Local mesh refinement sensor for the lattice Boltzmann method

Y. Thorimbert[a,*], D. Lagrava[c], O. Malaspinas[b], B. Chopard[a], C. Coreixas[a], J. de Santana Neto[a], R. Deiterding[d], J. Latt[a]

[a] *Department of Computer Science, University of Geneva, 1227 Carouge, Switzerland*
[b] *HEPIA - Haute école du paysage, d'ingénierie et d'architecture, 1202 Geneva, Switzerland*
[c] *University of Canterbury, Civil and Natural Resources Engineering, 8041 Christchurch, New Zealand*
[d] *School of Engineering, University of Southampton, Boldrewood Campus, SO16 7QF, United Kingdom*

Abstract

A novel mesh refinement sensor is proposed for lattice Boltzmann methods (LBMs) applicable to either static or dynamic mesh refinement algorithms. The sensor exploits the kinetic nature of LBMs by evaluating the departure of distribution functions from their local equilibrium state. This sensor is first compared, in a qualitative manner, to three state-of-the-art sensors: (1) the vorticity norm, (2) the Q-criterion, and (3) spatial derivatives of the vorticity. This comparison shows that our kinetic sensor is the most adequate candidate to propose tailored mesh structures across a wide range of physical phenomena: incompressible, compressible subsonic/supersonic single phase, and weakly compressible multiphase flows. As a more quantitative validation, the sensor is then used to produce the computational mesh for two existing open-source LB solvers based on inhomogeneous, block-structured meshes with static and dynamic refinement algorithms, implemented in the Palabos and AMROC-LBM software, respectively. The sensor is first used to generate a static mesh to simulate the turbulent 3D lid-driven cavity flow using Palabos. AMROC-LBM is then adopted to confirm the ability of our sensor to dynamically adapt the mesh to reach the steady state of the 2D lid-driven cavity flow. Both configurations show that our sensor successfully produces meshes of high quality and allows to save computational time.

Keywords: Lattice Boltzmann, Automatic Mesh Refinement, Refinement Sensor, Palabos, AMROC

1. Introduction

The lattice Boltzmann method (LBM) is widely used in the field of computational fluid dynamics, in which it has proven its importance with respect to other traditional numerical methods in multiple fields, including incompressible, weakly and fully compressible flows as well as complex multiphysics applications [1–6]. In all kinds of mesh-based approaches to computational fluid dynamics, a local adaptation of the size of mesh cells is of importance. Indeed, to save on computational expense, applications often require to resolve a wide range of physical scales (see [7, 8] for instance). In other cases, such as in external aerodynamics and aeroacoustics simulations, the computational domain may be much larger than the actual fluid domain of interest, to limit the effect of artificial domain boundaries on the flow structure [9–13]. In all these cases, simulations benefit substantially from a local increase of spatial resolution in critical regions of the computational domain, which are most often unknown in advance. There exists therefore a need for sensors capable of identifying areas in which the mesh requires local adjustments.

Mesh refinement is in itself a highly complex topic in the field of LB methods, which are traditionally designed to be executed on homogeneous meshes with uniform mesh density. In a canonical LB scheme based on a collision-streaming sequence, the streaming step copies data to neighboring cells at relative positions provided by lattice constants which are immutable and as such constrain the mesh to a uniform

[*] Corresponding author

shape. For a local adaptation of the mesh size, it is common practice to use structured meshes with cubic elements which are organized in a hierarchic structure such as an octree, in which the relative edge length of neighboring cells typically varies by powers of two. In this way, a normal LB scheme can be executed within groups of equally sized cells, and an additional mesh refinement algorithm is carried out to transfer data across refinement levels (see, e.g., [14–20]). Depending on the relative spatial arrangement of cells at different levels, refinement algorithms are split into vertex-centered and cell-centered schemes, which are both presented in Section 2.1.

Most works dedicated to grid refinement algorithms focus on how to adapt the mesh instead of providing a way to predict regions where mesh refinement is required. In addition, these works usually rely on static meshes obtained through *a priori* knowledge of the flow features, such as boundary layer development and separation for wall-bounded flows. In contrast, mesh refinement sensors and error estimators predict, with minimal or no *a priori* knowledge, areas where refinement is needed. By coupling these sensors with adaptive mesh refinement techniques, one then drastically reduces user inputs for efficient and accurate simulations. In that context, Crouse et al. [21] propose a sensor based on the velocity divergence as an error indicator in an incompressible flow. Eitel-Amor et al. [22] use an approach based on two sensors, one proportional to the velocity norm and the other providing a measure of the total pressure. Fakhari & Lee [23] compare different methods applied to a finite-difference LB scheme. The comparison includes the Q-criterion, which is proportional to the symmetric and the anti-symmetric part of the velocity-derivative tensor. Other interesting error indicators are the vorticity norm and the derivatives of the vorticity norm in specific directions. The latter criterion is used in the context of wall-bounded turbulent flows which require an accurate simulation of boundary layers. Beyond the literature on LBM, other error indicators can be found in the literature for finite-volume Navier-Stokes solver with block-structured AMR (SAMR). An overview is proposed in Kamkar et al. [24]. It includes the Q-criterion and so-called lambda-sensors which, based on eigenvalues of velocity-derivative tensors, provide an estimate for the presence of vortical or other coherent structures.

There have been similar applications of the idea presented here outside of the LBM literature, aiming to quantify the flows departure from the hydrodynamic continuum regime. Meng et al. [25] assess the consistency of the continuum-level description of a fluid, as a function of the Knudsen number, by using a criterion that shares similarities with the sensor proposed in this article. Likewise, Singh et al. [26], among others, propose various metrics to quantify the spurious contributions of first-order perturbations from the Chapman-Enskog expansion to macroscopic quantities of a continuum fluid. In comparison to these existing works, we propose a novel, fully local sensor which is computed from the available populations in a given cell and therefore maintains the parallelizability of the LBM algorithm. It detects the need for further mesh refinement based on an expectation of the ratio between off-equilibrium and equilibrium parts of the velocity distribution functions.

The article is structured as follows: The theoretical part first provides a summary of the theory of mesh refinement algorithms (Section 2.1). Although the choice of a mesh refinement sensor in principle depends on the physics of the considered problem rather than on the specifics of a given mesh refinement algorithm, this summary provides additional context to the numerical verification part, in which two codes with different mesh refinement strategies are used to validate the new sensor. The theoretical part then introduces some concepts and notations of the LBM used in the present study. The proposed sensor is presented in Section 3. The verification in Section 4 starts with a qualitative comparison of the proposed sensor against other sensors proposed in the literature as a way to predict its usefulness for different types of fluid problems. Then, simulation results are presented in which the criterion is used to build a static mesh for a 3D cavity flow in Palabos (with vertex-centered grid refinement) and a dynamic mesh for a 2D cavity flow in the SAMR code AMROC-LBM (with cell-centered grid refinement). The quality of the produced results is discussed in both cases.

2. Theory

2.1. Mesh refinement

Lattice Boltzmann refinement algorithms can be split into vertex-centered schemes [14, 15, 19, 27–32], in which some of the coarse and fine nodes are situated at coinciding coordinates along the refinement interface, and cell-centered schemes [17, 22, 33, 34], in which coarse and fine meshes have a staggered relative arrangement. The latter category is compatible with the point of view adopted by the community of SAMR [35], a field traditionally dedicated to finite volume codes that has in recent years been applied to LB schemes [36–40]. Another important distinguishing feature between mesh refinement algorithms is the choice of the value for the time step δt in mesh levels with different cell edge length δx. In diffusive scaling, the ratio $\delta x/\delta t^2$ remains constant across mesh levels, thus guaranteeing asymptotic mesh convergence for incompressible flow problems, while in acoustic scaling, which is naturally adapted to compressible flows, the ratio $\delta x/\delta t$ is constant. Many authors however adopt acoustic scaling for both compressible and incompressible flows, because only this choice can guarantee the continuity of both density and pressure across the interface of a refined mesh [27]. In addition, the diffusive scaling usually leads to very small time steps, as the number of refinement levels increases, because moving from coarse to finer levels decreases the time step by a factor four. As a last remark, it is worth noting that the various proposed algorithms further differ in other aspects, including the presence or absence of time interpolation schemes.

In this work, our refinement sensor is used to generate the mesh in the two LB codes Palabos [41] and AMROC-LBM [36, 38, 39]. Palabos adopts the vertex-centered-based mesh-refinement algorithm described in Ref. [27], which is based on: (1) third-order polynomial interpolation, (2) coarse-to-fine filtering, and (3) acoustic scaling. This algorithm has been re-used and further improved in recent publications [11, 19, 31, 32]. In the AMROC framework [42], dynamic structured mesh adaptation is implemented generically on finite volume meshes and with the same recursive algorithm [35] that makes use of ghost cells to prescribe boundary conditions and achieve communication between refinement blocks. The LB variant AMROC-LBM uses a cell-centered scheme, in which streaming is performed first, followed by collision. Acoustic scaling is used to couple levels; interface populations propagating across refinement boundary are meticulously tracked, cf. [36, 38] for details. Palabos and AMROC-LBM are used to generate the data compiled in Section 5.

2.2. Lattice Boltzmann method

LBM is by now a well-known numerical method in computational fluid dynamics. This section presents only basic concepts, while the reader is referred to Refs. [1–6] for further details.

The Boltzmann equation (BE) describes the time evolution of large numbers of particles in a region of the space $\boldsymbol{x} \in \mathbb{R}^3$ with a given microscopic velocity $\boldsymbol{\xi} \in \mathbb{R}^3$, which are represented by the particle mass distribution function $f(\boldsymbol{x}, \boldsymbol{\xi}, t)$. The Boltzmann equation for a gas without external force reads

$$\partial_t f + \boldsymbol{\xi} \cdot \boldsymbol{\nabla} f = -\Omega(f, f^{\mathrm{eq}}), \tag{1}$$

with the collision operator Ω and where f^{eq} is the equilibrium distribution function, given by the Maxwell-Boltzmann distribution. For the rest of this discussion, the BGK collision operator is used: $\Omega = \frac{1}{\tau}(f - f^{\mathrm{eq}})$, with τ a relaxation time [43].

Following the ideas presented in the works by Shan et al. [44, 45], one can discretize the velocity space to a set of q velocities $\boldsymbol{\xi}_i$, leading to velocity-discrete populations $f_i(\boldsymbol{x}, t) \equiv f(\boldsymbol{x}, \boldsymbol{\xi}_i, t)$, $i \in [0, q-1]$. In this work, a D2Q9 and a D3Q27 models have been used.

As will be seen later, it is convenient to reformulate Eq. (1) as a function of non-dimensional quantities; from now on in this section, variables expressed in physical units will be written with a star for the sake of clarity (*e.g.* the physical macroscopic velocity u^*), whereas non-dimensional variables are written normally. The reformulation is done using characteristic time t_0^*, length l_0^*, density ρ_0^* and velocity magnitude c_0^* (the isothermal speed of sound). The velocity-discrete version of Eq. (1) with the BGK operator then reads:

$$\frac{1}{t_0^*}\partial_t f_i + \frac{c_0^*}{l_0^*}\boldsymbol{\xi}_i \cdot \boldsymbol{\nabla} f_i = -\frac{1}{\tau^*}\left(f_i - f_i^{\mathrm{eq}}\right), \tag{2}$$

where $f_i = f_i^*/\rho_0^*$ are the non-dimensional populations and $\boldsymbol{\xi}_i = \boldsymbol{\xi}_i^*/c_0^*$ are the non-dimensional mesoscopic velocities. Similarly, non-dimensional operators have been used : $\partial_t = \partial_t^* t_0^*$ and $\boldsymbol{\nabla} = \boldsymbol{\nabla}^* l_0^*$. The condition $l_0^* = c_0^* t_0^*$ is required in order to represent the correct macroscopic behaviour. Eq. (2) will be used below in order to relate the Knudsen number of the flow to the particle distribution functions.

A non-dimensionalization and a time and space discretization (with time step δt^* and spacing δx^*) lead to the LBM equation that is actually solved [46]:

$$\bar{f}_i(\boldsymbol{x} + \boldsymbol{\xi}_i \delta t^*/t_0^*, t + \delta t^*/t_0^*) - \bar{f}_i(\boldsymbol{x}, t) = -\frac{1}{\bar{\tau}}(\bar{f}_i(\boldsymbol{x}, t) - \bar{f}_i^{\text{eq}}(\boldsymbol{x}, t)), \tag{3}$$

where $\bar{\tau} = \tau^*/\delta t^* + 1/2$, and

$$\bar{f}_i = f_i + \frac{\delta t^*}{2\tau^*}\left(f_i - f_i^{\text{eq}}\right). \tag{4}$$

Note that the condition $l_0^* = c_0^* t_0^*$ mentioned above, together with the geometric requirement $|\boldsymbol{\xi}_i| \delta t^*/t_0^* = \delta x^*/l_0^*$ for straight velocities, implies that $\delta x^*/\delta t^* = c_0^*/c_s$, where $c_s = 1/\sqrt{3}$ represents the magnitude of the straight velocities. The choices $\delta t^* = t_0^*$ and $\delta x^* = l_0^*/c_s$ simplify the form of the actual implementation for homogeneous grids.

The discrete equilibrium distribution function is expressed by the truncated Maxwellian equilibrium [44, 45]

$$\bar{f}_i^{\text{eq}} = f_i^{\text{eq}} = w_i \rho \left(1 + \frac{\boldsymbol{\xi}_i \cdot \boldsymbol{u}}{c_s^2} + \frac{1}{2c_s^4} \boldsymbol{Q}_i : \boldsymbol{u}\boldsymbol{u}\right), \tag{5}$$

where ρ is the density, $\boldsymbol{u}$ is the macroscopic velocity field, $\boldsymbol{Q}_i = \boldsymbol{\xi}_i \boldsymbol{\xi}_i - c_s^2 \boldsymbol{I}$, and w_i the lattice weights. From now on, the bar on the populations is omitted. The density and the velocity fields are computed by the distribution function through the relations

$$\rho = \sum_{i=0}^{q-1} f_i = \sum_{i=0}^{q-1} f_i^{\text{eq}}, \tag{6}$$

$$\rho \boldsymbol{u} = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i = \sum_{i=0}^{q-1} \boldsymbol{\xi}_i f_i^{\text{eq}}. \tag{7}$$

For implementation purposes, a time-step is decomposed into two parts that are applied successively on the whole computational domain. The two steps are called the “collide-and-stream” operation.

1. The collision, which *locally* modifies the value of the populations according to

$$f_i^{\text{out}}(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) - \Omega_i(f_i, f_i^{\text{eq}}). \tag{8}$$

2. The streaming, which moves the populations to their neighbors according to their microscopic velocity

$$f_i(\boldsymbol{x} + \boldsymbol{\xi}_i \delta_t^*/t_0^*, t + \delta_t^*/t_0^*) = f_i^{\text{out}}(\boldsymbol{x}, t). \tag{9}$$

Performing a multi-scale Chapman-Enskog (CE) expansion (see [2, 47] for more details), one can show that the LBM BGK scheme is asymptotically equivalent to the weakly compressible Navier-Stokes equations

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0, \tag{10}$$

$$\partial_t(\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot (2\nu \boldsymbol{S}), \tag{11}$$

with p being the pressure, $\boldsymbol{S}$ the viscous stress tensor and ν the kinematic viscosity defined by

$$p = c_s^2 \rho, \tag{12}$$

$$\boldsymbol{S} = \frac{1}{2}\left[\boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \boldsymbol{u})^{\mathrm{T}}\right], \tag{13}$$

$$\nu = c_s^2(\bar{\tau} - 1/2). \tag{14}$$

The CE expansion is done under the assumption that f_i can be expanded with respect to a small parameter ε

$$f_i = f_i^{(0)} + \varepsilon f_i^{(1)} + \mathcal{O}(\varepsilon^2), \tag{15}$$

where $\varepsilon \ll 1$ can further be identified with the Knudsen number [48]. Using Eq. (2) and keeping only the lowest orders after replacing f_i by the CE *Ansatz* (15), one ends up at $\mathcal{O}(1)$ and $\mathcal{O}(\varepsilon)$ with [45]

$$f_i^{(0)} = f_i^{(eq)}(\rho, \boldsymbol{u}, T), \tag{16}$$

and

$$f_i^{(1)} = -\bar{\tau}\left[\partial_t f_i^{(0)} + \boldsymbol{\xi}\cdot\boldsymbol{\nabla} f_i^{(0)}\right] \tag{17}$$

respectively. From this, the non-equilibrium contribution $f_i^{(1)}$ naturally depends on time and space derivatives of all macroscopic quantities (density ρ, velocity $\boldsymbol{u}$ and temperature T) through the zeroth-order contribution $f_i^{(0)}$. One can then expect $f_i^{(1)}$ to become non-negligible for strong, local variations of density, velocity or even temperature fields, whether they be of physical or numerical nature. In the following sections of this document, we refer to the non-equilibrium component of the distribution as $f^{\mathrm{neq}} \equiv f - f^{\mathrm{eq}}$, which is approximed in first order by $f^{(1)}$ with respect to the expansion parameter.

Finally, by further noticing that the derivatives should scale like the characteristic quantities of the system, the above relationship (17) becomes

$$f_i^{(1)} \sim \tau^* \left(\frac{1}{t_0^*} + \frac{c_0^*}{l_0^*}\right) f_i^{(0)}. \tag{18}$$

By interpreting $\lambda^* = c_0^*\tau^*$ as the mean free path of particles, and using $c_0^* = l_0^*/t_0^*$, the Knudsen number $\mathrm{Kn} = \lambda^*/l_0^*$ is introduced, yielding

$$\frac{f_i^{(1)}}{f_i^{(0)}} \sim \mathrm{Kn}. \tag{19}$$

We propose in Appendix A an interpretation of this quantity and its use as a sensor in the case of single phase and weakly compressible flows. This ratio remains small in the continuum limit for smooth flows but is expected to become large close to discontinuities and multiphase/multicomponent interfaces, or when the flow is underresolved, as it may happen close to the wall due to thin boundary layers. In these cases, the assumption that f^{neq} is properly approximated by $f^{(1)}$ may no longer be valid. This limits the validity of our analysis, in principle to properly resolved flows. The numerical data provided in the forthcoming sections, which includes evaluation of the Knudsen criterion in case of boundary layers in a cavity wall, the case of supersonic shock waves, and liquid/gas interfaces, suggest however that the criterion remains a valid option beyond this limitation.

3. Refinement sensor

3.1. Principles

In this section, the refinement sensor is proposed independently of the actual algorithm used to implement grid refinement. Also, the stars are from now on omitted for variables expressed in physical units.

The proposed sensor is based on the fact that the off-equilibrium and equilibrium parts of the distribution function are linked to each other through the Knudsen number as defined above:

$$f^{\mathrm{neq}} \sim f^{\mathrm{eq}}\mathrm{Kn}. \tag{20}$$

Consider now a grid G_c with mesh spacing δx_c, and a finer grid G_f with mesh spacing δx_f linked by the relationship $\delta x_f = \delta x_c/n$, where n is a positive integer. It is assumed that the same simulation is executed on both grids and that the Reynolds number Re is the same in both cases (the lattice relaxation time is adapted to ensure this constraint). Also, the ratio $\delta x/\delta t$ is set to be constant in G_c and G_f, hence

$\delta t_c/\delta t_f = \delta x_c/\delta x_f$. With this constraint, called "convective scaling" or "acoustic scaling", quantities that are proportional to the velocity are invariant in lattice units, and the Mach number does not depend on the grid resolution. Acoustic scaling however yields a proper convergence rate for compressible flow only and fails to approximate incompressible flow solutions with second-order convergence rate, as Eqs. (10-11) that are recovered with the CE expansion contain error terms of order $\mathcal{O}(\delta t^2) + \mathcal{O}(\mathrm{Kn}^2)$.

3.2. Implementation

To formalize the computation of the sensor, the following algorithm is proposed:

1. Choose a uniform spatial resolution δx.
2. Run the simulation.
3. For all cells, compute the local Knudsen number averaged over all discrete velocities

$$C = \frac{1}{q}\sum_{i=0}^{q-1}\left|\frac{f_i^{\mathrm{neq}}}{f_i^{\mathrm{eq}}}\right|. \tag{21}$$

It can be noted that the idea of using the ratio between off-equilibrium population (to a given power $n \in \mathbb{N}$) and their equilibrium counterpart was recently propose to locally stabilize simulations through local entropic filtering [49] and artificial viscosity [50, 51] with $n = 2$ and 1 respectively. Similarly, the ratio between non-equilibrium and equilibrium contributions is also used as a criterion to identify regions of the simulation domain where the continuum assumption falls apart. This criterion is usually computed at the level of populations [25] or their moments [52, 53]. The interested reader can refer to the letter by Singh et al. [26] for a brief review on breakdown criteria.

3.3. Comparison with other sensors

As discussed in the introduction, the proposed sensor will be compared to different other sensors in Section 4. The first one is the "vorticity norm sensor":

$$K_\omega = |\boldsymbol{\omega}| = |\boldsymbol{\nabla} \times \boldsymbol{u}|, \tag{22}$$

and the second one is based on the incompressible formulation of the "Q-criterion" [54]:

$$K_{\mathrm{Q}} = \frac{1}{2}\left(|\boldsymbol{\Omega}|^2 - |\boldsymbol{S}|^2\right), \tag{23}$$

with $\Omega = \frac{1}{2}\left[\boldsymbol{\nabla u} - (\boldsymbol{\nabla u})^{\mathrm{T}}\right]$ the rotation rate tensor. Then, a sensor proposed by Fakhari & Lee in [23] is also tested here:

$$K_{\mathrm{F}} = \frac{\sqrt{(\partial_x \omega)^2 + (\partial_y \omega)^2}}{|\boldsymbol{\omega}|}. \tag{24}$$

Note that this latter sensor has been proven to yield accurate results in some cases, in particular for the 2D lid-driven cavity benchmark. Finally, the proposed "Knudsen sensor" reads

$$K_{\mathrm{Kn}} = \frac{1}{q}\sum_{i=0}^{q-1}\left|\frac{f_i^{\mathrm{neq}}}{f_i^{\mathrm{eq}}}\right|. \tag{25}$$

3.4. Practical considerations for the use of a refinement sensor

Whereas the quantity K for all refinement sensors listed in the previous section denotes refinement factors (*i.e.* $\delta x_c = K \cdot \delta x_f$), it is convenient to work with the logarithm of K,

$$\phi = \log_2 K, \tag{26}$$

which is linked linearly with the desired refinement level l:

$$l = \begin{cases} l_{min} & \text{if } [a \cdot \phi] < l_{min} \\ l_{max} & \text{if } [a \cdot \phi] > l_{max} \\ [a \cdot \phi] & \text{otherwise,} \end{cases} \tag{27}$$

where the bracket $[x]$ denotes the integral part of a value x, and l_{min} and l_{max} stand for the minimal (coarsest) and maximal (finest) refinement level respectively. The scale factor a is chosen empirically (with our sensor as well as with other sensors) to fit the values of ϕ encountered in the simulation to the desired range of refinement levels. In this way, the different degrees of refinement correspond to powers of 2, as required by the implementation:

$$\delta x = 2^{l_{max}-l}\delta x_f, \tag{28}$$

where δx_f denotes the size of the finest cells.

In practice, the accuracy and efficiency of a code using static or dynamic mesh refinement depends therefore not only on the choice of the refinement sensor, but also on a fine tuning of these scale parameters. This manuscript is not focusing on the technical aspects of this exercise in parameter tuning, nor on the relative accuracy obtained by different refinement sensors, which all need to be tuned individually for optimal results. Instead, Section 4 provides a qualitative comparison of different sensors and discusses their capability to capture specific physical phenomena that are usually considered to require mesh refinement. In Section 5, where the Knudsen sensor is used to implement static grid-refinement in a 3D LB simulation and AMR in a 2D LB simulation, the scale parameter a is fixed choosing the finest refinement level δ_{xf} and the overall number of levels.

4. Qualitative comparison of sensors for different types of physics

Hereafter, we compare the ability of each sensor to highlight under-resolved regions for a wide range of physical phenomena: flow past a cylinder, lid-driven cavity flow, Riemann problem and droplet at rest. By further comparing these results with *a priori* knowledge about each configuration, it is possible to identify which sensor is more likely to provide meaningful information to grid-refinement algorithms.

4.1. 2D lid-driven cavity

Wall-bounded flows are commonly encountered in the context of internal fluid dynamics. For these, it is of paramount importance to correctly capture the development of the boundary layers, and their interaction with the bulk flow. To investigate the ability of each sensor to properly identify these flow features, we propose to simulate a 2D lid-driven cavity flow for Re = 1000. It consists of a square domain resolved by $N \times N$ lattice cells. A no-slip boundary condition $\boldsymbol{u} = 0$ is applied on each wall except the top wall located at $y = N$, where the condition $\boldsymbol{u} = (u_{LB}, 0)^T$ is applied. The moving bounce-back condition as described in [5] was used on the walls to impose the velocity boundary conditions. In order to perform the tests at Reynolds number Re $= 10^3$, a Smagorinsky subgrid-scale model [55] is added to the collision step, with Smagorinsky parameter $c_{\text{smago}} = 0.14$.

A simulation with parameters Re $= 1000$, $u_{LB} = 5 \cdot 10^{-2}$ and $N = 120$ has been performed in order to compare the different sensors. For this benchmark, no sensor is used as a reference, since no criterion yielded results more relevant (by visual assessment) than the Knudsen criterion.

Figure 1 displays the normalized average refinement level ϕ in the whole simulation domain and for each sensor. It is found that the Knudsen sensor is similar to the one of Fakhari & Lee at a global scale, as both suggest a low resolution in the center of the domain. Close to the borders on the other hand, the values proposed by the Knudsen sensor are similar to those of the vorticity-norm sensor.

While the sensor of Fakhari & Lee, which was designed for this purpose, may produce a mesh somewhat better adapted to the needs of the boundary layer, the Knudsen sensor once again suggests overall reasonable mesh levels matching the expectations qualitatively.

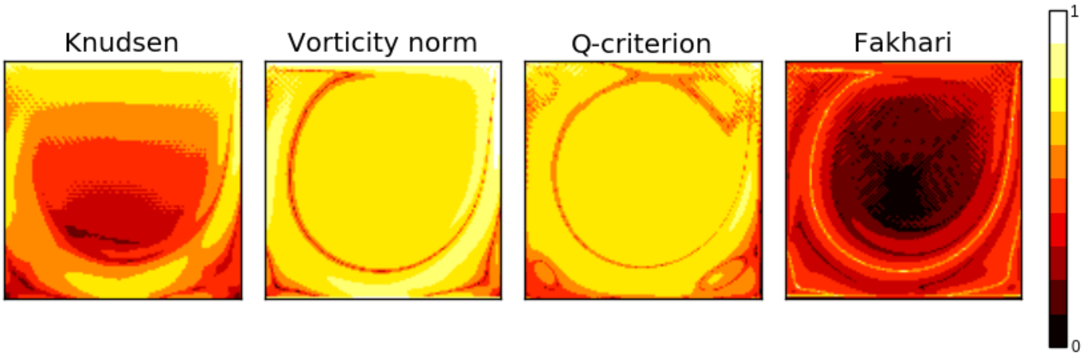

Figure 1: Average refinement level in the case of the 2D lid-driven cavity with $\mathrm{Re} = 10^3$, $u_{LB} = 5 \cdot 10^{-2}$ and $N = 120$.

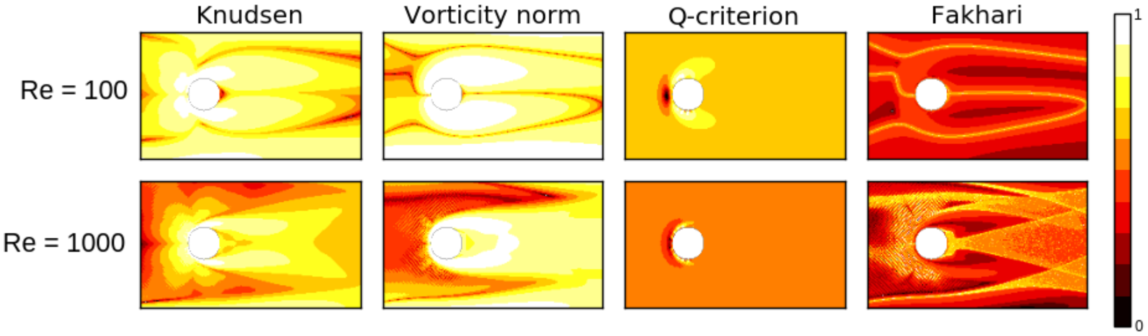

Figure 2: Comparison of the sensors for the flow around a 2D cylinder, at (top) Re = 100 and (bottom) Re = 1000. The normalized ϕ value for each sensor is shown in the subdomain $x \in [x_c - 4R, x_c + 10R]$, $y \in [y_c - 4R, y_c + 4R]$. The value of the sensor inside the boundary nodes of the cylinder is not plotted here.

4.2. Flow past a 2D cylinder

We investigate here how sensors identify physical phenomena related to the simulation of a flow past bluff bodies, e.g., boundary layer development and wake formation. More precisely, we consider the flow past a cylinder with a simulation domain that consists of a $N_x \times N_y$ rectangle with a bottom wall at $y = 0$ and a top wall at $y = N_y$. A cylinder of radius $R = N_y/10$ is centered in $x_c = N_x/3$. To trigger the unsteadiness of the flow, the y-coordinate of the cylinder center is slightly offset from the centerline and set to $y_c = 9N_y/20$. Simulations have been performed with $N_x = 300$, and for Re = 100 and Re = 1000. In all cases, $N_y = N_x/2$. A no-slip boundary condition is applied to the cylinder surface and the lateral walls, using the same bounce-back method as in the 2D cavity benchmark described above. A Zou/He velocity boundary condition [56] is used at the inlet in $x = 0$ to impose a velocity u_{LB} chosen to match the target Reynolds number. The outlet consists of a Neumann boundary condition : $f_i(t, x = N_x)$ is set to $f_i(t, x = N_x - 1)$ for every population i and for every time step t. A Smagorinsky subgrid model [55] was adopted to perform the simulations, with Smagorinsky parameter $c_{\mathrm{smago}} = 0.14$. For this benchmark, the vorticity norm is chosen as the reference sensor, as suggested in [23] for the case of a the flow past a square cylinder.

Figure 2 depicts the time average value of ϕ for each sensor at Re = 100 and Re = 1000. It is observed that the Knudsen sensor is essentially similar to the vorticity norm in this case, for Re = 100 as well as for Re = 1000. The spatial average of the refinement factor, on the other hand, is comparable for the Knudsen sensor and the vorticity norm at high Reynolds number, whereas the vorticity norm yields lower values at low Reynolds number (see Tab. 4).

4.3. 3D flow past a sphere

In this test the domain size is $N_x = 400$, $N_y = 160$ and $N_z = 160$. In this problem, the domain is periodic along y-axis and z-axis. The inlet is situated at $x = 0$, and an outlet sponge zone, in which the relaxation parameter is linearly increased, spreads from $x = 300$ to $x = 400$. The velocity at the inlet $\boldsymbol{u}(x = 0, y, z) = (u_{LB}, 0, 0)^T$ is imposed through a regularized boundary condition [57], with $u_{LB} = 3 \cdot 10^{-2}$. A sphere of diameter $D = 20$ lattice nodes, modelled through a bounce-back boundary condition, is located at $x_c = 100$, $y_c = 80$, $z_c = 80$. The simulation is executed with a BGK collision model and Smagorinsky subgrid-scale modeling [55] (Smagorinsky parameter $c_{\text{smago}} = 0.14$). Palabos library [41] was used to implement the benchmark. The results were averaged after transient state was reached, from iteration $4 \cdot 10^4$ to iteration $5 \cdot 10^4$. Just like for the corresponding 2D benchmark case, the vorticity norm serves as a reference sensor, because it is known to properly single out the area around the sphere and vortex-shedding phenomena in its wake.

The normalized value of ϕ for each sensor is displayed in Figure 3 for Re $= 1887$ and Figure 4 for Re $= 75$. Both the Knudsen sensor and the Q-criterion follow the lead of the vorticity and properly predict the requirement for high mesh resolution around the obstacle and in its wake. Solely, the Q sensor appears less useful at high Reynolds numbers, as it pushes for highest-level mesh resolution in large swaths of the full computational domain. The sensor of Fakhari & Lee finally does not exhibit useful values, neither at low nor at high Reynolds number, as expected for a sensor that is mainly targeted at interior flow problems.

The resolution of the simulations is kept intentionally low to test the capability of a sensor to predict the need for further resolution while running in an underresolved regime. In this case, spurious numerical patterns occur in the area between the obstacle and the inflow, which are first visible in the pressure field and velocity gradients (on Figure 3, they are most visible in the Knudsen and the vorticity-norm sensors). They can be traced back to the boundary representation of the obstacle, here a low-resolution bounce-back scheme, and can lead to numerical instabilities. Clearly, the occurrence of such patterns calls for better mesh resolution. Thus, the Knudsen sensor, which singles out these patterns very neatly, turns out to be particularly useful in this respect.

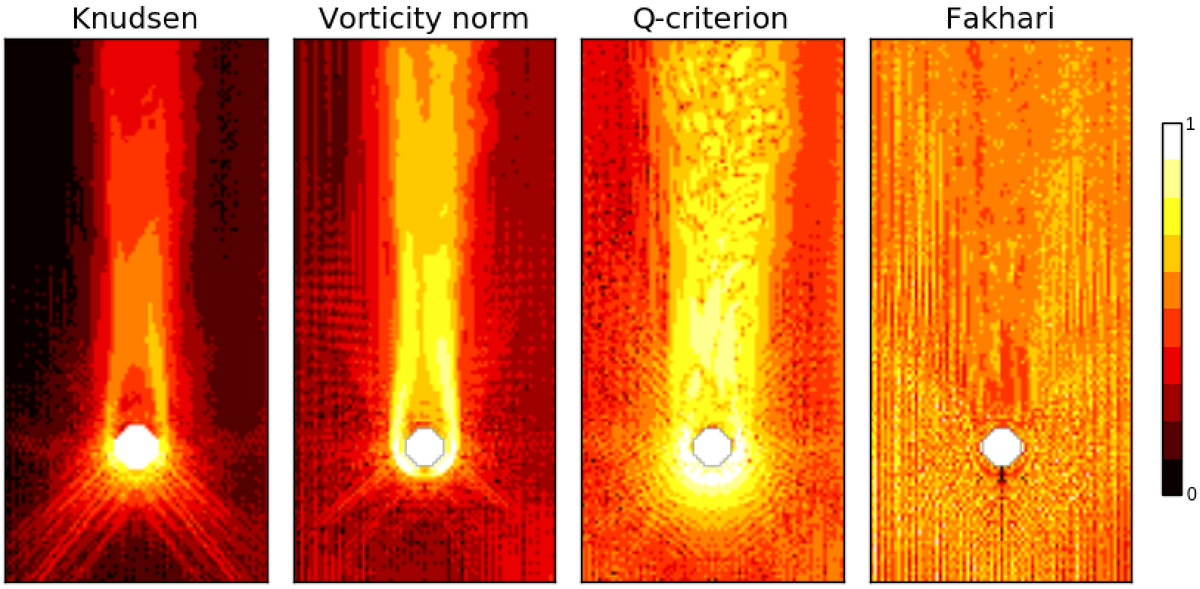

Figure 3: Comparison of time-averaged refinement sensors for a flow around a sphere at Re = 1887, at a deliberately poor resolution. The normalized ϕ value for each sensor is shown in the subdomain $x \in [0.4N_x, 0.8N_x]$, $y \in [0.25N_y, 0.75N_y]$ and $z = 0.5N_z$. The value of the sensor inside the boundary nodes of the sphere is not plotted here.

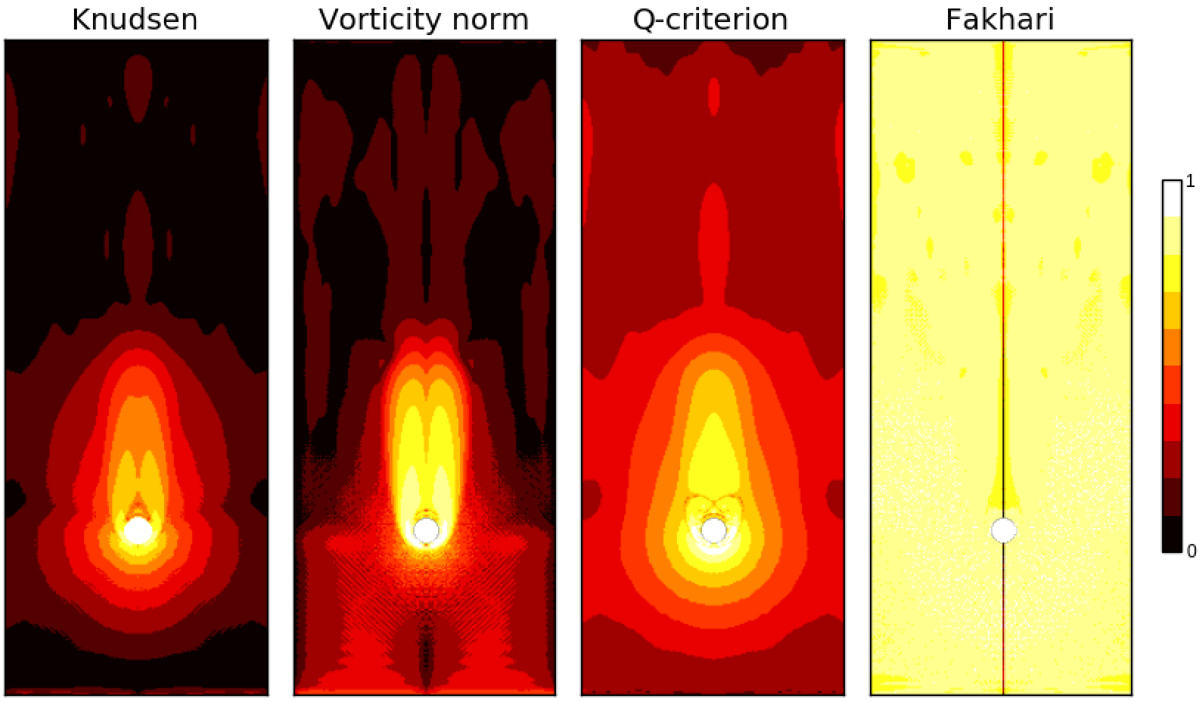

Figure 4: Comparison of the time average of the refinement sensor for the flow around a sphere at Re = 75. The value of the sensor inside the boundary nodes of the sphere is not plotted here.

4.4. 2D Riemann problem

This 2D benchmark case of a compressible flow (see [58] for a detailed description) enables the study of flows with sharp density, velocity and temperature variations. It is considered as a challenging test case due to the complex wave patterns arising in the solution [59]. Recently, Coreixas & Latt proposed a shock capturing technique that locally adds artificial viscosity in regions highlighted by the Knudsen sensor. This helped stabilizing the simulation while properly capturing the key flow features of this Riemann problem, in particular the shock wave fronts and their interplay [51]. As a consequence, the Knudsen sensor is assumed to provide reference data on this configuration.

A square domain of $N \times N$ lattice nodes is set with $N = 1000$, and the initial conditions described in Tab. 1. The domain is periodic along $X-$axis and $Y-$axis avoid the need for time-dependent boundary conditions. Owing to the compressible nature of this problem, the methodology described in Ref [50] has been used to run the simulation. More precisely, the D2Q49 lattice was used along with a BGK collision model based on a numerical equilibrium, that enforces all 13 constraints required to recover the Navier-Stokes-Fourier equations in an *exact manner*. The simulation corresponds to an inviscid flow ($\nu = 0$), with a reference temperature $T_0 = 0.7$, and results are plotted for $(x, y) \in [N/4, 3N/4]^2$ at iteration number 150.

Figure 5 displays on the left-most image the normalized logarithm of the density ρ to help identify regions that naturally require high grid resolution due to sharp density variations. The four remaining images show the ϕ value of the four sensors, again represented by a normalized version ϕ_n that is rescaled to the $[0, 1]$ range. By visual assessment, it is clearly determined that vorticity norm and the Fakhari & Lee sensors do not properly capture shock wave fronts propagating in quadrant 1 toward the upper right corner, as opposed to the Knudsen sensor and the Q-criterion. Moreover, the Knudsen sensor appears better adapted to the needs of mesh refinement than the Q-criterion, as it calls for finer meshes in smaller, more sharply depicted areas of the domain. This is confirmed by computing the spatial average of each sensor: $\bar{\phi}_{n,\mathrm{Kn}} = 0.14$,

	quadrant 1	quadrant 2	quadrant 3	quadrant 4
ρ	0.513	1	0.8	1
u_x	0	0.7276	0	0
u_y	0	0	0	0.7276
P	0.4	1	1	1

Table 1: Initial conditions of the 2D Riemann problem, with the pressure denoted by P. In the visual representations of the simulation, quadrant 1 is located on the upper right corner of the domain, quadrant 2 in the upper left corner, quadrant 3 in the bottom left corner and quadrant 4 in the bottom right corner. Note that, as a result of the compressible nature of this problem, Eq. (12) does not hold.

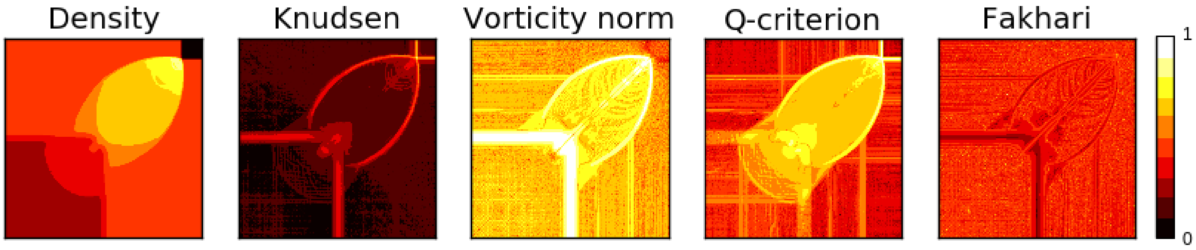

Figure 5: Comparison of the sensors for the Riemann problem. The normalized ϕ value for each sensor is shown in the center of the simulation domain ($x, y \in [0.25, 0.75]$), along with $\log(\rho)$ for comparison.

whereas $\bar{\phi}_{n,\mathrm{Q}} = 0.49$. For a more in-depth discussion of the characteristics of this problem, the reader is referred to [51].

It is interesting to emphasize that in this benchmark case, the Knudsen sensor proves to be very different, and substantially more useful than the vorticity-norm sensor, although the two sensors produced rather similar results in previous single phase incompressible flow cases. The Knudsen sensor proves more versatile, by virtue of the gradients of different macroscopic variables present in the non-equilibrium population (Eq. A.1). While velocity gradients are dominant in single phase incompressible flows, leading to an expression akin to the one of the vorticity norm, all gradients of macroscopic quantities (density, velocity and temperature) should be properly captured for this compressible example.

4.5. 2D droplet

Hereafter, a weakly compressible multiphase flow is simulated to understand how sensors react in the presence of non-negligible (1) density gradients and (2) spurious currents. More precisely, this multiphase benchmark consists in a 2D droplet which is at rest in a periodic square domain of size $N \times N$, with a resolution $N = 200$ cells. The droplet is simulated as a circular region of the domain centered about $(x_c, y_c) = (N/2, N/2)$, with a radius $R = N/4 = 50$ cells. The liquid density is set to 1.95, whereas the rest of the domain is initialized with a gas density equal to 0.15, eventually leading to a density ratio of 13. Lattice populations are initialized at equilibrium with the aforementioned densities and with zero velocity. The simulation is stopped after a stationary state is reached. A Shan-Chen single component model [60] with parameter $G = -5$ was used to simulate the phases of the problem. The pseudo-potential force was accounted for through Guo's forcing methodology [61] along the BGK collision model.

Figure 6 shows the refinement levels predicted by each sensor at the end of the simulation, along with the density gradient for reference. It is observed that the Knudsen sensor captures the interface accurately, while suggesting a low resolution inside the droplet and isotropy issues due to spurious currents, as the Q-criterion and the Fakhari & Lee sensor do. However, the contour lines demonstrate that the Knudsen sensor puts a more consistent focus on the interface of the droplet than other criteria. This can be explained by the presence of non-negligible spurious currents which more severely impact Fakhari & Lee, Q-criterion, and the vorticity norm sensors, with the latter one being of little interest for such a configuration. To isolate the impact of parasitic currents on the above conclusion, one could rely on the well-balanced LB formulation recently proposed by Guo [62]. The latter allows for the drastic reduction of the amplitude of spurious currents, even with the simple BGK collision model. Such an investigation is left for future work.

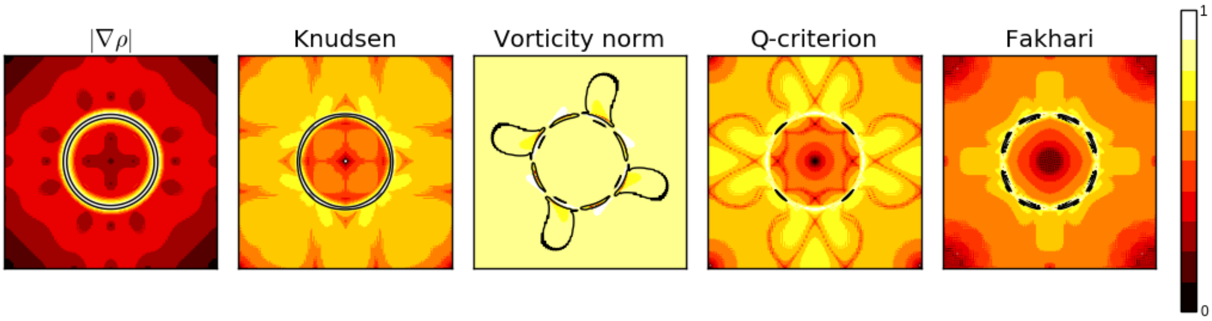

Figure 6: Normalized refinement level yielded by the different sensors in the case of the 2D droplet in the stationary state using a single component Shan-Chen model. The norm of the density gradient is used as a reference in this simulation. The contour line (black) corresponds to a value of 0.45

5. Use of the Knudsen sensor as a grid refinement sensor

This section aims at confirming the accuracy and efficiency improvement induced by the use of our kinetic based refinement criterion. With this idea in mind, we first rely on this criterion to build a static mesh density field that is further used to simulate the 3D lid-driven cavity flow at $\mathrm{Re} = 12000$ on a non-uniform mesh in an efficient and accurate manner. Then, it is shown in the 2D context that such an approach can be adopted for adaptive mesh refinement. In the latter case, the mesh density is (re)computed in a dynamic manner to speedup the convergence toward steady state.

5.1. 3D cavity at $\mathrm{Re} = 12000$ *with static mesh refinement*

In this first test simulation with grid refinement, the Knudsen sensor is used to generate a mesh for a 3D lid-driven cavity at a Reynolds number $\mathrm{Re} = 12000$. The accuracy of the results is verified by comparison against a Chebyshev spectral simulation reported in [63]. The Palabos library has been used to implement this test case. To set up this problem, a no-slip condition was imposed on each wall, except on the lid (x-z plane at $y = h$) where an x-velocity was imposed with a smooth velocity transition on the lid edges, as described in [64]:

$$u(x, y = h, z) = U_0 \left[1 - \left(\frac{x}{h} - 1\right)^{18}\right]^2 \left[1 - \left(\frac{z}{h} - 1\right)^{18}\right]^2, \qquad (29)$$

where U_0 is the maximum lid velocity. Here, $h = 1$ and U_0 and the lattice relaxation parameter is chosen in order to match the Reynolds number For $\mathrm{Re} = 12000$, $\mathrm{Ma} \approx 0.14$ and $\mathrm{Kn} \approx 1.15 \cdot 10^{-5}$.

A simulation with $200 \times 200 \times 200$ lattice nodes has first been performed on a uniform, relatively coarse mesh in order to compute the Knudsen sensor and build a corresponding, refined mesh. For this purpose, the refinement sensor was averaged over the duration of one dimensionless time period (defined as L/u, with L and u the characteristic length and velocity of the problem) starting at the end of the initial, transient flow. In this coarse, homogeneous simulation, $u_{LB} = 0.048$ and $\bar{\tau} = 0.5024$. Along the walls, the generated grid was set uniformly to the highest resolution to prevent the boundary conditions from crossing refinement interfaces. In the rest of the simulation domain, the number of lattice nodes per domain length is given by $N(l) = 2^{7+l}$, where $l \in [0, 3]$ is the local refinement level yielded by the sensor. The generated grid is represented in Figure 7 with cells that are coarser than the actual computational cells, for the sake of

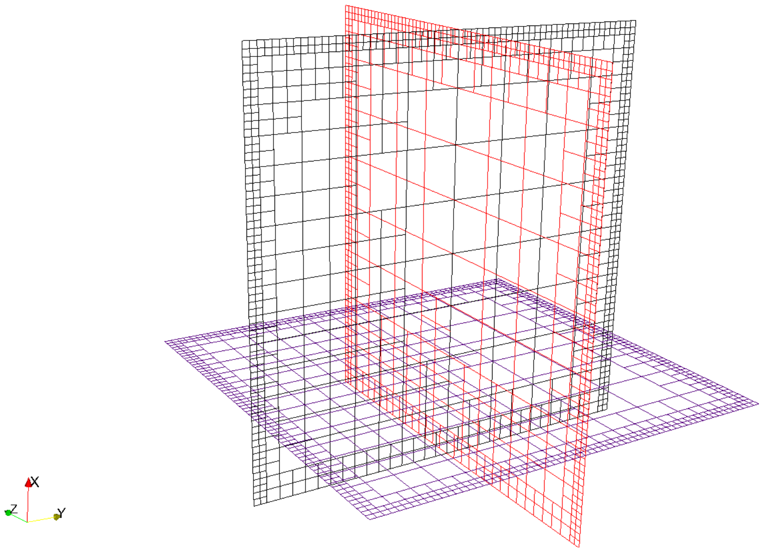

Figure 7: Schematic representation of the generated grid for the 3D cavity. For the sake of clarity, 3 slices are represented here, where one cell in the picture represents a slice of 16 × 16 simulation cells.

visualization. In total, the generated grid contains approximately 11 million cells, whereas a uniform grid at the finest mesh resolution would contain approximately 134 million cells.

Figure 8 shows a comparison of the mean u_x and u_y velocities in the reference solution and in the LB simulation with the refined grid. The velocity values from the LB simulation are averaged from $t = 150$ to $t = 225$ dimensionless time periods. This long simulation time is required to gather sufficient data to reach convergence, and it is one of the reasons for which this benchmark case is known to be very challenging. The comparison shows an excellent match and as such demonstrates the adequacy of the mesh used to run the problem.

5.2. 2D cavity with adaptive mesh refinement

In this second practical use case of the Knudsen sensor, a dynamic, time-dependent mesh is generated for the 2D cavity problem as described in Section 4.1, adopting the AMROC-LBM software [36–38]. The employed LB scheme is the BGK operator in two space dimensions, as described in Section 2.2. The AMROC-LBM results here use linear temporal interpolation, second-order accurate spatial interpolation and re-scaling of the non-equilibrium part f^{neq} of populations at the interface between levels, cf. [39], as originally proposed by Dupuis & Chopard [15].

The simulation is performed at a Reynolds number of $\mathrm{Re} = 1000$ with a multi-level structured AMR grid, based on a coarsest grid resolution with $N = 50$ cells across the width of the cavity. While several large eddy simulation models have been implemented for AMROC-LBM [65], the present results do not apply any such model. Although AMROC allows arbitrary refinement factors, 2 is used here for all additional refinement levels. Evaluation of the Knudsen criterion and mesh reorganization is carried out at the beginning of every level-0 time step. One additional layer of buffer cells is marked for refinement around cells tagged by the criterion. Refinement blocks are created with an efficiency of 80%, meaning that at least 80% of cells in each block need to be flagged. A block is further sub-divided otherwise. Proper nesting is also enforced, ensuring that a refinement is fully contained in the next coarser level with at least one row of coarse buffer cells. The detailed topological algorithms in AMROC can be found in [42].

Figure B.11 displays a snapshot of the sensor values superimposed with the velocity norm of the flow. Figure B.12 shows a snapshot of the grid generated by AMROC-LBM, based on the Knudsen sensor for both

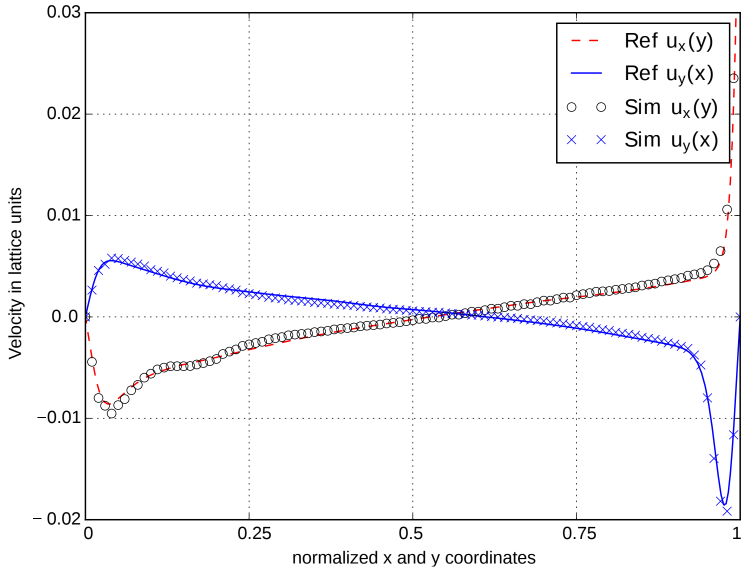

Figure 8: Comparison of the averaged centerline u_x and u_y velocity for the 3D cavity, for the cases of the reference solution and the static, refined grid.

Simulation setup	RMSE u_x	RMSE u_y	$N_{\text{effective}}$
Homogeneous mesh ($N = 50$)	0.060	0.053	2500
Homogeneous mesh ($N = 200$)	0.015	0.018	40000
Adaptive mesh using K_{Kn}	0.014	0.027	27672

Table 2: Root-mean-square error of homogeneous mesh simulation and adaptive mesh simulation from the reference solution.

3 levels and 5 levels of refinement for comparison. A video of a simulation is provided as supplementary material to this article, where the velocity norm is shown along with the refined grid levels.

For quantitative comparison, the velocity profiles of u_x along the Y-axis centerline and of u_y along the X-axis centerline were evaluated after convergence of the flow and compared to reference values presented in [66]. Table 2 lists the root-mean-square error between the computed velocities and the reference values along the two profiles. The number of cells for the refined mesh in Tab. 2 is computed in terms of effective cells, weighted by their frequency of execution in the convective scaling regime:

$$N_{\text{effective}} = \sum_{i=1}^{L} \frac{n_i}{2^{L-i}}, \tag{30}$$

where n_i stands for the total number of cells at level i and L the finest grid level. This comparison shows that AMR using our sensor allows accurate results while significantly saving the number of lattice points, as compared to a homogeneous grid made of cells the same size as the finest level of the refined grid.

Finally, it is worth noting that, even during the initial transient phase of the flow, the error remains small. The root-mean-square error can be computed between the velocity profiles obtained with the AMR method using the Knudsen criterion and the velocity profiles obtained with the homogeneous, 200 × 200 cells grid. This error is shown, from the beginning of the simulation to the convergence, on Figure 9. It is found that the peak root-mean-square error is approximately 6%, while it quickly reaches a stable value between 1% and 2%.

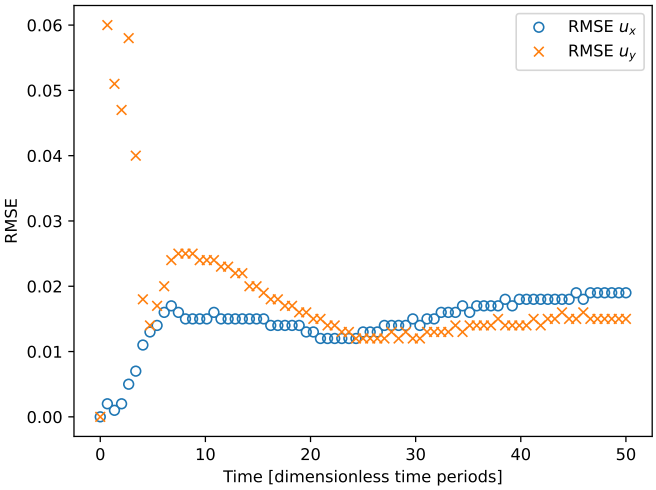

Figure 9: Time evolution of root-mean-square error between the velocity profiles of the 2D cavity. The RMSE are relative to the velocity profiles obtained with the AMR method using the Knudsen criterion and the velocity profiles obtained with the homogeneous, 200 × 200 cells grid, for u_x along Y-axis centerline and u_y along X-axis centerline.

6. Discussion and conclusion

A novel, local grid refinement sensor has been proposed. It benefits from the kinetic nature of LBMs through the local evaluation of the departure of distribution functions from their equilibrium state, hence providing valuable information on under-resolved regions of the simulation domain for a wide range of physical phenomena. This sensor was compared to three state-of-the-art criteria over five benchmarks and compared qualitatively with expectations based on a physical understanding of the investigated flows.

Table 3 summarizes the outcome of this qualitative discussion by assigning a grade to each of the sensor in every case. Furthermore, to support the discussion below, Tab. 4 summarizes the space average of the refinement factor obtained with each sensor on the different benchmarks, as a measure of the reduction of the number of grid cells that can be achieved by using the sensor.

Our proposed sensor is found to produce either excellent or acceptably good results in all executed test cases. As shown in the summary in Tab. 3, it is the only one among the four tested sensors that would produce an acceptable mesh across the full range of investigated physical situations. It therefore constitutes a sensor of choice for simulations in which too little *a priori* knowledge of the problem is available to produce the computational mesh manually. Moreover, this method has the advantage to rely only on local information (i.e., the populations) and is therefore very efficient in terms of parallelism, as compared to sensors that require the computation of gradients of macroscopic quantities through finite differences or similar means. Since it needs no *a priori* knowledge of the system, this method is also an excellent candidate for adaptive grid refinement techniques in time-dependent problems, as demonstrated by the preliminary results given in Section 5.2.

As a direct extension of this work, we plan to apply the Knudsen sensor to produce static and AMR meshes for more advanced problems and assess its efficiency in comparison with other approaches. Preliminary results for the supersonic flow past a 2D NACA0012 airfoil at Ma = 1.5 and Re = 10^4 are shown in Figure 10. Here, the proposed sensor identifies the expected regions of mesh refinement at (1) bow and secondary shock waves, (2) airfoil surface, and (3) Von Kàrman street. The mesh is computed using a 2D LB code with AMR capability, and based on C++ parallel algorithms [67]. More details on the performance and accuracy of such an approach will be proposed in a forthcoming work. Finally, as pointed out in Appendix A, there exist certain flow cases in which the dependence of the sensor on the relaxation time may be significant given the use of a single relaxation time collision model. Further assessment of this is out

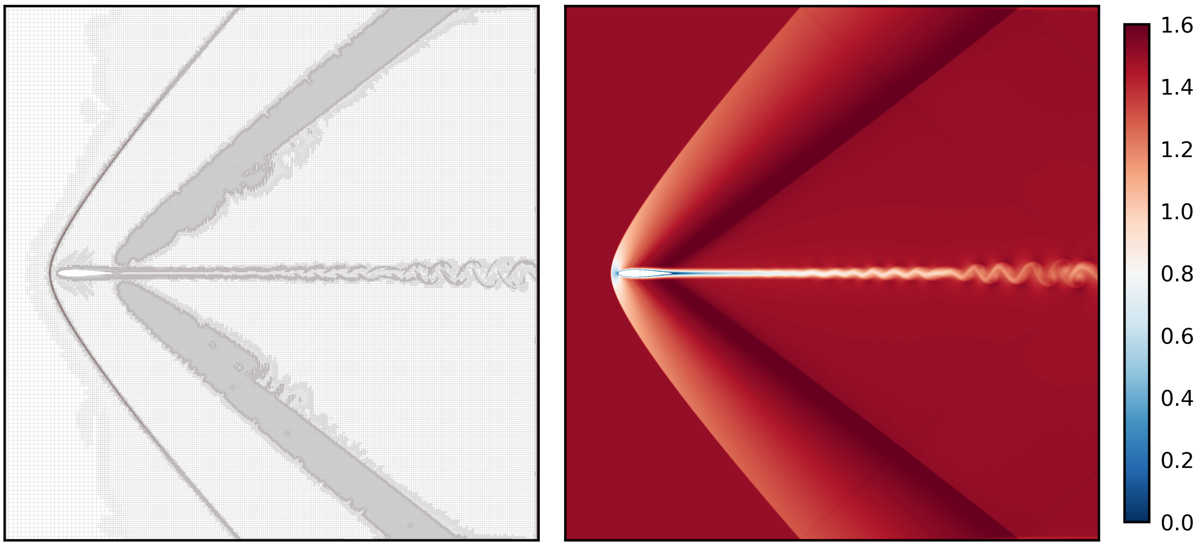

Figure 10: Supersonic flow past a 2D NACA0012 airfoil at $(\mathrm{Ma}, \mathrm{Re}) = (1.5, 10^4)$ with a simulation domain $[10C, 10C]$ discretized using 200 points per chord C. Right: Mach number; Left: Mesh computed using the proposed criterion. The finest level correspond to a resolution of $1/(10C)$. More details about the compressible LBM used for this test can be found in Ref. [50].

of the scope of this study, but provides an area of future research where perhaps a more complex collision model may be required (see [68] for example).

Acknowledgment

We acknowledge financial support from the Swiss National Science Foundation (SNSF) through project grants 200020_197223 and 200021_165984.

Appendix A. Derivation of δt_f as a function of δt_c for incompressible flows

To give the reader an insight of the relationship between coarse and fine grid parameters, we restrict the discussion to the single phase and weakly compressible case in this appendix.

	K_{Kn}	K_{ω}	K_{Q}	K_{F}
Cylinder (Re=100)	✓	ref.	✗	✗
Cylinder (Re=1000)	✓	ref.	✗	✗
Lid-driven Cavity	~	~	~	~
Riemann	ref.	✗	~	✗
Droplet	✓	✗	~	✓

Table 3: Summary of the accuracy of the sensors for each benchmark. A check mark (✓) indicates a good match with the reference sensor, a tilde (~) indicates a poor though relevant match, and a cross mark (✗) indicates a poor match and the fact that the sensor cannot be used as a refinement sensor for this problem. Where a sensor has been used as the reference, it is indicated as “ref.” in the corresponding line.

	$\bar{K}_{\mathrm{Kn}}$	$\bar{K}_{\omega}$	$\bar{K}_{\mathrm{Q}}$	$\bar{K}_{\mathrm{F}}$
Cylinder (Re=100)	0.77	0.82	0.77	0.38
Cylinder (Re=1000)	0.71	0.70	0.67	0.46
Lid-driven Cavity	0.54	0.71	0.66	0.32
Riemann	0.14	0.67	0.49	0.41
Droplet	0.63	0.85	0.64	0.55

Table 4: Summary of the space average of the refinement factor for each sensor on the different benchmarks. If the sensor is able to correctly identify areas of the simulation domain that require local refinement patches, then the lower the spatial average value, the better.

The *computed* Knudsen number in the coarse grid, noted Kn_c, does not correspond to the actual, physical Kn. On the other hand, the computed Knudsen number Kn_f in the refined grid corresponds to the physical Knudsen number Kn by definition, as one requires that the refined grid is such that it doesn't need to be refined anymore. Hence a term $\mathrm{Kn}/\mathrm{Kn}_c$ is expected to be included in the conversion factor between the grids.

We now want to express δt_f as a function of δt_c, Kn_c and Kn. The off-equilibrium population, in a given lattice, takes the form

$$f^{\mathrm{neq}} \sim \left(\frac{\tau^*}{\delta t^*} + \frac{1}{2}\right) A, \tag{A.1}$$

where A denotes the linear combination of the time and space gradients on macroscopic quantities. Eq. (A.1) originates from the Chapman-Enskog expansion (discussed in Section 2.2 and described for instance in [45]), where our definition $\bar{\tau} = \tau^*/\delta t^* + 1/2$ has been used. Note that the operator ∇ does not depend on the resolution.

When acoustic scaling is employed, the velocity is the same in both lattices as $f_c^{\mathrm{eq}} = f_f^{\mathrm{eq}}$, and it follows that $\mathrm{Kn}/\mathrm{Kn}_C = f_f^{\mathrm{neq}}/f_c^{\mathrm{neq}}$. Hence,

$$\frac{\mathrm{Kn}}{\mathrm{Kn}_c} = \frac{2\tau^* + \delta t_f^*}{2\tau^* + \delta t_c^*}. \tag{A.2}$$

According to Eq. (14), and using $c_0^* = c_s \delta x^*/\delta t^*$, the physical viscosity can be expressed as

$$\nu^* = c_0^{*2} \tau^*, \tag{A.3}$$

which, using Eq. (A.2), finally leads to

$$\delta t_f^* = \delta t_c^* \frac{\mathrm{Kn}}{\mathrm{Kn}_c} + \frac{2\nu^*}{c_0^{*2}} \left(\frac{\mathrm{Kn}}{\mathrm{Kn}_c} - 1\right), \tag{A.4}$$

where the second term can be neglected. Indeed, since $\nu^*/c_0^{*2} = \tau^*$, the condition $\mathrm{Kn} \ll 1$, under which is done the perturbation expansion of Eq. (15), can be formulated as $c_0^* \tau^* \ll l_0^*$, hence $\tau^* \ll \delta t^*$ for a given lattice.

Another important remark is that, as a consequence of the recursive dependence of the distribution functions (see for instance Eq. (43) from [45]), the unknown population of the order $n+1$ contains a τ^n coefficient : $f^{(n)} \approx \tau^n$. It is then clear that the proposed criterion contains a dependence on the relaxation time, that is not studied in this work.

Appendix B. Supplementary visualizations of the 2D cavity problem with adaptive mesh refinement

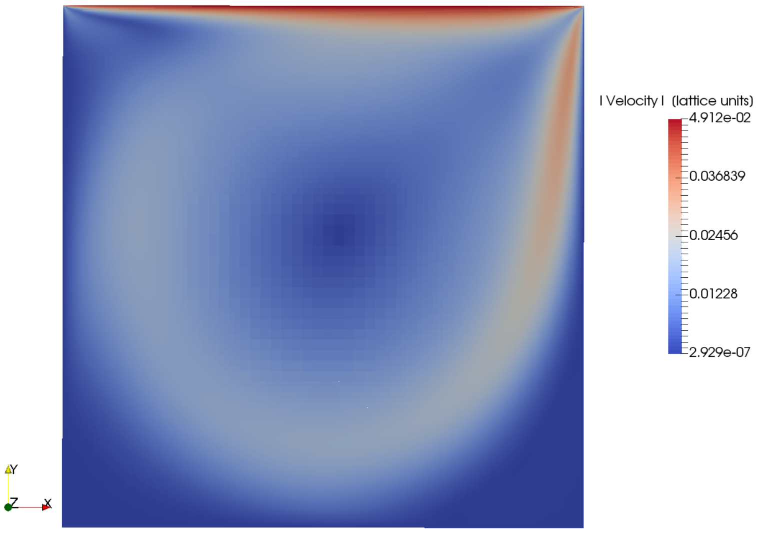

Figure B.11: Snapshot of a 2D lid-driven cavity simulation using adaptive mesh refinement with four refinement levels, after convergence of the flow ($t = 50$ dimensionless time periods). The velocity norm of the fluid is superimposed to the refined grid.

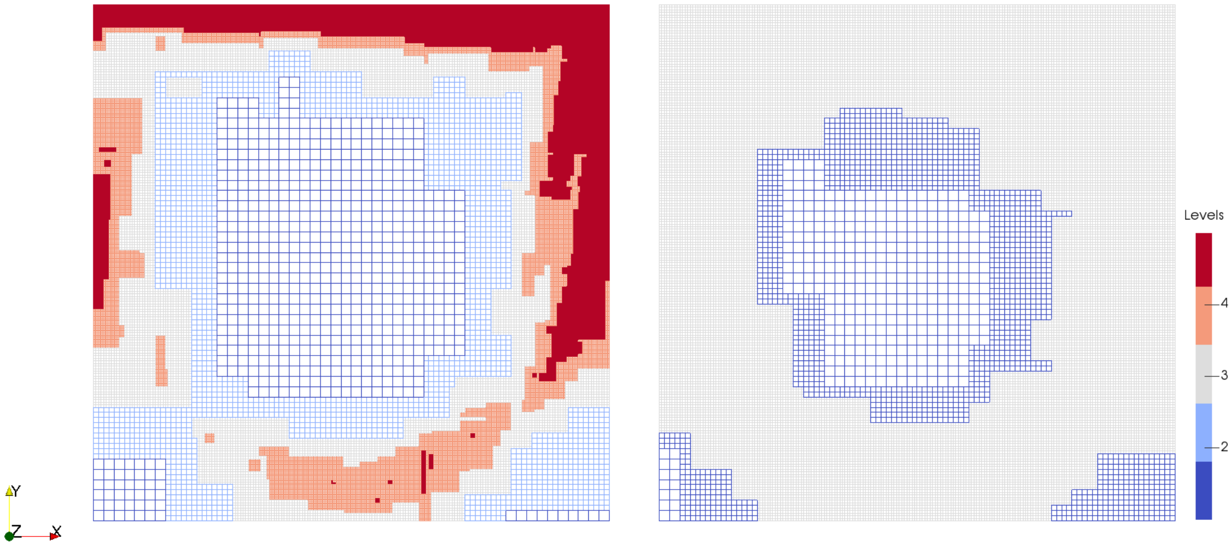

Figure B.12: Snapshot of 2D lid-driven cavity simulations using adaptive mesh refinement with (left) five and (right) three refinement levels, after convergence of the flow ($t = 50$ dimensionless time periods).

References

[1] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.*, 30(1):329–364, 1998.

[2] B. Chopard and M. Droz. *Cellular automata modeling of physical systems.* Cambridge University Press, June 2005.

[3] Z. Guo and C. Shu. *Lattice Boltzmann Method and Its Applications in Engineering.* World Scientific, 2013.

[4] H. Huang, M. Sukop, and X. Lu. *Multiphase lattice Boltzmann methods: Theory and application.* John Wiley & Sons, 2015.

[5] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. Viggen. *The Lattice Boltzmann Method: Principles and Practice.* Graduate Texts in Physics. Springer International Publishing, 2016.

[6] S. Succi. *The Lattice Boltzmann Equation: For Complex States of Flowing Matter.* Oxford University Press, 2018.

[7] A. Parmigiani, J. Latt, M. B. Belgacem, and B. Chopard. A lattice Boltzmann simulation of the Rhone river. *Int. J. Mod. Phys. C*, 24(11):1340008, 2013.

[8] S. Kobel, A. Valero, J. Latt, P. Renaud, and M. Lutolf. Optimization of microfluidic single cell trapping for long-term on-chip culture. *Lab Chip*, 10:857–863, 2010.

[9] C. Coreixas. Round cavity noise simulations using lattice-Boltzmann solver. In *11th PEGASUS-AIAA Student Conference*, April 2015.

[10] A. Sengissen, J.-C. Giret, C. Coreixas, and J.-F. Boussuge. Simulations of LAGOON landing-gear noise using lattice Boltzmann solver. In *21st AIAA/CEAS Aeroacoustics Conference*, page 2993, 2015.

[11] F. Brogi, O. Malaspinas, B. Chopard, and C. Bonadonna. Hermite regularization of the lattice Boltzmann method for open source computational aeroacoustics. *J. Acoust. Soc. Am*, 142(4):2332–2345, 2017.

[12] J. Appelbaum, B. M. Duda, E. Fares, and M. R. Khorrami. Airframe noise simulations of a full-scale aircraft. In *AIAA/CEAS Aeroacoustics Conference*, page 2973, 2018.

[13] C. Feuchter. Direct aeroacoustic simulation with a cumulant lattice-Boltzmann model. *Comput. Fluids*, 224:104970, 2021.

[14] O. Filippova and D. Hänel. Grid refinement for lattice-BGK models. *J. Comput. Phys.*, 147(1):219 – 228, 1998.

[15] A. Dupuis and B. Chopard. Theory and applications of an alternative lattice Boltzmann grid refinement algorithm. *Phys. Rev. E*, 67(6):066707, June 2003.

[16] H. Chen, O. Filippova, J. Hoch, K. Molvig, R. Shock, C. Teixeira, and R. Zhang. Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Physica A*, 362(1):158 – 167, 2006. Discrete Simulation of Fluid DynamicsProceedings of the 13th International Conference on Discrete Simulation of Fluid Dynamics13th International Conference on Discrete Simulation of Fluid Dynamics.

[17] M. Rohde, D. Kandhai, J. J. Derksen, and H. E. A. van den Akker. A generic, mass conservative local grid refinement technique for lattice-Boltzmann schemes. *Int. J. Numer. Meth. Fluids*, 51(4):439–468, 2006.

[18] D. W. Lagrava Sandoval. *Revisiting grid refinement algorithms for the lattice Boltzmann method.* PhD Thesis, University of Geneva, 2012.

[19] F. Gendre, D. Ricot, G. Fritz, and P. Sagaut. Grid refinement for aeroacoustics in the lattice Boltzmann method: A directional splitting approach. *Phys. Rev. E*, 96:023311, Aug 2017.

[20] T. Astoul. *Towards improved lattice Boltzmann aeroacoustic simulations with non-uniform grids: application to landing gears noise prediction.* PhD thesis, Aix-Marseille Université, 2021.

[21] B. Crouse, E. Rank, M. Krafczyk, and J. Tölke. A LB-based approach for adaptive flow simulations. *Int. J. Mod. Phys. B*, 17(01n02):109–112, January 2003.

[22] G. Eitel-Amor, M. Meinke, and W. Schröder. A lattice-Boltzmann method with hierarchically refined meshes. *Comput. Fluids*, 75:127–139, 2013.

[23] A. Fakhari and T. Lee. Finite-difference lattice Boltzmann method with a block-structured adaptive-mesh-refinement technique. *Phys. Rev. E*, 89(3):033310, March 2014.

[24] S. Kamkar, A. Wissink, V. Sankaran, and A. Jameson. Feature-driven Cartesian adaptive mesh refinement for vortex-dominated flows. *J. Comput. Phys.*, 230(16):6271–6298, 2011.

[25] J.-P. Meng, N. Dongari, J. Reese, and Y. Zhang. Breakdown parameter for kinetic modeling of multiscale gas flows. *Physical Review E*, 89(6), June 2014.

[26] N. Singh and M. Kroells. Quantitative uncertainty metric to assess continuum breakdown for nonequilibrium hydrodynamics. *Phys. Rev. Fluids*, 6:L111401, Nov 2021.

[27] D. Lagrava, O. Malaspinas, J. Latt, and B. Chopard. Advances in multi-domain lattice Boltzmann grid refinement. *J. Comput. Phys.*, 231(14):4808 – 4822, 2012.

[28] D. Yu and S. S. Girimaji. Multi-block lattice Boltzmann method: Extension to 3D and validation in turbulence. *Physica A*, 362(1):118 – 124, 2006. Discrete Simulation of Fluid Dynamics Proceedings of the 13th International Conference on Discrete Simulation of Fluid Dynamics 13th International Conference on Discrete Simulation of Fluid Dynamics.

[29] S. Geller, M. Krafczyk, J. Tölke, S. Turek, and J. Hron. Benchmark computations based on lattice-Boltzmann, finite element and finite volume methods for laminar flows. *Comput. Fluids*, 35(8):888–897, 2006. Proceedings of the First International Conference for Mesoscopic Methods in Engineering and Science.

[30] N. Pellerin, S. Leclaire, and M. Reggio. An implementation of the Spalart-Allmaras turbulence model in a multi-domain lattice Boltzmann method for solving turbulent airfoil flows. *Comput. Math. Appl.*, 70(12):3001–3018, 2015.

[31] T. Astoul, G. Wissocq, J.-F. Boussuge, A. Sengissen, and P. Sagaut. Analysis and reduction of spurious noise generated at grid refinement interfaces with the lattice Boltzmann method. *J. Comput. Phys.*, 418:109645, 2020.

[32] T. Astoul, G. Wissocq, J.-F. Boussuge, A. Sengissen, and P. Sagaut. Lattice Boltzmann method for computational aeroacoustics on non-uniform meshes: A direct grid coupling approach. *J. Comput. Phys.*, 447:110667, 2021.

[33] Z. Yu and L.-S. Fan. An interaction potential based lattice Boltzmann method with adaptive mesh refinement (AMR) for two-phase flow simulation. *J. Comput. Phys.*, 228(17):6456–6478, 2009.

[34] F. Schornbaum and U. Rüde. Massively parallel algorithms for the lattice Boltzmann method on nonuniform grids. *SIAM J. Sci. Comput*, 38(2):C96–C126, 2016.

[35] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82(1):64–84, 1989.

[36] R. Deiterding and S. L. Wood. An Adaptive Lattice Boltzmann Method for Predicting Wake Fields Behind Wind Turbines. In A. Dillmann, G. Heller, E. Krämer, C. Wagner, and C. Breitsamter, editors, *New Results in Numerical and Experimental Fluid Mechanics X*, volume 132, pages 845–857. Springer International Publishing, Cham, 2016.

[37] R. Deiterding and S. L. Wood. Predictive wind turbine simulation with an adaptive lattice Boltzmann method for moving boundaries. *J. Phys.: Conf. Ser.*, 753:082005, sep 2016.

[38] K. Feldhusen, R. Deiterding, and C. Wagner. A dynamically adaptive lattice Boltzmann method for thermal convection problems. *International Journal of Applied Mathematics and Computer Science*, 26:735 – 747, 2016.

[39] M. Grondeau and R. Deiterding. Direct prediction of flow noise around airfoils using an adaptive lattice Boltzmann method. In B. Stoevesandt, G. Schepers, P. Fuglsang, and Y. Sun, editors, *Handbook of Wind Energy Aerodynamics*. Springer, 2022.

[40] M. F. Barad, J. G. Kocheemoolayil, and C. C. Kiris. Lattice Boltzmann and Navier-Stokes Cartesian CFD Approaches for Airframe Noise Predictions. In *23rd AIAA Computational Fluid Dynamics Conference*, Denver, Colorado, June 2017. American Institute of Aeronautics and Astronautics.

[41] J. Latt, O. Malaspinas, D. Kontaxakis, A. Parmigiani, D. Lagrava, F. Brogi, M. B. Belgacem, Y. Thorimbert, S. Leclaire, S. Li, F. Marson, J. Lemus, C. Kotsalos, R. Conradin, C. Coreixas, R. Petkantchin, F. Raynaud, J. Beny, and B. Chopard. Palabos: Parallel lattice Boltzmann solver. *Comput. Math. Appl.*, 81:334 – 350, 2021.

[42] R. Deiterding. Block-structured adaptive mesh refinement - theory, implementation and application. *ESAIM: Proceedings*, 34:97–150, 2011.

[43] P. Bhatnagar, E. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94:511–525, May 1954.

[44] X. Shan and X. He. Discretization of the velocity space in the solution of the Boltzmann equation. *Phys. Rev. Lett.*, 80:65–68, Jan 1998.

[45] X. Shan, X.-F. Yuan, and H. Chen. Kinetic theory representation of hydrodynamics: A way beyond the Navier–Stokes equation. *J. Fluid Mech.*, 550:413–441, 3 2006.

[46] P. J. Dellar. An interpretation and derivation of the lattice Boltzmann method using Strang splitting. *Comput. Math. Appl.*, 65(2):129 – 141, 2013.

[47] S. Chapman and T. G. Cowling. *The mathematical theory of nonuniform gases*. Cambridge University Press, Cambridge, 1960.

[48] K. Huang. *Statistical Mechanics*. J. Wiley, New York, 1987.

[49] A. Gorban and D. Packwood. Enhancement of the stability of lattice Boltzmann methods by dissipation control. *Physica A*, 414:285 – 299, 2014.

[50] J. Latt, C. Coreixas, J. Beny, and A. Parmigiani. Efficient supersonic flow simulations using lattice boltzmann methods based on numerical equilibria. *Phil. Trans. R. Soc. A*, 378(2175):20190559, 2020.

[51] C. Coreixas and J. Latt. Compressible lattice Boltzmann methods with adaptive velocity stencils: An interpolation-free formulation. *Phys. Fluids*, 32(11):116102, 2020.

[52] W.-L. Wang and I. D. Boyd. Predicting continuum breakdown in hypersonic viscous flows. *Phys. Fluids*, 15(1):91–100, 2003.

[53] D. A. Lockerby, J. M. Reese, and H. Struchtrup. Switching criteria for hybrid rarefied gas flow solvers. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2105):1581–1598, 2009.

[54] V. Kolář and J. Šístek. Corotational and compressibility aspects leading to a modification of the vortex-identification q-criterion. *AIAA J.*, 53(8):2406–2410, 2015.

[55] J. Smagorinsky. General circulation model of the athmosphere. *Mon. Wheather Rev.*, 91:99–164, 1963.

[56] Q. Zou and X. He. On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Phys. Fluids*, 9(6):1591–1598, 1997.

[57] J. Latt, B. Chopard, O. Malaspinas, M. Deville, and A. Michler. Straight velocity boundaries in the lattice Boltzmann method. *Phys. Rev. E*, 77:056703, May 2008.

[58] C. W. Schulz-Rinne, J. P. Collins, and H. M. Glaz. Numerical solution of the Riemann problem for two-dimensional gas dynamics. *SIAM J. Sci. Comput.*, 14(6):1394–1414, 1993.

[59] A. Kurganov and E. Tadmor. Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers. *Numer. Methods Partial Differ. Equ.*, 18(5):584–608, 2002.

[60] X. Shan and H. Chen. Lattice Boltzmann model for simulating flows with multiple phases and components. *Phys. Rev. E*, 47:1815–1819, Mar 1993.

[61] Z. Guo, C. Zheng, and B. Shi. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys. Rev. E*, 65:046308, Apr 2002.

[62] Z. Guo. Well-balanced lattice Boltzmann model for two-phase systems. *Phys. Fluids*, 33(3):031709, 2021.

[63] E. Leriche. Direct numerical simulation in a lid-driven cubical cavity at high reynolds number by a Chebyshev spectral method. *J. Sci. Comput.*, 27(1):335–345, 2006.

[64] E. Leriche and S. Gavrilakis. Direct numerical simulation of the flow in a lid-driven cubical cavity. *Phys. Fluids*, 12(6):1363–1376, 2000.

[65] C. Gkoudesnes and R. Deiterding. Verification of the WALE large eddy simulation model for adaptive lattice boltzmann

methods implemented in the AMROC framework. In R. Deiterding, M. Domingues, and K. Schneider, editors, *Cartesian CFD Methods for Complex Applications: Thematic mini-symposium contributions from ICIAM 2019*, pages 123 – 144. Springer, 2021.

[66] U. Ghia, K. Ghia, and C. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.*, 48(3):387 – 411, 1982.

[67] J. Latt and C. Coreixas. *GPU performance of lattice Boltzmann simulations on non-uniform meshes.* 2021.

[68] I. Ginzburg, F. Verhaeghe, and D. d'Humieres. Two-relaxation-time lattice boltzmann scheme: About parametrization, velocity, pressure and mixed boundary conditions. *Communications in computational physics*, 3(2):427–478, 2008.