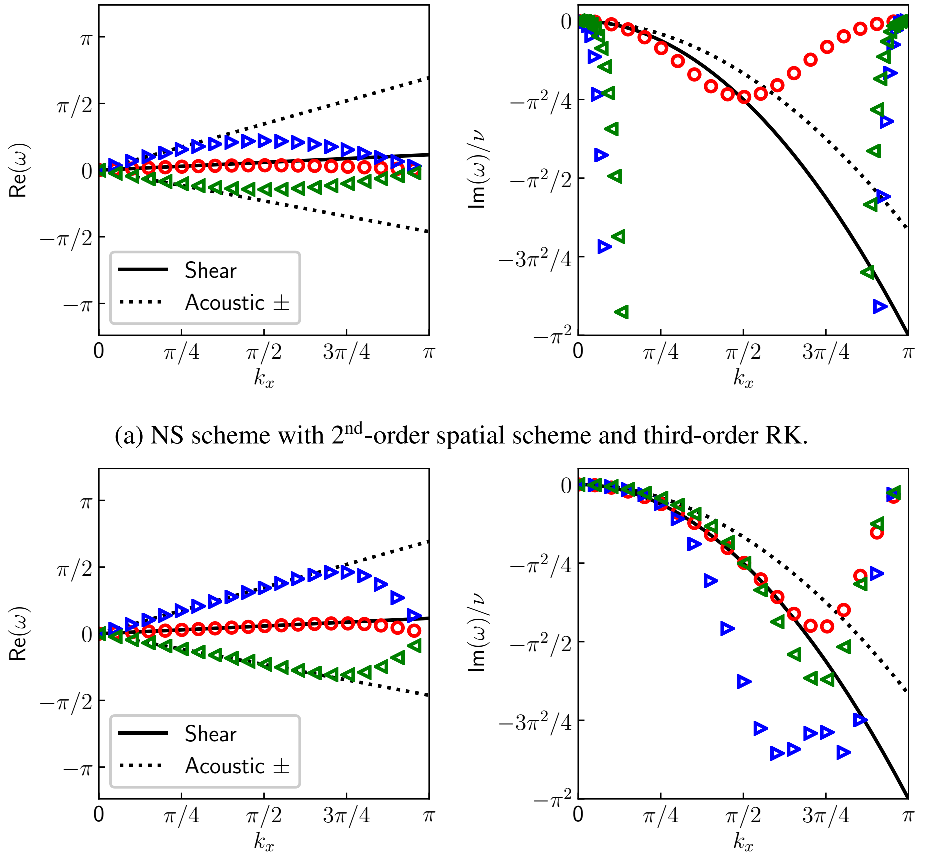
$$\mathbf{\Lambda} = -\frac{\Delta x}{c_0}\left[\mathcal{K}_{\mathrm{scheme},x_1}\mathbf{M}_{x_1} + \mathcal{K}_{\mathrm{scheme},x_2}\mathbf{M}_{x_2}\right]. \tag{39}$$

Note that $\mathcal{K}_{\mathrm{scheme},x_1}$ and $\mathcal{K}_{\mathrm{scheme},x_2}$ are also present in the definition of $\mathbf{M}_{x_1}$ and $\mathbf{M}_{x_2}$ since they involve derivatives in their generic expressions.

The eigenvalue problem of Equation (38) is solved numerically thanks to an in-house Python code, using the NumPy and SciPy libraries. 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 given step Δk in both directions which has been set to $\Delta k = 0.01$.

Figure 4 displays the propagation and dissipation curves of the space and time discrete Navier-Stokes schemes for horizontal plane waves (i.e $ky = 0$) with a mean flow at $\overline{\mathrm{Ma}} = 0.2$. The viscosity has been set to $\nu = 10^{-5}$ which is a commonly encountered value for air flow simulations. The numerical modes (represented by symbols is Figure 4) are compared to their theoretical counterparts (drawn in black lines). Note that the same color and symbols convention as in [7] is employed so as to ease the comparison of the present results with other existing studies.

For both schemes, the acoustic modes are clearly more dissipated than the shear mode which is in accordance with previous results [4]. In can be seen that perturbations having a wavenumber of π (i.e. two



(a) NS scheme with 2$^{\text{nd}}$-order spatial scheme and third-order RK.

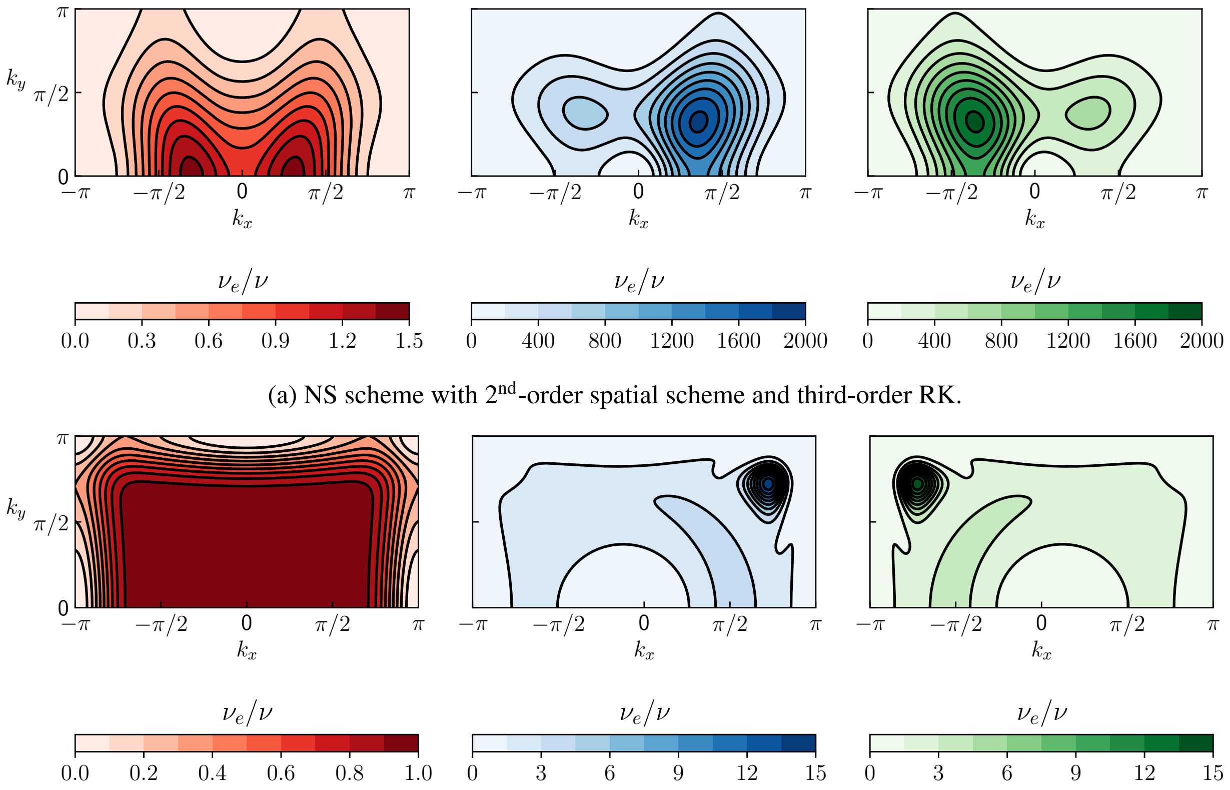
(b) NS scheme with the 6$^{\text{th}}$-order centered optimised spatial scheme and sixth-order RK.

Figure 4: Propagation (left) and dissipation (right) curves of the shear (○), Ac+ (▷) and Ac- (◁) modes for the space and time discrete Navier-Stokes schemes. The perturbations are superimposed to a mean flow at $\overline{\text{Ma}} = 0.2$ along the x-axis.

points per wavelength) are not propagated since Re(ω) vanishes for all the modes. In addition, for high wave numbers, an anti-dissipative behavior is evidenced. As expected, increasing the order of the space and time discretisations tends to improve the correspondance between the theoretical and numerical modes.

The information displayed in Figure 4 is of very limited use since perturbations are not always aligned with the main axes. To complete these analyses, spectral maps of the effective viscosity of each mode (shear, downstream and upstream acoustics) are displayed in Figure 5 for $\overline{\text{Ma}} = 0.2$. By definition, the effective viscosity is computed as $\nu_e = -\omega_i/||\boldsymbol{k}||^2/\nu$. Surprisingly, a rather anisotropic behavior is observed for both schemes on a broad range of wavenumbers. Inasmuch as centered schemes do not induce any numerical dissipation, the anisotropy is attributed to the coupling between the spatial and temporal discretisations. Nevertheless, by increasing the order of the discretisations, one retrieves a rather isotropic behavior for low wavenumbers. Moreover, the conclusions draw by the analysis of Figure 4 can be generalised. Indeed, regardless of the direction of propagation, the shear waves are far less dissipated than the acoustic ones. The anti-dissipative behavior is also recovered when the perturbations are under-resolved in a specific direction, i.e. when k_x and/or k_y are close to π.

In the light of the anisotropy of classical Navier-Stokes schemes, spectral maps of the eigenmodes of



(a) NS scheme with 2$^{\text{nd}}$-order spatial scheme and third-order RK.

(b) NS scheme with the 6$^{\text{th}}$-order centered optimised spatial scheme and sixth-order RK.

Figure 5: Effective viscosity plot ν_e/ν of the shear (left), Ac+ (center) and Ac- (right) modes. The perturbations are superimposed to a mean flow at $\overline{\text{Ma}} = 0.2$ along the x-axis.

the LB schemes have to be investigated to fully compare the spectral properties of both methods.

3.3. von Neumann analysis of lattice Boltzmann schemes

In the following, the linear stability analysis is applied to both the DVBE (8) and the LB scheme (16). While the first one allows to evaluate the impact of the velocity discretization on the macroscopic behavior, the second one highlights the influence of the space and time discretization on the spectral properties of the related LBM.

Analysis of the DVBE. Since the DVBE describes the evolution of the distribution functions, the state vector $\mathbf{q}$ is now defined as:

$$\mathbf{q}_f = (\widehat{f_1}, \widehat{f_2}, ..., \widehat{f_{q-1}}, \widehat{f_q}) \tag{40}$$

where q is the number of discrete velocities. In order to distinguish the state vector of Equation (40) with the one used in the NS formalism, the former is written with the letter f as subscript. This difference being noted, the general methodology introduced in Section 3.1 is applied in the same way.

By replacing Equations (29) and (31) into the dynamical form of the DVBE, the following eigenvalue problem of size q :

$$\omega\widehat{\mathbf{q}} = \mathbf{M}^{\mathrm{C}}\widehat{\mathbf{q}} \tag{41}$$

where the definition of the time evolution matrix $\mathbf{M}^{\mathrm{C}}$ is given in Appendix B. Equation (41) highlights one main difference between the LSA of NS and LB schemes. Indeed, for the Navier-Stokes schemes, the eigenvalue problem leads to 4 (resp. 5) eigenmodes. However, in the lattice Boltzmann framewok, the eigenvalue problem is of size q meaning that q eigenmodes are obtained. Owing to the fact that $q > 4$, the dynamics of the DVBE include more information than at the Navier-Stokes level.

In order to give a physical interpretation to modes resulting from the von Neumann analysis in the lattice Boltzmann framework, an extended von Neumann analysis methodology [6] is used. The latter relies on the study of the LB eigenvector $\hat{\mathbf{q}}$ and more particularly on its moments defined as:

$$\widehat{\rho} = \sum_{I=1}^{q} \widehat{f_i} \quad \text{and} \quad \widehat{\rho u} = \sum_{I=1}^{q} \xi_i \widehat{f_i} \tag{42}$$

The resulting macroscopic vector $\mathbf{V} = [\hat{\rho}, \widehat{\rho u}]^T$ 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 last years for a very large set of problems [7, 39, 52, 53].

Figure 6 displays the propagation and dissipation curves of the DVBE for horizontal plane waves (i.e $ky = 0$) with a mean flow at $\overline{\mathrm{Ma}} = 0.2$ and a viscosity of $\nu = 10^{-5}$. It can be seen that, based on the standard second-order equilibrium, the DVBE is able to recover the correct macroscopic behavior of the isothermal Navier-Stokes equations. Indeed, 3 hydrodynamic modes are identified while the 6 remaining ones, are said to be non-observable since they do not propagate any macroscopic information (they are also highly dissipated since $\mathrm{Im}(\omega)/\nu = -10^{-10}$). While the propagation speed of the shear and acoustic modes matches its theoretical value, deviations in their dissipation rate start appearing for non-vanishing mean flow (i.e. $\overline{\mathrm{Ma}} \neq 0$) as shown in Figure 6. Such deviations were already observed in [4] and are related to the $O(\mathrm{Ma}^3)$ errors in the momentum equation that stem from the velocity discretisation (see discussion in Section 2.2).

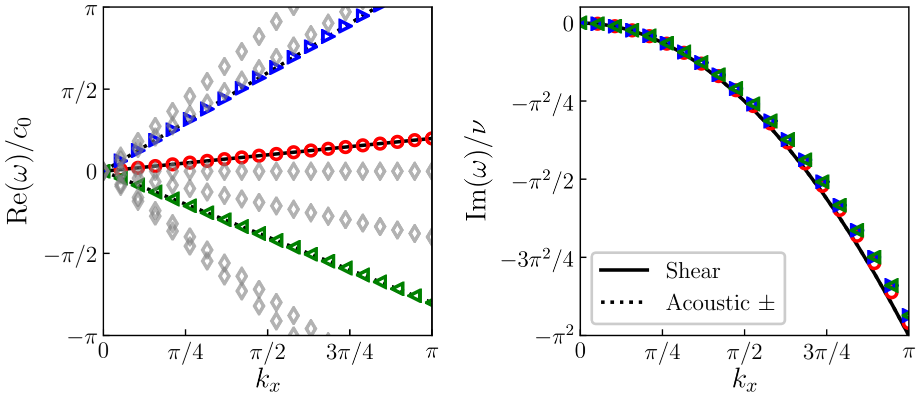


Figure 6: Propagation (left) and dissipation (right) curves of the shear (○), Ac+ (▷), Ac- (◁) and non-hydrodynamic modes (◊) for the discrete velocity Boltzmann equation. The perturbations are superimposed to a mean flow at $\overline{\mathrm{Ma}} = 0.2$ along the x-axis.

The curves in Figure 6 highlight the fact that, independently from the numerical discretisation, the DVBE recovers the Navier-Stokes linear behavior only in the limit of weakly compressible flows (i.e. Ma $\lesssim$ 0.3) for which the $O(\mathrm{Ma}^3)$ error term is seen as negligible. Nevertheless, in order to retrieve the right NS behavior, Feng *et al.* introduced a well designed correcting term ψ_i to restore the Galilean invariance of low-quadrature lattices [54]. By taking this correction into account in the time-advance matrix of the DVBE (see

Appendix B), the correct dissipative behavior is recovered as highlighted in Figure 7. In the following, the corrective term will be neglected since only low-mach number flows are studied. However, in the case of the HRR collision operator, the latter is mandatory to obtain stable results as discussed in [39]. Therefore, the HRR model will be denoted as HRR$_\psi$.

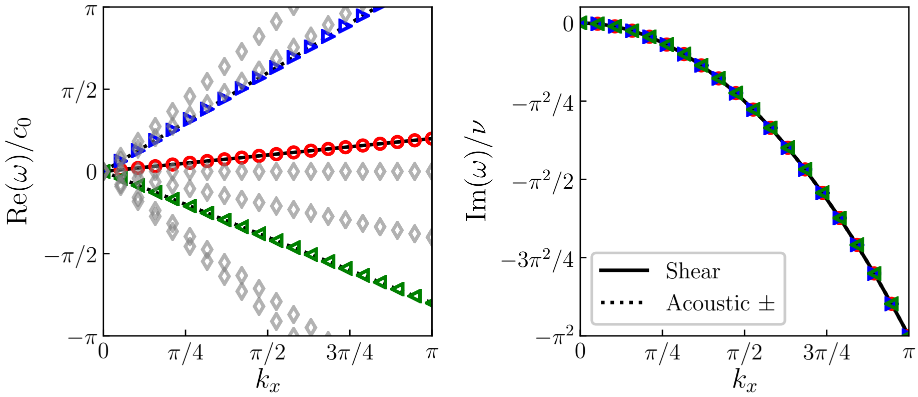


Figure 7: Propagation (left) and dissipation (right) curves of the shear (○), Ac+ (▷), Ac- (◁) and non-hydrodynamic modes (◊) for the corrected discrete velocity Boltzmann equation. The perturbations are superimposed to a mean flow at $\overline{\mathrm{Ma}} = 0.2$ along the x-axis.

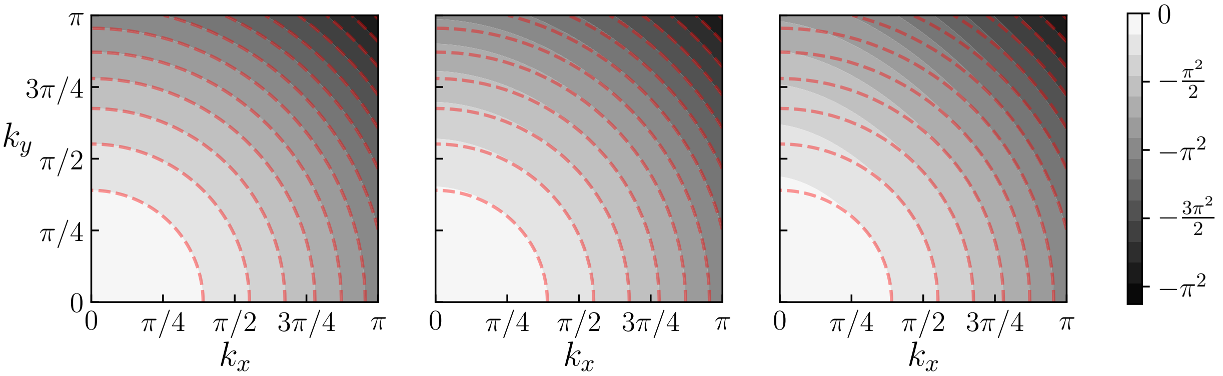
To conclude this preliminary discussion, the isotropy of the discrete velocity Boltzmann equation is studied. As shown in Figure 3, the linear modes of the Navier-Stokes equations are perfectly isotropic regardless of the mean flow Mach number. Figure 8 displays the dissipation plane $\mathrm{Im}(\omega)/\nu$ for different mean flow Mach numbers ranging from 0. to 0.4 by an increment of 0.2. As indicated by the dashed red lines representing the Navier-Stokes behavior, increasing the mean flow Mach number induces isotropy defects on the linear modes of the DVBE. Such result is surprising since the D2Q9 lattice used to discretise the velocity space is isotropic. It can be shown that higher-order lattices allow to recover an isotropic behavior regardless of the mean flow Mach number [55]. However, as only low-mach number flows ($M \leq 0.2$) are of concern is this study, the isotropy defect of the DVBE on D2Q9 lattices can be ignored.

Spectral properties of lattice Boltzmann schemes. The von Neumann analysis of the space and time discrete lattice Boltzmann schemes is performed by replacing Equations (29) and (31) into the dynamical form of the lattice Boltzmann equation. Consequently, the following eigenvalue problem is obtained:

$$e^{-i\omega}\widehat{\mathbf{q}} = \mathbf{M}^{\mathrm{D}}\widehat{\mathbf{q}} \tag{43}$$

where the time evolution matrix $\mathbf{M}^{\mathrm{C}}$ depends on the collision operator. For the BGK, RRN_r and HRR$_\psi$ collision models, its general expression is derived in Appendix B. As for the DVBE q eigenmodes are obtained.

Figure 9 displays the propagation $\mathrm{Re}(\omega)$ and dissipation $\mathrm{Im}(\omega)$ curves for perturbations traveling in the horizontal direction ($k_y = 0$) for $\nu = 10^{-5}$ and a horizontal mean flow at $\overline{\mathrm{Ma}} = 0.2$. Three different collision models are studied, namely the BGK, RR3 and HRR$_\psi$. For the latter, two values of σ are considered : $\sigma = 0$ and 0.995 which is commonly used for industrial applications [52]. Contrarily to the results of the von Neumann analysis of the DVBE, five type of modes are represented. In addition to the theoretical Navier-Stokes modes and to the non-hydrodynamic ones, non-identified modes arise. The latter are found to be a



(a) $\overline{\mathrm{Ma}} = 0$ (b) $\overline{\mathrm{Ma}} = 0.2$ (c) $\overline{\mathrm{Ma}} = 0.4$

Figure 8: Dissipation map of the shear mode of the discrete velocity Boltzmann equation in the (k_x, k_y) plane for an increasing Mach number. (a) $\overline{\mathrm{Ma}} = 0$, (b) $\overline{\mathrm{Ma}} = 0.2$, and (c) $\overline{\mathrm{Ma}} = 0.4$.

linear superposition of physical waves (shear and acoustics) according to when projecting the macroscopic vector $\mathbf{V} = [\hat{\rho}, \widehat{\rho u}]^T$ onto the Navier-Stokes one [6].

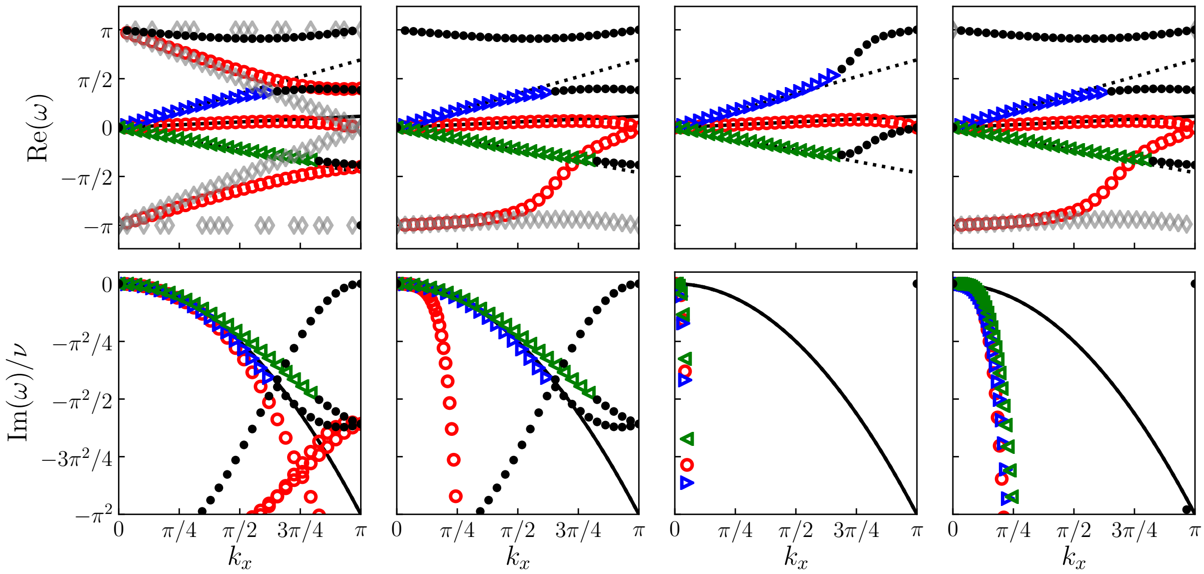


Figure 9: Propagation (top) and dissipation (bottom) curves of the shear (○), Ac+ (▷), Ac- (◁), non-hydrodynamic modes (◊) and non-identified modes (•) for the corrected discrete velocity Boltzmann equation. The perturbations are superimposed to a mean flow at $\overline{\mathrm{Ma}} = 0.2$ along the x-axis.

From Figure 9 it can be seen that changing the collision model has a non-negligible influence on the spectral properties of the LBM. The first thing to notice is the mode-filtering property of the regularised collision models [7]. Indeed, even though all the eigenvalues problems are of size $q = 9$, both the RR3 and HRR$_\psi$ with $\sigma = 0.995$ only six modes are present for these collision models. It is interesting to notice that when $\sigma = 0$, the HRR$_\psi$ model filters out all non-hydrodynamics modes. Another specificity of the LBM is that there are multiple modes carrying a shear information (three for the BGK model and 2 for RR3) but they are highly dissipated, at least for $k_x \leq \pi/2$.

Regarding the dissipative behavior of the LBM, it becomes clear that the BGK collision model is far less dissipative than regularised collision models [7, 39]. The RR3 collision operator seems to be suited for acoustic computations since the acoustic modes are dissipated at the same rate as for the BGK case. Nevertheless, if there is a non-vanishing mean flow, the shear mode is highly dissipated. This property is shared by all regularised collision models and is attributed to a hyperviscous degeneracy [39]. Moving to the HRR$_\psi$ collision model, it becomes clear that the value of $\sigma = 0$ is of no practical use since all the modes are experiencing an over-dissipation. Yet, by increasing the value of σ towards unity, one tends to recover the behaviour of the recursive regularised models where the acoustic modes are slightly less dissipated than the shear one.

To complete these analyses, as for the NS schemes, spectral maps of the effective viscosity of shear and acoustics modes are displayed in Figure 10. The same methodology as in [7] is applied: when several modes eventually carry a similar macroscopic information, only the one of maximal amplification rate ω_i is displayed. Moreover, contrarily to the effective viscosity plots of the NS schemes, these plots evidence some grey regions indicating no physical information was identified (see the non-identified modes in Figure 9). The plot is separated in two parts each having its own color scale. This is motivated by the fact that the BGK collision operator is far less dissipative than regularised models.

For the BGK model at $\overline{\mathrm{Ma}} = 0.2$ a rather anisotropic behavior is observed. Such observations were already made in [7]. However, in comparison to the effective viscosity maps of the NS schemes, one can see that the LBM-BGK is really suited for acoustic applications since it has an effective viscosity about four time smaller than the one of sixth-order optimised NS schemes. Yet, when it comes to the shear mode, it can be seen that the conclusions are not completely the same.

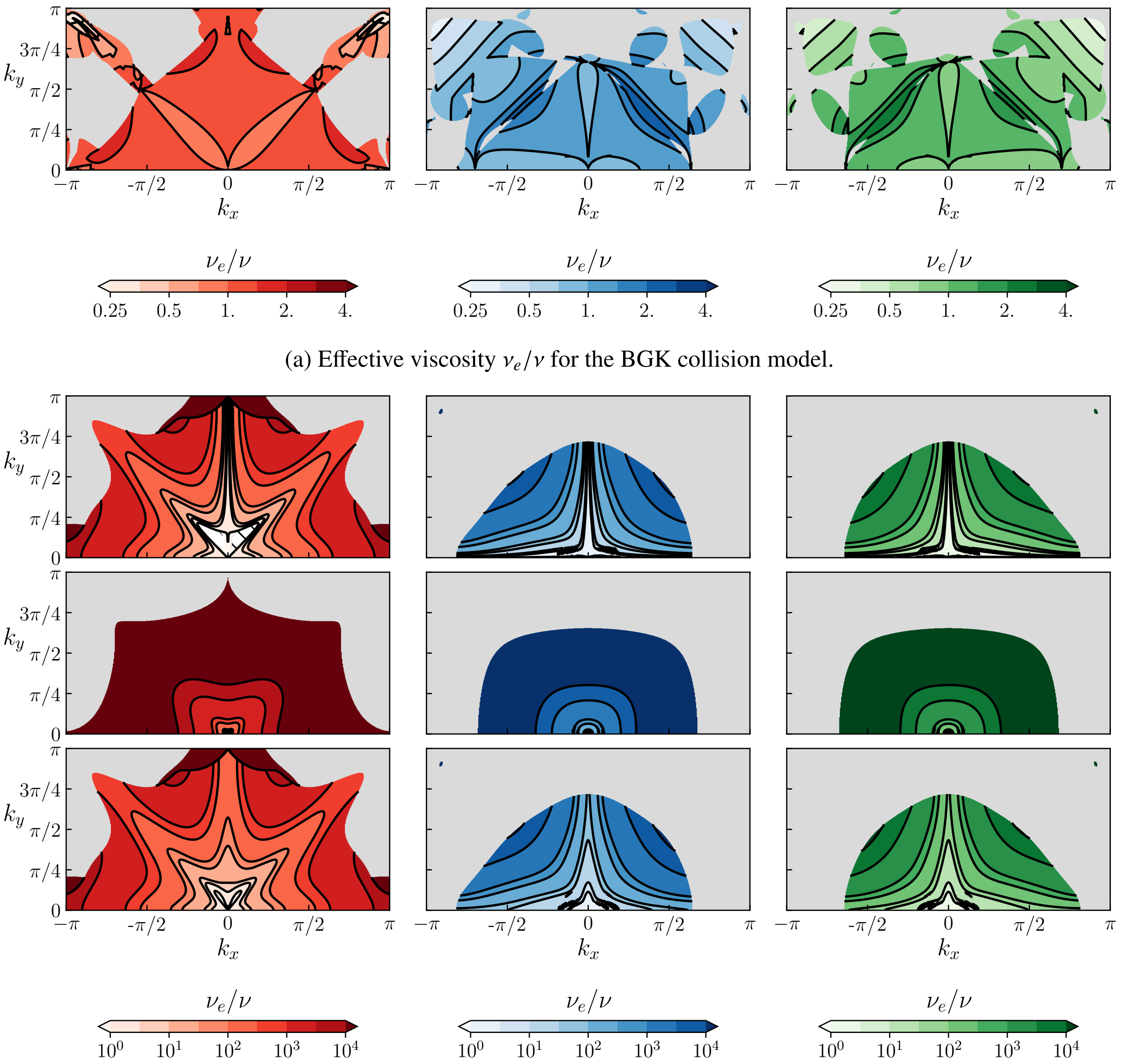
On the other hand, the effective viscosity plots of regularised collision models immediately reports an overall more dissipative behavior on all the physical modes. Although the RR3 and HRR$_\psi$ with $\sigma = 0.995$ schemes are highly anisotropic, some privileged directions are observed and seem to be aligned with the lattice main directions (including diagonals for the shear mode). Surprisingly, in the limit of $\sigma = 0$, the HRR$_\psi$ model has a rather isotropic behavior which might be related to the fact that only the finite difference part is involved into the computation of the non-equilibrium moments of the regularised distribution functions. Nevertheless, the 1D results are confirmed, the latter collision model is of very little interest in practice since the hyperviscous degeneracy tends to erase the benefits in terms of stability. Therefore, in the following, this model will be left aside for the comparison with Navier-Stokes schemes.

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

Now that the theoretical framework of the von Neumann analysis of both the lattice Boltzmann and Navier-Stokes schemes is introduced, the focus is made on the comparison of their dispersive and dissipative properties. In the following, the comparison criteria is the error committed on Re(ω) and Im(ω) as a function of the wavenumber [4]:

$$\left\{ \begin{array}{lcl} \mathrm{Err}^{\mathrm{Re}}(\boldsymbol{k}) & = & |\mathrm{Re}(\omega^{th}) - \mathrm{Re}(\omega)| \\ \mathrm{Err}^{\mathrm{Im}}(\boldsymbol{k}) & = & |\mathrm{Im}(\omega^{th}) - \mathrm{Im}(\omega)| \end{array} \right. \tag{44}$$

where ω^{th} refers to the solutions of the exact linearised Navier-Stokes equations, and ω refers to the exact solutions of the eigenvalue problems of both the NS and LB schemes. These criteria will be computed for the same CFL number. Since the lattice Boltzmann models operate at a fixed CFL number given by CFL $= 1/\sqrt{3}$, the same value is chosen for the Navier–Stokes schemes. Other values of the CFL number



(a) Effective viscosity ν_e/ν for the BGK collision model.

(b) Effective viscosity ν_e/ν for the RR3 (top row) HRR$_\psi$ with $\sigma = 0$ (middle row) and HRR$_\psi$ with $\sigma = 0.995$ (bottom row) collision models.

Figure 10: Effective viscosity plot ν_e/ν of the shear (left), Ac+ (center) and Ac- (right) modes. The perturbations are superimposed to a mean flow at $\overline{\mathrm{Ma}} = 0.2$ along the x-axis.

have been tested without changing the conclusions. Moreover, two mean flows are investigated : $\overline{\mathrm{Ma}} = 0$ and $\overline{\mathrm{Ma}} = 0.2$. The viscosity is set to $\nu = 10^{-5}$.

455 Figure 11 compares the dispersion and dissipation errors of the lattice Boltzmann and Navier–Stokes schemes for $\overline{\mathrm{Ma}} = 0$. Only the Ac+ mode is represented since the result is exactly the same for the Ac- mode.

First, we can note that the all the LB models have the same dispersion error which is between a global second order scheme and an optimized third order in space with a 3-step Runge–Kutta in time. However,

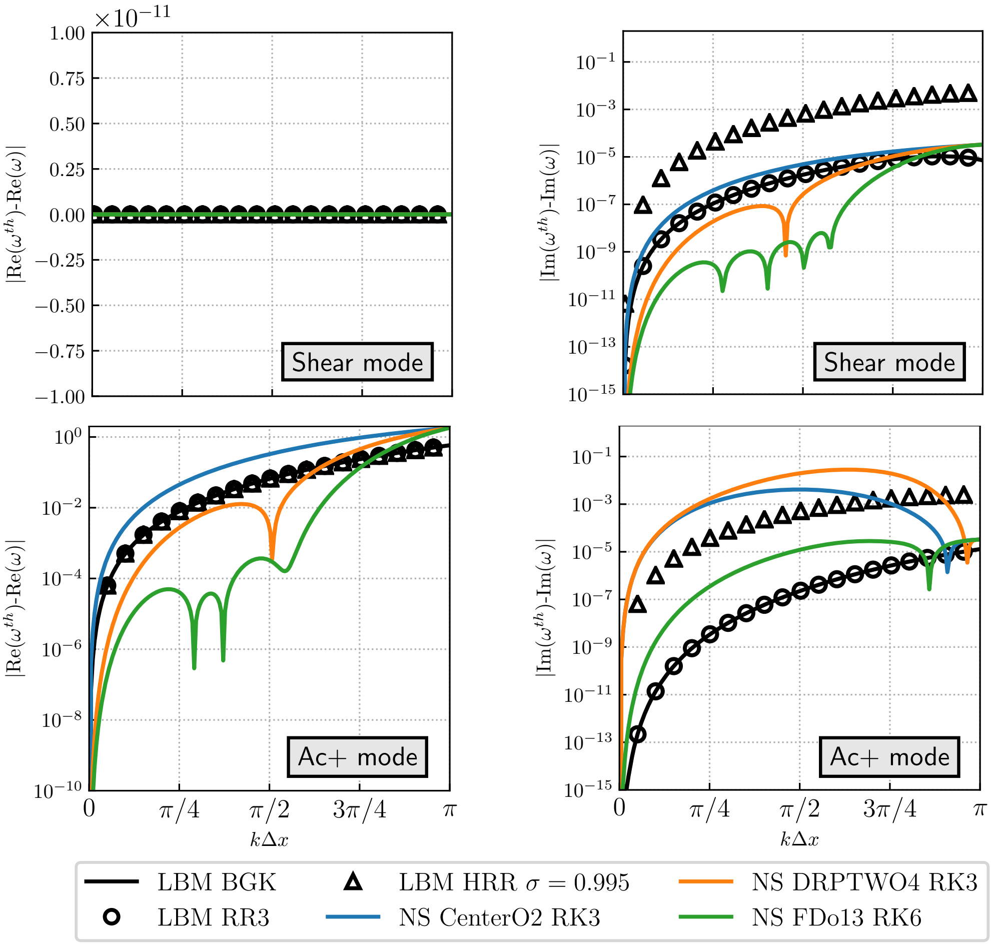


Figure 11: Dispersion (left), and dissipation (right) of the lattice Boltzmann and Navier–Stokes schemes for $\overline{\mathrm{Ma}} = 0$ and $\nu = 10^{-5}$.

the dissipation error curves highlight some discrepancies between the LB schemes. While the BGK and RR3 collision models have the same dissipation error on both the shear and acoustic modes, the HRR$_\psi$ collision operator has an higher dissipation error. More generally, the LBM-BGK and LBM-RR3 have less numerical dissipation than a second-order scheme on the shear mode and are even less dissipative than sixth-order optimised Navier-Stokes schemes for acoustics. Now, these results have to be tempered for the HRR$_\psi$ collision model with $\sigma = 0.995$. Indeed, the shear mode is more dissipated than with a second-order NS scheme and its dissipation error when considering acoustic fluctuations is rather between optimised third order and sixth-order Navier-Stokes centered schemes.

When increasing the Mach number of the mean flow up to $\overline{\mathrm{Ma}} = 0.2$, dissymmetries appear in the Ac+ and Ac- modes. Therefore, Figure 12 represents the dispersion and dissipation errors of the lattice Boltzmann and Navier–Stokes schemes at $\overline{\mathrm{Ma}} = 0.2$ for all three modes of interest.

Considering the dissipation error curves, the same conclusions can be drawn: all the LBM models exhibit the same dispersive properties and lie between a global second order scheme and an optimized third order in space with a 3-step Runge–Kutta in time. In the same way, while both the dissipation error curves for the Ac+ and Ac- are in favor of the LBM-BGK and LBM-RR3 models, the HRR$_\psi$ model is still better in between an optimised third order and sixth-order Navier-Stokes centered scheme. Nevertheless, results

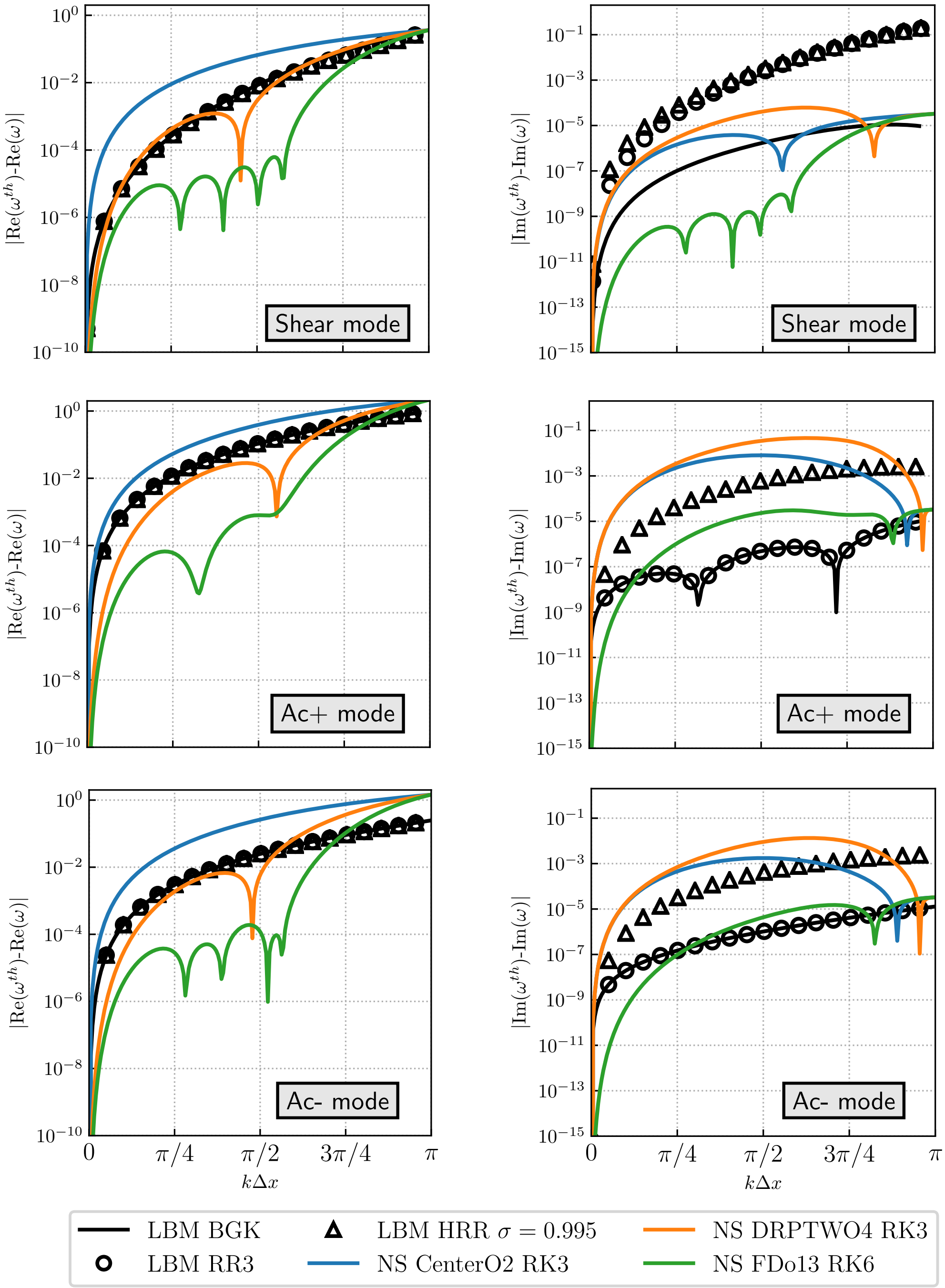


Figure 12: Dispersion (left), and dissipation (right) of the lattice Boltzmann and Navier–Stokes schemes for $\overline{\mathrm{Ma}} = 0.2$ and $\nu = 10^{-5}$.

drastically change concerning the dissipation of the shear mode. As previously discussed, the hyperviscous degeneracy induced by regularised collision models make these models highly dissipative on the shear mode. In comparison, Navier-Stokes schemes tend to be less dissipative. The only scheme 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.

To conclude this section on the comparison between Navier-Stokes and lattice Boltzmann schemes through linear stability analyses, planar plots of the dissipation errors are provided in Figure 13 for $\overline{\text{Ma}} = 0$, similar tendencies can be observed for $\overline{\text{Ma}} \neq 0$.

As already stated, despite its strong anisotropy, the LBM-BGK is the least dissipative method over a broad range of wavenumbers. When considering more stable collision operators such as regularised ones, the low-dissipation argument of the LBM need to be revised. Indeed, for perturbations aligned with the lattice main directions, the LBM remains advantageous in comparison with traditional Navier-Stokes schemes. However, in other direction, the results highly depend on the physical mode of interest. Therefore, when considering shear modes, the Navier-Stokes schemes seem to be better candidates to propagate such information over long distances. Now, for acoustic problems, even though regularised LBMs have a higher dissipation error w.r.t. the classical LBM-BGK, at low wave numbers, i.e. $||\boldsymbol{k}|| \leq \pi/2$, they present an lower dissipation error than 2nd-oder and 3rd-order Navier-Stokes schemes.

All in all, the conclusion of [4] are still valid for the BGK collision operator even when considering 2D perturbations. Moreover, on acoustic modes, even LBMs with regularised collision operators remain competitive with respect to 2nd-order Navier-Stokes schemes. Nevertheless, for vorticity driven flow, only the LBM-BGK presents advantages in comparison with traditional Navier-Stokes schemes.

4. Numerical simulations

The lattice Boltzmann and Navier-Stokes methods are now compared through numerical simulations on four different test cases. In the first place, the computation of plane monochromatic waves aims at validating the results of the linear stability analyses performed in Section 3. Then, two more realistic test cases, representative of unsteady aerodynamics problems, are studied: the convection of a vortex and the 3D Taylor Green vortex. This relative comparison between the LBM and the finite-volume Navier-Stokes method will give some insight on their numerical behavior and draw some guidelines on their optimal operating conditions.

As stated earlier, all the simulations are performed in the framework of ONERA's Cassiopee/Fast CFD environment. Only full 3D computational domains using periodic boundary conditions are considered.

4.1. Plane monochromatic acoustic wave

This first test case focuses on the propagation of a downstream plane monochromatic acoustic wave in order to assess the capabilities of the lattice Boltzmann and Navier-Stokes schemes introduced in Section 2 when dealing with aeroacoustic computations. For this purpose, the initial flow field is given by [7]:

$$\begin{aligned} \rho(x,y)|_{t=0} &= \overline{\rho} + \rho', \quad \text{where } \rho' = \epsilon\overline{\rho}\cos(k_x x + k_y y) \\ u_x(x,y)|_{t=0} &= \overline{\text{Ma}} c_s + \rho' c_s \cos(\theta_k)/\overline{\rho} \\ u_y(x,y)\big|_{t=0} &= \rho' c_s \sin(\theta_k)/\overline{\rho} \end{aligned} \tag{45}$$

where $\overline{\rho}$ 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 wave numbers in direction x and y respectively. 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 points per wavelength in the i direction. $\theta_k = \arctan(k_y/k_x)$ is the propagation angle of the wave.

In the following, the grid size Δx is constant and the computational domain corresponds to one wavelength in the propagating direction with five cells in the other directions. The simulations are run for 50,000

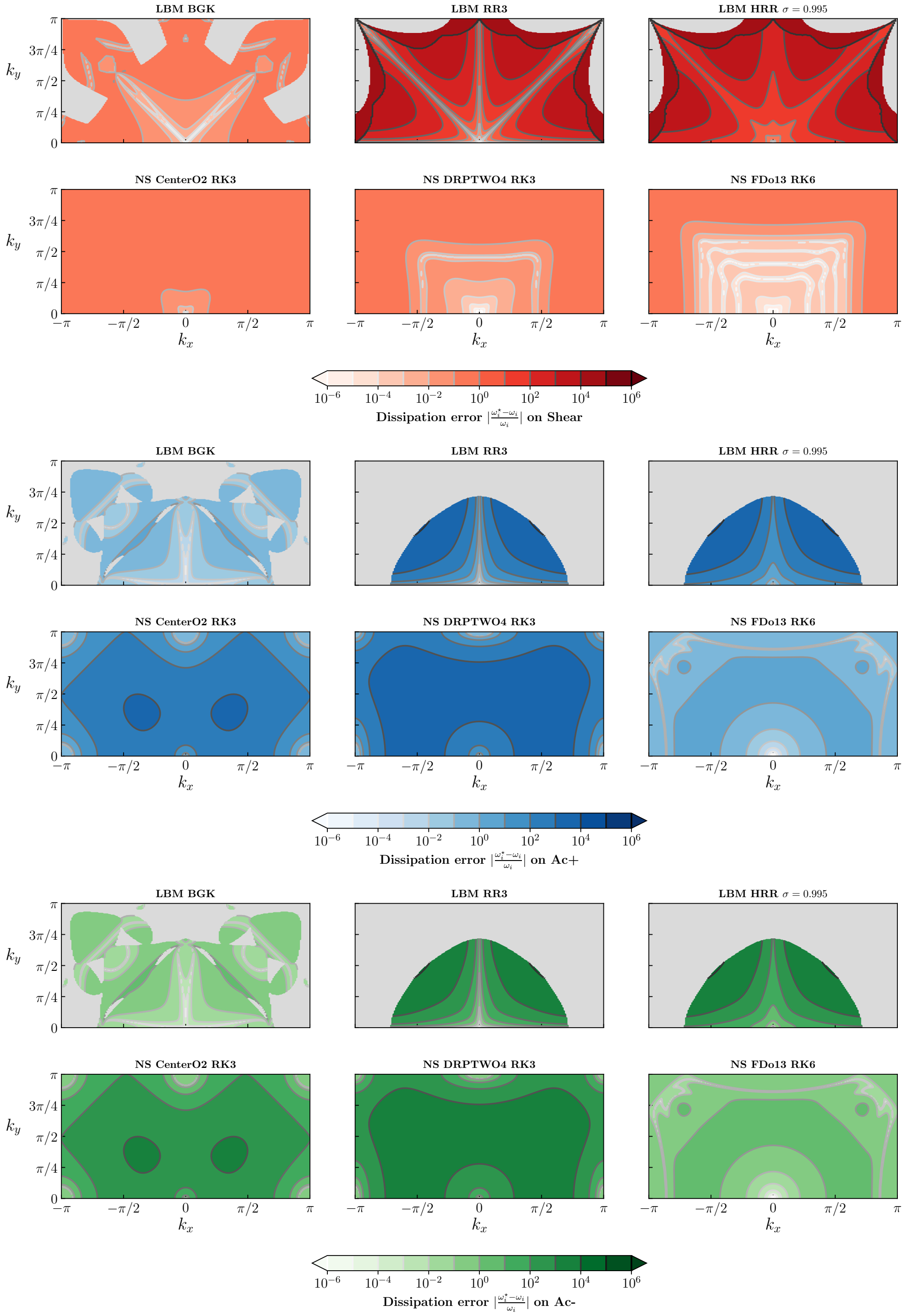


Figure 13: Spectral maps of the dissipation errors for the shear (top), Ac+ (middle) and Ac- (bottom) modes.

time-steps so as to observe significant effects of dispersion and dissipation. It can be shown that the density field at any given time t is given by:

$$\rho(x, y, t) = \overline{\rho} + \epsilon\overline{\rho} \exp\left[\mathrm{Im}(\omega)t\right] \cos(\boldsymbol{k} \cdot \boldsymbol{x} - \mathrm{Re}(\omega)t) \tag{46}$$

where the theoretical real and imaginary part of the angular frequency ω are given by Equation (34). Note that, for this study, the viscosity is set to $\nu = 10^{-5}$ m^2/s. The numerical dispersion and the numerical dissipation induced by each method are estimated by least-square fitting Equation (46) to the density time signal. Consequently, the study of the acoustic properties of both approaches is made through the dispersion and dissipation ratios:

$$E_\omega = \frac{\mathrm{Re}(\omega)}{\mathrm{Re}(\omega^{th})} \quad \text{and} \quad E_\nu = \frac{\mathrm{Im}(\omega^{th})}{\mathrm{Im}(\omega)}. \tag{47}$$

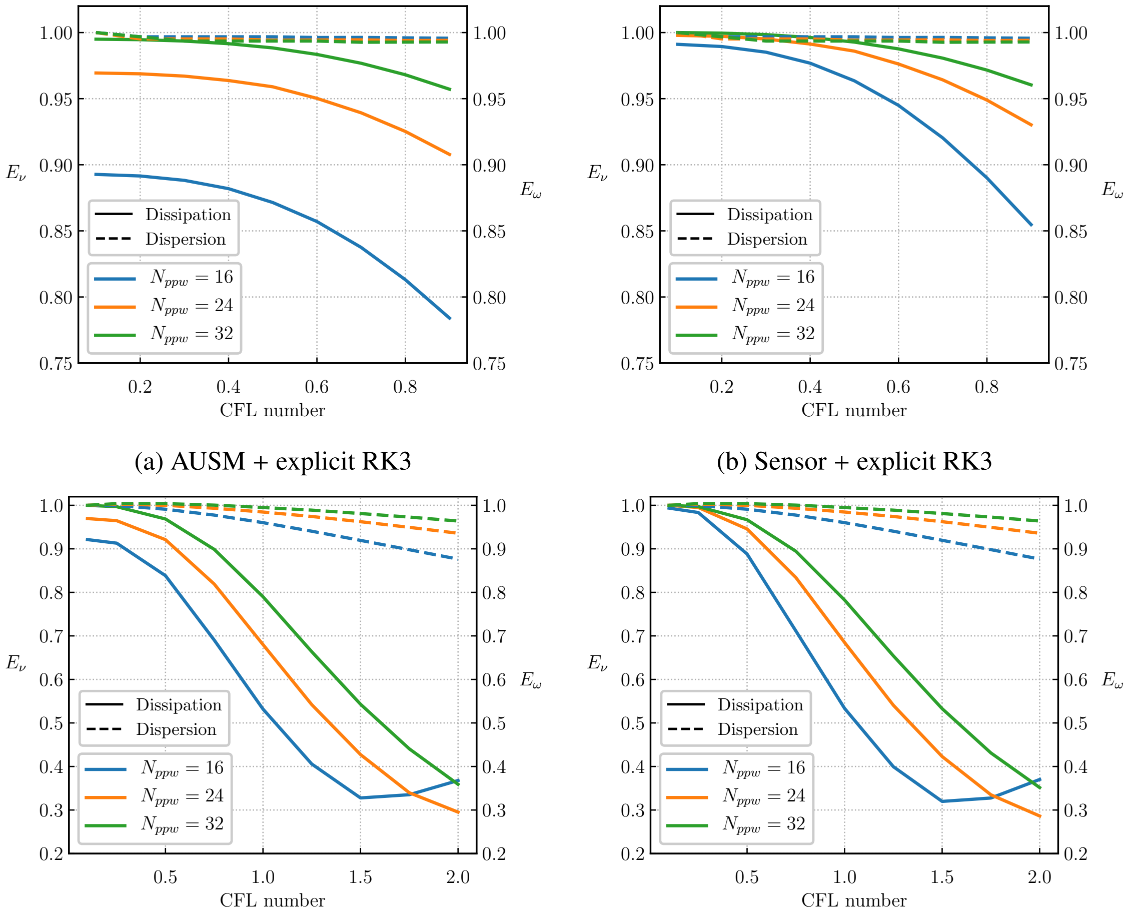
Parametric study of the Navier-Stokes schemes. Before comparing the acoustic capabilities of the lattice Boltzmann and Navier-Stokes schemes, a little discussion regarding the tuning parameters of the latter has to be made. Indeed, finite-volume Navier-Stokes solvers offer a higher set of degrees of freedom than LBMs. For instance, the influence of the various combinations of spatial and temporal schemes as well as the effect of the CFL number on the acoustic properties of Navier-Stokes methods have to be discussed. 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);
- the value of the CFL number.

Figure 14 shows the results obtained for different CFL numbers, points per wavelength and time-stepping schemes for the AUSM and Sensor convective fluxes approximations in the case of a 1D plane monochromatic acoustic wave without mean flow (i.e. $k_y = 0$ and $\overline{\mathrm{Ma}} = 0.$).

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 expected, 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 aeroacoustics (CAA) applications. Indeed, for typical CFL numbers encountered in industrial applications, 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 14 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 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. Now that the best combinations of spatial and temporal schemes for finite-volume Navier-Stokes computations have been highlighted, one can compare their numerical properties with the lattice Boltzmann models. Figure 15 shows the dispersion and dissipation ratios as a function of the non-dimensional wavenumber $k\Delta x = 2\pi/N_{\mathrm{ppw}}$ without mean flow. Since the values of E_ω and E_ν fall in the



(a) AUSM + explicit RK3

(b) Sensor + explicit RK3

(c) AUSM + implicit Gear

(d) Sensor + implicit Gear

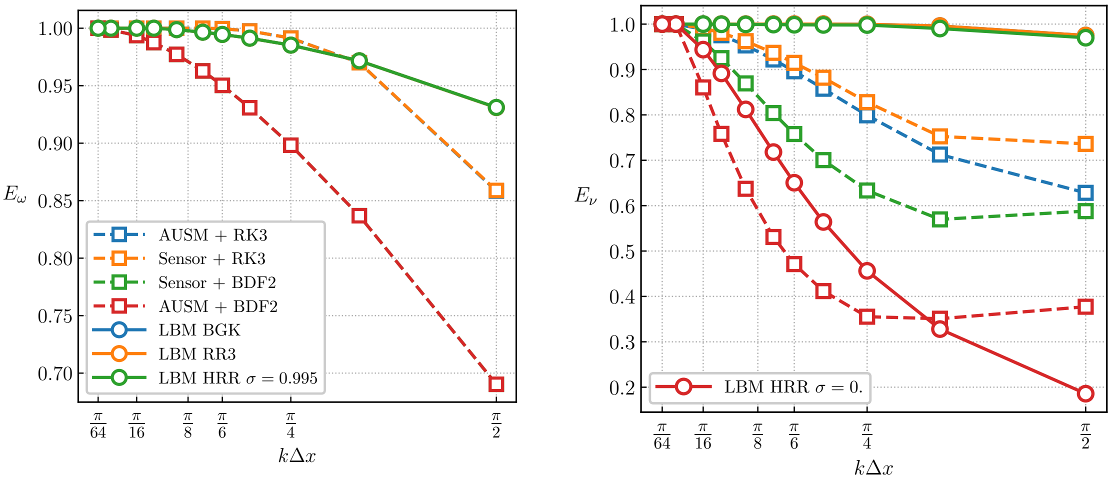
Figure 14: Dispersive (dashed lines) and dissipative (solid lines) behavior of the finite-volume Navier-Stokes schemes for different CFL numbers and points per wavelength.

range [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.

As one can see from Figure 15, the NS implicit schemes are not suited for CAA simulations as their values of Re(ω) ands Im(ω) strongly deviate from their theoretical counterpart. In the same way, the LBM HRR with $\sigma = 0.0$ has to avoided when focusing on acoustic waves since its dissipative behavior is much the same as that of implicit NS schemes.

Regarding the dispersive properties of the schemes, all the LB schemes have the same values of E_ω, which confirm the trend of Figure 11. However, up to 6 points per wavelength, the finite-volume Navier-Stokes schemes are slightly less dispersive than the LB ones. This result does not in agreement with Figure 11 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 equivalent to centered schemes thereby modifying their dispersion relations.

When it comes to dissipation, the results of the linear stability analyses are retrieved. Indeed, all the LB schemes (except the LBM HRR with $\sigma = 0.$), are by far less dissipative than the finite-volume Navier-



(a) Dispersion

(b) Dissipation

Figure 15: 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 $\overline{\mathrm{Ma}} = 0.$).

Stokes schemes. 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.

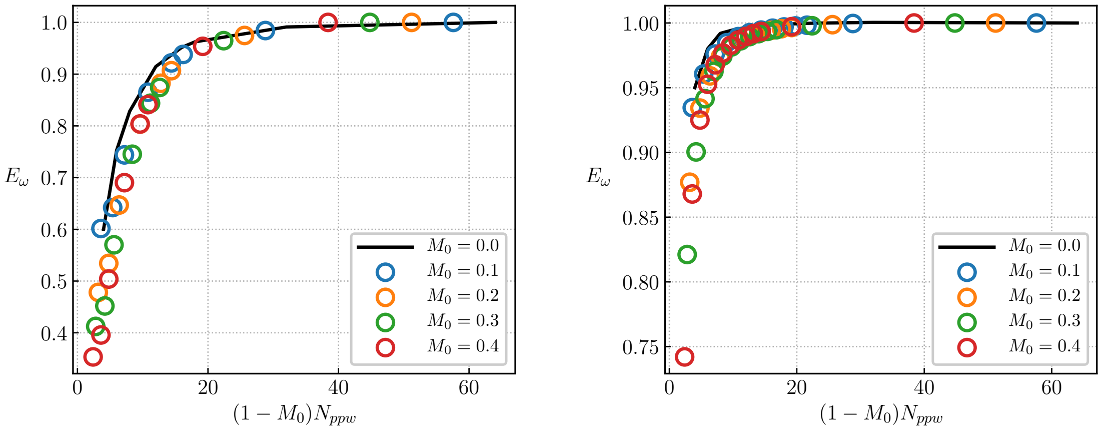
While Figure 15 compares the acoustic properties of the NS and LB schemes without mean flow, the same tendencies are observed when considering acoustic waves superimposed to a mean flow. Figure 16 compares the dissipative behavior with and without mean flow as a function of the effective number of points per wavelength $(1 - \overline{\mathrm{Ma}})N_{\mathrm{ppw}}$. It can be seen that the results with $\overline{\mathrm{Ma}} \neq 0$ are coherent with the ones obtained without mean flow if the number of points per wavelength is accordingly corrected.

All in all, the curves of Figure 15 can be transposed to cases with a mean flow by increasing the number of points per wavelength to $(1 + \overline{\mathrm{Ma}})N_{\mathrm{ppw}}$.

Until now, only 1D acoustic waves were considered. In order to compare the schemes for waves traveling in all possible directions (i.e. $k_x \neq 0$ and $k_y \neq 0$), simulations were performed for $k_x \in [4, 32]$ and $k_y \in [4, 32]$ with a step $\Delta k = 2$ leading to approximately 200 computations per scheme. Figure 17 shows the spectral maps of effective viscosity in the absence of mean flow.

As one can see from Figure 17, there is a good agreement between the results of the 2D linear stability analysis introduced in Section 3 and the numerical results. Indeed, the LBM with BGK collision model is the least dissipative scheme for all wavenumbers in the $[0, \pi/2]^2$ plane. For the regularised kernels, the computations highlight some preferential directions corresponding to main directions of the lattice. Even though they have an increased dissipation, they still are competitive with respect to the explicit Sensor scheme in all the considered directions. Moreover, the spectral map of the HRR collision model with $\sigma = 0$ confirms the fact that its high hyperviscosity makes it of very little use in aeroacoustics.

From all the results presented above, it is now possible to draw some guidelines about the most promising methods and schemes for acoustic propagation. Since the computational cost directly depends on the number of points per wavelength, Figure 18 displays the minimal value of N_{ppw} for the LB and NS schemes required to achieve a given tolerated dispersion or dissipation error level. For the sake of clarity, the minimal



(a) Sensor + explicit RK3

(b) LBM HRR $\sigma = 0.995$

Figure 16: Influence of the mean flow Mach number on the dissipative behavior of the lattice Boltzmann and finite-volume Navier-Stokes schemes.

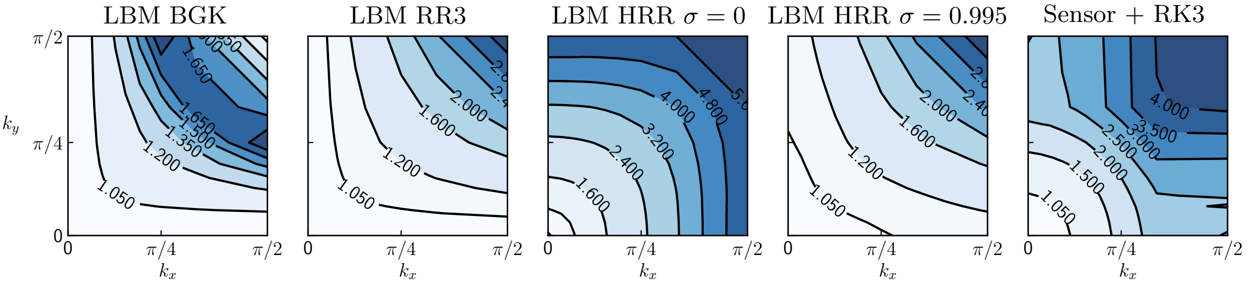


Figure 17: Spectral maps of effective viscosity ν_e/ν of the downstream acoustic mode with $\overline{\mathrm{Ma}} = 0$. Each map compiles the results of approximately 200 computations.

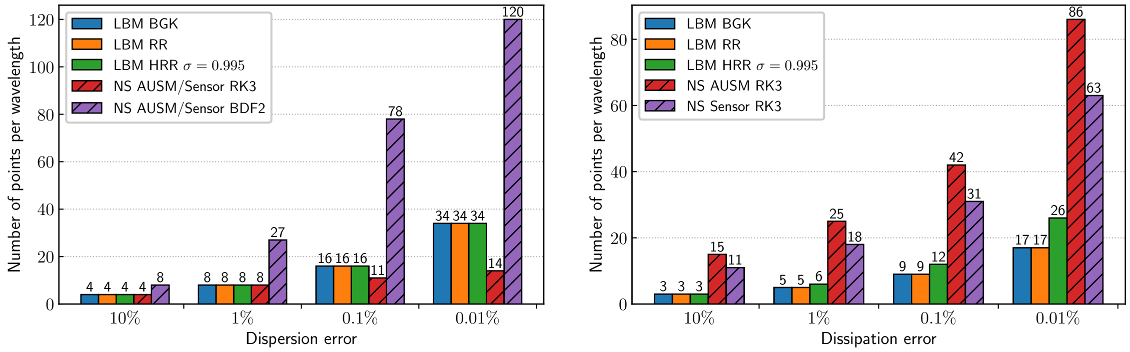
value of N_{ppw} is reported on top of each bar.

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 schemes require only 70% to 50% of the points needed by lattice Boltzmann schemes. Note that some values are provided for the implicit case in order to underline the fact that regardless of the spatial scheme, the implicit Gear scheme is not suitable for aeroacoustic computations.

When it comes to the minimal value of N_{ppw} required to achieve a given tolerated dissipation error level, Figure 18 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 5 times less points per wavelength than explicit Navier-Stokes finite volume schemes.

4.2. Plane monochromatic shear wave

The second type of hydrodynamic linear mode contained in the Navier-Stokes equations in the shear or vortical mode. Therefore, in order to assess the capabilities of the lattice Boltzmann and Navier-Stokes schemes when dealing with shear-driven flows the case of a plane monochromatic shear wave is considered.



(a) Dispersion

(b) Dissipation

Figure 18: 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.

For this purpose, the initial flow field is given by [7]:

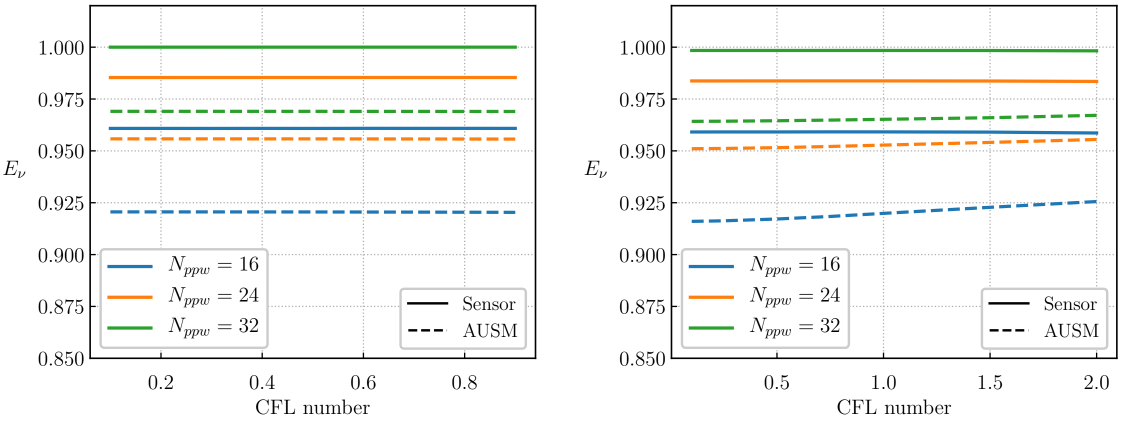
$$\begin{aligned} \rho(x,y)|_{t=0} &= \overline{\rho} \\ u_x(x,y)|_{t=0} &= \overline{\mathrm{Ma}}c_s - \epsilon\overline{\mathrm{Ma}}c_s \sin(\theta_k)\cos(k_x x + k_y y) \\ u_y(x,y)\big|_{t=0} &= \overline{\mathrm{Ma}}c_s \cos(\theta_k)\cos(k_x x + k_y y) \end{aligned} \tag{48}$$

where $\overline{\rho}$ and $\overline{\mathrm{Ma}}$ denote the mean flow density and Mach number respectively. As in Section 4.1, the amplitude of the perturbation is set to $\epsilon = 10^{-3}$. Moreover, the same computation domain is used and the simulations are run for 50,000 time-steps so as to observe significant effects dissipation. For the sake of clarity, only the dissipative behavior of each method on the shear wave is studied here. According to the results of the LSA, the decay of the transverse velocity u_y is expected to follow: $u_y \propto e^{-\nu k^2 t}$ where $k = ||\boldsymbol{k}||$. Therefore, the dissipation ratio E_ν is the only parameter investigated. Note that, for this study, the viscosity is set to $\nu = 10^{-5}$ m^2/s. In the following, a horizontal mean flow at $\overline{\mathrm{Ma}} = 0.2$ is considered.

Parametric study of the Navier-Stokes schemes. Following the same approach as above, the influence of the different parameters of the finite-volume Navier-Stokes is discussed. The results are presented in Figure 19 in the case of a 1D plane monochromatic shear wave (i.e. $k_y = 0$).

It can be seen that the numerical behavior of the finite-volume Navier-Stokes schemes is quite different in comparison to the acoustic case. Even if increasing the number of points per wavelength tends to recover the correct dissipation rate of shear waves, the CFL number only has very little influence on the numerical dissipation. For both the implicit and explicit time-stepping schemes, the low dissipation of the Sensor scheme in comparison to the AUSM is once again highlighted. Regarding the implicit time-stepping scheme, a slightly increased dissipation is observed for low CFL values which appears to be anti-dissipative when increasing the latter parameter. In this particular case, it is therefore more difficult to conclude whether one scheme is more appropriate than another even though the combination Sensor + explicit RK3 once again seems to be the best behaving one.

Comparison. In order to compare the lattice Boltzmann schemes with the finite-volume Navier-Stokes ones, Figure 20 shows the dissipation ratio of shear waves as a function of the non-dimensional wavenumber $k\Delta x$. As for the acoustic case, it can be concluded that both numerical methods to over-dissipate the shear waves inasmuch as $0 \leq E_\nu \leq 1$. This conclusion is in accordance with the results of the linear stability



(a) Explicit RK3

(b) Implicit Gear

Figure 19: Dissipative behavior of the finite-volume Navier-Stokes schemes for different CFL numbers and points per wavelength.

analyses of Section 3. However, in the present case, the dissipation gap between the LB and NS schemes is much smaller than in the acoustic case. Indeed, the low dissipation of the BGK collision model is lost as soon as one picks a more stable collision operator. This observation confirms the results of the linear stability analysis of Section 3. Yet, the lattice Boltzmann methods still seems to be a good candidate to propagate shear waves resolved by at least 6 points per wavelength.

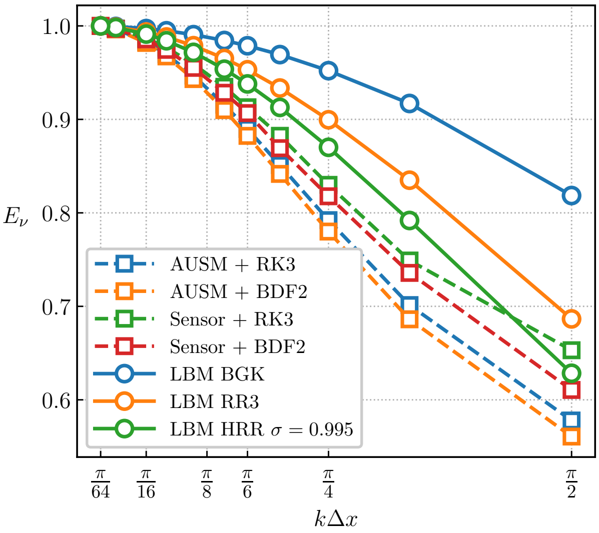


Figure 20: Comparison of the dissipative behavior of the lattice Boltzmann (solid lines) and finite-volume Navier-Stokes (dashed lines) schemes for a 1D plane monochromatic shear wave with mean flow ($k_y = 0$ and $\overline{\mathrm{Ma}} = 0.2$).

The schemes are now compared for shear waves traveling in all possible directions (i.e. $k_x \neq 0$ and $k_y \neq 0$). In the same way as for the spectral maps of the acoustic waves, simulations were performed for $k_x \in [4, 32]$ and $k_y \in [4, 32]$ with a step $\Delta k = 2$. The resulting effective viscosity maps for each schemes are given in Figure 21.

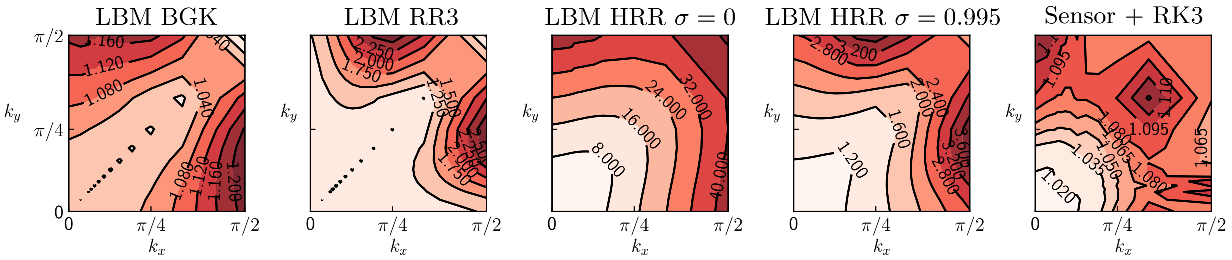


Figure 21: Spectral maps of effective viscosity ν_e/ν of the shear mode. Each map compiles the results of approximately 200 computations.

Once again, the numerical results are in agreement with the ones of the 2D linear stability analysis introduced in Section 3. While the LBM with BGK collision model has a numerical dissipation close to the one of Navier-Stokes schemes, the regularised lattice Boltzmann schemes suffer from a higher dissipation. As for the downstream acoustic mode, the effective viscosity remains close to one in the main directions of the lattice. Concerning the spectral map of the explicit Sensor scheme, some discrepancies with the centered scheme of the LSA are observed. This might indicate that for the shear wave the binary function Φ is not always equal to zero leading to an upwind formulation of the convective fluxes. All in all, for the whole range of wavenumbers studied here, when the shear waves do not propagate along the main lattice directions, the Navier-Stokes scheme seems to lead to a lower numerical dissipation.

To conclude this discussion on plane monochromatic shear waves, the minimal number of points per wavelength N_{ppw} for the LB and NS schemes required to achieve a given tolerated dispersion or dissipation error level is studied. Figure 22 compiles all these values. The first thing to notice is that the gap between of the mesh points required by both methods is narrower in comparison to the acoustic case. For all the error levels, the LBM BGK model requires about the same number of points per wavelength than the Sensor scheme. However, when switching to regularised collision models, the a-value of N_{ppw} tends to increase and matches the one of the AUSM scheme. All in all, the lattice Boltzmann method has the same numerical dissipation as classical finite-volume schemes (such as the AUSM) when considering shear waves. However, the Sensor scheme which was especially designed to achieve very little numerical dissipation seems to slightly outperform the LBM over a broad range of wavenumbers especially in comparison to regularised kernels.

4.3. Convected vortex

The convected vortex is classic benchmark for CFD methods. This problem is ideal for comparing the lattice Boltzmann and Navier-Stokes approaches since their theoretical accuracy should allow them to propagate the vortex without distorsion for an indefinite amount of time. Moreover, being able to sustain vortical flow structures with minimal numerical dissipation is crucial for advanced turbulence modeling like LES. For all these reasons, the advection of a vortex is considered here.

Most of the time, an isentropic formulation is adopted in the context of Navier-Stokes based schemes. However, as discussed in dedicated publications [56, 57], 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 [57] is chosen. It

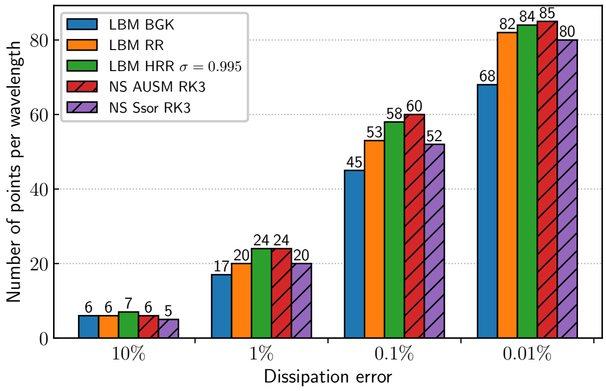


Figure 22: Minimal value of N_{ppw} for the LB and NS schemes required to achieve a given tolerated dispersion or dissipation error level on the shear mode.

reads:

$$\begin{cases} \rho(x,y)|_{t=0} &= \overline{\rho}\exp\left[-\dfrac{\epsilon^2}{2c_s^2}\exp\left(\dfrac{-r^2}{R_c^2}\right)\right], \\ u_x(x,y)|_{t=0} &= \overline{\text{Ma}}c_s - \epsilon\left(\dfrac{y-y_c}{R_c}\right)\exp\left[-\dfrac{(x-x_c)^2+(y-y_c)^2}{2R_c^2}\right], \\ u_y(x,y)\big|_{t=0} &= \epsilon\left(\dfrac{x-x_c}{R_c}\right)\exp\left[-\dfrac{(x-x_c)^2+(y-y_c)^2}{2R_c^2}\right], \end{cases} \tag{49}$$

where $\overline{\rho} = 1.1765$ 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 $R_c = 0.1$ m is the characteristic radius of the vortex. The center of the vortex is initially positioned at (x_c, y_c).

For this test case, a $3D$ periodic box of size $[L, L, 10\Delta x]$ is considered, 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 simulations are performed for a range of grid resolutions such as $N \in \{25; 50; 100; 200; 400\}$ and the computations are run over 5 advection cycles defined by the normalised time $5t^\star$ where $t^\star = t\overline{\text{Ma}}c_s/L$. The viscosity is set to $\nu = 10^{-5}$ such as to mimic a vortex convection in air. 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 $e^{-\nu k^2 t}$. This numerical dissipation rate G is estimated by averaging the norm of the vorticity field over one advection cycle and by computing the ratio between the 2 last and the 2 first cycles:

$$G = \left(\frac{\int_{3t^\star}^{5t^\star} ||\omega_z(t)||dt}{\int_0^{2t^\star} ||\omega_z(t)||dt}\right)^{1/4t^\star} \tag{50}$$

Figure 23 displays the evolution of the vorticity norm dissipation as a function of the numerical wavenumber $k\Delta x$. The latter is computed by noticing that the vortex has a global characteristic length of $8R_c$ [57]. In addition, to ease the comparison between the numerical solutions and the theoretical dissipation rate, a reference curve is shown in Figure 23.

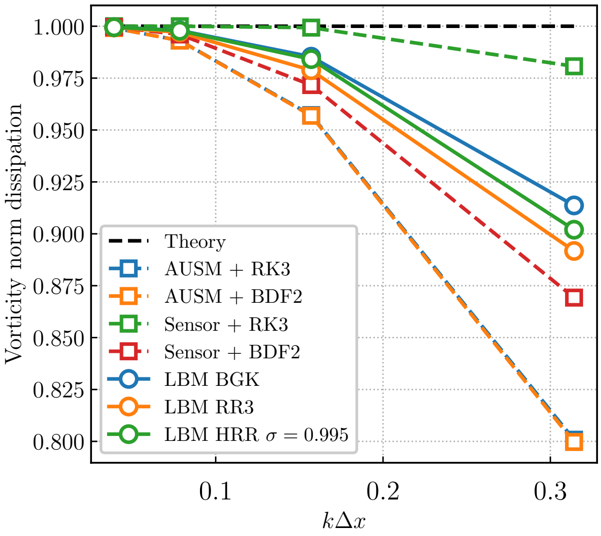


Figure 23: Numerical dissipation of the norm, with respect to the numerical wavenumber. The dashed line represents the theoretical dissipation rate of the vorticity mode.

From a general point of view, when refining the mesh (i.e. by decreasing the value of $k\Delta x$), all the numerical schemes tend to recover the theoretical dissipative behavior. If one focuses on the Navier-Stokes schemes, on main difference 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, contrary to the case of the plane monochromatic shear wave, their numerical dissipation is slightly greater than the one of the Sensor scheme with explicit time-stepping scheme. Moreover, as indicated by the linear stability analyses, switching to regularised collision models further increases the numerical dissipation. Yet, the numerical dissipation of the LB schemes is still less important than the one of the classic 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 the whole range of wavenumbers considered here for this test case.

Quality of the solution. Figure 24 compares the vortex shapes after 5 advection cycles. As one can see, for highly resolved cases ($N = 200$), all the numerical schemes converge to the same solution and match the analytical profile. Now, for low resolution, only the LB and explicit Sensor schemes are able to recover the analytical solution with limited numerical dissipation and distortion. Indeed, as already mentioned, by employing the AUSM scheme or an implicit time-stepping scheme, an increased numerical dissipation is observed. Moreover, the results in Figure 24 confirm the fact that the error of the Sensor scheme depends on the time-stepping scheme while the AUSM solutions are driven by the error of the convective fluxes approximation.

For $N = 25$ and $N = 50$, all the LB schemes outperform the AUSM and implicit Navier-Stokes solutions. Nevertheless, when $N = 100$ and $N = 200$, all the solutions are almost indistinguishable since they are all superimposed to the analytical profiles. Therefore, in order to distinguish their numerical behavior, one has to consider the convergence curve.

Convergence. The convergence of both approaches is discussed through the analysis of their L^2-error norm on the density field (the exact same conclusion were obtained when reasoning with the velocity fields). Figure 25 displays the evolution of the L^2-error norm as a function of the grid size.

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 their are designed as being second-order schemes. This discrepancy can be explained by the fact that all the computations where performed on cartesian grids with a third-order MUSCL reconstruction thereby biasing the results. Notwithstanding this unexpected behavior, the conclusions of Figures 24 and 23 are recovered.

Indeed, for low resolutions, the lattice Boltzmann schemes provide a more accurate solution than classical finite-volume schemes. However, if one is interested in well converged solution in the sense of the L^2-error norm, Navier-Stokes schemes seem to tend more rapidly to high-precision solution owing to their third-order MUSCL reconstruction.

4.4. 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 now 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 asses 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 [58, 59, 60].

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.* [61], the initial velocity and pressure fields are given by:

$$\begin{aligned}
u_x(x,y,z)|_{t=0} &= U_\infty \sin\left(\frac{x}{L}\right)\cos\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right)\\
u_y(x,y,z)\big|_{t=0} &= -U_\infty \cos\left(\frac{x}{L}\right)\sin\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right)\\
u_z(x,y,z)|_{t=0} &= 0\\
p(x,y,z)|_{t=0} &= p_\infty + \frac{\rho_\infty U_\infty^2}{16}\left[\cos\left(\frac{2z}{L}\right)+2\right]\left[\cos\left(\frac{2x}{L}\right)+\cos\left(\frac{2y}{L}\right)\right]
\end{aligned} \tag{51}$$

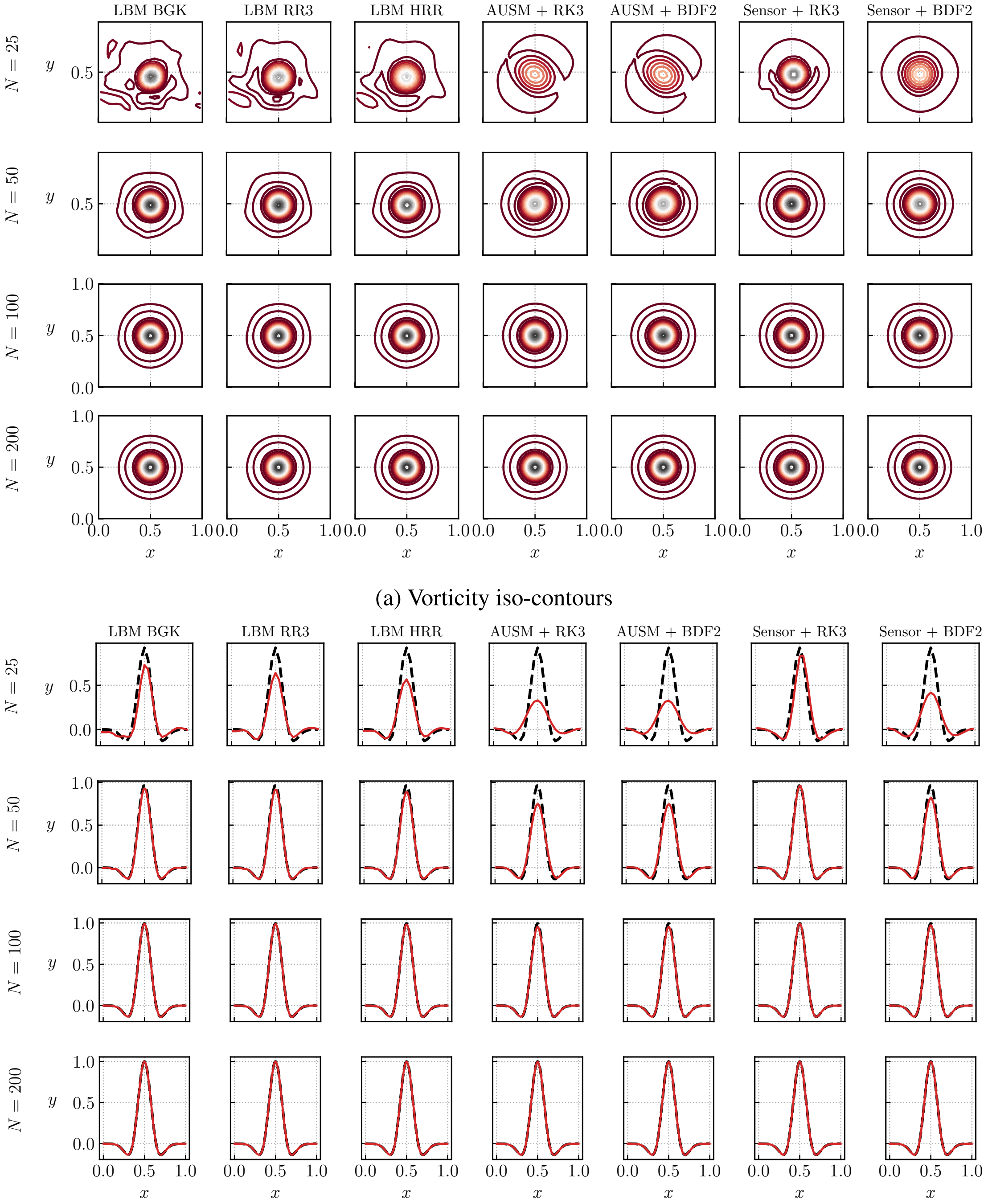
where U_∞, p_∞ and ρ_∞ denote the reference velocity, pressure, and density respectively. All these parameters are chosen such as to impose a Reynolds number $\mathrm{Re} = 1600$. Also note that for the LB computations, the initial distribution field is computed by calculating the velocity gradient [62].

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:

$$E_k(t) = \frac{1}{2|\Omega|U_\infty^2}\int_\Omega ||\boldsymbol{u}||^2 \,\mathrm{d}\Omega, \tag{52}$$

and the non-dimensional enstrophy evolution :

$$\mathcal{E}(t) = \frac{L^2}{2|\Omega|U_\infty^2}\int_\Omega ||\boldsymbol{\omega}||^2 \,\mathrm{d}\Omega. \tag{53}$$



(a) Vorticity iso-contours

(b) Vorticity profiles

Figure 24: Comparison of the vortex shapes in terms of in vorticity fields after 5 advection cycles. (a) Vorticity isocontours. 30 levels are displayed ranging from -1 (red) to 1 (grey). (b) Vorticity profiles for $y = 0$. The reference solution is shown by the dashed black line.

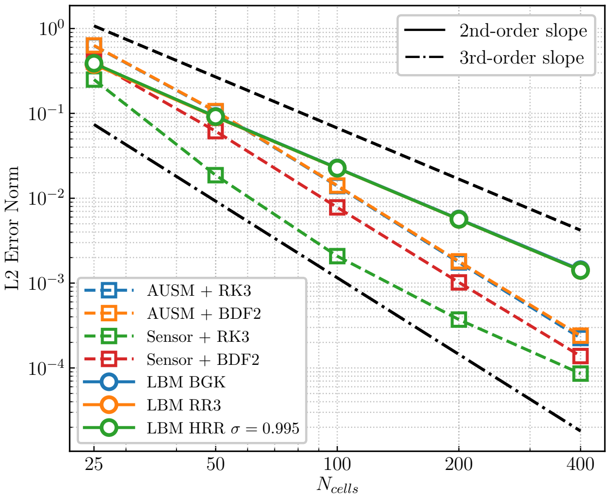


Figure 25: Convergence plot of the lattice Boltzmann (solid lines) and NS (dashed lines) schemes for the convected vortex test case. Two reference slopes are also represented.

The enstrophy (53) is computed from the vorticity field $\boldsymbol{\omega}$ for which a fourth-order reconstruction is used by both numerical methods. The temporal evolution of E_k and $\mathcal{E}$ is expressed by means of the non-dimensional time scale defined by: $t^\star = L/U_\infty$.

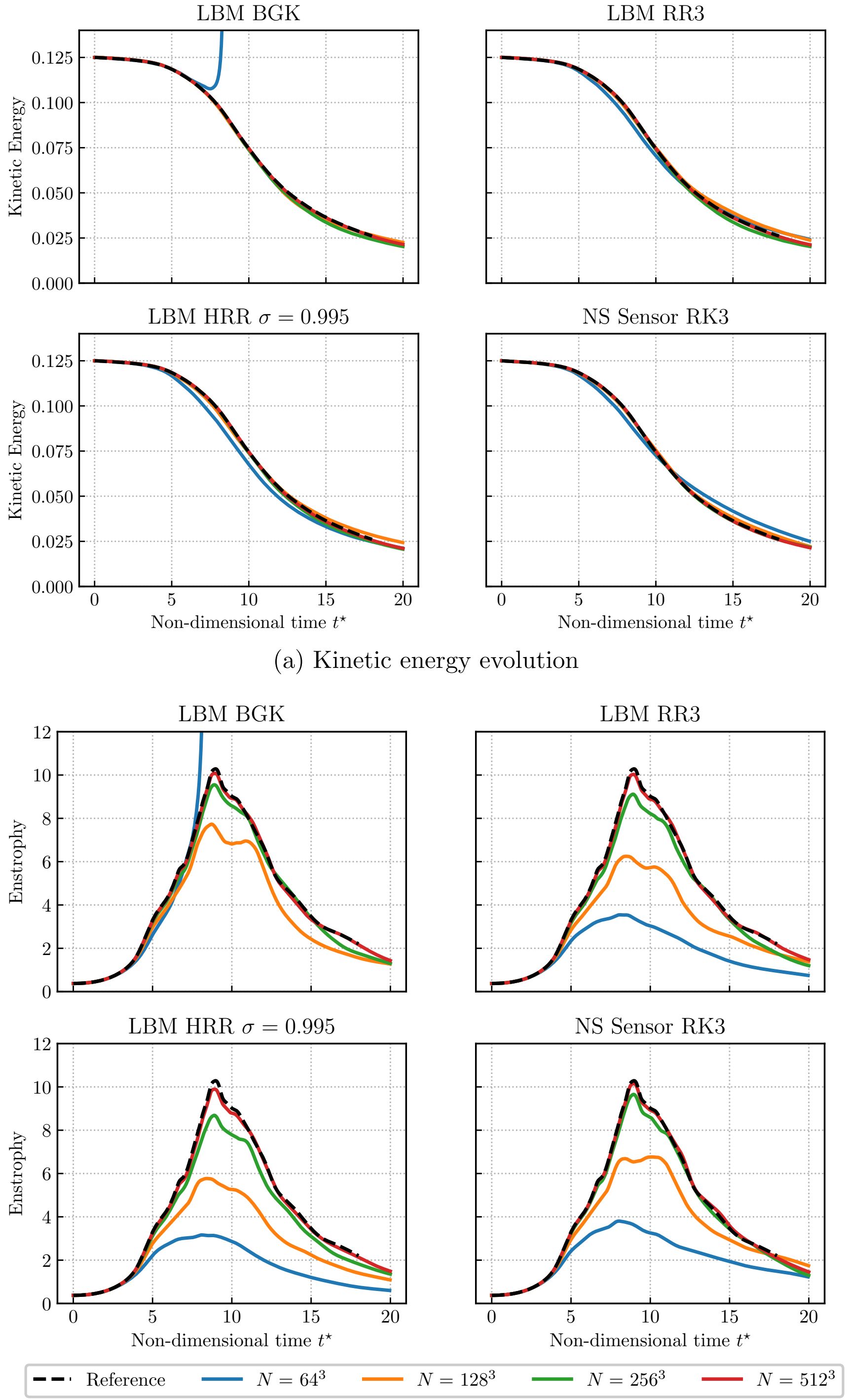
Figure 26 displays the temporal evolution of the kinetic energy and the enstrophy for three LBM computations (with BGK, RR3 and HRR collision models) and one explicit NS computation with the Sensor scheme. Indeed, the latter is especially designed such as to mitigate dissipative effect for turbulent flow computations. Different grid resolutions are investigated, namely 64^3, 128^3, 256^3 and 512^3. The results are compared to the spectral solution from [58] for a resolution of 512^3.

The first thing to notice from Figure 26 is that the computation carried out by the lattice Boltzmann method with BGK collision operator in the highly under-resolved case (i.e. on the 64^3 grid) is unstable. This justifies the fact of considering regularised schemes in this study since the latter remain stable for all the grids considered here. Consequently, one should keep in mind that even if stability issues are not fully studied here, the ability of numerical schemes to stay stable is fundamental in CFD.

Both the LBM and the finite-volume NS scheme converge towards the reference solution. In particular, the enstrophy peak is correctly recovered at $t^\star = 9$ 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 [63]. Therefore, comparing both numerical methods for this particular grid resolution gives some insight into their quasi-DNS capabilities. As can be seen on Figure 26, all the numerical schemes reache the spectral solution with very little error.

Accuracy. In order to compare more quantitatively the accuracy and convergence behavior of both methods, the are confronted through their L^2-error norm on the temporal evolution of the kinetic energy and enstrophy.

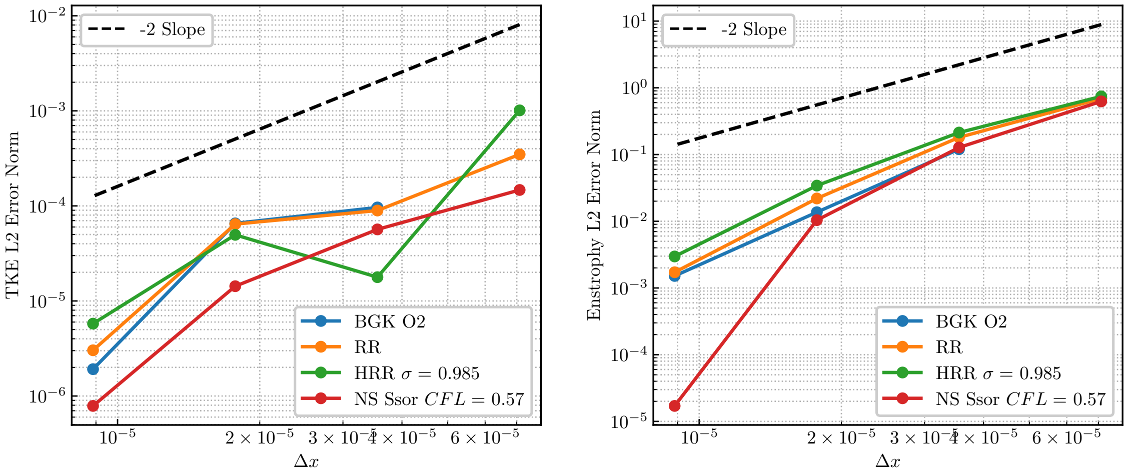


(a) Kinetic energy evolution

(b) Enstrophy evolution

Figure 26: 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 [58].

Figure 27 shows that the lattice Boltzmann and Navier-Stokes 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 27 confirm the tendencies of Figure 26. 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. However, 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 error

(b) Enstrophy error

Figure 27: Accuracy and convergence study for the 3D Taylor-Green vortex test case at Re = 1600. All the L^2-norms are computed with respect to the spectral solution of [58].

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

$$E(k,t) = \frac{1}{2}\sum_{k} |\widehat{\boldsymbol{u}}(\boldsymbol{k},t)|^2 \tag{54}$$

where $\widehat{\boldsymbol{u}}$ is the complex Fourier transform of the velocity field, $\boldsymbol{k}$ the wavenumber vector, and $k = |\boldsymbol{k}| = \sqrt{k_x^2 + k_y^2 + k_z^2}$ its norm. Equation (54) 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 Duraisamy [64] at $t = 10t^{\star}$.

Figure 28 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 [64] even though they tend to slightly over-estimate the kinetic energy associated to very low wavenumbers. Such discrepancies with the reference solution can also originate from the digitisation process of the low-resolution figure in [64]. For all the schemes considered here, the energy cascade in the intertial 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. Finally, for the 128^3 grid, the LBM-BGK model shows better convergence than the Navier-Stokes computation, thereby confirming the fact that lattice Boltzmann methods are efficient when considering coarse resolutions.

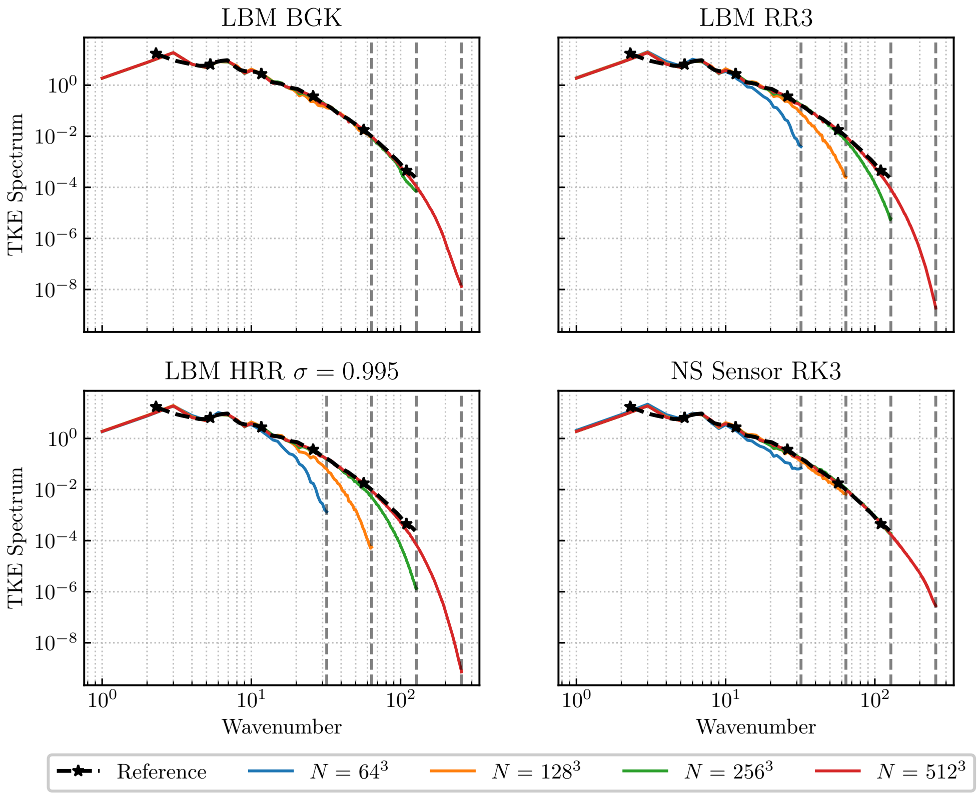


Figure 28: Turbulent kinetic energy spectrum for the 3D Taylor-Green vortex test case benchmark at Re = 1600. The vertical dashed lines indicate the cut-off wavenumbers associated to the smallest resolved scales.

5. CPU Performance

To conclude the comparison between the lattice Boltzmann and Navier-Stokes methods, their CPU performance has to be discussed. Indeed, in Section 4 some guidelines about the numerical schemes and grid-points requirements were provided. However, in the context of massively parallel systems, the ability of a numerical method to take advantage of modern computing resources is a crucial issue. While most of the existing runtime comparisons rely on different large-scale codes, the modularity of ONERA's Cassiopee/Fast CFD environment as well as the factorisation of the HPC layer provide the infrastructure for such endeavors.

5.1. Implementation details

This section aims as giving a short overview of the general implementation of the lattice Boltzmann and Navier-Stokes solvers in the framework of ONERA's Cassiopee/Fast CFD environment. Some of these details have already been presented in a communication by Alferez *et al.* [65].

Here the focus is mainly made on the HPC layer dedicated to shared memory nodes. Indeed, the transfers on distributed memory nodes uses functions of the Cassiopee/Connector module, based on MPI asynchrone point to point tranfers and, with the helps of these functions, it is relatively easy to obtain a good scalability at a cluster level.

Data layout. Both the Navier-Stokes and lattice Boltzmann equations are solved using a domain decomposition technique where ghostcells are used at the interfaces between two subdomains. The simulation domain is therefore represented by a multi-dimensional array featuring a flag flied for distinguishing between the computed cells centers and the not computed ones (e.g. ghostcells or obstacles). Consequently, all variables can be accessed by simple index arithmetic.

The choice has been made to store the variables in a Structure of Array mode (see Figure 29) for vectorisation purposes and to guarantee contiguous memory access in the NS and LBM schemes updates [66]. In the case of the NS solver, these variables consist in the primitive state vector $\mathbf{W} = (\rho, u_x, u_y, u_z, T)^t$ at 2 or 3 time levels depending on the time-stepping scheme. On the other hand, the LB solver uses two sets of distribution functions are used, namely the pre- and post-collision distribution functions.

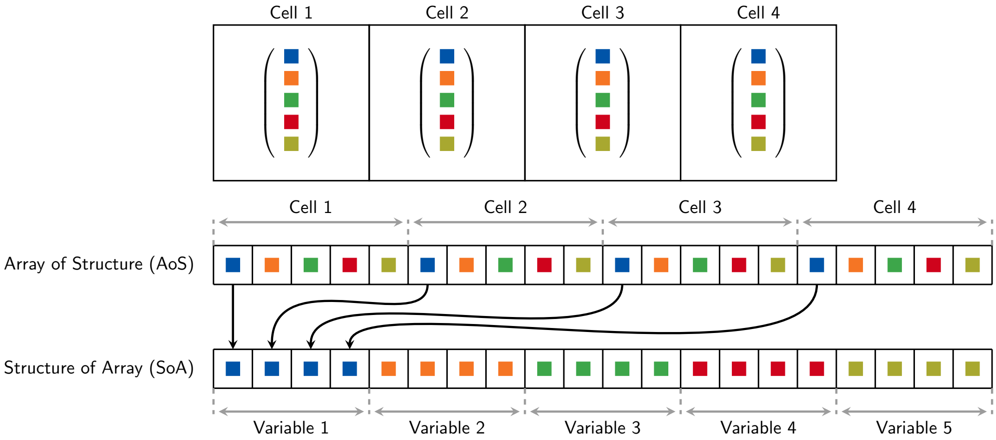


Figure 29: Graphical representation of the Array-of-Structure (AoS) and Structure-of-Array (SoA) data layouts. A unidimensional mesh composed of four cells is illustrated here. Each cell has five variables symbolised by squares.

Owing to its non-dimensional form, the LB solvers do not require any metrics to be stored. However, for the Navier-Stokes solvers, informations about each cell face normals and surface are needed for each topological direction of the mesh. For HPC reasons, 3 different versions of the NS structured solver FastS are implemented in order to take advantage of specific grid topology:

- `3dcart` for three-dimensional cartesian grids where all the metrics are reduced to scalar values.
- `3dhomo` for curvilinear meshes in the (x, y) plane and cartesian in the z direction.
- `3dfull` for three-dimensional fully curvilinear meshes.

Factorisation of the compute kernels. The factorisation process corresponds to the minimization of the number of functions needed to solve the Navier-Stokes and lattice Boltzmann equations. Therefore, one improves the cache temporality and minimizes the size of the work arrays.

For the structured Navier-Stokes equations solver, the number of functions has been reduced to 5 calls of subroutines [65] where the computation of fluxes balance is responsible for approximately 80% of the overall computational time in the explicit case.

Regarding the lattice Boltzmann solver, a straightforward implementation would lead to separate functions, namely the streaming step and the collision step. However, the number of data transfers can be reduced by executing the collision and propagation step in the same loop [66]. Moreover, since three nested loops over the three spatial dimensions are involved, an additional level of optimisation can be introduced by splitting the innermost loop in smaller ones and by storing common subexpression into buffer arrays [67].

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 layout of the threads if designed such as to match the layout of the memory. As stated earlier, the three-dimensional arrays are stored as multi-dimensional arrays. The memory alignment follows the standard alignment in Fortran: first, the x-direction is aligned, then the y-direction and finally the z-direction. Accordingly, the threads are organised as shown in Figure 30 indicating that one thread is aligned along the x-axis while blocks 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.

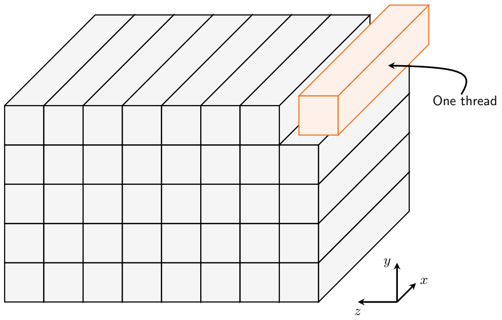


Figure 30: Computational domain decomposition strategy for threading. Each sub-domain corresponds to one thread.

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. For this purpose, the calculation area is divided into sub-parts whose size is processor dependent. This 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".

OpenMP parallelisation. All the previous optimisation techniques are embedded within an OpenMP thread parallelisation strategy at the node-level. Each thread computes its own local subset of indexes and the workload is distributed among the cores of the node (one thread per core). If the resulting workload is too large, cache-blocking is also performed. The cache blocking strategy also provides a way to reduce synchronisation constrains at the border between 2 threads.

5.2. Roofline Performance model

Before discussing the CPU peformance of the lattice Boltzmann and Navier-Stokes solvers implemented in ONERA's Cassiopee/Fast CFD environment, the Roofline model [68] is introduced in an effort to derive an upper performance limit of each method on the present hardware system. Throughout this section, the performances are measured on ONERA's supercomputer SATOR[1] featuring two types of CPUs with different micro-architecture. The first system is an Intel Xeon E5-2680v4 Broadwell dual-socket CPU (BDW) with 14 physical cores per socket operating at a fixed frequency of 2.4GHz. The second system is an Intel Gold 6152 Skylake dual-socket CPU (SKL) that features 22 cores per scoket and runs at a frequency of 2.1GHz. Table 3 summarizes the main specifications of both CPUs investigated in the present work.

Name		BDW	SKL
Processor		Intel Xeon E5-2680v4	Intel Xeon Gold 6152
micro-arch.		Broadwell	Skylake
frequency	[GHz]	2.4	2.1
cores per sock.		14	22
sockets		2	2
L1 cache	[KB]	32	32
L2 cache	[KB]	32	32
L3 cache	[MB]	18	30
ISA		AVX2	AVX512

Table 3: Specifications of the BDW and SKL systems.

The Roofline model states that the performance of a given application is either bounded by the floating point performance (expressed in floating point operations per second or FLOPS) or bounded by the memory bandwidth (in bytes/s). Thereby, the achievable peformance P is expressed by:

$$P = \min(P_{\text{peak}}, B_m / B_\ell) \tag{55}$$

where P_{peak} denotes the processors peak floating point performance, B_m is the memory bandwidth and B_ℓ is the loop balance which depends on the studied algorithm. The latter corresponds to the number of bytes that must be transferred between the processor and memory per floating point operation required to execute on step of the algorithm.

In order to determine the effective memory bandwidth, the STREAM `copy` benchmark is used. It consists in copying two array without non-temporal stores and including write-allocate. Moreover, to refine this estimate, an adjusted benchmark mimicking the LB streaming step is designed: `copy-19` which concurently copies 19 arrays [69]. Table 4 reports the measured bandwidths of all evaluated systems. From Table 4 it becomes clear that memory bandwidth provided by the manufacturer differs from the effective one. Indeed, the STREAM `copy` benchmark recovers only 70% of the specified bandwidth. Moreover, the introduction of the adjusted `copy-19` benchmark is totally justified by the fact the bandwidth is decreased when copying concurrently 19 arrays in comparison to one. All in all, only half of the manufacturer bandwidth is expected for the lattice Boltzmann scheme.

[1] `https://www.onera.fr/en/high-performance-computer`

Name		BDW	SKL
Processor		Intel Xeon E5-2680v4	Intel Xeon Gold 6152
socket bandwidth			
Intel	[GB/s]	78.6	128
`copy`	[GB/s]	58.8	92.7
`copy-19`	[GB/s]	43.6	70.8
full node bandwidth			
Intel	[GB/s]	157.2	256
`copy`	[GB/s]	117.0	186.4
`copy-19`	[GB/s]	87.2	140.4
Floating point op. limits	(full node)		
P_{max} scalar	[GFLOPS]	57.6	92.5
P_{max} vect.	[GFLOPS]	230.8	369.3

Table 4: Memory bandwidth and peak floating point performance for the BDW and SKL systems.

In its classical formulation, the Roofline model in expressed in FLOPS. However, for the LBM and finite-volume Navier-Stokes algorithms it is better to think about performance in terms of “work performed per time unit” where “work” corresponds to cell updates and the “time unit” to one time-step. The associated metric is Cell Updates Per Second (CUPS). In order to perform this change of units, the key figure to determine is the number of floating point operations per cell update:

$$\gamma_{\text{FP}} = \left[\frac{\text{FLOP}}{\text{CUP}}\right] \tag{56}$$

The value of the parameter γ_{FP} has to be determined carefully. Indeed, the number of operations in the mathematical description and high-language implementation differ. However, the only valid number for an estimation is the number of generated assembly instructions and its corresponding number of operations. Manually counting the number of floating point operations is tedious. Therefore, an automated analysis with the Intel Advisor tool is employed. The total number of floating point for an application is then computed and divided by the number of cells and iterations. The values of γ_{FP} obtained for the different solvers are reported in Table 5.

Solver	Version	γ_{FP}
FastLBM	D3Q19-BGK	204
	D3Q19-RR3	325
	D3Q19-HRR	490
FastS	`3dcart`	1012
	`3dhomo`	1295
	`3dfull`	1774

Table 5: Number of floating point operations per cell update for each numerical scheme.

The γ_{FP} metric already points out one main difference between the lattice Boltzmann method and finite-volume Navier-Stokes algorithms. Indeed, regardless of the collision models, the LBM is less computationally intense than traditional finite-volume schemes. This directly comes from the simplicity of the stream and collide algorithm. Therefore, from a purely operational point of view, the LBMs require less floating point operations per cell updates than Navier-Stokes schemes. Naturally, when implementing more robust collision model (such as the regularised ones), more operations have to be made in comparison to the classic BGK model.

In order to determine whether the lattice Boltzmann and finite-volume Navier-Stokes solvers are memory or compute bound and to determine their theoretical performance, one has to compute their respective arithmetic intensity. This is equivalent to calculating the number of bytes which are transferred from the main memory to the CPU for one cell update.

Arithmetic intensity of the D3Q19 LBM schemes. A straightforward implementation of the LB algorithm requires two data arrays each of which has the characteristic data footprint of 19 distribution functions f_i. For a cell update, a first set of 19 distribution functions has to be loaded from memory. Then, wiriting to the second array requires another load of 19 distribution functions and finally a store instruction for the latter. This leads to a number of bytes being transferred for a single cell update:

$$19 \times 8 \text{ bytes} \times 3 = 456 \text{ bytes/CUP} \tag{57}$$

The estimate of Equation (57) assumes that the macroscopic variables $(\rho, \mathbf{u})$ are not stored at each time-step. Indeed, for the classic LBM-BGK and LBM-RR algorithms (without turbulence model) these are unnecessary and only required for post-processing. However, in the case of the HRR collision operator, the shear-stress tensor and other gradients involved in the definition of the corrective term have to be computed. Therefore, the storage of the macroscopic variables $(\rho, \mathbf{u})$ is mandatory leading to the following arithmetic intensity:

$$19 \times 8 \text{ bytes} \times 3 + 4 \times 8 \text{ bytes} \times 2 = 520 \text{ bytes/CUP} \tag{58}$$

It is however possible to mitigate the number of bytes being transferred from memory to CPU by using non-temporal stores. These bypass the writing step to the cache. In addition, other strategies using only one set of distribution functions naturally give rise to a decrease in the number of bytes transferred from memory to CPU for one CUP. However, these will not be studied here. The interested reader can refer to [70].

Arithmetic intensity of the finite-volume Navier-Stokes solver. The memory footprint of a cell update of the finite-volume Navier-Stokes solver is a bit more complicated to characterise since it involves many different variables. In the case of an explicit computation, using FastS `3dcart` solver, the metric fields are reduced to scalar values and only two set of primitive variables are needed for the update. According to the fact that the stencil of the second-order finite-volume schemes involves 25 cells for each variable, one can infer that its arithmetic intensity is about: 1152 bytes/CUP (which means that the arithmetic intensity is about 384 bytes per sub-iteration of the third-order explicit Runge-Kutta scheme).

By comparing the arithmetic intensity of the lattice Boltzmann and Navier-Stokes solvers, it becomes clear that the LBM is memory driven meaning that for one cell update, more bytes are being transferred than computations performed. On the other hand, the inverse is true for the Navier-Stokes evolution where computations are predominant.

Figure 31 displays the roofline model for the LBM-BGK, LBM-RR3, LBM-HRR as well as for the 3D cartesian version of the Navier-Stokes solver of FastS on a full BDW node. The vertical lines indicate the

Solver	LBM BGK	LBM RR	LBM HRR	FastS `3dcart`
Achievable Performance [MCUPS]	285	283	234	101
Relative Performance	2.82	2.8	2.3	1.

Table 6: Maximum performance achievable by each of the numerical method according to the results of the Roofline model (see Figure 31). The relative performance w.r.t. the FastS `3dcart` solver is also provided.

arithmetic intensity of each solver and therefore allow to determine their maximum achievable performance. These values are also reported in Table 6 for the sake of clarity.

All the models indicate that both the Navier-Stokes and lattice Boltzmann algorithms are memory bound. This statement is true with the help of vectorisation techniques which fully justifies their use here. Moreover, the influence of the γ_{FP} parameter is clearly evidenced by the values of the horizontal plateaus. Indeed, a higher value of γ_{FP} tends to decrease the maximum performance under the computing power limit. Since γ_{FP} represents the number of FLOPs per cell update, it can be seen from the comparison of Figures (a) and (d) that a cell update within the NS framework is more expensive than a LB iteration.

All in all, regardless of the collision model, the LBM is expected to outperform the FV-NS solver from a purely CPU performance point of view. It can be seen that a well coded FV-NS solver on cartesian grids is far less than 10× slower than LB codes [11]. Here, the ratio between their performance is shown to be around 2.3 to 3. Note that this is only valid for the `3dcart` version and that the ratio might increase when considering curvilinear versions such as `3dhomo` and `3dfull`.

5.3. Single Node scaling

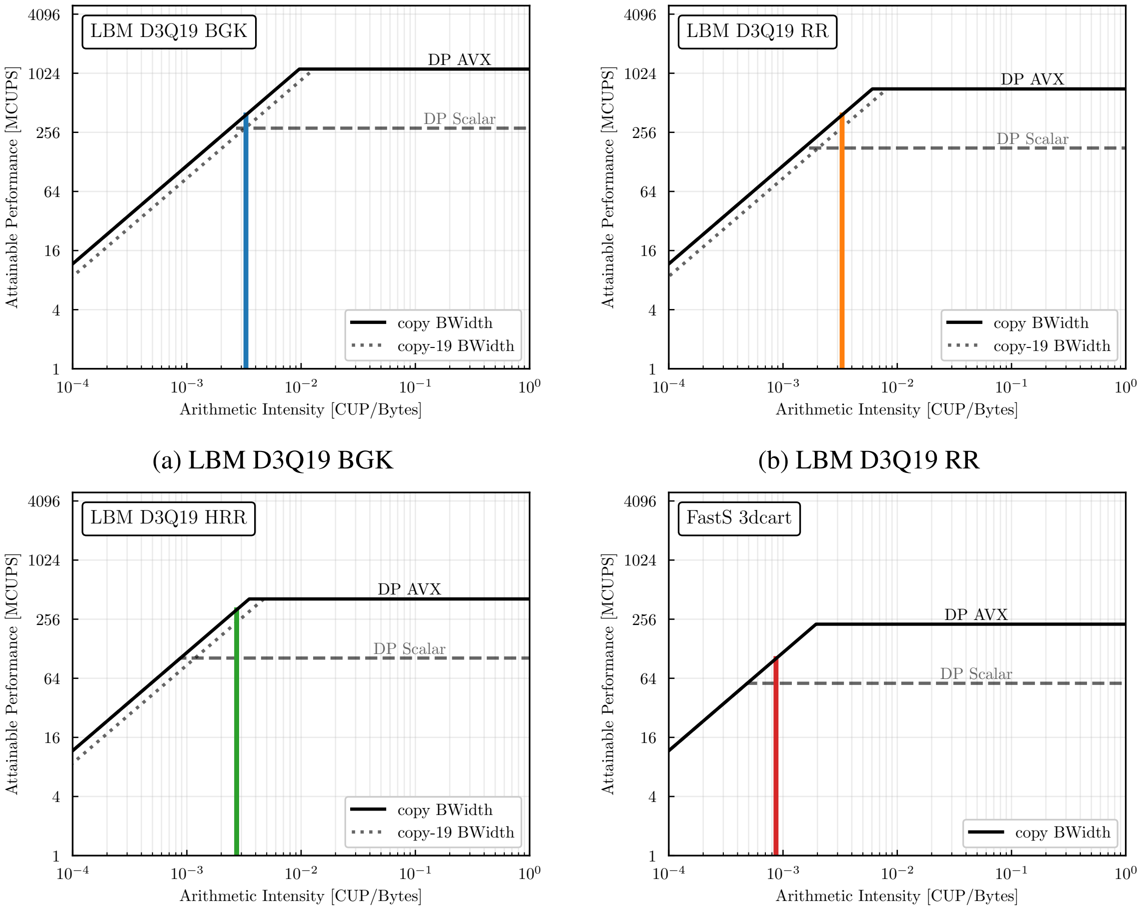
Now that the maximum achievable performance of each solver have been determined, it is compared to effective performance of the kernels implemented in ONERA's Cassiopee/Fast CFD environment. The performances are evaluated by running computations of the three dimensional Taylor Green vortex on grids consisting of 200^3 points which is representative of the workload one would encounter at a node level for distributed memory clusters on large scale problems.

Figure 32 shows the strong scaling of the performance of the Navier-Stokes and lattice Boltzmann solvers depending on the number of cores of the BDW system. The performance is evaluated in MCUPS and is defined by:

$$P_{\text{MCUPS}} = \frac{n_{\text{substep}} \times 200^3}{t_{\text{CPU}} \times 10^6} \tag{59}$$

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). The horizontal lines show the Roofline model predictions for each kernel.

It can be seen from Figure 32 that the full-node performance of the Navier-Stokes and lattice Boltzmann solvers 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 structured solver, the curves of Figure 32 highlight the gain achieved by implementing grid specific subroutines. The cartesian version is naturally the fastest and is about 2.5 times faster than its full curvilinear counterpart. The `3dhomo` version of the Navier-Stokes solver is only 1.5 times slower than the cartesian version making it a good candidate for LES or DNS in reasonable CPU times. Moving to the lattice Boltzmann modes, they all are more than twice as fast as the Navier-Stokes solvers. As indicated by the Roofline model, when the full node is used, both the



(a) LBM D3Q19 BGK

(b) LBM D3Q19 RR

(c) LBM D3Q19 HRR

(d) FastS `3dcart`

Figure 31: 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.

LBM-BGK and LBM-RR achieve the same performance. On the other hand, due to the introduction of the computation of non-local operations in the algorithm of the LBM-HRR, the performance is decreased with respect to the other model. Yet, the LBM-HRR is still about 2.25 times faster than the cartesian version of the Navier-Stokes solver. All these results are in perfect agreement with the tendencies outlined by the Roofline model (see Table 6). Note that the performance figures of the NS solvers 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 mores expensive than the explicit ones.

In order to get more insight into the performance of each solver, Figure 33 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 architecture

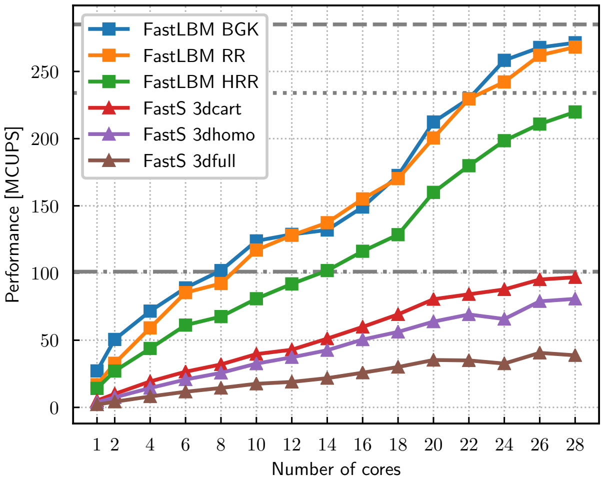


Figure 32: Comparison of the performance of each numerical scheme on one single BDW node. The horizontal lines correspond to the rooflines given in Table 6. (- - - -) FastLBM BGK and RR, (⋯⋯) FastLBM HRR, and (— · —) FastS `3dcart`.

to perform one iteration on one cell of the mesh:

$$t_{\text{eff}} = \frac{t_{\text{CPU}} \times N_{\text{threads}}}{n_{\text{substep}} \times 200^3}. \tag{60}$$

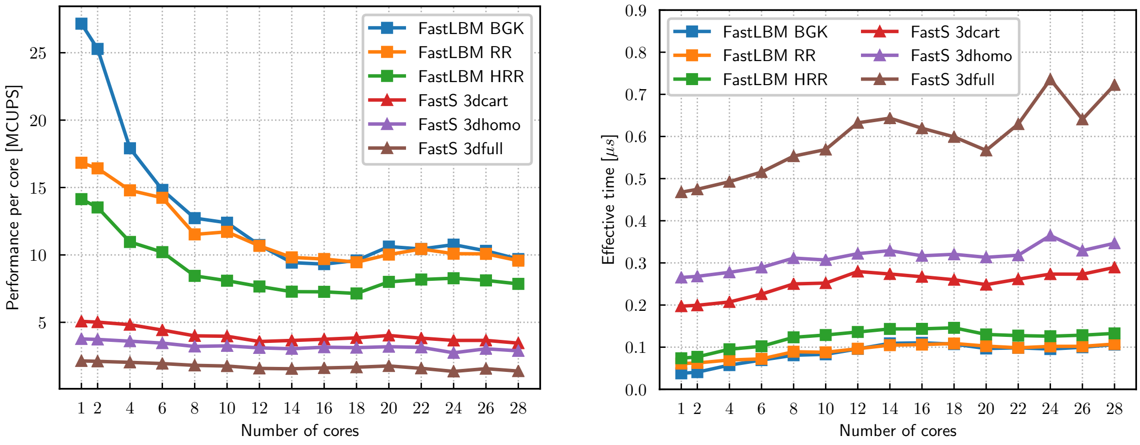
The results of Figure 33 (a) indicate that the lattice Boltzmann models can iterate over 10 million cells per core when the node is fully loaded. In comparison, the cartesian kernel of the Navier-Stokes solver only performs one iteration over 3.3 million cells per core. One should keep in mind that in 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 solver but not one entire iteration as it is the case for lattice Boltzmann solvers.

Figure 33 (b) shows that a cell iteration is performed in approximately 0.1 μs which confirms the fact that the performance, expressed in MLUPS, is the inverse of the effective time. 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%. However, when the node is fully loaded, the figures of Table 6 are perfectly recovered. Therefore, one can state that, in comparison to a well-optimised Navier-Stokes cartesian core, lattice Boltzmann methods are about 2.2 to 3 times faster. Now, with respect to a fully curvilinear NS solver, LB methods are between 5 to 8 times faster.

In terms of strong scaling, Figure 33 (a) suggests that the Navier-Stokes and lattice Boltzmann solvers do not behave in the same way. Figure 34 displays the speedup curve as well as the efficiency of the solvers as functions of the number of cores. The efficiency E is computed by relating the computational time per iteration t_{CPU} on a given number of cores N_P its counterpart on one single core t_1:

$$E = \frac{N_1}{N_P}\frac{t_1}{t_{\text{CPU}}} = \frac{1}{N_P}\frac{t_1}{t_{\text{CPU}}}. \tag{61}$$

The speedup curve of Figure 34 clearly shows that the Navier-Stokes solvers, regardless of their grid-specific version, have a better scaling than the lattice Boltzmann models. Indeed, while the efficiency is about 70%

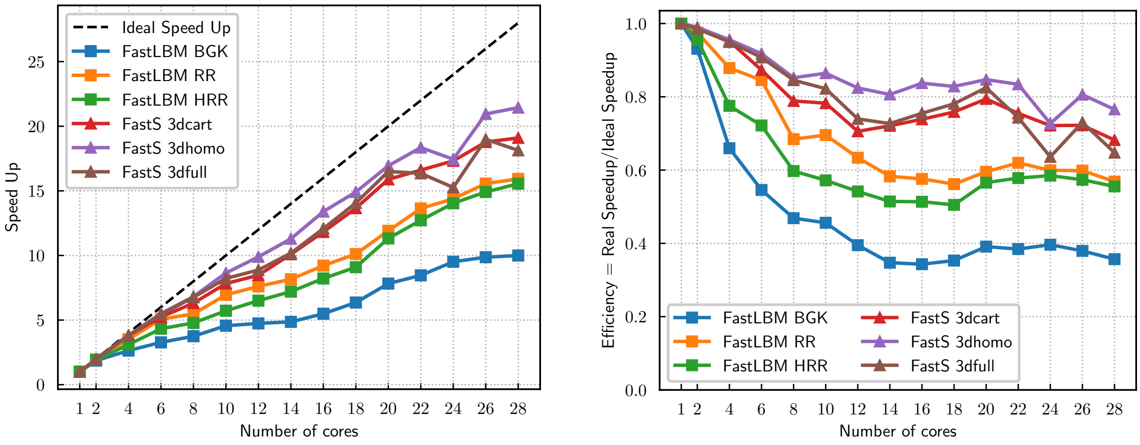


(a) Performance [MCUPS] per core

(b) Effective time [μs]

Figure 33: Performance comparison of the solver in terms of MCUPS per core and effective time.

to 80% on a full node for the Navier-Stokes algorithm, it achieves only 40% to 60% depending on the collision model used by the lattice Boltzmann solver. Such disparity might be explained by the fact that the LB "Stream & Collide" scheme is memory driven thereby increasing the number of cores does not lead to the same 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 classic BGK collision model.



(a) Scalability (strong scaling)

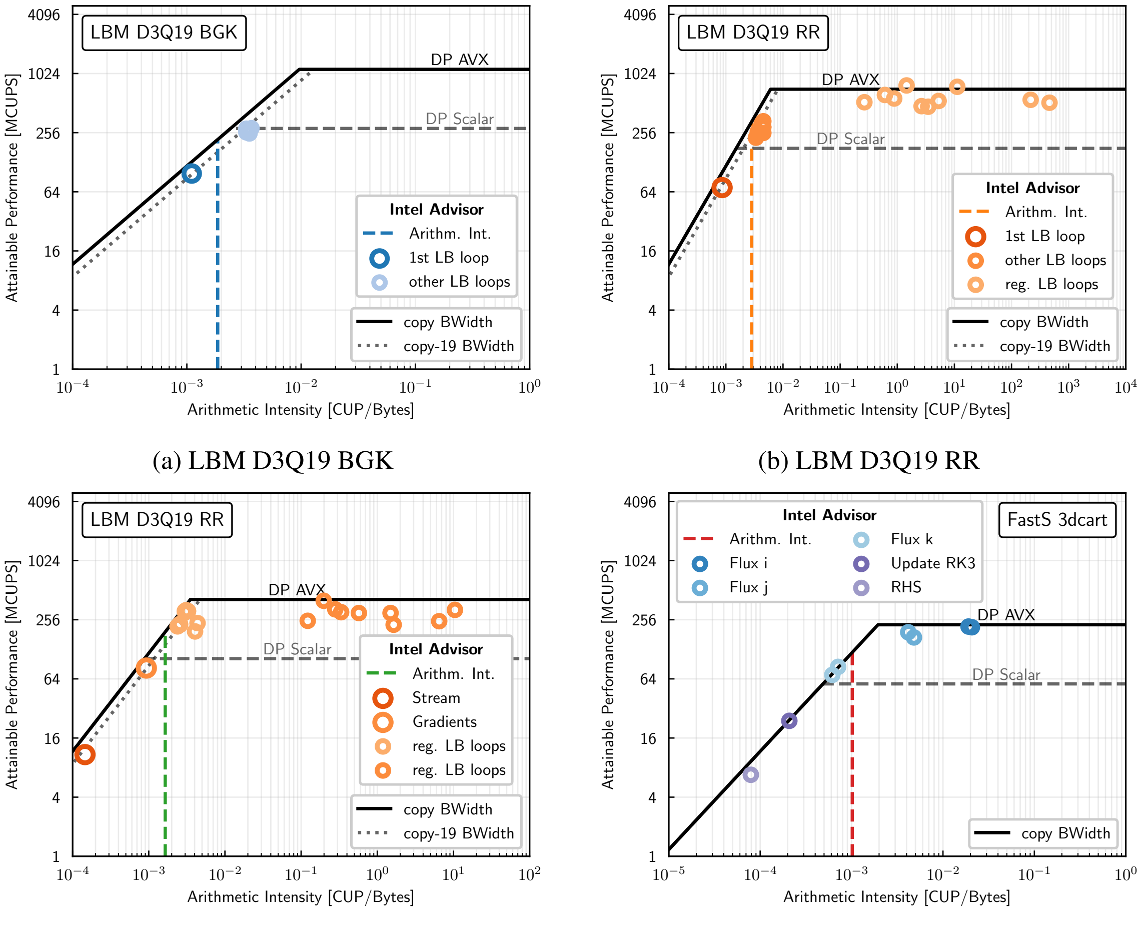
(b) Parallel efficiency

Figure 34: Comparison of the Navier-Stokes and lattice Boltzmann solvers in terms of scalability (a) and parallel efficiency (b).

To conclude the single-node performance measurements, the Roofline models obtained by profiling the present implementations with the Intel's Advisor software are compared to the theoretical ones derived in Section 5.2. One of the major advantages of Intel Advisor is that it can measure the performance of each loop in the code and therefore place them on the plot of the Roofline model. Figure 35 shows the corresponding results for each of the numerical schemes studied in this section. The theoretical results of the Roofline model derived in Section 5.2 are represented by the grey and black lines while the measurements made with

Intel Advisor are depicted by circles.

From a general point of view, it can be seen that all the implemented loops are located on the rooflines meaning that their corresponding code is optimised up to the hardware limit. Figure 35 (a) confirms the fact that the LBM-BGK is completely memory bound since all the loops hit 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 are initialised, the computations reach the peak FLOP performance roofline. Finally, Figure 35 (d) 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 according to Figure 30), then in the j-direction (or y respectively) and in the i-direction (corresponding to the x coordinate). The values being loaded 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.



(a) LBM D3Q19 BGK

(b) LBM D3Q19 RR

(c) LBM D3Q19 HRR

(d) FastS `3dcart`

Figure 35: Comparison between the theoretical Roofline models and the measurement performed with Intel Advisor.

5.4. *Time to solution study*

This last section aims at bridging all the results of the previous sections. Indeed, in Sections 3 and 4 the influence of the numerical scheme as well as the number of grid points per characteristic flow structure were discussed. However, from an engineering point of view, the relevant metric for comparing the lattice

Boltzmann and Navier-Stokes methods is the “time to solution” which corresponds to the time needed for each method to achieve a certain result. In [4], the authors proposed to compute the number of operation required by each numerical scheme in order to compute the same physical time. Here, it is proposed to extend this idea since competitive time to solution not only depends on the number of grid points (or equivalently the number of operations) but also on the update cost per mesh point.

The time to solution can be defined as follows:

$$T_{\text{CPU}} = \frac{T c_0}{\Delta x} \frac{t_{\text{eff}} N_{\text{ppw}}}{\text{CFL}} \tag{62}$$

where T is the physical time intended to be simulated, c_0 is the speed of sound, Δx the grid spacing, t_{eff} the effective cost per iteration and per mesh point of the corresponding method introduced in Equation (60), N_{ppw} is the number of grid points and CFL is the CFL number of the method. Note that the CFL number is only relevant for NS schemes since it is fixed for the LB schemes.

It is now proposed to return on the numerical test cases of Section 4 and to complete the comparisons by taking into account the results of the discussion on CPU performance. In the following, the time to solution metric for the Navier-Stokes method is only provided for the cartesian version of ONERA’s FastS solver. The results can then easily be extrapolated to the other versions by applying the corresponding multiplying factor (see Section 5.3).

Plane monochromatic acoustic wave. In the case of acoustic waves, it has been shown that in order to rigorously compare each numerical methods, one has to compare them at a given tolerated dispersion or dissipation error. Indeed, numerical schemes in CAA necessitate very low dispersion or dissipation errors to propagate the acoustic fluctuations over long distances. Therefore, the definition of the time to solution has to be revised leading to:

$$T_{\text{CPU,err}} = \frac{T c_0}{\Delta x} \frac{t_{\text{eff}} N_{\text{ppw,err}}}{\text{CFL}_{\text{err}}} \tag{63}$$

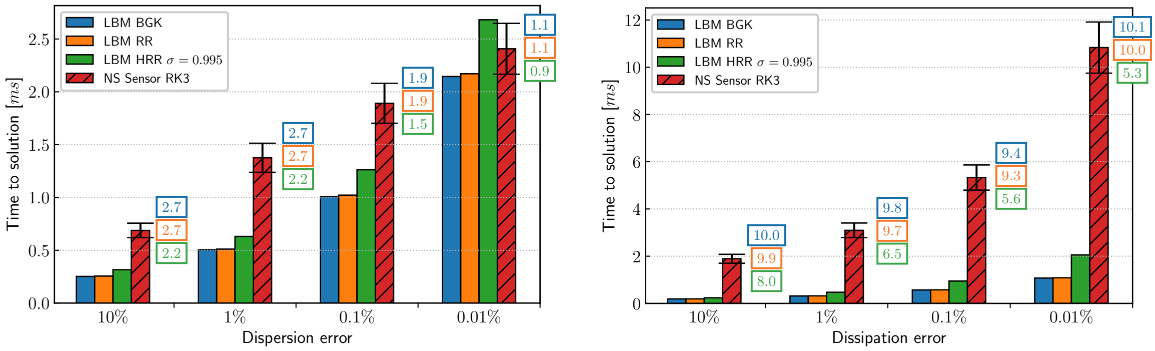
where the “err” subscript denotes the required value of the parameter to reach the corresponding error level. As one can see, the time to solution $T_{\text{CPU,err}}$ directly depends on the ratio between the number of points per wavelength and the CFL number.

In the case of Navier-Stokes schemes, where the CFL number is a free parameter, the $\frac{N_{\text{ppw,err}}}{\text{CFL}_{\text{err}}}$ ratio has to be taken as small as possible to minimise the total CPU time. According to the results of Figure 14, this ratio varies by only 10% around its value when $\text{CFL}_{NS} = \text{CFL}_{LBM} = 1/\sqrt{3}$. For this reason, the results are given with a 10% uncertainty margin to account for this slight variability.

Figure 36 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.

In terms of dissipation, 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 18 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 if offer speedups up between 5 and 10 over the whole range of error levels considered. This shows that the



(a) Dispersion error

(b) Dissipation error

Figure 36: Time to solution for the plane monochromatic acoustic wave.

LBM contains intrinsic and serious acoustic capabilities thereby enabling the method to propagate acoustic fluctuations at very low computational cost.

Plane monochromatic shear wave. Concerning shear waves, the methods are also compared at a given tolerated error level. However, this time, according to the results of Figure 19, the CFL number seems to have little influence on the dissipative behavior of the NS schemes. Therefore, the value of CFL = 0.8 is retained in the following since it ensures a minimal $\frac{N_{\mathrm{ppw,err}}}{\mathrm{CFL}_{\mathrm{err}}}$ ratio.

Figure 37 shows the time to solution for the LB and NS methods when propagating the shear wave for 100 periods. From a global point of view, the lattice Boltzmann methods are once again faster than the Navier-Stokes solver. However, unlike the case of the acoustic wave, the relevance of the LBM solver w.r.t the Navier-Stokes solver highly depends on the collision model. Indeed, while the standard LBM-BGK model is almost two times faster than the NS scheme, this model is also more unstable in areas of high shear. Therefore, when switching to regularised collision models for which the stability is improved, the final speedup is only between 1.1 to 1.7. The LBM is therefore still faster but to a lesser extent than in the case of acoustic waves.

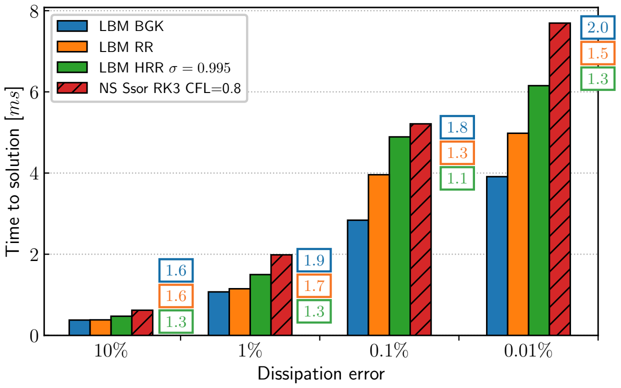


Figure 37: Time to solution for the plane monochromatic shear wave.

Convected vortex. The time to solution obtained when considering plane monochromatic waves help to draw some conclusion on the intrinsic behavior of each method. However, in real applications, all the modes are mixed and flow conditions might be non-linear. Therefore, the time to solution metric is now applied in the context of a convected vortex.

First, the methods are compared on the same mesh. Indeed, as shown by Figure 24, for well resolved cases (i.e. $N \geq 100$) all the the solutions are superimposable without being distinguishable to the naked eye. Therefore, from an engineering point of view were general flow tendencies are studied, this comparison at fixed mesh totally makes sense. Table 7 reports the CPU time required by each method to perform 5 advection cycles in a fully periodic box of size $[N\Delta x, N\Delta x, 10\Delta x]$. To ease the comparison, the relative time $T_{\text{CPU,NS}}/T_{\text{CPU,LBM}}$ is also provided. As one would expect, when considering the same mesh size, the results of the single-node scaling are retrieved: the LBM is about 2.3 to 2.8 times fast than the Navier-Stokes scheme. Thus, the general flow patterns can be expected to be computed much faster with an LB solver than with a Navier-Stokes solver on Cartesian grids. This last remark makes sense in the context of complex aerodynamic simulations where immersed boundary conditions are used.

Solver	CFL	T_{CPU} [s]		$T_{\text{CPU,NS}}/T_{\text{CPU,LBM}}$
		$N = 100$	$N = 200$	
FastLBM BGK	$1/\sqrt{3}$	91.9	735.3	2.86
FastLBM RR	$1/\sqrt{3}$	93.0	744.2	2.82
FastLBM HRR	$1/\sqrt{3}$	114.9	919.4	2.28
FastS `3dcart`	$1/\sqrt{3}$	262.5	2103.8	-

Table 7: Time to solution for the convected vortex test case when all the solvers use the same mesh.

Another requirement of numerical methods is their ability to resolve flow fields with very little error compared to a reference solution. Therefore, Table 7 reports the CPU time required by each method to achieve a certain error target on the L^2 norm of the velocity field as well as the number of grid points required in each direction N_{err}.

As a matter of fact, the greater the constraint, the greater the number of points and therefore the longer the computational time. Surprisingly, compared to the monochromatic plane shear wave, the NS scheme with sensor performs best for all error levels. However, this is only true because of the presence of the binary function Φ acting on the dissipation terms of the AUSM scheme. Indeed, if ones compared the FastS `3dcart` - AUSM line with the ones of the LB schemes, it can be seen that once again the results depend on the desired error level. For low error constraints $L^2(\mathbf{u}) \leq 10^{-2}$, the lattice Boltzmann models are slightly faster than the NS-AUSM scheme. However, when it comes to convergence down to the order of machine accuracy, it is clear that Navier-Stokes methods implemented in ONERA's FastS solver are more appropriate.

It is essential to note that this conclusion is 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 the merit of underlining the fact that by taking advantage of the combinations of spatial and temporal patterns and MUSCL-type reconstructions in the framework of finite-volume Navier-Stokes schemes, one can sometimes possible to obtain more efficient patterns than those offered by the LBM.

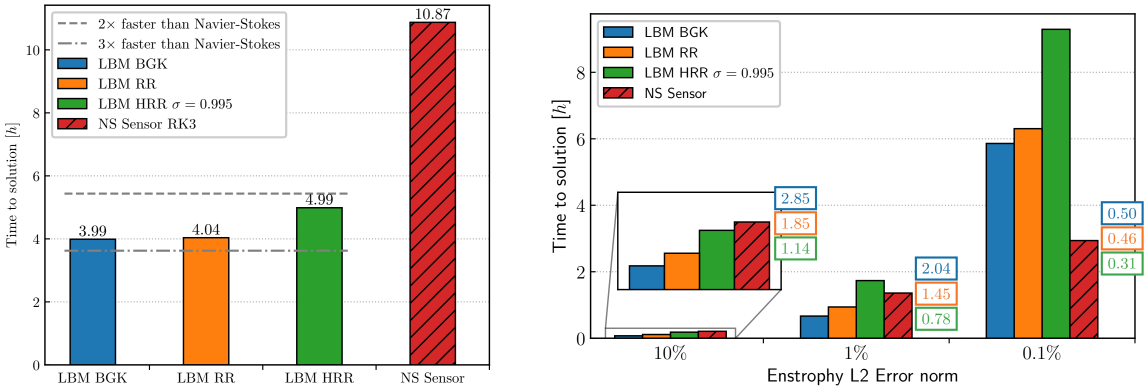
Solver	T_{CPU} [s] at fixed error level and (N_{err})			
	$L^2(\mathbf{u}) \leq 10^{-1}$	$L^2(\mathbf{u}) \leq 10^{-2}$	$L^2(\mathbf{u}) \leq 10^{-3}$	$L^2(\mathbf{u}) \leq 10^{-4}$
FastLBM BGK	9.5 (47)	310.2 (150)	9850.2 (475)	310196 (1500)
FastLBM RR	9.6 (47)	313.9 (150)	9969.5 (475)	313954 (1500)
FastLBM HRR	11.9 (47)	387.6 (150)	12317.0 (475)	387880 (1500)
FastS `3dcart` - AUSM	32.9 (50)	349.9 (110)	3726.9 (242)	36977.2 (520)
FastS `3dcart` - Sensor	7.01 (30)	56.8 (60)	604.8 (132)	13105.9 (368)

Table 8: Time to solution for the convected vortex test case for different error levels.

Taylor-Green Vortex. Following the discussion of the time to solution in the context of a convected vortex computation, the same metric is applied to the simulation of turbulent flows with implicit LES techniques such as for the 3D Taylor Green vortex.

Figure 38 (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. As in the case of the convected vortex, all the LB models, regardless of their collision operator, are at least two times faster than the Navier-Stokes solver. Referring to the results in Figures 26 and 28 it can be concluded that the main flow phenomena are well captured with a slight advantage in terms of computation time for the LBM even though the regularised models tend to introduce some numerical dissipation in the high wave-number range. This confirms the tendencies outline with the convected vortex test case.

In Figure 38 (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. 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 shown in Figure 27.



(a) TGV on a 512^3 grid.

(b) Comparison at fixed error level

Figure 38: Time to solution for the 3D Taylor Green vortex test case.

6. Conclusion

In the present work an extended comparison between the lattice Boltzmann method and standard finite-volume Navier-Stokes schemes was performed. Three complementary aspects of the numerical methods were studied.

First, an extended Von Neumann analysis was applied to the discretised Navier-Stokes equations and to lattice Boltzmann models in order to their dispersion and dissipation properties. Thanks to its natural extension in 2D, spectral maps of the dissipation error of each scheme were analysed and compared. This highlighted anisotropic behavior for both the lattice Boltzmann and Navier-Stokes schemes. Moreover, the low dissipative capabilities of the lattice Boltzmann models have been shown to highly depend on the collision model as well as on the physical mode of interest. All in all, the lattice Boltzmann method shows very little dissipation on acoustic modes, even with regularised collision operators. However, when it comes to the shear mode, only the BGK model remains competitive with the NS schemes while the regularised ones suffer from important hyperviscous effects. These results were confirmed over the whole spectral plane.

The numerical simulation of plane monochromatic waves helped to recover the trends of the linear stability analysis as well as to validate their 2D extension. In addition, two more realistic test cases were considered in order to draw some conclusions in the context of unsteady computational fluid dynamics. For each of these test case, a thorough discussion on the tuning parameters of the Navier-Stokes schemes was made and general guidelines about their optimal value were drawn. As a result, the minimal number of points per characteristic flow structure was outlined. From a general point of view, for acoustic problems, regardless of the collision operator LB schemes require far less points per wavelength than Navier-Stokes models. In the case of shear-driven flows, the LBM is very advantageous at low resolutions.

Finally, the CPU performance of both numerical methods was investigated. The Roofline model allowed to show theoretically that the LB algorithm can achieve speedups up to 3 with respect to a well tuned Navier-Stokes cartesian solver. These trends were confirmed by performing a single node scaling of all the solvers. The conclusion of the comparison between both approaches was given by introducing a time to solution metric. It has been shown that speedups up to 10 can be achieved on aeroacoustic problems. Moreover, general flow patterns can be computed much faster with an LB solver than with a Navier-Stokes solver on Cartesian grids. However, when looking for machine precision, Navier-Stokes methods seem to be more competitive because of their flexibility since one can combine different order in space and time to achieve the desired order of accuracy.

7. Acknowledgements

This work was performed by using ONERA's HPC resources on the Sator cluster. The authors are grateful to N. Alferez for his support on Intel Advisor.

Appendix A. Linearized athermal Navier–Stokes equations

The aim of this appendix is to give some details about the linearisation of the 3D isothermal Navier-Stokes equations introduced in Section 3.1.

By substituting the decomposition of Equation (29) into Equations (25) and (27) and neglecting the high-order fluctuations terms, one obtains the following system of equations:

$$\left\{\begin{array}{rcl}
\dfrac{\partial \rho'}{\partial t} + \dfrac{\partial}{\partial x_1}\left[\overline{\rho} u_1' + \rho' \overline{u_1}\right] + \dfrac{\partial}{\partial x_2}\left[\overline{\rho} u_2' + \rho' \overline{u_2}\right] & = & 0 \\
\overline{\rho}\dfrac{\partial u_1'}{\partial t} + \overline{\rho}\,\overline{u_1}\dfrac{\partial u_1'}{\partial x_1} + \overline{\rho}\,\overline{u_2}\dfrac{\partial u_1'}{\partial x_2} + \frac{\partial p'}{\partial x_1} & = & \frac{\partial}{\partial x_j}(\tau'_{1j}) \\
\overline{\rho}\dfrac{\partial u_2'}{\partial t} + \overline{\rho}\,\overline{u_1}\dfrac{\partial u_2'}{\partial x_1} + \overline{\rho}\,\overline{u_2}\dfrac{\partial u_2'}{\partial x_2} + \frac{\partial p'}{\partial x_2} & = & \frac{\partial}{\partial x_j}(\tau'_{2j}) \\
\dfrac{\partial p'}{\partial t} + \dfrac{\partial}{\partial x_1}\left[p' \overline{u_1}\right] + \dfrac{\partial}{\partial x_2}\left[p' \overline{u_2}\right] & = & -\gamma \dfrac{\partial}{\partial x_j}\left[\overline{p} u_j'\right]
\end{array}\right. \tag{A.1}$$

where the linearised shear stress tensor is given by:

$$\tau'_{ij} = \overline{\rho}\nu\left[\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) - \frac{2}{3}\frac{\partial u_k'}{\partial x_k}\delta_{ij}\right] + \overline{\rho}\zeta\frac{\partial u_k'}{\partial x_k}\delta_{ij}. \tag{A.2}$$

Introducing the perturbed vector of unknowns $\mathbf{U}' = (\rho', \overline{\rho} u_1', \overline{\rho} u_2', p')^t$, the system of Equations (A.1) can be recast as:

$$\frac{\partial \mathbf{U}'}{\partial t} + \frac{\partial}{\partial x_1}\left[\mathbf{E}_e - \mathbf{E}_v\right] + \frac{\partial}{\partial x_1}\left[\mathbf{F}_e - \mathbf{F}_v\right] = \mathbf{0} \tag{A.3}$$

where $\mathbf{E}_e'$ and $\mathbf{F}_e'$ are the Eulerian fluxes and $\mathbf{E}_v'$ and $\mathbf{F}_v'$ the viscous flux given by:

$$\mathbf{E}_e' = \begin{pmatrix} \rho'\overline{u_1} + \overline{\rho} u_1' \\ p' + \overline{\rho}\,\overline{u_1} u_1' \\ \overline{\rho}\,\overline{u_1} u_1' \\ \overline{u_1} p' + \gamma \overline{p} u_1' \end{pmatrix} \quad \mathbf{F}_e' = \begin{pmatrix} \rho'\overline{u_2} + \overline{\rho} u_2' \\ \overline{\rho}\,\overline{u_2} u_1' \\ p' + \overline{\rho}\,\overline{u_2} u_2' \\ \overline{u_2} p' + \gamma \overline{p} u_2' \end{pmatrix}, \tag{A.4}$$

and

$$\mathbf{E}_v = \begin{pmatrix} 0 \\ \tau'_{11} \\ \tau'_{12} \\ 0 \end{pmatrix} \quad \mathbf{F}_v = \begin{pmatrix} 0 \\ \tau'_{21} \\ \tau'_{22} \\ 0 \end{pmatrix}. \tag{A.5}$$

The last step towards the linearisation of the resulting equations is to express the spatial derivatives in Equation (A.3) as functions of the unknown vector $\mathbf{U}'$. After some algebra, one finally gets:

$$\frac{\partial \mathbf{U}'}{\partial t} + \mathbf{M}_{x_1}\frac{\partial \mathbf{U}'}{\partial x_1} + \mathbf{M}_{x_2}\frac{\partial \mathbf{U}'}{\partial x_2} = \mathbf{0} \tag{A.6}$$

where the $\mathbf{M}_{x_1}$ ans $\mathbf{M}_{x_2}$ are matrices whose generic expressions are:

$$\mathbf{M}_{x_1} = \begin{pmatrix} \overline{u_1} & 1 & 0 & 0 \\ 0 & \overline{u_1} - \left(\frac{4}{3}\nu + \zeta\right)\frac{\partial}{\partial x} & \left(\frac{2}{3}\nu - \zeta\right)\frac{\partial}{\partial y} & 1 \\ 0 & -\nu\frac{\partial}{\partial y} & \overline{u_1} - \nu\frac{\partial}{\partial x} & 0 \\ 0 & c_0^2 & 0 & \overline{u_1} \end{pmatrix}, \tag{A.7}$$

and

$$\mathbf{M}_{x_2} = \begin{pmatrix} \overline{u_2} & 0 & 1 & 0 \\ 0 & \overline{u_2} - \nu\frac{\partial}{\partial y} & -\nu\frac{\partial}{\partial x} & 0 \\ 0 & \left(\frac{2}{3}\nu - \zeta\right)\frac{\partial}{\partial x} & \overline{u_2} - \left(\frac{4}{3}\nu + \zeta\right)\frac{\partial}{\partial y} & 1 \\ 0 & 0 & c_0^2 & \overline{u_2} \end{pmatrix}. \tag{A.8}$$

where $c_0 = \gamma \frac{\overline{p}}{\overline{\rho}}$.

By analogy with the semi-discrete form of Equation (30), Equation (A.9) can be recast so as to reveal the Jacobian:

$$\frac{\partial \mathbf{U}'}{\partial t} = \mathcal{J}\mathbf{U}' \quad \text{where} \quad \mathcal{J} = -\left[\mathbf{M}_{x_1}\frac{\partial}{\partial x_1} + \mathbf{M}_{x_2}\frac{\partial}{\partial x_2}\right] \tag{A.9}$$

Using Equation (31), the perturbed vector of unknowns $\mathbf{U}' = (\rho', \overline{\rho}u_1', \overline{\rho}u_2', p')^t$ is written under the form of monochromatic plane waves:

$$\mathbf{U}' = \widehat{\mathbf{U}} \exp\left(\mathrm{i}\left(\boldsymbol{k}\cdot\boldsymbol{x} - \omega t\right)\right) \tag{A.10}$$

where $\mathrm{i}^2 = -1$, $\widehat{\mathbf{U}} = (\widehat{\rho}, \overline{\rho}\widehat{u_1}, \overline{\rho}\widehat{u_2}, \widehat{p})^t$ is the vector of the complex amplitudes. Then, injecting (**??**) in (A.9) leads to the following eigenvalue problem:

$$\omega\hat{\mathbf{U}}' = \mathbf{M}^{\mathrm{NS},\dagger}\hat{\mathbf{U}}' \tag{A.11}$$

where the $\mathbf{M}^{\mathrm{NS},\dagger}$ matrix is defined by:

$$\mathbf{M}^{\mathrm{NS},\dagger} = -\left[k_{x_1}\widetilde{\mathbf{M}_{x_1}} + k_{x_2}\widetilde{\mathbf{M}_{x_2}}\right]. \tag{A.12}$$

$\widetilde{\mathbf{M}_{x_1}}$ and $\widetilde{\mathbf{M}_{x_2}}$ have the exact same expressions as in Equations (A.7) and (A.8) except that ∂_{x_i} has been replaced by $\mathrm{i}k_{x_i}$.

Appendix B. von Neumann analysis of LB schemes

The von Neumann analysis of the lattice Boltzmann scheme requires the linearization of all nonlinear terms around a global equilibrium state . Therefore, the distribution functions are expanded as:

$$f_i = \overline{f_i} + f_i' \tag{B.1}$$

where the global state defined by $\overline{f_i}(\overline{\rho}, \overline{\boldsymbol{u}})$ is constant in both space and time, and where f_i' are the fluctuating distribution functions.

In the lattice Boltzmann framework non-linearities stem from the collision operator since the latter involves the equilibrium state f_i^{eq} which depends on f_i through the macroscopic quantities ρ and $\boldsymbol{u}$. By performing a first-order Taylor expansion around the global state, one gets:

$$\Omega_i(f_i) = \Omega_i(\overline{f_i}) + \underbrace{\left.\frac{\partial \Omega_i}{\partial f_j}\right|_{f_j=\overline{f_j}}}_{J_{ij}^{\Omega}} f_j' + O\left(f_i'^2\right) \tag{B.2}$$

where Einstein's summation convention is used on index j. The linearization process involves a Jacobian matrix $\mathbf{J}^{\mathbf{\Omega}}$ which depends on the collision operator.

By injecting Equations (B.1) and (B.2) in both the DVBE (8) and the lattice Boltzmann scheme (16) two perturbed equations are obtained. The first one is the perturbed discrete velocity Boltzmann equation:

$$\frac{\partial f_i'}{\partial t} + \boldsymbol{\xi}_i \frac{\partial f_i'}{\partial x_i} = J_{ij}^{\Omega} f_j', \tag{B.3}$$

and the second corresponds to the perturbed space and time discrete lattice Boltzmann scheme:

$$f_i'(\boldsymbol{x} + \boldsymbol{\xi}_i, t+1) = \left[\delta_{ij} + J_{ij}^{\Omega}\right] f_j'. \tag{B.4}$$

In order to obtain the plane wave solution of Equations (B.3) and (B.4), the perturbations are sought as complex monochromatic plane waves:

$$f_i' = \hat{f}_i \exp\left(i\left(\boldsymbol{k}\cdot\boldsymbol{x} - \omega t\right)\right) \tag{B.5}$$

where $\hat{f}_i \in \mathbb{C}$, $\boldsymbol{k}$ is a real dimensionless wavevector and ω is the complex dimensionless pulsation of the wave. Injecting Equation (B.7) into the perturbed Equations (B.3) and (B.4) leads to two eigenvalue problems of size q :

$$\omega\widehat{\mathbf{F}} = \mathbf{M}^{\mathrm{C}}\widehat{\mathbf{F}} \tag{B.6}$$

in the continuous case, and:

$$e^{-i\omega}\widehat{\mathbf{F}} = \mathbf{M}^{\mathrm{D}}\widehat{\mathbf{F}} \tag{B.7}$$

in the space and time discrete case. The definition of the corresponding $\mathbf{M}^{\mathrm{C}}$ and $\mathbf{M}^{\mathrm{D}}$ can be found in the following sections.

Appendix B.1. Discrete velocity Boltzmann equation

Starting from the perturbed discrete velocity Boltzmann equation:

$$\frac{\partial f_i'}{\partial t} + \boldsymbol{\xi}_i \frac{\partial f_i'}{\partial x_i} = J^{\Omega}_{ij} f_j', \tag{B.8}$$

the monochromatic plane wave form of the perturbed distribution functions defined by Equation (B.7) is adopted, leading to:

$$\left(-i\omega + i\boldsymbol{\xi}_i\cdot\boldsymbol{k}\right) f_i' = J^{\Omega}_{ij} f_j'. \tag{B.9}$$

By recasting the latter Equation under the form of an eigenvalue problem, on gets:

$$\omega\widehat{\mathbf{F}} = \left[(\boldsymbol{\xi}_i\cdot\boldsymbol{k})\delta_{ij} + J^{\Omega}_{ij}\right]\widehat{\mathbf{F}} \tag{B.10}$$

where the generic term of the collision jacobian J^{Ω}_{ij} has to be determined. As it is classically done, the BGK approximation is used for Ω_i:

$$\Omega_i = -\frac{1}{\tau}\left(f_i - f_i^{eq}\right) \tag{B.11}$$

In this case, only the equilibrium distribution function is non-linear in $\overline{f_i}$ which yields:

$$J_{ij} = -\frac{1}{\tau}\left(\delta_{ij} - J^{eq}_{ij}\right) \text{ where } J^{eq}_{ij} = \left.\frac{\partial f_i^{eq}}{\partial f_j}\right|_{f_j=\overline{f_j}}. \tag{B.12}$$

$\mathbf{J}^{eq}$ is often referred to as the equilibrium jacobian. Finally, the eigenvalue problem of the discrete velocity Boltzmann equation is:

$$\omega\widehat{\mathbf{F}} = \left[(\boldsymbol{\xi}_i\cdot\boldsymbol{k})\delta_{ij} - \frac{i}{\tau}\left(\boldsymbol{\delta} - \mathbf{J}^{eq}\right)\right]\widehat{\mathbf{F}} = \mathbf{M}^{\mathrm{C}}\widehat{\mathbf{F}}. \tag{B.13}$$

Appendix B.2. BGK collision operator

Starting from the perturbed lattice Boltzmann equation where the collision term has already been linearised:

$$f_i'(\boldsymbol{x} + \boldsymbol{\xi}_i, t + 1) = \left[\delta_{ij} + J_{ij}^{\Omega}\right] f_j'. \tag{B.14}$$

the monochromatic plane wave form of the perturbed distribution functions defined by Equation (B.7) is adopted, leading to:

$$\exp\left[\mathrm{i}\left(\boldsymbol{\xi}_i \cdot \boldsymbol{k} - \omega\right)\right] f_i' = J_{ij}^{\Omega} f_j'. \tag{B.15}$$

which can be recast under the form of an eigenvalue problem:

$$\exp\left(-\mathrm{i}\omega\right) \widehat{\mathbf{F}} = \left[\exp\left(-\mathrm{i}\boldsymbol{\xi}_i \cdot \boldsymbol{k}\right) \boldsymbol{\delta}\right] \left[\boldsymbol{\delta} + \mathbf{J}\right] \widehat{\mathbf{F}}. \tag{B.16}$$

When chossing the BGK collision model, the collision model in the discrete case is given by:

$$\Omega_i = -\frac{1}{\tau + 1/2}\left(f_i - f_i^{eq}\right) \tag{B.17}$$

which yields to the following Jacobian:

$$J_{ij} = -\frac{1}{\tau + 1/2}\left(\delta_{ij} - J_{ij}^{eq}\right) \quad \text{where} \quad J_{ij}^{eq} = \left.\frac{\partial f_i^{eq}}{\partial f_j}\right|_{f_j = \overline{f_j}}. \tag{B.18}$$

Finally, one gets:

$$\exp\left(-\mathrm{i}\omega\right) \widehat{\mathbf{F}} = \left[\exp\left(-\mathrm{i}\boldsymbol{\xi}_i \cdot \boldsymbol{k}\right) \boldsymbol{\delta}\right] \left[\boldsymbol{\delta} - \frac{1}{\tau + 0.5}\left(\boldsymbol{\delta} - \mathbf{J}^{eq}\right)\right] \widehat{\mathbf{F}} = \mathbf{M}^{\mathrm{D}} \widehat{\mathbf{F}}. \tag{B.19}$$

Appendix B.3. Regularised collision operators

The derivation of the time-advance matrices of the recursive and hybrid recursive regularised collision models can be found in [7, 53] and follows the same methodology as in the BGK case.

The matrices appearing in the eigenvalue problems of regularised collision models rely on the following definition which corresponds to the regularisation of the second non-equilibrium moment:

$$M_{ij}^{\mathrm{PR}} = e^{-i\mathbf{k}\cdot\mathbf{e}_i}\left[J_{ij}^{eq,N} + \left(1 - \frac{1}{\overline{\tau}}\right)\left(\delta_{ij} - J_{ij}^{eq,N}\right) h_{ik}\right]. \tag{B.20}$$

In this equation, the implicit summation is done over the index k and one has:

$$h_{ik} = \frac{w_i}{2c_s^4} \boldsymbol{\mathcal{H}}_i^{(2)} : \boldsymbol{\mathcal{H}}_k^{(2)}. \tag{B.21}$$

Consequenty, the time-advance matrix of the RRN_r collision model is given by:

$$M_{ij} = M_{ij}^{\mathrm{PR}} + e^{-i\mathbf{k}\cdot\mathbf{e}_i}\left(1 - \frac{1}{\overline{\tau}}\right)\sum_{n=3}^{N_r} \frac{w_i}{n!c_s^{2n}} \boldsymbol{\Lambda}_{1,j}^{(n)} : \boldsymbol{\mathcal{H}}_i^{(n)} \tag{B.22}$$

where the full expression of $\boldsymbol{\Lambda}_{1,j}^{(n)}$ can be found in [7].

For the HRR collision model (including the corrective term), one has:

$$M_{ij} = e^{-i\mathbf{k}\cdot\mathbf{e}_i}\left[J_{ij}^{eq,N} + \left(1 - \frac{1}{\overline{\tau}}\right) G_{ij} + \frac{\Delta t}{2}\Psi_{ij}\right]. \tag{B.23}$$

where the full expression of G_{ij} and Ψ_{ij} are given in [53].

References

[1] P. R. Spalart, V. Venkatakrishnan, On the role and challenges of CFD in the aerospace industry, The Aeronautical Journal 120 (1223) (2016) 209–232. doi:10.1017/AER.2015.10.

[2] R. Löhner, Towards overcoming the LES crisis, International Journal of Computational Fluid Dynamics 33 (3) (2019) 87–97. doi:10.1080/10618562.2019.1612052.

[3] 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.

[4] 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.

[5] P. Lallemand, L. S. Luo, Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability, Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics 61 (6) (2000) 6546–6562. doi:10.1103/PhysRevE.61.6546.

[6] 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.

[7] 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.

[8] M. B. Giles, I. Reguly, Trends in high-performance computing for engineering calculations, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 372 (2022) (2014) 20130319. doi:10.1098/rsta.2013.0319.

[9] 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:10.1177/10943420211006169.

[10] E. Manoha, B. Caruelle, Summary of the LAGOON solutions from the benchmark problems for airframe noise computations-III workshop, in: 21st AIAA/CEAS Aeroacoustics Conference, American Institute of Aeronautics and Astronautics Inc, AIAA, 2015. doi:10.2514/6.2015-2846.

[11] 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, American Institute of Aeronautics and Astronautics Inc, AIAA, 2019. doi:10.2514/6.2019-2555.

[12] M. F. Barad, J. G. Kocheemoolayil, C. C. Kiris, Lattice boltzmann and navier-stokes cartesian CFD approaches for airframe noise predictions, 23rd AIAA Computational Fluid Dynamics Conference, 2017 (2017). doi:10.2514/6.2017-4404.

[13] 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.

[14] A. Aniello, D. Schuster, P. Werner, J. Boussuge, M. Gatti, C. Mirat, L. Selle, T. Schuller, T. Poinsot, 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.

[15] M. Fiore, Influence of cavity flow on turbine aerodynamics, Ph.D. thesis, Institut Supérieur de l'Aéronautique et de l'Espace (2019).

[16] 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.

[17] 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.

[18] http://elsa.onera.fr/Cassiopee/.

[19] https://w3.onera.fr/FAST/.

[20] P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (2) (1981) 357–372. doi:10.1016/0021-9991(81)90128-5.

[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.

[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] 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).

[24] 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).

[25] P. S. Lowery, W. C. Reynolds, Numerical simulation of a spatially-developping, forced, plane mixing layer, Tech. Rep. TF26, Stanford University (1986).

[26] F. Daude, I. Mary, P. Comte, Self-Adaptive Newton-based iteration strategy for the LES of turbulent multi-scale flows, Computers & Fluids 100 (2014) 278–290. doi:10.1016/J.COMPFLUID.2014.04.028.

[27] 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.

[28] T. J. Coakley, Implicit upwind methods for the compressible Navier-Stokes equations, AIAA Journal 23 (3) (1985) 374–380. doi:10.2514/3.8923.

[29] 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.

[30] N. Alferez, I. Mary, E. Lamballais, Study of Stall Development Around an Airfoil by Means of High Fidelity Large Eddy Simulation, Flow, Turbulence and Combustion 2013 91:3 91 (3) (2013) 623–641. doi:10.1007/S10494-013-9483-7.

[31] 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.

[32] 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.

[33] T. Krueger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, E. M. Viggen, The Lattice Boltzmann Method: Principles and Practice, Graduate Texts in Physics, Springer, 2016.

[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.

[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] 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.

[38] 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.

[39] 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.

[40] 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.

[41] D. D'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, L. S. Luo, Multiple-relaxation-time lattice Boltzmann models in three dimensions, in: Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, Vol. 360, Philos Trans A Math Phys Eng Sci, 2002, pp. 437–451. doi:10.1098/rsta.2001.0955.

[42] 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.

[43] N. Frapolli, S. S. Chikatamarla, I. V. Karlin, Entropic lattice Boltzmann model for gas dynamics: Theory, boundary conditions, and implementation, Physical Review E 93 (6) (2016) 063302. doi:10.1103/PhysRevE.93.063302.

[44] J. Latt, B. Chopard, Lattice Boltzmann method with regularized pre-collision distribution functions, Mathematics and Computers in Simulation 72 (2) (2006) 165–168. doi:10.1016/j.matcom.2006.05.017.

[45] C. Coreixas, G. Wissocq, G. Puigt, J.-F. Boussuge, P. Sagaut, Recursive regularization step for high-order lattice Boltzmann methods, Physical Review E 96 (3) (2017) 033306. doi:10.1103/PhysRevE.96.033306.

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

[47] D. Ricot, S. Marié, P. Sagaut, C. Bailly, Lattice Boltzmann method with selective viscosity filter, Journal of Computational Physics 228 (12) (2009) 4478–4490. doi:10.1016/j.jcp.2009.03.030.

[48] S. Marié, X. Gloerfelt, Adaptive filtering for the lattice Boltzmann method, Journal of Computational Physics 333 (2017) 212–226. doi:10.1016/j.jcp.2016.12.017.

[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] 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.

[51] C. Bogey, C. Bailly, A family of low dispersive and low dissipative explicit schemes for flow and noise computations, Journal of Computational Physics 194 (1) (2004) 194–214. doi:10.1016/j.jcp.2003.09.003.

[52] 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.

[53] 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.

[54] Y. Feng, P. Boivin, J. Jacob, P. Sagaut, Hybrid recursive regularized thermal lattice Boltzmann model for high subsonic compressible flows, Journal of Computational Physics 394 (2019) 82–99. doi:10.1016/j.jcp.2019.05.031.

[55] C. Coreixas, High-order extension of the recursive regularized lattice Boltzmann method, Ph.D. thesis, Institut National Polytechnique de Toulouse (INP Toulouse) (2018).

[56] 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.

[57] 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.

[58] 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 (jul 2013). doi:10.1002/fld.3767.

[59] 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.

[60] 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.

[61] 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.
URL https://doi.org/10.1017/S0022112083001159

[62] 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.

[63] 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, Theoretical and Computational Fluid Dynamics (2020) 1–40doi:10.1007/s00162-020-00555-7.

[64] D. Foti, K. Duraisamy, An investigation of an implicit large-eddy simulation framework for the vorticity transport equations, in: 2018 Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics Inc, AIAA, 2018. doi:10.2514/6.2018-3407.

[65] N. Alferez, 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: Proceedings of November 11-12 2017, USA, Denver (Intel HPC Developer Conference), 2017.

[66] 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.

[67] M. Bauer, H. Köstler, U. Rüde, lbmpy: Automatic code generation for efficient parallel lattice Boltzmann methods, Journal of Computational Science 49 (2021) 101269. arXiv:2001.11806, doi:10.1016/j.jocs.2020.101269.

[68] 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.

[69] J. Habich, A performance engineering process for developing high performance lattice Boltzmann implementations, Ph.D. thesis, Technischen Fakultat der Universitat Erlangen-Nürnberg (2015).

[70] M. Wittmann, V. Haag, T. Zeiser, H. Köstler, G. Wellein, Lattice Boltzmann benchmark kernels as a testbed for performance analysis, Computers & Fluids 172 (2018) 582–592. arXiv:1711.11468, doi:10.1016/J.COMPFLUID.2018.03.030.