A comparison of semi-Lagrangian Vortex method and Lattice Boltzmann method for incompressible flows

Chloé Mimeau[a,c], Simon Marié[a,b], Iraj Mortazavi[a,c]

[a] *Conservatoire National des Arts et Métiers, 2 rue Conté 75003 Paris, France*

[b] *Laboratoire DynFluid*

[c] *Laboratoire M2N*

Abstract

The semi-Lagrangian Vortex method (VM) and the Lattice Boltzmann method (LBM) are used to investigate flows simulations in the incompressible regime. In this study, a proven version of each method is used and compared on different three dimensional benchmarks in terms of numerical accuracy, convergence, numerical diffusion and dissipation. The first comparisons are made on a convected vortex to study and compare the numerical dissipation of LBM and VM. Then the Taylor-Green vortex is investigated to compare the dissipation rate of the kinetic energy of each method. It is shown that both methods converge to the same solution but in a different way. The VM performs better than the LBM for the lowest resolution whereas LBM appears to be more accurate for the growing resolutions. These results are confirmed on 3D simulations with wall boundaries for the stiff test case of the wake behind a 3D cube at $Re = 290$ and $Re = 570$.

Keywords: Vortex Methods, Lattice Boltzmann, Comparison, Numerical simulation, method accuracy, Taylor-Green Vortex, flow around a cube

Introduction

The design of numerical methods to study fluid flows has had a tremendous development during past decades. A large family of these methods such as finite difference, finite volume or finite element approaches as well as spectral/pseudo-spectral methods that deal with primitive variables and purely Eulerian frameworks, have been extensively studied both from consistency/stability point of view as well as numerical diffusivity and dissipation characterization. The later aspect is important to explore how numerical methods discretization properties affect the numerical efficiency and robustness and is highly dependent on the link between the scheme and the grid.

Two other classes of methods, namely Lattice Boltzmann and particle approaches, have met a large development recently in the context of incompressible or weakly-compressible flows. The first and major common thread shared by these two methods relies on the fact that they intrinsically differ from the traditional approaches previously cited. In particular they do not directly

[0] **Abbreviations:** LBM, Lattice Boltzmann Method; VM, Vortex Method

deal with primitive variables contrary to previous methods and, in particular, the pressure field is not directly computed in their primary discretization. Moreover, they shortcut the non-linearities related to the advection phenomenon. For these reasons, they represent for many flow problems a promising alternative of software design. Now, if one considers Lattice Boltzmann and particle approaches with respect to each other, they interestingly show complementary aspects: indeed, on one side Lattice Boltzmann methods lie on a mesoscopic approach: they follow the evolution of probability distribution functions of fluid particles, thank to a fixed lattice, instead of calculating the usual macroscopic variables involved in Navier-Stokes equations. On the other side, particle methods are Lagrangian approaches: the particles, playing the role of discretiazation elements and computational "points", move with the material velocity and the evaluation of the macroscopic quantities are evaluated on these numerical particles. Lattice Boltzmann is a mesoscopic Eulerian approach, whereas particle methods are macroscopic and Lagrangian. A recent focus on this kind of approaches has been investigated [1] for Lattice Boltzmann and SPH (Smoothed Particle Hydrodynamics) methods to solve 2D problems in multiphase flows, which demonstrates the current interest of such type of alternative and non-traditional methods.

The present work aims at describing and comparing a semi-Lagrangian Vortex particle method and a Lattice Boltzmann method, in order to try to numerically highlight the above statements in the case of various physical 3D problems in CFD (Computational Fluid Dynamics). All the computations made in this study are based on in-house and research codes, developed or co-developed by the authors of the present paper.
Vortex methods (VM) belong to particle methods. They are based on a Lagrangian or semi-Lagrangian description of the governing equations (Euler equations, linear convection-diffusion equation, Navier-Stokes equations) which, when they are resolved, provide the dynamics and the evolution of the fluid elements. In the case of Vortex methods, the fluid elements are numerical particles, characterized by their spacial position and the vorticity they carry. With the first vortex sheet computations in the 1930's [2, 3], the Vortex methods correspond to one of the first numerical method ever used in the Computational Fluid Dynamics community. This can be explained by their very natural framework provided by the particle approach, mimicking the physics, which make them particularly well suited for advection dominated flow problems in particular because the Lagrangian treatment of the convective term is free of numerical dissipation. In the 70's, a lot of efforts have been devoted to propose numerical developments that overcome the main intrinsic difficulties of Lagrangian Vortex methods, mostly relying on the modeling of the viscous effects in Navier-Stokes equations [4, 5] and the treatment of boundary conditions [6]. Significant developments were also made in the last decade in order to provide to Lagrangian Vortex methods (also called Particle Vortex methods) an efficient evaluation of the velocity field. Indeed, for N_p particles in the computational domain, the classical resolution of the Biot-Savart law (which gives the velocity from the vorticity) implies to compute the interactions between all the particles, leading to a $O(N_p^2)$ computational cost. The development of the Fast Multipole Method (FMM) allowed to drastically reduce the cost of such operation [7, 8, 9, 10, 11]. Moreover, the issue related to the distortion of the particle distribution, which is one of the major drawback of pure Lagrangian methods, has been subject to deep researches in order to design accurate Particle Vortex methods preventing from high clustering or rarefaction of the vortex particles in the domain [10, 12, 13].
Thanks to the remeshing technique introduced in the 90's [14, 15], it exists another main variant of Vortex methods, which relies on a semi-Lagrangian approach. The remeshing technique was originally proposed to bypass the inherent problem of the distortion of Lagrangian particle

distribution. It consists in periodically redistributing the particles onto an underlying Cartesian grid in order to ensure their overlapping and thus the convergence of the solution. Following the introduction of this remeshing procedure, semi-Lagrangian Vortex methods emerged, also called remeshed Vortex methods (or Vortex Particle-Mesh method or Vortex-in-Cell method). They are characterized by the fact that the vorticity transport equations and the velocity equation are both handled on the particles field and on a Cartesian grid. They allow one to benefit from the strengths of Particle Vortex schemes to handle the flow advection and from the one of grid-based methods, like immersed boundary methods to model boundary conditions or FFT to solve the Poisson equation. Based on these different improvements, Vortex methods have matured and now offer a robust framework able to compete with pure Eulerian methods in the handling of challenging problems like interface tracking for colliding obstacles [16], fluid-structure interaction [17, 18, 19], shape optimization [20], flows past bluff bodies [19], passive control using porous media [21], wind turbine aerodynamics [22] or reinforcement learning [23]. In the present study, the VM denomination will be restricted to the remeshed (semi-Lagrangian) Vortex method. All the VM simulations presented in this work are based on an in-house parallel and object-oriented library, implemented in Python/Fortran language.

The Lattice Boltzmann Method [24, 25] (LBM) is nowadays recognized as a fast and reliable algorithm to numerically solve the Boltzmann equation. The physics of this kind of model is led by a mesoscopic description of the collision between particles. Hence, if a given collision operator is chosen with a reliable equilibrium state, a wide variety of physical modeling could be obtained, from turbulent to relativistic flows [26, 27, 28]. In order to describe fluid dynamics, governed by the Navier-Stokes equation, the BGK collision operator, based on a relaxation towards the equilibrium, has been shown to be an efficient mesoscopic description [29]. The Lattice Boltzmann methods then perform a discretization of the velocity space in which the fluid particles are allowed to displace. This discretization has to be highly connected to the mesh and induces strong constrains in the choice of the velocity lattice. These constrains are often coupled to the algorithmic advection which basically relies on a collision and a propagation step. The propagation step is led by the mesh and the collision step depends on the description of the relaxation process. The traditional way to describe this step is to assign a relaxation parameter to the main statistical moments when they relax to their equilibrium state. This model, also referred to as *MRT* model for Multiple Relaxation Times [30] has been shown to recover the behavior of the weakly compressible Navier-Stokes equation. It has been shown [31, 32] that this kind of method has a lower dissipation error compared to traditional finite-difference schemes. As a counterpart, LBM suffers from numerical instabilities when Reynolds number becomes high. The origins of LBM instabilities have been actively studied and remain an open subject [33, 34, 35]. Some modern collision models have now emerged and could improve those aspects by changing the moments definition [36], by re-normalizing the post-collision step [37], or by enforcing energy conservation [38, 39, 40]. A theoretical comparison of a wide variety of collision models has been recently reviewed in [41, 35]. In the present study, the LBM method will be restricted to the standard *MRT* model with optimized relaxation times defined in [30]. All the LBM simulations exposed in this work are based on this approach, which is implemented in a proper in-house parallel scientific Python/Fortran code.

Despite their increasing applications, the Lattice Boltzmann and Vortex methods suffer from a lack of extensive computational characterization in the literature (dissipation, diffusivity, parameter dependency, etc.) and deserve a better focus on such issues. This paper is devoted to the numerical characterization of a Lattice Boltzmann and a remeshed Vortex method. On one hand

this work aims at enlightening the effect of the time and space discretization refinement on the accuracy and robustness of these techniques and on the other hand at clarifying their grid vulnerability with a quantitative evaluation of the numerical diffusion and dissipation. Finally, this study is an attempt to classify the strong and weak points of both methods in order to offer an understanding of their range of efficiency.

This paper is organized as follows, the first section is dedicated to the presentation of some basic theoretical background of each method, where the differences and similarities of LBM and VM algorithms are clearly highlighted and discussed. Then, in section 2, the two methods are compared in terms of numerical dissipation on classical test cases : first, the simulation of a simple convected vortex is investigated in a 3D periodic domain and then the three-dimensional Taylor-Green vortex flow is performed, followed by a discussion on the evolution of enstrophy and kinetic-energy. Then, section 3 discusses the effect of wall boundary condition for each method on the three-dimensional flow past a cube at different Reynolds numbers.

1. Theoretical backgrounds

1.1. Vortex method

1.1.1. Governing equations

Vortex methods are based on the velocity-vorticity formulation of the incompressible Navier-Stokes equations in a domain D, which reads:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} = \frac{1}{Re}\Delta\boldsymbol{\omega} \quad \text{in } D. \tag{1}$$

In this equation $\boldsymbol{\omega}$, $\mathbf{u}$ and Re respectively denote the vorticity, the velocity and the Reynolds number. The first term corresponds to the advection of the vorticity $\boldsymbol{\omega}$ carried by the particles at the velocity $\mathbf{u}$. The second non-linear term $(\boldsymbol{\omega}\cdot\nabla)\mathbf{u}$ models the stretching of the flow structures (it vanishes in 2D) and the right hand side term represents the diffusion of $\boldsymbol{\omega}$ under viscous effects. This equation has to be coupled to the system giving the velocity in terms of the vorticity. Using the incompressibility condition, the velocity may be directly linked to the vorticity through the following Poisson equation:

$$\Delta\mathbf{u} = -\nabla \times \boldsymbol{\omega}. \tag{2}$$

The system (1)-(2) has to be complemented by appropriate boundary conditions at artificial boundaries and at solid boundaries (if present). The prescription of such solid boundary condition may be done by adding a forcing term in the right hand side of equation (1). This issue will be specifically addressed in section 3.

1.1.2. Discretization method

To solve the $(\boldsymbol{\omega}, \mathbf{u})$ Navier-Stokes equations (1)-(2), the flow is discretized onto particles that carry the vorticity field $\boldsymbol{\omega}$ transported at the velocity $\mathbf{u}$ and the resolution of the governing equations is based on a splitting algorithm, which consists at each time step in successively solving

the following equations:

$$\Delta \mathbf{u} = -\nabla \times \boldsymbol{\omega} \tag{3}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \mathrm{div}(\boldsymbol{\omega} : \mathbf{u}) \tag{4}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \frac{1}{Re}\Delta \boldsymbol{\omega} \tag{5}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\, \boldsymbol{\omega} = 0 \tag{6}$$

$$\Delta t_{\mathrm{adapt}} = \frac{\mathrm{LCFL}}{\|\nabla \mathbf{u}\|_\infty} \tag{7}$$

The discretization of each equation of this fractional step algorithm is realized in this study by using a remeshed vortex method.
The advection of vorticity field (eq. (6)) is performed in a Lagrangian way using a vortex method:

$$\begin{cases} \dfrac{dx_p}{dt} = u_j^n(x_p), \quad j \in \{1,2,3\} & \\ x_p^n = x_i^n, & \text{(advection)} \\ \dfrac{d\omega_p}{dt} = 0, & \\ \omega_i^{n+1} = \displaystyle\sum_p \omega_p^n \Lambda_{4,2}\left(\frac{x_p^{n+1} - x_i}{h}\right) & \text{(remeshing)} \end{cases} \tag{8}$$

At each time step of the method, numerical particles are created on the nodes i of an underlying uniform Cartesian grid ($x_p^n = x_i^n$) and the new position x_p^{n+1} of each particle p is obtained by solving $d_t x_p = u_j^n(x_p)$, while the transported vorticity remains constant ($d_t \omega_p = 0$). This Lagrangian treatment of the advection step is close to the physics and provides a flexible resolution of the non-linearities, decreasing drastically the numerical diffusion. In this work, we numerically integrate the particle positions in time ($d_t x_p = u_j^n(x_p)$) by using an explicit 2$^{\mathrm{nd}}$ order Runge-Kutta method. The only difficulty of this step relies on the interpolation of the velocity field at the intermediate position of the particles in the RK2 scheme, since this intermediate position will not always be aligned with the grid. In the present case, it is performed by using bilinear interpolation.
Once the particle positions x_p have been updated according to the flow velocity, the vorticity carried by each particle is redistributed on the neighbouring points of the underlying Cartesian grid using a remeshing kernel of type $\Lambda_{p,r}$ [42] (cf last equation of system (8)). The $\Lambda_{p,r}$ remeshing kernels are piecewise polynomial functions of regularity C^r, satisfying the conservation of the first p moments. The one used in this work is $\Lambda_{4,2}$: this kernel is of regularity C^2, it satisfies the conservation of 4 moments and includes 6 grid points by direction in its support on which each particle can be redistributed. In this work, the particle advection and the remeshing procedure are performed using a directional splitting approach [43]. It consists in successively solving 1D convection/remeshing problems, direction by direction, as written in eqs. (8). This directional splitting allows to save significant computational efforts compared to a classical tensorial approaches, especially in 3D.
The systematic remeshing of particles onto an Eulerian grid at each time step after the advection

stage (eq. (6)), enables to ensure the overlapping of particles required for the convergence of the method. Moreover the presence of the grid allows to discretize the other equations using efficient and/or fast grid methods (finite differences and spectral method based on FFT evaluations). In the present algorithm, equations (3) to (5) are solved on the grid.
The Poisson equation (3) is resolved in the Fourier space with periodic boundary conditions according to the following expression :

$$\widehat{\mathbf{u}}(\xi) = \frac{1}{|\xi|^2}(\widehat{\boldsymbol{\nabla}\times\boldsymbol{\omega}}) \tag{9}$$

In the presence of an underlying mesh that is uniform and Cartesian (like in the present VM), the use of FFT-based evaluations for the velocity computation may be considered as one of the most appropriate and efficient approach [44, 45]. However, if the grid is non-uniform then the use of other type of algorithm is mandatory. In that case, the most famous and efficient one, which is widely used in meshless (i.e. purely Lagrangian) Vortex methods [46, 11], is the Fast Multipole Method (FMM) like in the works dealing with adaptive mesh refinement (AMR) [47, 48].

Regarding the stretching problem (4), it is considered here in its conservative formulation:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \mathrm{div}(\boldsymbol{\omega} : \mathbf{u}), \tag{10}$$

where $\mathrm{div}(\boldsymbol{\omega} : \mathbf{u}) := (\boldsymbol{\omega}\cdot\nabla)\,\mathbf{u} + \mathbf{u}\,\mathrm{div}(\boldsymbol{\omega})$. The time integration scheme chosen here to discretize this equation is the 3$^{\mathrm{rd}}$ order Runge-Kutta scheme. Throughout this time discretization, the velocity field involved in the divergence operator is not modified. The divergence operator is discretized through a 4$^{\mathrm{th}}$ order centered finite-differences scheme on the grid.
Concerning the diffusion equation (5), it is discretized in time using an implicit 1^{st} order Euler scheme and then solved in the Fourier space.
An adaptive time-step $\Delta t_{\mathrm{adapt}}$ (7) is computed at the end of the fractional step algorithm. It is based on the non-linear stability of the advection/remeshing scheme in Vortex methods:

$$\Delta t_{\mathrm{adv}} \leq \frac{\mathrm{LCFL}}{\|\nabla \mathbf{u}\|_\infty}, \tag{11}$$

where the LCFL denotes the Lagrangian CFL [49]. This number must satisfy LCFL < 1 [42], which, from a physical point of view, imposes that particles trajectories do not cross. As the time step defined by this stability condition (11) is not constrained by the grid size or the distance between the particles but only by the flow strain, it often provides larger time steps compared to Eulerian schemes, based on CFL conditions.

Table 1 summarizes the time and space discretization schemes used in this work to solve each equation of the present fractional step algorithm.

The fractional construction of this algorithm offers a flexibility in the choice of the discretization schemes of each step. The algorithm exposed in this work is one of the different existing remeshed vortex algorithms in literature. We can cite for instance the remeshed vortex algorithms used in [50] and [16], which differ from the present one by the nature of the remeshing kernel ($\Lambda_{2,1}$ instead of $\Lambda_{4,2}$ here), by a tensorial approach for the advection/remeshing step (contrary to the directional one proposed here), or by the use of a centered fourth-order [50] or second-order [16] finite differences scheme for the evaluation of the viscous term. In the algorithm established

Equation	Time discretization method	Space discretization method
Poisson equation (3)	-	spectral method
Stretching (4)	RK3 scheme	4^{th} order centered FD
Diffusion (5)	implicit Euler scheme	spectral method
Advection + Remesh (8)	RK2 scheme (particles advec.)	remeshing with $\Lambda_{4,2}$ kernel
Adaptive time step (7)	-	4^{th} order centered FD (LCFL < 1)

Table 1: Time and space discretization methods used for the resolution of the viscous splitting VM algorithm (eqs. (3) to (7)).

by [22], the main difference with respect to [50, 16] or the present one relies on the fact that the remeshing operation is not performed every time step but only every 5 time steps (using a $\Lambda_{2,1}$ kernel, with a tensorial approach). This choice implies a particle-to-mesh and a mesh-to-particle interpolation operation for the time steps where remeshing in not applied.
Providing the stability and consistency of all the numerical schemes used in each sub-steps, remeshed Vortex method algorithms are proved to converge numerically as shown in the above literature reference and as it will be highlighted in the next sections of this paper.

1.2. Lattice Boltzmann method

The Lattice Boltzmann method [24], used to perform fluid flow simulations, is not directly based on the resolution of the Navier-Stokes equations but is a particular discretization of the Boltzmann equation, describing the dynamics of gas:

$$\frac{\partial f(\mathbf{c},\mathbf{x},t)}{\partial t} + c_i \frac{\partial f(\mathbf{c},\mathbf{x},t)}{\partial x_i} = \left(\frac{\partial f}{\partial t}\right)_{\text{coll}} \tag{12}$$

where $f(\mathbf{c},\mathbf{x},t)$ is the distribution of particles density with a given velocity $\mathbf{c}$ at a given position $\mathbf{x}$ at time t. The left hand side terms corresponds to the propagation (advection) of the particles and the right hand side term represents the time evolution of the distribution function f due to the collisions between particles. In this work, the collision between particles is given by the BGK [29] collision operator which describes an average collision effect through the relaxation to a local equilibrium f^{eq} with a relaxation parameter τ:

$$\frac{\partial f(\mathbf{c},\mathbf{x},t)}{\partial t} + c_i \frac{\partial f(\mathbf{c},\mathbf{x},t)}{\partial x_i} = -\frac{1}{\tau}(f - f^{eq}) \tag{13}$$

In order to solve (13) numerically, one should restrict the velocity space to a discrete one. This part is very important and gives the LBM its numerical originality. To perform this discretization, the standard Gauss quadrature is used and is detailed in [26]. The number of lattice points needed to achieve a given dynamics is directly connected to this latter step and converts $f(\mathbf{c},\mathbf{x},t)$ into $f(\mathbf{c}_\alpha,\mathbf{x},t)$ where α denotes the discrete velocities indices. The usual 19 velocities lattice (D3Q19) allows to recover the dynamics described by the 3D isothermal Navier-Stokes equations for small Mach numbers. From this lattice, one could define an equilibrium function in its incompressible polynomial form:

$$f_\alpha^{eq}(\mathbf{x},t) = \rho\omega_\alpha + \rho_0\omega_\alpha\left(\frac{\mathbf{u}.\mathbf{c}_\alpha}{\tilde{c}_0^2} + \frac{(\mathbf{u}.\mathbf{c}_\alpha)^2}{2\tilde{c}_0^4} - \frac{|\mathbf{u}|^2}{2\tilde{c}_0^2}\right) \tag{14}$$

where ρ_0 is a unity constant and the coefficients ω_α and $\tilde{c_0}$ are defined by:

$$\begin{cases} \omega_\alpha = & \frac{1}{3}, \frac{1}{18}, \frac{1}{36} & , \quad \alpha = 0, \alpha = 1..6, \alpha = 7..18 \\ \tilde{c_0}^2 = & \frac{1}{3} & \end{cases} \tag{15}$$

The macroscopic variables, $\rho, \mathbf{u}$ are linked to the distribution functions f by their moments:

$$\begin{aligned} \rho &= \sum_\alpha f_\alpha \\ \rho\mathbf{u} &= \sum_\alpha \mathbf{c}_\alpha f_\alpha \end{aligned} \tag{16}$$

Then, the final step to get the LBM algorithm is to perform a space and time discretization. This is achieved by using the advective properties of the left-hand side of equation (13) which can be integrated along the characteristic $\mathbf{c}_\alpha$ to get the following LBM algorithm:

$$\begin{cases} g_\alpha^{coll}(\mathbf{x}, t) &= g_\alpha(\mathbf{x}, t) - \frac{dt}{\tau_g}(g_\alpha(\mathbf{x}, t) - g_\alpha^{eq}(\mathbf{x}, t)) \\ g_\alpha(\mathbf{x}, t) &= g_\alpha^{coll}(\mathbf{x} - \mathbf{c}_\alpha dt, t - dt) \end{cases} \tag{17}$$

where the g_α distribution function comes from the integration step to get an explicit formulation and is related to the distribution f_α with the relation $g_\alpha = f_\alpha + \frac{dt}{2\tau}(f_\alpha - f_\alpha^{eq})$ which implies $g_\alpha^{eq} = f_\alpha^{eq}$ and $\tau_g = \tau + \frac{dt}{2}$.

From this, the algorithm imposes $dt = dx = 1$ in order to be consistent with a uniform grid size imposed by the lattice. Then one could define some physical time and grid steps in order to compute physical quantities from lattice quantities. This is done by introducing the physical speed of sound c_0 which defines:

$$\Delta t = \frac{\tilde{c}_0 \Delta x}{c_0} \tag{18}$$

where Δx is the physical grid step obtained by discretizing a reference length scale L with a given number of points N. Based on these parameters, the LBM algorithm can recover the Navier-Stokes dynamics with a second-order accuracy in space and time.

In the BGK collision operator, the distribution functions relax toward the equilibrium according to a single relaxation time. A more sophisticated idea is to relax each moment according to a proper relaxation time. This method is called multiple relaxation time (MRT) [30] and is known to alleviate some stability issues encountered with the BGK operator. The implementation of the MRT model is based on the modification of the collision step of equation (17) which is done in the momentum space:

$$\begin{cases} \mathbf{m}^{coll}(\mathbf{x}, t) &= \mathbf{m}(\mathbf{x}, t) - \mathcal{S}(\mathbf{m}(\mathbf{x}, t) - \mathbf{m}^{eq}(\mathbf{x}, t)) \\ \mathbf{g}(\mathbf{x}, t) &= \mathcal{M}^{-1}\mathbf{m}^{coll}(\mathbf{x} - \mathbf{c}_\alpha dt, t - dt) \end{cases} \tag{19}$$

where the matrix $\mathcal{M}$, transforms the distribution functions into moments:

$$\mathbf{m} = \mathcal{M}\mathbf{g}. \tag{20}$$

$\mathcal{M}$ is a square transformation matrix. If the first line of the matrix is filled only by 1, then the first moment is the density. The invert transformation from the moments to distribution functions is simply $\mathbf{g} = \mathcal{M}^{-1}\mathbf{m}$. Further details about the construction of the matrix $\mathcal{M}$ can be found in [30]. The equilibrium moments are obtained from $\mathbf{m}^{eq} = \mathcal{M}\mathbf{g}^{eq}$. The diagonal of $\mathcal{S}$ corresponds to the inverse of the relaxation time, also called relaxation rate, associated with each moment:

$$\mathcal{S} = diag\,(0, s_1, s_2, 0, s_3, 0, s_3, 0, s_3, s_\nu, s_2, s_\nu, s_2, s_\nu, s_\nu, s_\nu, s_4, s_4, s_4)) \tag{21}$$

where s_ν is related to the fluid viscosity:

$$\frac{1}{s_\nu} = 3\nu - \frac{1}{2} \tag{22}$$

The other relaxation rates, s_1, s_2, s_3 and s_4 do not appear in the macroscopic equations and are chosen according to stability optimization [51, 30] leading to $s_1 = 1.19, s_2 = 1.4, s_3 = 1.2, s_4 = 1.98$. The BGK operator is recovered if all the relaxation rates are the same.

Then the LBM algorithm used in this study will rely on equations (14), (16) and (19), imposing the physical parameters with (18) and (22). It is to be noticed that other forms of the collision operator are possible [36, 37, 52, 53, 54, 55, 56] but would give very similar results for the test-cases considered in this study. Moreover, a high order formulation of eq. (14) could be adopted to enhance stability issues [57] and to reach higher Mach numbers [58]. A detailed theoretical comparison of the different collision models and their impact on the physics can be found in [41] and [35]. Then, the massively used D3Q19 lattice with the MRT collision operator will serve as the reference LBM model in the following comparisons.

1.3. Algorithmic comparison

In order to summarize the theoretical backgrounds exposed in the previous sections, the basic steps of each algorithm are detailed in Table 2.

Step	LBM	VM
Initialization ($t^* = 0$)	Compute g_α from ρ and $\mathbf{u}$	Compute $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ from $\mathbf{u}$
RHS update	Collision from (19)-1	Stretching (4) and diffusion of $\boldsymbol{\omega}$ (5)
Advection	Streaming from (19)-2	Particles advection + remeshing (8)
Macro state update	Compute ρ and $\mathbf{u}$ from g_α and update (14)	Compute $\mathbf{u}$ from $\boldsymbol{\omega}$ (3)

Table 2: Comparison of the basic algorithmic steps for present LBM and VM methods.

In order to compare this two different kind of methods, some important features should be pointed out. The computed quantities of each method are basically different. Indeed, LBM computations compute the distribution functions and give direct access to density and velocity whereas VM methods compute the vorticity and the velocity. Then because of the incompressible nature of the VM, only velocity and vorticity will be compared in this study. The Mach number of the LBM method will always be chosen to a low value and the equilibrium will be computed with relation (14). Finally, it should be mentioned that the vorticity is not a native quantity in the LBM algorithm and must be reconstructed. Then when a time evolution of the vorticity will be needed in LBM computations, it will be reconstructed inside the algorithm with only second order in space to preserve the global order of computation.

Then, it should be highlighted that a fine computational cost comparison is delicate to handle in the present study. Indeed, these types of considerations are very dependent on the level of implementation of the algorithms (pedagogic, academic, industrial, optimized) and should be considered in a dedicated study. However, some basic features of the implementation can be pointed out for each method (they will be confirmed in section 2.1 with Table 3 giving indicative CPU-times for a well chosen test case). Concerning LBM, the classical implementation of the algorithm is spread out into a local collision step which represents the main computational cost and an advection step which is generally very fast due to the low stencil of the D3Q19 lattice. The counterpart of this efficiency is that the LBM timestep is generally limited by the grid step, meaning that it should be low for high resolution. Concerning VM, the computational time dedicated to the resolution of the advection/resmehing step is also rather fast due to the directional splitting (successive resolutions of 1D problems in each spacial direction), as well as the resolution of the Poisson equation and diffusion (by using the optimized FFTW library). On the other side, the resolution of the stretching equation and the evaluation of Δt_{adapt} represent a non negligible part or the total computational time within one time step. However, the use of such adaptive time step enables to significantly reduce the number of total iterations needed to complete simulations, which leads to a net reduction of total computational time compared to simulations based on classical CFL conditions.

2. Numerical dissipation

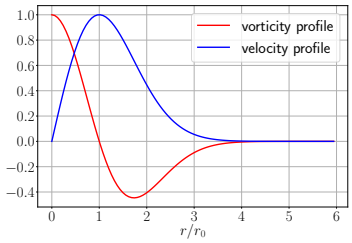
In this section some standard test-case computations will be performed to characterize the level of numerical dissipation induced by each method. To investigate this kind of numerical characteristics, the linear stability analysis (LSA) of the scheme could be a powerful and efficient theoretical tool. The LSA of the LBM has been the purpose of numerous studies [59, 31, 60]. The main conclusion of these studies relies on the low dissipation rate of the perturbative (or acoustic) mode due to the mesoscopic nature of the method and a level of numerical dissipation for the vorticity mode comparable to 3^{rd} order of macroscopic method such as finite-differences. Concerning remeshed Vortex methods, according to the authors knowledge, there is no LSA or complete numerical analysis of convergence for fractional step algorithms like the one used in this paper (cf Tab. 1) due to its heterogeneous aspect. However, it is important to mention the theoretical studies carried out in the pure Langrangian framework of the Vortex methods: Hald in 1979 [61] and then Beale and Majda in 1982 [62] proved the theoretical convergence of pure Lagrangian Vortex methods in the context of the 2D Euler equations (inviscid flows). Later, the time discretization was added in the convergence analysis of the 2D and 3D Euler equations by Anderson and Greengard in 1985 [63]. Besides the existence of a theoretical convergence analysis, a very interesting feature of Lagrangian Vortex methods relies on the fact that they conserve many inviscid flow invariants. For the 2D Euler equations, Vortex methods guarantee the conservation of 4 invariants, namely the total circulation, the linear and angular impulses as well as the kinetic energy, as proved in [64]. The conservation properties of Lagrangian Vortex methods ensure that they are naturally free of numerical dissipation which implies that, even for underresolved simulations, they provide correct qualitative solutions. Concerning the semi-Lagrangian aspect of the method and the particle remeshing, Cottet et. al [42] proved recently the consistency and linear stability of the advection/remeshing scheme (eqs. (8)) with remeshing kernels $\Lambda_{p,r}$ until $p = 8$.

A relative comparison of the present LBM and remeshed VM on well-known test cases will

therefore give some insight on the numerical dissipation induced by these non fully macroscopic methods.

2.1. Advection of a simple vortex

For this first test case, the simple and widely used Taylor vortex is investigated. Here, the convection of a viscous vortex is used to characterize the effects of each discretization strategies on the dissipation of a simple coherent structure. For this test case, a periodic $[L, L, L/4]$ domain is used and the velocity field is initialized by equations (23) where $r^2 = (x-x_0)^2+(y-y_0)^2$, $r_0 = L/10$ and



$$\begin{cases} u_x &= U_\infty - \dfrac{U_t}{r_0}(y-y_0)\exp\left[\dfrac{1}{2}\left(1-\dfrac{r^2}{r_0^2}\right)\right] \\ u_y &= \dfrac{U_t}{r_0}(x-x_0)\exp\left[\dfrac{1}{2}\left(1-\dfrac{r^2}{r_0^2}\right)\right] \\ u_z &= 0 \end{cases} \tag{23}$$

Figure 1: Velocity and vorticity profiles of the convected Taylor vortex.

$U_t = U_\infty/10$. The LBM pressure is initialized so as to ensure the isotropic condition and avoid some spurious oscillations. This latter point is particularly discussed in dedicated publications [65, 66]. The Taylor vortex has the particularity to be surrounded by a negative vorticity region between $r = r_0$ and $r = 4r_0$. Moreover, the velocity profile has a compact form and reaches a very small value for $r > 4r_0$ (Fig. 1). For this first comparison, all the numerical parameters such as grid size and time-step will be the same for both LBM and VM. Then in order to get rid of the peculiar normalization procedure of each method and for the sake of clarity, the vortex will be defined in physical units. Thus, the characteristic dimensions are taken to $U_\infty = 34$m/s and $L = 1.28$m. The Reynolds number based on r_0 is set to 100 in order for the diffusion term to be effective in the VM algorithm. Indeed, this test case is usually performed in the inviscid form to get rid of the viscous dissipation and directly compare the numerical dissipation. But for the present study, the numerical schemes are compared in the three presented test cases with all the features described in Table 2. From these parameters, the grid size is set to $\Delta x = L/N$ where N is the grid resolution and the time-step is chosen so as to enforce a CFL number based on the upstream velocity $\text{CFL} = U_\infty \frac{\Delta t}{\Delta x} = 1/\sqrt{3} \approx 0.057$ for both LBM and VM models.
The center of the vortex is initially positioned at the center of the 3D periodic $[L, L, L/4]$ domain and its convection is observed through a given number of domain crossings while the grid resolution $N \times N \times N/4$ in the whole domain is varying from $N = 16$ to 256. Concerning the present VM, since it is a semi-Lagrangian Vortex method, one recalls that at each time step of the algorithm the particles are redistributed on the background grid nodes and then convected in a Lagrangian way (cf eqs 8), thus implying that the total number of particles in the domain is always equal to the underlying grid resolution.

The velocity and vorticity signals recorded in the center point of the domain are plotted on Figures 2, 3 and 4 with respect to the normalized time $t^\star = tU_\infty/L$.

First of all, the convergence behavior is clearly different in VM and LBM. Indeed, from Figure

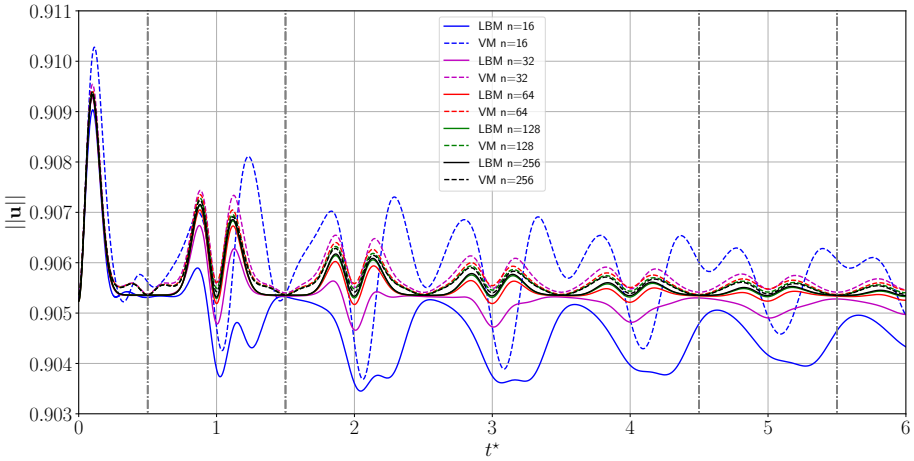


Figure 2: Time evolution of the velocity norm at the center of the computational domain.

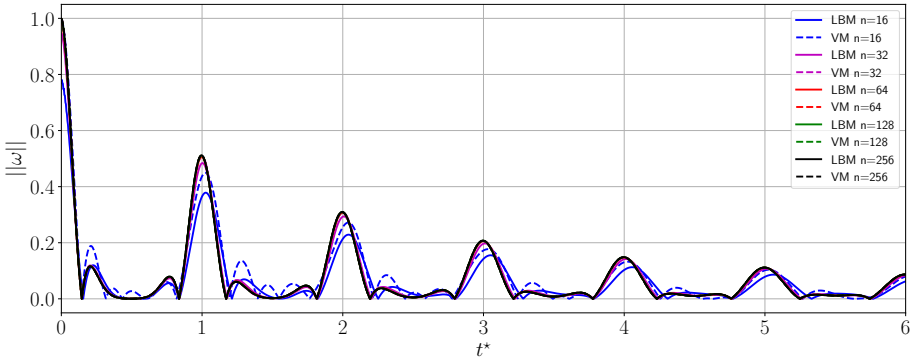


Figure 3: Time evolution of the vorticity norm at the center of the computational domain.

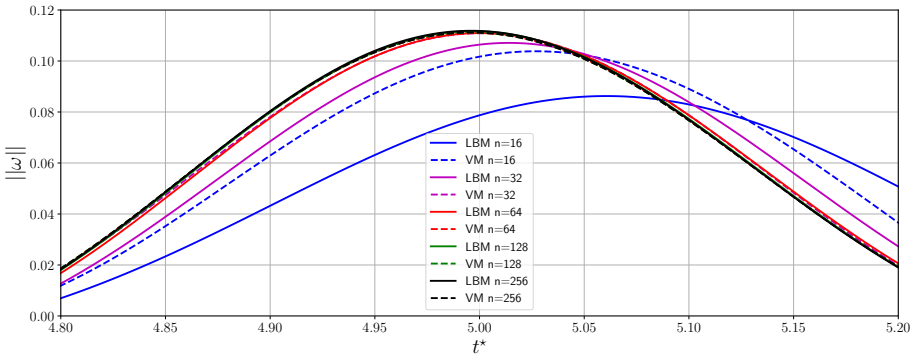


Figure 4: Time evolution of the vorticity norm at the center of the computational domain. Closer view of Figure 3

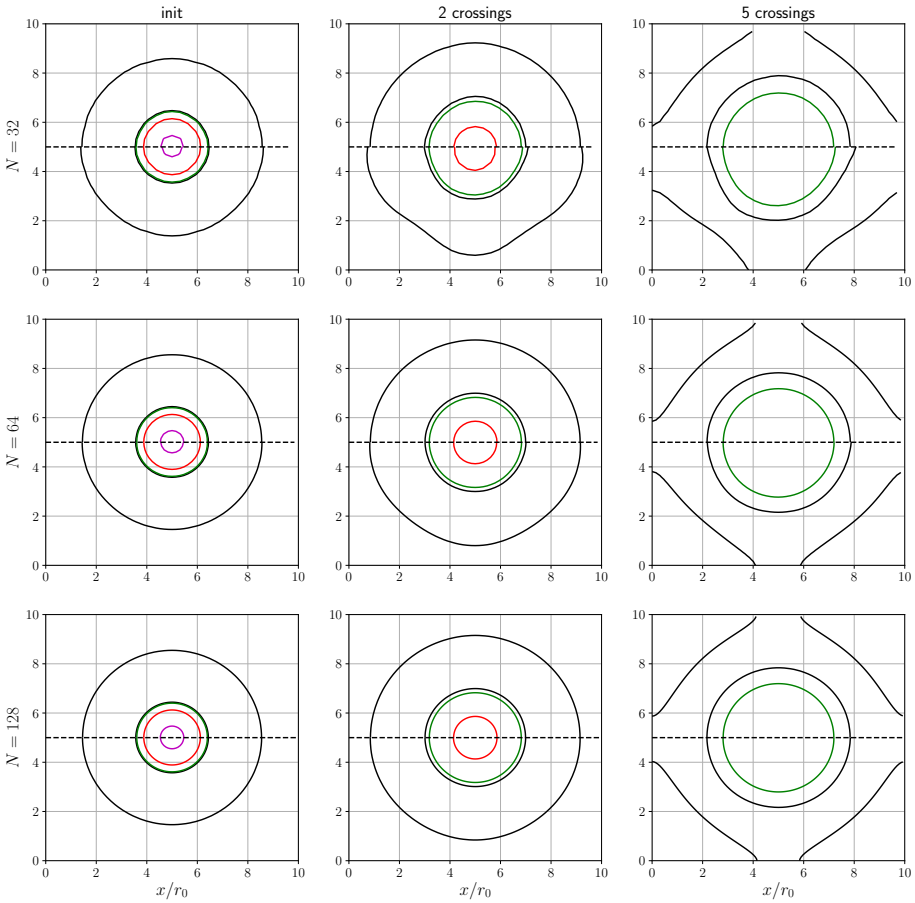


Figure 5: Vorticity isocontours at different resolutions (rows) and different times (columns) (initial time, 2nd domain-crossing and 5th domain-crossing). Each thumbnail shows the face-to-face results for VM (top part) and LBM (bottom part). The isocontours levels are set to −0.01 (black), 0.01 (green), 0.2 (red) and 0.8 (magenta).

2, one can see that the coarse resolution underestimates the LBM velocity whereas it is overestimated with VM. The LBM results exhibit a large dispersion for coarse resolution.

Then, it should be noticed that the LBM vorticity is a reconstructed quantity which explains that the initial vorticity for the lowest resolution is slightly lower than the theoretical one because of the second order reconstruction (see the solid blue curve at $t^* = 0$ on Fig. 3). Thus the vorticity level for the LBM results at the lowest resolution should be interpreted with this initial reconstruction error. Then the first global results of Figure 3 clearly show a similar behavior for LBM and VM results. The stronger difference is observed for the lowest resolution where the LBM results exhibit a higher numerical dispersion of the convected vortex, which is not at the expected position. The close-up view of Figure 4 highlights that the VM vortex has less dispersion for the overall resolutions.

The comparisons of the vortex shapes in Figure 5 confirm these observations by highlighting a

strong deformation of the LBM vortex for the lowest resolution after 2 domain-crossings (second column of Fig. 5). Then the vortex dissipates and the limit of the negative vorticity ring reaches the end of the domain (last column of Fig. 5). We note that the vortex shapes obtained for both methods with $N = 256$ do not show qualitative change compared to the case $N = 128$ and are consequently not represented in Figure 5.

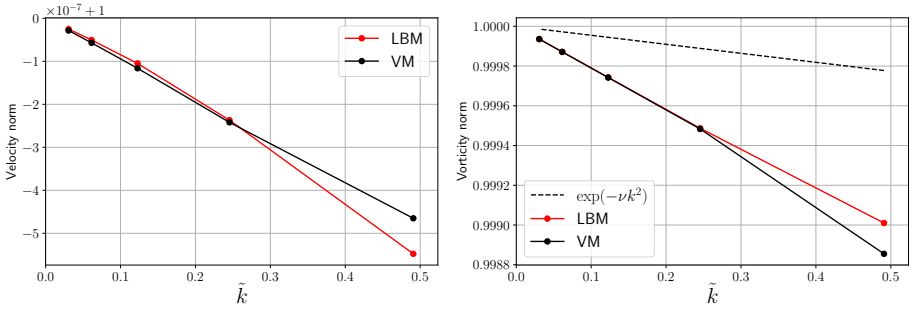


Figure 6: Numerical dissipation of the velocity (left) and vorticity (right) norm, with respect to the numerical wavenumber. The dashed line represents the theoretical dissipation rate of the vorticity mode [67] $-\nu k^2$.

To take into account the overall results for both velocity and vorticity, the evolution of the numerical dissipation with the numerical wavenumber is shown in Figure 6. This dissipation rate G is estimated by averaging the norm of the desired quantity $\mathbf{q}$ during the period $T = L/U_\infty$ and by computing the ratio between the last 2 periods and the first 2 periods:

$$G_q = \left(\frac{\int_{9T/2}^{11T/2} \|\mathbf{q}(t)\| dt}{\int_{T/2}^{3T/2} \|\mathbf{q}(t)\| dt} \right)^{1/4T}, \tag{24}$$

where the intervals of time integration $[\frac{T}{2}, \frac{3T}{2}]$ and $[\frac{9T}{2}, \frac{11T}{2}]$ are graphically represented by vertical dotted lines in Figure 2.
Figure 6 displays the evolution of G_u and G_ω (velocity and vorticity dissipation respectively) against the numerical wavenumber $\tilde{k}$, which is computed by assuming from Figure 1 that the vortex have a global wavelength of $8r_0$. As the grid resolutions vary among $N = 16, 32, 64, 128, 256$ they correspond to numerical wavenumbers respectively equal to:

$$\tilde{k} = k\Delta x = \frac{2\pi}{8r_0}\Delta x = \frac{\pi}{2\frac{N}{5}} = 0.491, 0.245, 0.123, 0.061, 0.031.$$

The observed dissipation have similar level for both methods with a slight trend for VM to better propagate the low resolved vortex due to its low dispersion brought by the Lagrangian advection step. For high resolved vortex, the numerical dissipation produced by both methods tends to zero: the total dissipation of the solution tends to the physical viscous dissipation, as can be seen on the right hand side of Figure 6 where a comparison of the numerical vorticity dissipation with respect to the theoretical dissipation rate of the vorticity mode $-\nu k^2$ is given.

Table 3 gives indicative computational times, obtained respectively with LBM and VM for the convected eddy test case. For the coarsest resolutions, namely $N = 16, 32, 64$, the data reported

in the table correspond to the mean of the CPU-times obtained over 10 simulations. Both algorithms have been compiled with the standard *gfortran* compiler with similar options. All the simulations presented in this table have been performed on the same hardware with only 1 core (Xeon E7-8860 v4 2.2-3.3 GHz) for all the resolutions. The choice of a unique core allows to carry out a comparison detached from any influence of the parallelization level of the two associated codes (which, one recalls, have been developed independently). In terms of absolute comparison between the LBM and VM CPU-times (in seconds), one can conclude from Table 3 that the present implementation of LBM is faster than the present implementation of VM by a factor of 2 approximately for the finest resolutions. This can be explained by the two respective algorithms themselves, where the "RHS update" and "Advection" steps contain more substeps in VM than in LBM (see Table 2). However, for the reasons explained in section 1.3, these data are to handle very cautiously and might be essentially considered as indicative.

Resol. N	CPU-time **LBM**		CPU-time **VM**	
	absolute (sec)	normalized	absolute (sec)	normalized
16	0.276	1	1.80	1
32	3.12	11 (×11)	10.72	6 (×6)
64	53.57	194 (×17.5)	116.08 (∼ 2 min)	65 (×11)
128	1176.26 (∼ 20 min)	4262 (×22)	1921.40 (∼ 32 min)	1070 (×16.5)
256	19112.48 (∼ 5h 20 min)	69248 (×16)	36598.16 (∼ 10h 10 min)	20378 (×19)

Table 3: Comparison of the CPU times required for LBM and VM approaches on the convected eddy test case for the time range $t^* \in [0, 1]$. All simulations are performed on a single processor and the CPU times are given in absolute values (in seconds) and normalized according to the CPU-time associated to the $N = 16$ simulation.

2.2. Taylor-Green Vortex

In order to study the effect of numerical dissipation on a fully 3D turbulent and highly documented test-case, the decaying Taylor-Green vortex (TGV) is now considered. It is a fundamental benchmark used as prototype for vortex stretching and production of small-scale eddies which therefore allows to study the dynamics of transition to turbulence. This test-case has been widely used to study the dissipation errors of numerical schemes or the impact of collision operators in LBM [68].

For this test-case, the simulations are performed on a 2π-periodic cubic domain Ω defined as $0 \leq x, y, z \leq 2\pi$, with a Reynolds number equal to $Re = 1600$. The initialization of the Taylor-Green vortex is done by setting velocity and pressure variables as follows:

$$\begin{cases} p &= p_\infty + \dfrac{\rho_\infty U_\infty^2}{16}[\cos(2z) + 2][\cos(2x) + \cos(2y)] \\ u_x &= U_\infty \sin(x)\cos(y)\cos(z) \\ u_y &= -U_\infty \cos(x)\sin(y)\cos(z) \\ u_z &= 0 \end{cases} \tag{25}$$

For the LBM simulation, a peculiar attention has to be made for the initialization of the distribution functions when the initial velocity gradients are not negligible [69, 70].

In order to work with non-dimensional quantities, all the displayed quantities are normalized by an arbitrary length scale L and velocity scale U_∞. From this the non dimensional time is defined by $t^\star = tU_\infty/L$ and the Reynolds number by $Re = U_\infty L/\nu$.

2.2.1. Time step definition

For this benchmark, the time steps of each method are set differently by taking advantage of each algorithm. For VM, the adaptive time step is used whereas the LBM time step is fixed. Thus, in contrast with the first test-case where both methods were set with the same CFL, this benchmark will impose different CFL and will highlight the methods for a set of parameter naturally used in the literature.

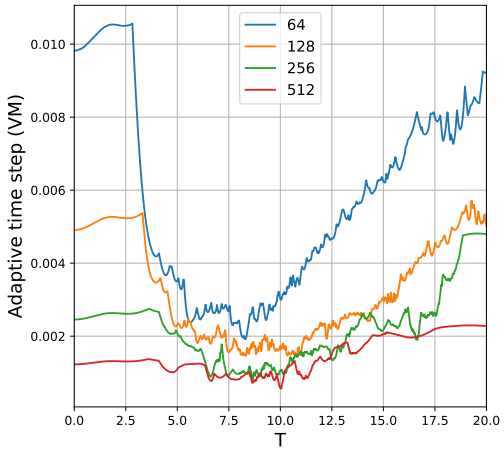
All the simulations realized for this test-case are performed in a $t^\star \in [0, 20]$ time range. Different grid resolutions, denoted N^3, will be studied, namely $64^3, 128^3, 256^3$ and 512^3. For LBM computations, the time step is constant and defined by relation (18) (assuming that $\Delta x = 2\pi/N$) and then multiplied by U_∞/L to get dimensionless values. This definition gives the following dimensionless time steps, with respect to the different resolutions : $\Delta t_{64} \approx 4.8 \cdot 10^{-3}, \Delta t_{128} \approx 2.4 \cdot 10^{-3}, \Delta t_{256} \approx 1.2 \cdot 10^{-3}, \Delta t_{512} \approx 6.0 \cdot 10^{-4}$.

Regarding the VM simulations, the time step is adaptive, based on relation (7), namely $\Delta t^n_{\text{LCFL}} \leq \dfrac{\text{LCFL}}{\|\nabla \mathbf{u}\|_\infty}$ with LCFL < 1. More precisely, in these TGV computations, the global adaptive time step chosen all along VM simulations is defined by :

$$\Delta t^n_{\text{adapt}} = \min(\Delta t^n_{\text{CFL}}, \Delta t^n_{\text{LCFL}}) = \min\left(\frac{\text{CFL} \cdot \Delta x}{\|\mathbf{u}\|^n_\infty}, \frac{\text{LCFL}}{\|\nabla \mathbf{u}\|^n_\infty}\right) \tag{26}$$

where n denotes the current iteration and where Δt^n_{CFL} refers to the classical Eulerian stability condition based on the grid size Δx. In the following TGV simulations, the values of CFL and LCFL numbers will be respectively set to CFL $= 0.1$ and LCFL $= 1/32$. The left handside of Figure 7 shows the temporal evolution this adaptive time step along VM simulations depending on the different grid resolutions under study. For each resolution, one can clearly distinguish between $t^\star = 2.5$ and $t^\star = 5$ the switch from the CFL stability condition to the LCFL one. The evolution of $\Delta t^n_{\text{adapt}}$ follows the flow dynamics and shows minimum values in the time range corresponding to the peak of energy dissipation, at $t^\star \approx 9$ (see next subsection for a clear observation of this peak).

For purpose of comparison, the table located on the right hand-side of Figure 7 reports the number of time steps required respectively in LBM and VM computations to perform a TGV simulation until $t^\star = 20$. In LBM, since Δt is governed by a CFL condition, the number of total time steps is successively multiplied by 2 when increasing the grid resolution. In VM, it can be noticed that, when Δt_{adapt} is driven by the LCFL condition, its value at a given resolution is not divided by two with respect to the previous coarser resolution (as it is the case when the CFL condition governs). This can be explained by the fact that the LCFL condition is based on velocity gradients, not on grid step. Thus, the finer is the grid resolution, the better is the evaluation of these gradients and higher is the gain in terms of total time steps.



Grid resolution	# time steps LBM	# time steps VM
64^3	4153	4411
128^3	8306	7395
256^3	16612	10849
512^3	33224	15324

Figure 7: (Left) Temporal evolution of the adaptive time step along VM simulations depending on grid resolution. (Right) Comparison of number of time steps required to achieve a TGV simulation until $t^\star = 20$ between LBM and VM.

2.2.2. *Time evolution of kinetic energy*

This subsection is dedicated to the study of the temporal evolution of the kinetic energy $E = \frac{1}{2}\int_\Omega \|\mathbf{u}\|^2 \, d\Omega$, the kinetic energy dissipation rate $\varepsilon = -\frac{dE}{dt}$ and the enstrophy $Z = \frac{1}{2}\int_\Omega \|\omega\|^2 \, d\Omega$. Note that the kinetic energy decays proportionally to enstrophy and ε and Z are linked by the following relation : $\varepsilon = -\frac{dE}{dt} = 2\nu Z$, where ν denotes the kinematic viscosity of the fluid.

For VM computations, the energy dissipation rate ε is reconstructed *a posteriori* using a second order time integration scheme ($\varepsilon = -\frac{(E(t+m\Delta t_{\text{adapt}})-E(t-m\Delta t_{\text{adapt}})}{2m\Delta t_{\text{adapt}}}$), where m is chosen to be equal to 3 in this study to prevent from spurious oscillations. For LBM calculations, the enstrophy Z is computed from the vorticity for which a second order reconstruction is used.

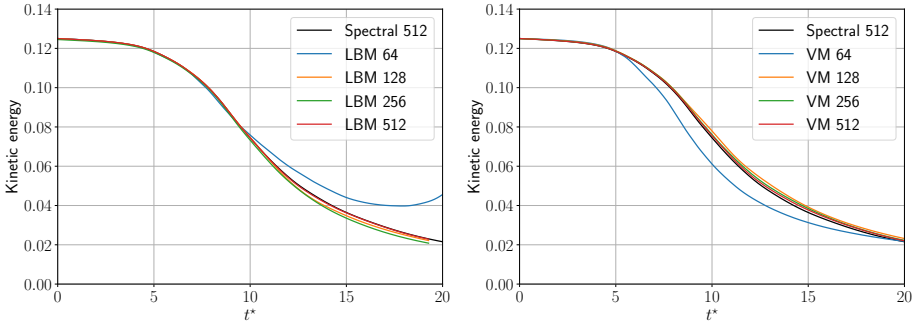


Figure 8: Kinetic energy decay for Taylor-Green vortex benchmark between $t^\star = 0$ and $t^\star = 20$. (Left) LBM, (Right) VM.

Figures 8, 9, 10 respectively show the temporal evolution of E, ε and Z for LBM (left) and VM (right) for different grid resolutions. The results are compared to the spectral solution at 512^3, taken as the reference solution [71]. As can be seen, both LBM and VM converge towards the

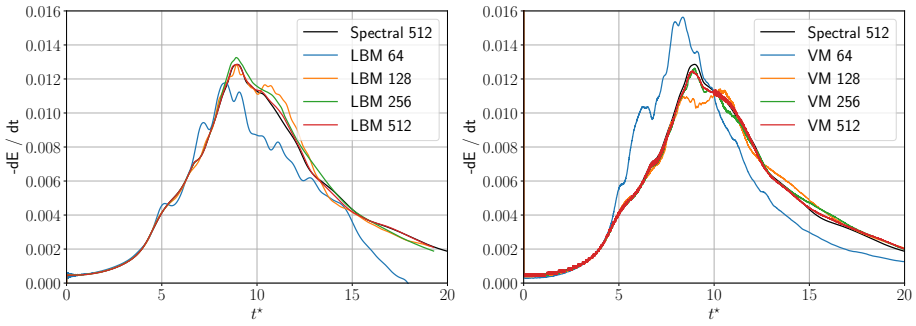


Figure 9: Time evolution of kinetic energy dissipation rate $\varepsilon = -\frac{dE}{dt}$ for Taylor-Green vortex benchmark between $t^\star = 0$ and $t^\star = 20$. (Left) LBM, (Right) VM.

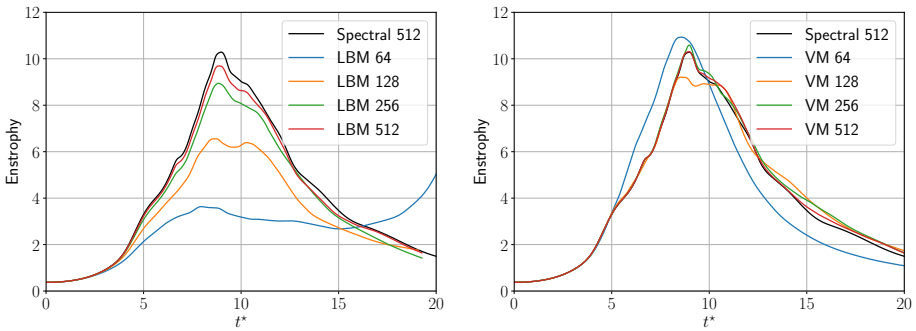


Figure 10: Time evolution of enstrophy $Z = \frac{1}{2}\int_\Omega \|\omega\|^2 \, d\Omega$ for Taylor-Green vortex benchmark between $t^\star = 0$ and $t^\star = 20$. (Left) LBM, (Right) VM.

reference solution, for each physical quantity. In particular, both approaches correctly recover the peak of energy dissipation as well as the peak of enstrophy reached at $t^\star \approx 9$ (Figs. 9, 10). However, this convergence behaves in a different way : with LBM, the simulation results converge from "bottom to top" towards the reference, which illustrates the numerical dissipation brought by the Lattice Boltzmann method. From the 512^3 grid resolution, the numerical diffusion of E and ε becomes negligible (cf Figs. 8, 9) and the LBM solution reaches the spectral solution with an error of 10^{-6}, as can be seen on the grid convergence curves, Figure 11, which represent the L^2-norm errors of the physical quantities based on the spectral solution on the time range $t^\star \in [0, 20]$. As regards enstrophy, which is a second-order-reconstructed quantity in LBM, the convergence order is, as expected, close to 2 (cf Figs 10 and 11). However the final enstrophy error reached with the finest mesh remains in the order of 10^{-3}. For the coarse resolution, the LBM solution is over-dissipated by the grid and the chosen collision model (MRT), despite its ability to get stable simulation on coarse grids, still overestimates molecular viscosity when grid step is large. A detailed comparison of LBM collision models on the Taylor-Green vortex benchmark could be found in [68].

Concerning VM, one can observe on Figures 8, 9, 10 a convergence of the solutions from "top to bottom" and, more specifically, Figures 9 and 10 reveal that VM tend to slightly over-estimate

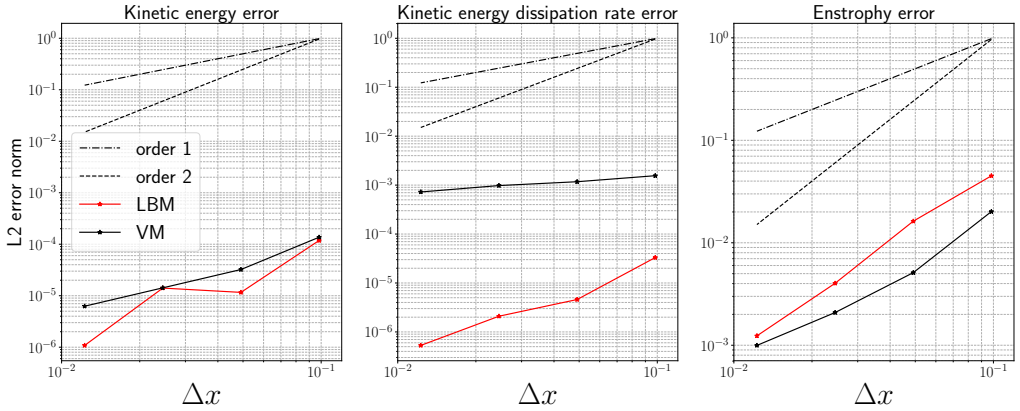


Figure 11: Grid-convergence study for Taylor-Green vortex benchmark between $t^\star = 0$ and $t^\star = 20$. The L^2-norm errors of kinetic energy (left), kinetic energy dissipation rate (center) and enstrophy (right) with respect to the spectral solution are plotted against the grid step.

the enstrophy for under-resolved simulations (this phenomenon is shown and proved in [72] and can be explained by the antidiffusion mechanisms embedded in the error resulting from the reconstruction of particle velocity in Vortex methods). As pointed out in the previous section dealing with the convected eddy, VM is a low dissipative method and even for highly under-resolved simulations (i.e 64^3) it manages to provide a rather correct time-evolution of energy decay (Fig. 8) and enstrophy (Fig. 10). At the 512^3 resolution, VM reaches the spectral solutions with the same ranges of error than LBM, except for the ε quantity (cf center of Fig. 11), which is only reconstructed at the first order in the present remeshed Vortex method.
Nevertheless, one can globally observe that the VM convergence order is lower than the one of LBM, regardless of the quantity studied (cf Figs 11). In particular, the enstrophy convergence order achieved with VM (approx 1.43) is slightly lower than LBM (approx 1.76) whereas the vorticity ω, involved in the enstrophy, represents the primary quantity solved by Vortex methods. On one side, as mentioned in section 1.1, the heterogeneous construction of the fractional VM algorithm offers a flexible framework to design arbitrarily a semi-Lagrangian numerical method, based on previous theoretical works, but on the other hand this heterogeneity makes the *a priori* determination of the global order of the method delicate. Numerically, the global order of the VM grid-convergence turns out to be a bit less than 2, despite the use of 4^{th} order or spectral schemes (cf Tab. 2). Some elements like the linear interpolation of the particle velocity in the Lagrangian transport or the first order evaluation of the integral quantities E and Z can explain this behavior.

2.2.3. Spectral analysis

Figure 12 shows turbulent kinetic energy spectra obtained with LBM (left) and VM (right). These spectra are plotted at $t^\star = 12$ in order to study the energy cascade throughout the different spatial scales at a time when the turbulent flow is developed. Note that the "wavenumber" label on the x-axis refers to the mean over the unit sphere of all the wavenumbers (k_x, k_y, k_z), that is to say to the quantity $|\mathbf{k}| = \sqrt{(k_x^2 + k_y^2 + k_z^2)}$. For both methods, a cutoff has been applied to the spectra below the cutoff wavenumber corresponding to the smallest resolved scale (vertical dotted lines).

One can notice that for the coarsest resolution 64^3, LBM slightly overestimates the kinetic energy in the resolved scales, while the opposite behavior is observed for VM. This confirms the results depicted in Figure 8.

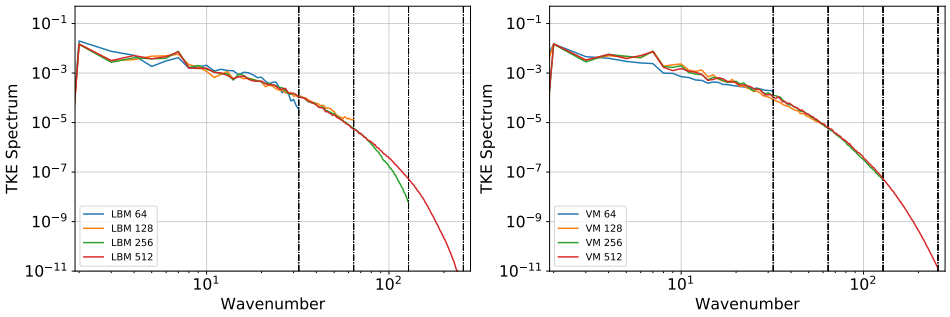


Figure 12: Turbulent kinetic energy spectrum for Taylor-Green vortex benchmark at $t^\star = 12$. (Left) LBM. (Right) VM. Vertical doted lines refer to the cutoff wavenumbers associated to the smallest resolved scales.

If we now consider the spectra at resolution 256^3 and 512^3, the energy cascade in the inertial range seems converged for both method. In the dissipation range, it can be seen that LBM tends to dissipate more energy when going towards the cutoff wavenumber, which confirms the remark formulated before, stating that VM is globally less dissipative than LBM.

2.2.4. Flow structures

This last subsection relies on a qualitative comparison between the vortical structures obtained in the LBM and VM solutions. Figures 13 and 14 respectively show the vorticity and velocity norm of the flow field in the symmetric and periodic half-plane $x = \pi$, at $t^\star = 9$ with a 512^3 resolution. As can be seen, at this resolution, the flow structures are significantly comparable between LBM and VM.

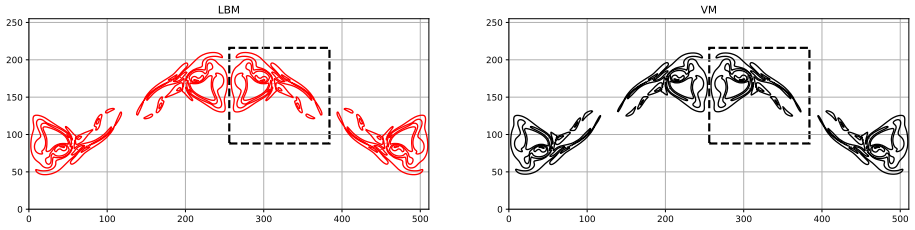


Figure 13: Isocontours of vorticity norm at time $t^\star = 9$ and at levels $1, 5, 10, 20, 30$ obtained with a 512^3 resolution on the periodic and symmetric half-plane $x = \pi$. (Left) LBM. (Right) VM.

An enlargement of the main vortical pattern (delimited by the dotted square in Figures 13 and 14) is proposed in Figure 15, and despite little discrepancies about the vorticity (left) and velocity (right) norms, both methods manage to correctly take into account the small scales of the flow. In particular it is interesting to look at the vorticity isocontours into more details since they bring most of the crucial information of the Taylor-Green flow. Indeed, at $t^* = 9$ the energy dissipation reaches its maximum (cf Fig. 9) and the coherent structures of the flow start to be destroyed,

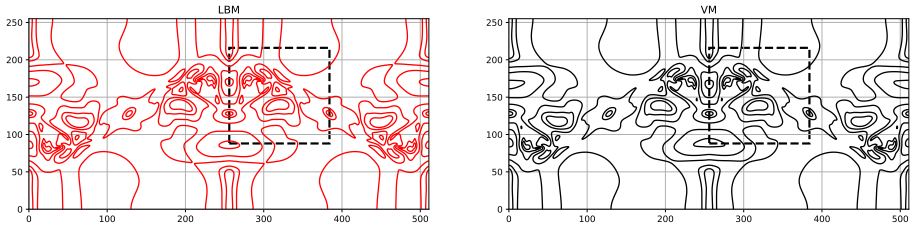


Figure 14: Isocontours of velocity norm at time $t^\star = 9$ and at levels $0.1, 0.2, 0.4, 0.6, 0.8, 1$ obtained with a 512^3 resolution on the periodic and symmetric half-plane $x = \pi$. (Left) LBM. (Right) VM.

leading to the development of the turbulent flow. On Figure 13, the phenomenon of rupture of the main vortical structures is clearly observable and very similar for LBM and VM. More precisely, on the left hand-side of Figure 15 one can see that both method manage to recover the regions of the flow corresponding to vortex tubes (thin elongated structures on the right) which are associated to strong vorticity and small scales, as well as the sheet-like structure (large "eye-like" structure on the left) that is associated to strong energy dissipation. The few discrepancies existing between the two solutions mainly rely on a little spacial shifting, rather than a capacity of catching the tears of the small scale vortex tubes or the contours of the sheet-like structure.

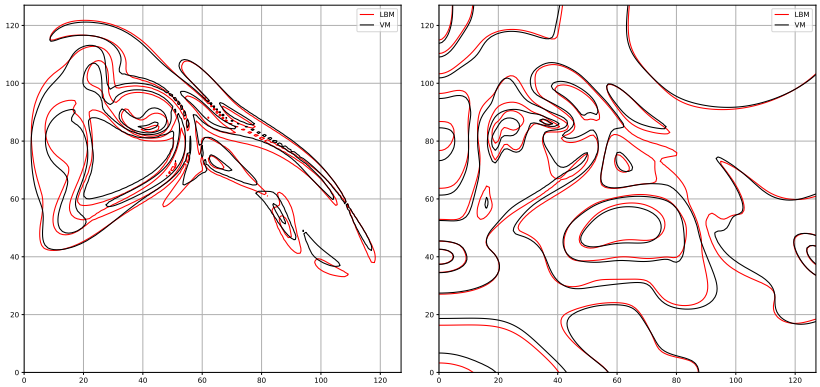


Figure 15: Isocontours at time $t^\star = 9$ with a 512^3 resolution on the plane $x = \pi$. (Left) Superimposition of isocontours of vorticity norm at levels $1, 5, 10, 20, 30$ obtained with LBM (red) and VM (black). (Right) Superimposition of isocontours of velocity norm at levels $0.1, 0.2, 0.4, 0.6, 0.8, 1$ obtained with LBM (red) and VM (black).

3. Flow past a solid cube

The last test case proposed in this paper is the flow past a 3D cube fully immersed in the fluid, for different regimes. It aims at providing to the present list of benchmarks a significant test where the treatment of no-slip boundary conditions is investigated and compared for both LBM and VM methods in the proper methodological context where they are usually applied. First of all, it has to be noted that, to the authors knowledge, the flow past a 3D cube test-case has been subject to rather few number of experimental or numerical studies compared to the case of flow past a surface-mounted cube or flow past a sphere. It therefore represents an interesting non-usual

benchmark for the present study, characterized in particular by the sensitivity of the numerical results to the sharp corners of such bluff body. Two different flow regimes will be handled in this study to directly compare the effects of wall treatment for both methods. The first chosen regime is at $Re = 290$, which corresponds to an unsteady and planar symmetric flow [73]. The second regime of study is at $Re = 570$, where no symmetry is observed in the wake and for which the flow becomes fully unsteady [73, 74].

3.1. Boundary conditions

3.1.1. Vortex Method framework

The approach used in the present remeshed VM method to handle the no-slip boundary condition at the solid cube interface is the Brinkman penalization method. The latter, firstly proposed by [75] and further developed by [76, 77], is part of the immersed boundary methods. It consists in extending the fluid velocity inside the body and then to penalize it through an extra term in the Navier-Stokes equations. This penalization term, added as a forcing term, models the no-slip boundary conditions and is driven by a penalization factor, which can be related to the effective porosity of the body. Such approach is therefore fully compatible with the use of FFT-based evaluations for the computation of the velocity field (see eq. (9)), for any type of body geometry. The Brinkman penalization method has been widely used in the context of semi-Lagrangian Vortex methods dealing with complex body geometries, because of its efficiency and simplicity [16, 18, 78, 79, 80]. For the computation of the drag and lift forces, VM simulations use the so called momentum change approach based on [81] which gives the force acting on the wall surface. Some other approaches may also be used to compute the aerodynamic forces in the context of vortex methods with Brinkman penalization [79], where a Poisson equation is solved to evaluate the pressure field from the velocity. Concerning the output boundary conditions, the present VM method uses periodic boundary conditions. In order to prescribe the desired uniform flow at the inlet as well as proper outlet conditions, an absorption region is added at the outlet (cf the following subsection 3.2). Then a correction of the velocity field has to be performed in order to recover the desired flux at the inlet and to ensure a non-zero circulation in the computational domain. We refer the reader to [78] for further details about these aspects.

3.1.2. Lattice Boltzmann framework

For the lattice Boltzmann simulations, the inflow boundary conditions are set through the equilibrium distribution function by imposing a uniform streamwise velocity at the inlet and the outflow boundary conditions imposes a conservation of the non-equilibrium distribution functions at the outlet [82]. A sponge zone is used to increase the viscosity at the outlet and damp the outgoing structures. The shape of the sponge zone is the same as for VM and is described in the next subsection. For the computation of forces, the pressure is directly integrated on each elementary surface and projected in each direction to get the normalized coefficients of Tables 5 and 6. The no-slip condition is imposed with the non-equilibrium bounce-back condition which corrects the wall output non-equilibrium distributions with their symmetric counterpart from the wall normal direction [83, 84]. The treatment of no-slip boundary condition with bounce-back is not the only possibility [85] but could be considered as a standard and efficient way to investigate no-slip conditions on Cartesian geometries.

3.2. Numerical setup

The study of the flow past a 3D cube is performed for both methods on a uniform Cartesian grid by imposing a number of n grid points along the cube length D. The size of the computational

domain is defined in terms of the cube length and set to $[-L_u,\ L_d]\times[-H/2,\ H/2]\times[-H/2,\ H/2]$. The associated blockage ratio, BR, is defined as the ratio of the cube length and the domain cross-section, i.e. D/H^2. The center of the cube is located at the origin $(x, y, z) = (0, 0, 0)$ of the domain. The reference values U_∞ and D are set to unity such that the grid step is defined by $h = D/n$. The different grid parameters considered in the cube simulations are summarized in Table 4, depending on the Reynolds number, and a schematic representation of the computation domain is given in Figure 16.

	n	N_x	N_y	N_z	h	Domain	BR
$Re = 290$							
	25	320	128	128	0.04	$[-2,\ 10.8]\times[-2.56,\ 2.56]\times[-2.56,\ 2.56]$	3.81%
	50	640	256	256	0.02	$[-2,\ 10.8]\times[-2.56,\ 2.56]\times[-2.56,\ 2.56]$	3.81%
$Re = 570$							
	80	1024	512	512	0.0125	$[-2,\ 10.8]\times[-3.2,\ 3.2]\times[-3.2,\ 3.2]$	2.44%

Table 4: Grid parameters used for the cube simulation at $Re = 290$ and $Re = 570$ for both LBM and VM.

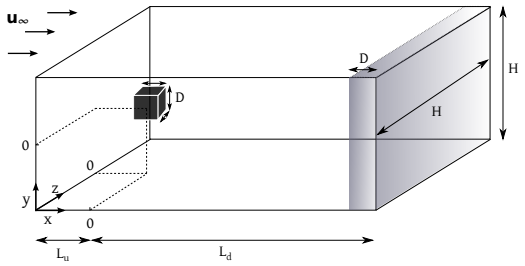


Figure 16: Computational domain, including the outlet sponge zone, for flow past a 3D cube.

For both LBM and VM methods, a sponge zone is added at the outlet of the domain $\mathcal{D}$ to prescribe the absorbing boundary condition. The absorption is performed according to the following one-dimensional smoothing function f defined in the flow direction (i.e. x-direction):

$$f(x) = \begin{cases} 1 & \text{if } x < x_b \\ \dfrac{\tanh(\alpha(x - x_c)) - \tanh(\alpha(x_e - x_c))}{\tanh(\alpha(x_b - x_c)) - \tanh(\alpha(x_e - x_c))} & \text{if } x_b \le x \le x_e \\ 0 & \text{if } x > x_e \end{cases} \tag{27}$$

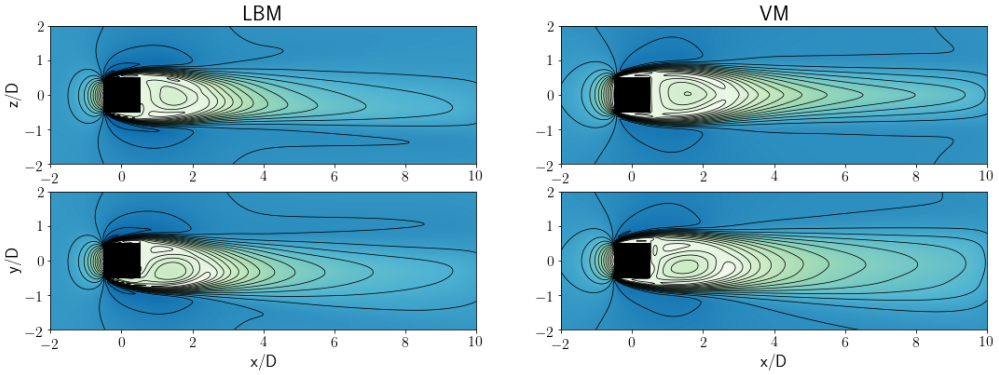
where x_b, x_c and x_e respectively refer to the beginning, the center and the end of the absorption band at the outlet. The parameter α allows to adjust the steepness of the absorption function f. In both LBM and VM approaches, one sets $\alpha = 10$, $x_b = L_d - D$ and $x_e = L_d$ (width equal to $1D$), which accounts for less than 8% of the total size, and thus of the total computational cost. A uniform velocity field $\mathbf{u}_\infty = (u_{x\infty}, u_{y\infty}, u_{z\infty}) = (U_\infty, 0, 0) = (1, 0, 0)$ is set at the inlet of the domain. In order to trigger the instability in a similar way for both methods, a perturbation is applied during the simulation between $t^\star = 3$ and $t^\star = 4$ on the y component of the velocity, defined by $u_{y\infty} = 0.1\ \sin(\pi(T - 3))$. For LBM, the time-step value is defined by (18) imposing a CFL number based on U_∞, equal to 0.1, and for VM, the time step is adaptive, according to relation (26), taking CFL= 0.5 and LCFL= 0.125.

Two Reynolds numbers are investigated here, $Re = 290$ and $Re = 570$. One notices that the aim of this section is not to carry out an exhaustive study of the physics related to this type of flows, but rather to compare the behavior of two numerical methods based on the same geometrical setup with boundary conditions usually applied in their proper computational context.

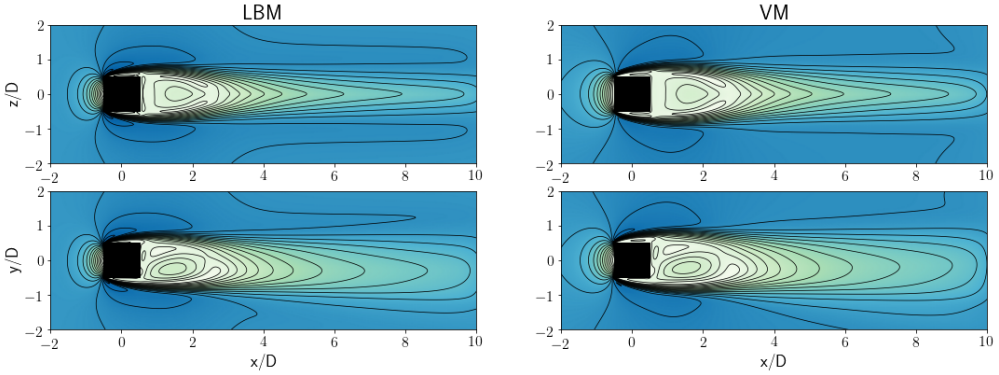
3.3. Results and discussion

Let us first investigate the results for the flow around a cube at $Re = 290$. As reported in Table 4, at this regime the simulations are performed for both methods at 2 grid levels, namely with $n = 25$ and $n = 50$. The simulations are realized on a total characteristic time of $t^* = tU_\infty/D = 150$. The mean quantities are averaged during the period from $t^* = 50$ to the end of the simulation. Figure 17 shows the norm of the mean velocity components. The solutions at the two grid levels are represented in the two orthogonal planes in the streamwise direction of the flow, namely the XZ and XY planes. These results on mean flows are complemented by Figure 18, giving the associated streamwise velocity profiles along x and y direction. The first observation that one can make from these two figures, is that both LBM and VM methods reach with $n = 50$ the planar symmetric flow in XZ plane, as expected and reported in literature [73, 74]; indeed, on Figure 17b the mean flows in XZ plane clearly show a symmetry with respect to the centerline of the wake and the non-planar symmetric wakes in XY plane exhibit comparable isocontours and more specifically a qualitatively similar ex-centred recirculation zone downstream. On Figure 18-right, the z-profiles of both methods for $n = 50$ (solid lines) are also distinctly symmetric. What is also interesting to notice, is the way the two methods converge to the planar symmetric flow state: as also highlighted in the two previous test cases of this paper, VM achieves better results at coarser grid resolution than LBM which fails at $n = 25$ in computing a planar symmetric flow for the same grid configuration (see XZ planes on Fig. 17a and the dotted curves on Fig. 18).

If we now focus more precisely on the mean velocity profiles in Figures 18, one can observe a velocity difference on Figure 18-left where the VM inlet velocity appears to be slightly lower than the expected far-field velocity, correctly restored by LBM. As a consequence, the levels of Figure 17 are noticeably different in the inlet region for the two methods. These discrepancies may be explained by the velocity correction performed in VM in order to account for non-periodic flow in the domain. This correction prescribes the uniform inlet flow rate with an error compared to the theoretical one based on the desired inlet velocity field ($\mathbf{u}_\infty = (1, 0, 0)$). However, despite these differences close to the domain walls, it is interesting to see that the solution close to the solid boundaries is comparable for LBM (with bounce-back conditions) and VM (with Brinkman penalization method), which highlights the ability of both methods to treat correctly no-slip boundary conditions. Among the differences in the obstacle region, one can first remark that VM results exhibit a slightly thicker recirculation zone induced by a larger detached flow region near the wall. This is more visible on Figure 18-right where the VM velocity profile is larger than LBM in the z-direction. Moreover, one can concentrate on the $x/D = [-0.5, 0.5]$ and $z/D = [-0.5, 0.5]$ regions in Figure 18, corresponding to the solid cube. At the extremities of these regions, that is to say at the cube surface, the mean velocity profiles of LBM indicate a 0 value which corresponds to the direct prescription of no-slip boundary conditions in the bounce-back approach. On the other hand, the profiles obtained with VM indicate that the velocity does not completely vanish inside the solid cube and at its surface. These non-zero values correspond to the Brinkman penalization model adopted in the present VM method, which consists, at each time step, in allowing the flow to enter the solid obstacle and then to penalize it by considering it as an extremely low permeable region according to a penalization coefficient, denoted λ. The



(a) $n = 25$



(b) $n = 50$

Figure 17: Flow past a cube at $Re = 290$. Norm of the mean velocity components ($\sqrt{\overline{u_x}^2 + \overline{u_y}^2 + \overline{u_y}^2}$) in XZ (top) and XY (bottom) planes for LBM (left) and VM (right). Levels correspond to 20 equispaced values between 0 and 1.4 included.

λ-convergence of the Brinkman penalization to real no-slip boundary conditions ($\mathbf{u} = 0$), is of order 1 [19]. The ability of both methods to correctly handle the boundary conditions, and thus the flow physics, is also confirmed by Figures 19 where are depicted the norm of the mean velocity component for flow past a cube at $Re = 570$. As reported in literature [74], the flow at such regime is unsteady and the wake does not show any symmetry, which is the result observed on Figures 19 and 20.

To complement the flow analysis, the two methods are compared in terms of instantaneous vorticity fields in Figure 21. This figure shows the close-up view of the ω_z isocontours for three different resolutions at $Re = 570$ in the XY plane near the solid walls, at $t^* = 120$. The interest of such view is to focus the comparison on the boundary layer region where the flow is laminar, rather than the wake region which is turbulent and whose analysis at a given time t^* strongly depends on the perturbation trigger. One can observe on this figure that the boundary layer thickness as well as the region of detachment points, located around downstream corners, are very similar between LBM and VM. The main discrepancy occurs at the upstream cube corners, where the different boundary conditions adopted by the two methods induce different levels

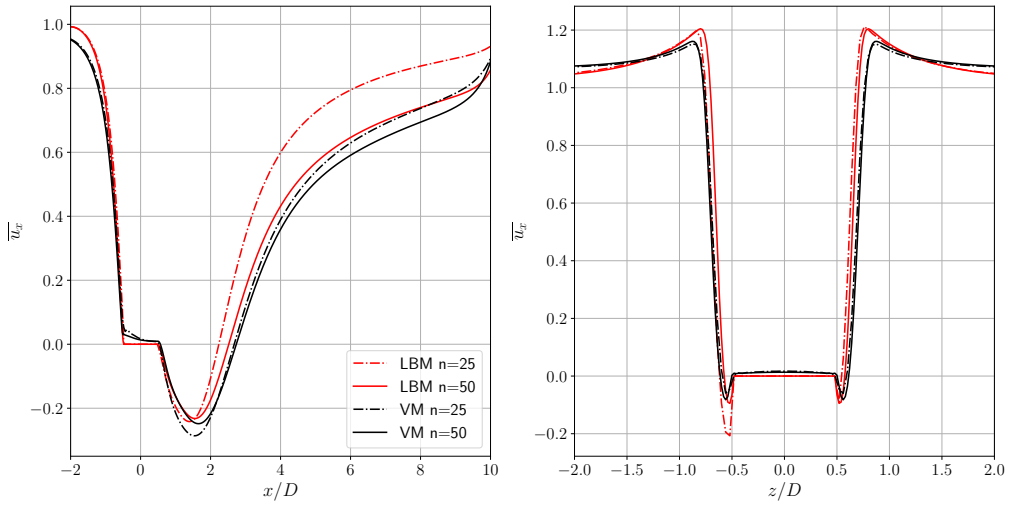


Figure 18: Average streamwise velocity profiles for flow past a cube at $Re = 290$. (Left) Along streamwise direction at $y = 0$ and $z = 0$. (Right) Along spanwise direction at $x = 0$ and $y = 0$.

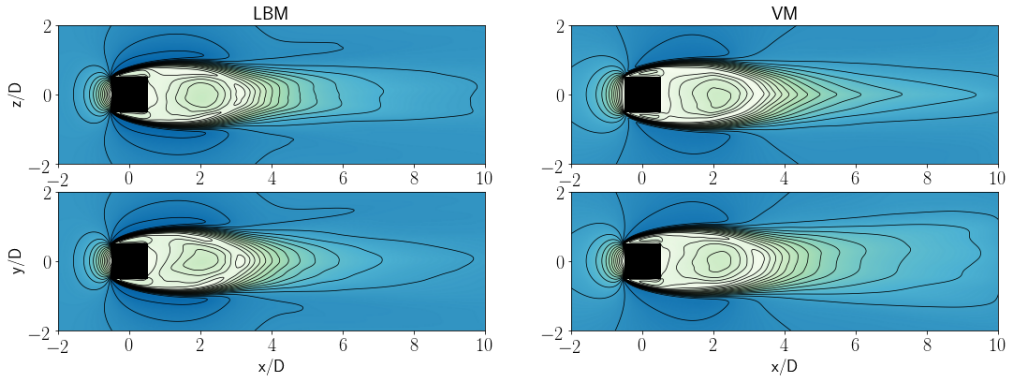


Figure 19: Flow past a cube at $Re = 570$ with $n = 80$. Norm of the mean velocity components ($\sqrt{\overline{u_x}^2 + \overline{u_y}^2 + \overline{u_y}^2}$) XZ (top) and XY (bottom) planes for LBM (left) and VM (right) (same isocontour levels as Fig. 17).

of spurious vorticity at these sharp edges. However, the spurious solution in this region turns out to be attenuated for both methods when refining the mesh; with $n = 80$ the thickness and shape of the upstream boundary layer are very comparable.

Regarding the evaluation of flow characteristics, Tables 5 and 6 report the mean drag, lift and side-lift coefficients ($\bar{c_D}$, $\bar{c_L}$, $\bar{c_S}$), the recirculation length (l_r) and the Strouhal number ($St = fD/U_\infty$) obtained at $Re = 290$ and $Re = 570$ by LBM and VM as well as those of selected references in literature. Concerning the recirculation length, it is defined as the axial distance, along the centerline of the wake, between the center of the cube and the point where the mean streamwise component of the velocity is zero. First of all, it has to be noted that the cube benchmark is not as widely studied in literature as the sphere benchmark, which explains the

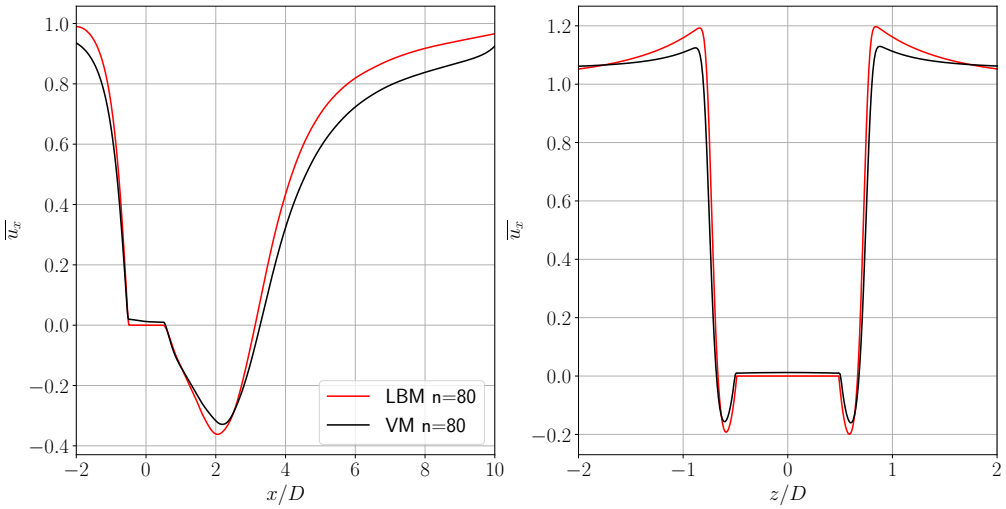


Figure 20: Average streamwise velocity profiles for flow past a cube at $Re = 570$. (Left) Along streamwise direction at $y = 0$ and $z = 0$. (Right) Along spanwise direction at $x = 0$ and $y = 0$.

rather few quantity of reference works about such flow. In terms of flow physics, and as explained in [74], it is important to mention that the values of $\bar{c_D}$ for sphere and cube differ as the Reynolds number increases: once the Reynolds number has reached the unsteady and planar-symmetric regime (ie $Re \gtrsim 276$), the $\bar{c_D}$ for cube increases with the Reynolds number, contrary to the one of the sphere. One can see in Table 5 ($Re = 290$) and Table 6 ($Re = 570$) that this statement is verified both by LBM and VM. In terms of comparison, in Table 5, the $\bar{c_D}$ results of LBM and VM are in good accordance with the evaluations of Haider & Levenspiel [86]. On the other hand, the results of Saha [73] and Khan *et al.* [74] predict a lower drag coefficient. Concerning the Strouhal number at $Re = 290$, the LBM and VM results are rather close to the experimental result of Klotz *et al.* [87], standing around 0.12, whereas the values reported by Saha and Khan *et al.* are lower than 0.1. These differences can be explain by the different numerical setup used in the studies with respectively 30 and 20 points along the cube for Saha and Khan *et al.* and on the other hand a different blockage ratio of the computational domain; indeed, the present BR for simulations at $Re = 290$ is equal to 3.81% against 0.44% in Khan *et al.* simulations and 0.51% for Saha. The present BR is closer to the one of Klotz *et al.* (1.44%) which may explain the closer results in terms of Strouhal number.
It has also to be noted that both LBM and VM methods recover a quasi-zero $\bar{c_S}$ value at $Re = 290$, which is consistent with the fact that the wake is symmetric in the XZ plane at such regime. The sign of the lateral and side lift coefficients has been removed due to the arbitrary asymmetry direction chosen by the flow. Indeed, the asymmetry balance can change due to infinitesimal computing artefacts without altering the flow behavior and topology. Finally one can note that the recirculation length is slightly lower for LBM results. Indeed, the profiles of Figure 18-left show that the LBM mean streamwise velocity in the recirculation area starts to increase earlier than in the VM results. As mentioned previously, this could be an effect of the prescribed inflow/outflow boundary conditions for the two methods.

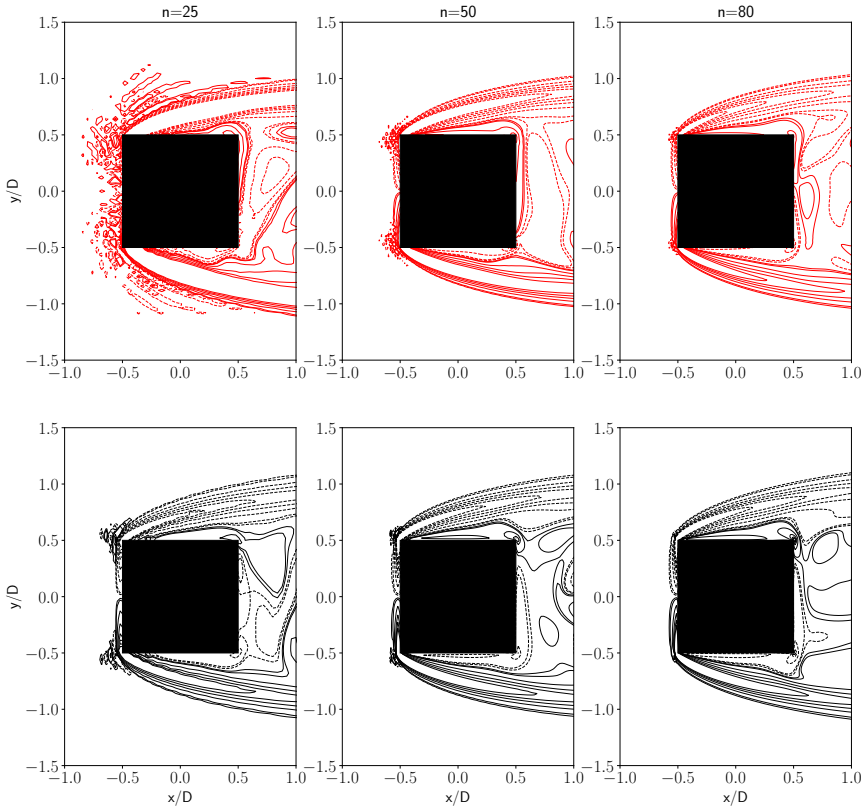


Figure 21: ω_z vorticity isocontours for flow past a cube at $Re = 570$ at $t^\star = 120$. Levels correspond to 10 equispaced values between -10 and 10 with additional contours at ±0.5 and ±0.25, where dashed lines represent negative values. Top(red) LBM and Bottom(Black) VM.

4. Conclusion

In this work an algorithmic and numerical comparative study of a Lattice Boltzmann Method (LBM) and a remeshed Vortex method (VM) was presented in the context of three dimensional incompressible flows. Both approaches belong to families of methods where the flow is not considered and discretized in a macroscopic way and where the notion of particles is a common aspect. In Lattice Boltzmann Methods, the local algorithms and low stencil schemes (lattice) contribute to their efficiency and allow one to easily implement and parallelize them. Regarding semi-Lagrangian (remeshed) Vortex methods, they couple optimally Lagrangian and Eulerian schemes in a fractional step algorithm, which contributes to their flexibility and specificity. In particular, the Lagrangian treatment of the advection term enables the use of an adaptive Δt, thus reducing the total number of time steps within a whole simulation.
The two methods were compared with respect to each others and validated with other experimental/numerical results in literature for three reference test cases: the advection of a simple vortex, the Taylor Green vortex and the flow around a cube in a free domain. The first observation for the first two benchmarks was the low dissipative and dispersive behaviour of the present remeshed Vortex method, especially for coarse grids, which offers better results for low resolutions. The

	$\bar{c_D}$	$\bar{c_L}$	$\bar{c_S}$	l_r	St
Saha [73]	0.783	0.064	0.0	-	0.094
Klotz *et al.* [87]	-	-	-	2.63	0.128 ($Re = 292$)
Khan *et al.* [74]	0.83	0.0053	0.01646	2.50	0.098 ($Re = 300$)
Haider & Levenspiel [86]	1.08	-	-	-	-
Present LBM $n = 25$	1.093	0.136	0.039	2.20	0.120
$n = 50$	0.900	0.073	$4.31 \cdot 10^{-4}$	2.50	0.120
Present VM $n = 25$	0.985	0.066	0.0041	2.64	0.120
$n = 50$	1.003	0.088	$8.14 \cdot 10^{-5}$	2.72	0.120

Table 5: Comparison of mean force coefficients, mean recirculation length and Strouhal number for flow past a cube at $Re = 290$.

	$\bar{c_D}$	$\bar{c_L}$	$\bar{c_S}$	l_r
Khan *et al.* [74]	0.91	0.0576	0.103	-
Haider & Levenspiel [86]	1.14	-	-	-
Present LBM ($n = 80$)	1.121	0.005	0.011	3.13
Present VM ($n = 80$)	1.014	0.003	0.013	3.26

Table 6: Comparison of mean force coefficients, mean recirculation length for flow past a cube at $Re = 570$.

second main observation relies on the fact that the present LBM offers a better accuracy at fine grid resolutions and a higher order of convergence, which also aligns with the conclusions already made in the literature as previously described.
For the cube test case, which is a stiff numerical problem with sharp edges and singularities, both LBM and VM were used with their own boundary conditions in order to illustrate a typical use of both algorithms in the context of a more complex and applied benchmark. Once again, the ability of VM to better compute the expected flow behavior on coarse grids was confirmed. For higher resolved grids the results of both methods show a good qualitative agreement, the quantitative discrepancies being mainly due to the different treatment of inlet/outlet and no-slip boundary conditions within the two methods.

All the results presented in this study should be interpreted in the specific context of the chosen models, which have been mostly tested and compared in terms of global and integral quantities. Even if some improved versions of these models exist in literature, a deeper analysis of local behaviors should increase the understanding of each item of the two presented approaches. Furthermore, in order to fully take advantage of the intrinsic performances of each method, a hybrid algorithmic implementation should be considered as a target for future developments in the framework of this family of numerical methods.

Acknowledgement

Part of this work was performed using HPC resources from GENCI-CINES (Grant 2019-A0072A10636). The authors would like to thanks M. Rafik Abdesselam for providing them the JupyterHUB-Cnam server access for efficient and shared post-processing.

References

[1] T. Douillet-Grellier, S. Leclaire, D. Vidal, F. Bertrand, F. De Vuyst, Comparison of multiphase SPH and LBM approaches for the simulation of intermittent flows, Computational Particle Mechanics 6 (2019) 695–720.

[2] W. Prager, Die druckverteilung an korpern in ebener potentialstromung, Physikalische Zeitschrift 29 (1928) 865–869.

[3] L. Rosenhead, The formation of vortices from a surface of discontinuity, Proc. R. Soc. London Ser. A 134(823) (1931) 170–192.

[4] A. J. Chorin, Numerical study of slighly viscous flow, J. Fluid Mech. 57 (1973) 785–796.

[5] G.-H. Cottet, S. Mas-Gallic, A particle method to solve the Navier-Stokes system, Num. Math. 57 (1990) 1–23.

[6] A. J. Chorin, Vortex sheet approximation of boundary layers, J. Comput. Phys. 27 (1978) 428–442.

[7] Y. M. Marzouk, A. F. Ghoniem, K-means clustering for optimal partitioning and dynamic load balancing of parallel hierarchical N-body simulations, J. Comput. Phys. 207(2) (2005) 493–528.

[8] Q. Hu, N. Gumerov, R. Yokota, R. Barba, L.A. Duraiswami, Scalable fast multipole methods for vortex element methods, in: IEEE Explore, 2012, pp. 1408–1409.

[9] R. Yokota, L. A. Barba, N. T., K. Yasuoka, Petascale turbulence simulation using a highly parallel fast multipole method on GPUs, Comput. Phys. Commun 184(3) (2013) 445–455.

[10] E. Rossi, A. Colagrossi, D. Durante, G. Graziani, Simulating 2D viscous flow around geometries with vertices through the Diffused Vortex Hydrodynamics method, Comput Methods Appl. Mech. Eng. 302 (2016) 147–169.

[11] O. Giannopoulou, A. Colagrossi, A. Di Mascio, C. Mascia, Chorin's approaches revisited: Vortex Particle method vs Finite Volume method, Eng. Anal. Bound. Elem. 106 (2019) 371–388.

[12] L. Barba, A. Leonard, C. Allen, Advances in viscous vortex methods – meshless spatial adaption based on radial basis function interpolation, Int. J. Numer. Methods Fluids 47(5) (2005) 387–421.

[13] E. Rossi, A. Colagrossi, G. Graziani, Numerical simulation of 2D-vorticity dynamics using Particle Methods, Comput. Math. with Appl. 69 (2015) 1484–1503.

[14] S. Huberson, A. Jollès, Correction de l'erreur de projection dans les méthodes particules/maillage, La recherche aérospatiale 4 (1990) 1–6.

[15] P. Koumoutsakos, A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using Vortex methods, J. Fluid Mech. 296 (1995) 1–38.

[16] M. Coquerelle, G.-H. Cottet, A Vortex level-set method for the two-way coupling of an incompressible fluid with colliding rigid bodies, J. Comput. Phys. 227 (2008) 9121–9137.

[17] M. Gazzola, P. Chatelain, W. M. Van Rees, P. Koumoutsakos, Simulations of single and multiple swimmers with non-divergence free deforming geometries, J. Comput. Phys. 230 (2011) 7093–7114.

[18] M. Gazzola, C. Mimeau, A. Tchieu, P. Koumoutsakos, Flow mediated interactions between two cylinders at finite re numbers, Phys. Fluids 24 (2012). doi:10.1063/1.4704195.

[19] C. Mimeau, F. Gallizio, G.-H. Cottet, I. Mortazavi, Vortex penalization method for bluff body flows, Int. J. Numer. Meth. Fluids doi: 10.1002/fld.4038 (2015).

[20] W. M. Van Rees, M. Gazzola, P. Koumoutsakos, Optimal shapes for anguilliform swimmers at intermediate reynolds numbers, J. Fluid Mech. 722 (2013) R31–R312.

[21] C. Mimeau, I. Mortazavi, G.-H. Cottet, Passive control of the flow around a hemisphere using porous media, Eur. J. Mech - B/Fluids 65 (2017) 213–226. doi:10.1016/j.euromechflu.2017.03.002.

[22] P. Chatelain, M. Duponcheel, D.-G. Caprace, Y. Marichal, G. Winckelmans, Vortex Particle-Mesh simulations of vertical axis wind turbine flows: from the airfoil performance to the very far wake, Wind Energ. Sci. 2 (2017) 317–328.

[23] M. Gazzola, B. Hejazialhosseini, P. Koumoutsakos, Reinforcement learning and wavelet adapted Vortex methods for simulations of self-propelled swimmers, SIAM J. Sci. Comput 36 - 3 (2014) B622–B639.

[24] Y.-H. Qian, D. d'Humières, P. Lallemand, Lattice BGK models for Navier-Stokes equation, Europhys. Lett. 17 (1992) 479–484.

[25] He, X. and Luo, L.S., A priori derivation of the lattice Boltzmann equation, Phys. Rev. E 55 (1997) R6333.

[26] X. Shan, X. Yuan, H. Chen, Kinetic theory representation of hydrodynamics: a way beyond the Navier-Stokes equation., J. Fluid Mech. 550 (2006) 413–441.

[27] L. Zhong, S. Feng, P. Dong, S. Gao, Lattice Boltzmann schemes for the nonlinear Schrödinger equation., Phys. Rev. E 74 (2006) 036704.

[28] P. Romatschke, M. Mendoza, S. Succi, Fully relativistic lattice boltzmann algorithm, Phys. Rev. C 84 (2011). doi:10.1103/PhysRevC.84.034903.

[29] P. Bhatnagar, E. Gross, M. Krook, A model for collision process in gases. I. Small amplitude process in charged and neutral one-component systems, Phys. Rev. 94(3) (1954) 511–525.

[30] D. d'Humière, I. Ginzburg, Y. Krafczyk, P. Lallemand, L. Luo, Multiple relaxation time lattice Boltzmann models in three dimensions, Phil. Trans. R. Soc. Lon. A 360 (2002) 437–451.

[31] S. Marié, D. Ricot, P. Sagaut, Comparison between lattice boltzmann method and navier-stokes high order schemes for computational aeroacoustics., J. Comput. Phys. 228 (2009) 1056–1070.
[32] Y. Peng, W. Liao, L.-S. Luo, L.-P. Wang, Comparison of the Lattice Boltzmann and pseudo-spectral methods for decaying turbulence: Low-order statistics, Comp. & Fluids 39 (2010) 568 – 591. URL: http://www.sciencedirect.com/science/article/pii/S0045793009001546. doi:https://doi.org/10.1016/j.compfluid.2009.10.002.
[33] J. Sterling, S. Chen, Stability analysis of Lattice Boltzmann methods, J. Comput. Phys. 123 (1996) 196–206.
[34] C. David, P. Sagaut, Structural stability of Lattice Boltzmann schemes, Physica A 444 (2016) 1–8.
[35] C. Coreixas, G. Wissocq, B. Chopard, J. Latt, Impact of collision models on the physical properties and the stability of lattice boltzmann methods, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 378 (2020) 20190397. URL: http://dx.doi.org/10.1098/rsta.2019.0397. doi:10.1098/rsta.2019.0397.
[36] M. Geier, M. Schönherr, A. Pasquali, M. Krafczyk, The cumulant lattice boltzmann equation in three dimensions: Theory and validation, Computers and Mathematics with Applications 70 (2015) 507 – 547. URL: http://www.sciencedirect.com/science/article/pii/S0898122115002126. doi:https://doi.org/10.1016/j.camwa.2015.05.001.
[37] J. Latt, B. Chopard, Lattice Boltzmann method with regularized pre-collision distribution functions, Mathematics and Computers in Simulation 72 (2006) 165–168.
[38] F. Tosi, S. Ubertini, S. Succi, H. Chen, I. Karlin, Numerical stability of Entropic versus positivity-enforcing Lattice Boltzmann schemes, Mathematics and Computers in Simulation 72 (2006) 227–231.
[39] P. Lallemand, F. Dubois, Some results on energy-conserving lattice Boltzmann models, Computers and Mathematics with Applications 65 (2013).
[40] F. Dubois, B. Graille, S. R. Rao, A notion of non-negativity preserving relaxation for a mono-dimensional three velocities scheme with relative velocity, Journal of Computational Science (2020) 101181. URL: http://www.sciencedirect.com/science/article/pii/S1877750320304828. doi:https://doi.org/10.1016/j.jocs.2020.101181.
[41] C. Coreixas, B. Chopard, J. Latt, Comprehensive comparison of collision models in the Lattice Boltzmann framework: Theoretical investigations, Phys. Rev. E 100 (2019). URL: http://dx.doi.org/10.1103/PhysRevE.100.033305. doi:10.1103/physreve.100.033305.
[42] G.-H. Cottet, J.-M. Etancelin, F. Perignon, C. Picard, High order semi-lagrangian particles for transport equations: numerical analysis and implementation issues, ESAIM: Mathematical Modelling and Numerical Analysis 48 (2014) 1029–1064.
[43] A. Magni, G.-H. Cottet, Accurate, non-oscillatory remeshing schemes for particle methods, J. Comput. Phys. 231(1) (2012) 152–172.
[44] G.-H. Cottet, P. Poncet, Advances in direct numerical simulation of 3D wall-bounded flows by Vortex-In-Cell methods, J. Comput. Phys. 193 (2003) 136–158.
[45] D.-G. Caprace, T. Gillis, P. Chatelain, FLUPS - a Fourier-based library of unbounded Poisson solvers, SIAM Journal on Scientific Computing 43 (2021) github.com/vortexlab–uclouvain/flups.
[46] R. Yokota, T. Narumi, R. Sakamaki, S. Kameoka, S. Obi, K. Yasuoka, Fast multipole methods on a cluster of GPUs for the meshless simulation of turbulence, Comp. Physics Comm. 180(11) (2009) 2066–2078.
[47] D. Rossinelli, B. Hejazialhosseini, W. M. van Rees, M. Gazzola, M. Bergdorf, P. Koumoutsakos, MRAG-I2D: Multi-resolution adapted grids for remeshed vortex methods on multicore architectures, J. Comput. Phys. 288 (2015) 1–18.
[48] G. Novati, S. Verma, D. Alexeev, D. Rossinelli, W. van Rees, P. Koumoutsakos, Synchronisation through learning for two self-propelled swimmers, Bioinspir. Biomim. 12 (2017) 036001.
[49] P. Koumoutsakos, Multiscale flow simulations using particles, Ann. Rev. Fluid Mechanics 37 (2005) 457–487.
[50] W. M. Van Rees, A. Leonard, D. Pullin, P. Koumoutsakos, A comparison of Vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high reynolds numbers, J. Comput. Phys. 230(8) (2011) 2794–2805.
[51] Lallemand, P. and Luo, L.S., Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance and stability, Phys. Rev. E 61(06) (2000).
[52] T. Février, Extension et analyse des schémas de Boltzmann sur réseau : les schémas à vitesse relative., Ph.D. thesis, Univeristé de Paris-Saclay, 2015.
[53] O. Malaspinas, Increasing stability and accuracy of the latticeboltzmann scheme: recursivity and regularization, arXiv e-prints (2015) arXiv:1505.06900. URL: https://arxiv.org/abs/1505.06900. arXiv:1505.06900.
[54] Y. Feng, P. Sagaut, W.-Q. Tao, A compressible Lattice Boltzmann finite volume model for high subsonic and transonic flows on regular lattices, Comp. & Fluids 131 (2016) 45 – 55. URL: http://www.sciencedirect.com/science/article/pii/S0045793016300652. doi:https://doi.org/10.1016/j.compfluid.2016.03.009.
[55] A. D. Rosis, K. H. Luo, Role of higher-order hermite polynomials in the central-moments-based lattice

boltzmann framework, Phys. Rev. E 99 (2019). URL: https://doi.org/10.1103/physreve.99.013301. doi:10.1103/physreve.99.013301.
[56] J. J., M. O., S. P., A new hybrid recursive regularised bhatnagar–gross–krook collision model for lattice boltzmann method-based large eddy simulation, Journal of Turbulence 19 (2018) 1051–1076. doi:10.1080/14685248.2018.1540879.
[57] F. Dubois, P. Lallemand, Towards higher order lattice boltzmann schemes, J. Stat. Mech. 2009 (2008). doi:10.1088/1742-5468/2009/06/P06006.
[58] S. Guo, Y. Feng, J. Jacob, F. Renard, P. Sagaut, An efficient lattice boltzmann method for compressible aerodynamics on d3q19 lattice, J. Comput. Phys. (2020).
[59] He, Xiaoyi and Luo, L.S., Lattice Boltzmann Model for the Incompressible Navier-Stokes Equation, J. Stat. Phys. 88 (1997) 927–944.
[60] G. Wissocq, P. Sagaut, J.-F. Boussuge, An extended spectral analysis of the lattice boltzmann method: modal interactions and stability issues, J. Comput. Phys. 380 (2019) 311–333. URL: https://doi.org/10.1016/j.jcp.2018.12.015. doi:10.1016/j.jcp.2018.12.015.
[61] O. Hald, Convergence of Vortex methods II, SIAM J. Num. Anal 16 (1979) 726–755.
[62] J. Beale, A. Majda, Vortex methods II: high order accuracy in 2 and 3 dimensions, Math. Comput. 32 (1982) 29–52.
[63] C. Anderson, C. Greengard, On Vortex methods, SIAM J. Num. Anal. 22 (1985) 413–440.
[64] G.-H. Cottet, P. Koumoutsakos, Vortex Methods - Theory and Practice, Cambridge University Press, 2000.
[65] F. Gendre, D. Ricot, G. Fritz, P. Sagaut, Grid refinement for aeroacoustics in the lattice boltzmann method: A directional splitting approach, Phys. Rev. E 96 (2017). doi:10.1103/PhysRevE.96.023311.
[66] G. Wissocq, J.-F. m. c. Boussuge, P. Sagaut, Consistent vortex initialization for the athermal lattice Boltzmann method, Phys. Rev. E 101 (2020) 043306. URL: https://link.aps.org/doi/10.1103/PhysRevE.101.043306. doi:10.1103/PhysRevE.101.043306.
[67] L. Kovasznay, Turbulence in supersonic flow., J. Aeronautical Science 20 (1953) 657–682.
[68] 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 (2019) 1950074. doi:10.1142/S0129183119500748.
[69] P. Skordos, Initial and Boundary Conditions for the lattice Boltzmann Method., Phys. Rev. E 48 (1993) 4823–4842.
[70] R. Mei, L.-S. Luo, P. Lallemand, D. d'Humières, Consistent initial conditions for Lattice Boltzmann simulations, Comp. & Fluids 35 (2006) 855 – 862. URL: http://www.sciencedirect.com/science/article/pii/S0045793005001477. doi:https://doi.org/10.1016/j.compfluid.2005.08.008.
[71] M. Brachet, D. Meiron, S. Orszag, B. Nickel, R. Morf, U. Frisch, Small-scale structure of the taylor-green vortex., J. Fluid Mech. 130 (1983) 411–452.
[72] G.-H. Cottet, Artificial viscosity models for vortex and particle methods, J. Comput. Phys. 127 (1996) 199–208.
[73] A. K. Saha, Three-dimensional numerical simulations of the transition of flow past a cube, Phys. Fluids 16 (2004) 1630–1646.
[74] M. H. Khan, A. Sharma, A. Agrawal, Simulation of Flow Around a Cube at Moderate Reynolds Numbers Using the Lattice Boltzmann Method, J. Fluids Eng. 142 (2019). URL: https://doi.org/10.1115/1.4044821. doi:10.1115/1.4044821, 011301.
[75] J. P. Caltagirone, Sur l'intéraction fluide-milieu poreux : Application au calcul des efforts exercés sur un obstacle par un fluide visqueux, C. R. Acad. Sci. Paris 318 (1994).
[76] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numer. Math. 81 (1999) 497–520.
[77] C.-H. Bruneau, I. Mortazavi, Passive control of the flow around a square cylinder using porous media, Int. J. Numer. Methods Fluids 46 (2004) 415–433.
[78] C. Mimeau, G.-H. Cottet, I. Mortazavi, Direct numerical simulations of three-dimensional flows past obstacles with a Vortex penalization method, Comp. & Fluids 136 (2016) 331–347.
[79] S. Verma, G. Abbati, G. Novati, P. Koumoutsakos, Computing the force distribution on the surface of complex, deforming geometries using Vortex methods and Brinkman penalization, Int. J. Numer. Methods Fluids 85 (2017) 484–501.
[80] C. Bernier, M. Gazzola, R. Ronsse, P. Chatelain, Simulations of propelling and energy harvesting articulated bodies via Vortex Particle-Mesh methods, J. Comput. Phys. 392 (2019) 34–55.
[81] F. Morency, H. Beaugendre, F. Gallizio, Aerodynamic force evaluation for ice shedding phenomenon using Vortex in cell scheme, penalisation and level set approaches, Int. J. Numer. Methods Fluids 26(9–10) (2012) 435–450.
[82] Q. Zou, X. He, On pressure and velocity boundary conditions for the lattice boltzmann BGK model, Phys. Fluids 9 (1997) 1591–1598. URL: https://doi.org/10.1063/1.869307. doi:10.1063/1.869307.
[83] X. He, Q. Zou, L.-S. Luo, M. Dembo, Analytic solutions of simple flows and analysis of nonslip bound-

ary conditions for the lattice Boltzmann BGK model, Journal of Statistical Physics 87 (1997) 1572–9613. doi:10.1007/BF02181482.

[84] T. Inamuro, M. Yoshino, F. Ogino, A non-slip boundary condition for lattice boltzmann simulations, Phys. Fluids 7 (1995) 2928–2930. doi:10.1063/1.868766.

[85] J. Wu, C. Shu, An improved immersed boundary-lattice boltzmann method for simulating three-dimensional incompressible flows, J. Comput. Phys. 229 (2010) 5022 – 5042. URL: http://www.sciencedirect.com/science/article/pii/S0021999110001348. doi:https://doi.org/10.1016/j.jcp.2010.03.024.

[86] A. Haider, O. Levenspiel, Drag coefficient and terminal velocity of spherical and nonspherical particles, Powder Technology 58 (1989) 63–70.

[87] L. Klotz, S. Goujon-Durand, J. Rokicki, J. Wesfreid, Experimental investigation of flow behind a cube for moderate reynolds numbers, J. Fluid Mech. 750 (2014) 73–98. doi:10.1017/jfm.2014.236.