POLITECNICO DI TORINO
Repository ISTITUZIONALE

Link-wise Artificial Compressibility Method

Original
Link-wise Artificial Compressibility Method / Asinari, Pietro; T., Ohwada; Chiavazzo, Eliodoro; DI RIENZO, ANTONIO FABIO. - In: JOURNAL OF COMPUTATIONAL PHYSICS. - ISSN 0021-9991. - 231:(2012), pp. 5109-5143. [10.1016/j.jcp.2012.04.027]

Availability:
This version is available at: 11583/2496923 since:

Publisher:
Elsevier

Published
DOI:10.1016/j.jcp.2012.04.027

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2024

Link-wise Artificial Compressibility Method

Pietro Asinari [a 1], Taku Ohwada [b], Eliodoro Chiavazzo [a] and Antonio F. Di Rienzo [a]

[a] *Dipartimento Energia, Politecnico di Torino, Torino 10129, Italy*

[b] *Department of Aeronautics and Astronautics, Graduate School of Engineering, Kyoto University, Kyoto 606-8501, Japan*

Abstract

The Artificial Compressibility Method (ACM) for the incompressible Navier-Stokes equations is (link-wise) reformulated (referred to as LW-ACM) by a finite set of discrete directions (links) on a regular Cartesian mesh, in analogy with the Lattice Boltzmann Method (LBM). The main advantage is the possibility of exploiting well established technologies originally developed for LBM and classical computational fluid dynamics, with special emphasis on finite differences (at least in the present paper), at the cost of minor changes. For instance, wall boundaries not aligned with the background Cartesian mesh can be taken into account by tracing the intersections of each link with the wall (analogously to LBM technology). LW-ACM requires no high-order moments beyond hydrodynamics (often referred to as *ghost* moments) and no kinetic expansion. Like finite difference schemes, only standard Taylor expansion is needed for analyzing consistency. Preliminary efforts towards optimal implementations have shown that LW-ACM is capable of similar computational speed as optimized (BGK-) LBM. In addition, the memory demand is significantly smaller than (BGK-) LBM. Importantly, with an efficient implementation, this algorithm may be one of the few which is compute-bound and not memory-bound. Two- and three-dimensional benchmarks are investigated, and an extensive comparative study between the present approach and state of the art methods from the literature is carried out. Numerical evidences suggest that LW-ACM represents an excellent alternative in terms of simplicity, stability and accuracy.

Key words: artificial compressibility method (ACM); lattice Boltzmann method (LBM); complex boundaries; incompressible Navier-Stokes equations

[1] Corresponding author: pietro.asinari@polito.it (P. Asinari). Example codes can be provided upon request to corresponding author.

Contents

1 Introduction **4**

2 Link-wise Artificial Compressibility Method **7**

2.1 The main algorithm: Link-wise formulation 7

2.2 The main algorithm: Finite difference formulation 11

2.3 Physical and lattice units 13

2.4 Optimized computer implementation 15

2.5 Simple boundary conditions 18

2.5.1 Isothermal Couette flow 18

2.5.2 Generalized Green-Taylor vortex flow 19

2.5.3 Minion & Brown flow 20

2.6 Link-wise wall boundary conditions 23

2.6.1 2D lid driven cavity flow 26

2.6.2 3D diagonally driven cavity flow 34

2.6.3 Circular Couette flow 37

3 Conclusions **46**

A Appendix: Equilibrium distribution functions **47**

B Appendix: Asympthotic analysis **49**

B.1 Diffusive scaling 51

B.2 Acoustic scaling 52

B.3 Forcing 53

B.4 Energy equation 54

C Appendix: Computing derivatives locally **56**

D Appendix: Equivalent finite-difference formulas **58**

1 Introduction

Despite a large variety of mesh generation techniques for numerical solvers of the fluid dynamics governing equations [1], addressing complex geometries remains a difficult duty. To this aim, several approaches were proposed for adapting computational grids to complex geometries by unstructured meshes. Generating unstructured meshes of high quality though, is a challenging computational task *per se*, which involves quite advanced algorithms (Rupperts algorithm, Chews second algorithm, Delaunay triangulation, etc.) [1]. While those approaches simplify the treatment of boundaries, in turn, each of them introduces new difficulties such as extra terms in the equations, extra interpolations, larger computational molecules, and problems associated with the transfer of information across grid interfaces. The added complexity makes code development even more difficult and increases computation time [2], with an additional risk that those algorithms may not lead to an acceptable solution.

An alternative approach which has attracted an increasing interest in recent years makes use of Cartesian grids for all cells with the exception of those that present intersections with boundaries, which are thus truncated according to the shape of the boundary surface. The advantages of Cartesian grids can be retained for all cells in the bulk fluid, and a special treatment is only reserved to boundary cells. On the contrary, cells fully outside the flow can be simply ignored during computations [2]. In the literature, this approach is typically referred to as the "embedded boundary method", the "Cartesian grid method" or the "cut-cell method" [3, 4, 5, 6, 7]. Clearly, the challenging point is to make the method accurate in dealing with curved and planar boundaries transversal to the grid, even though such boundaries are conveniently approximated in a staircase fashion. More specifically, after determining the intersection between the Cartesian grid and a boundary, cells whose center lies in the fluid are reshaped by discarding their part belonging to the solid wall, while pieces of cut cells with the center in the solid are absorbed by neighboring cells [5]. This results in the formation of control-volumes which are trapezoidal in shape.

Classical approaches to the incompressible limit of Navier-Stokes equations require (a) dedicated techniques for solving a pressure Poisson equation in order to take advantage of the underlying structured nature of the mesh and thus speed-up convergence [5]. Moreover, (b) compact multi-dimensional polynomial interpolating functions are used for obtaining a second-order accurate approximation of the fluxes and gradients on the faces of the trapezoidal boundary cells from available neighboring cell-center values [5]. Recent developments to this also follows a similar approach [6, 7].

Both (a) the need of a dedicated solver for the pressure Poisson equation and

(b) the use of compact multi-dimensional interpolations, can be overcome by the lattice Boltzmann method (LBM) [8], while preserving the main features of the Cartesian cut-cell method for mesh generation and boundary treatment. However, this comes at a price of dealing with specific features inherited from the kinetic theory of gases, which are unessential as far as the continuum description of incompressible Navier-Stokes equations is the only concern.

For this reason, we propose a novel formulation of the artificial compressibility method (ACM), which retains the convenient features of LBM, namely (a) the artificial compressibility and (b) the link-wise formulation based on the theory of characteristics, but concurrently gets rid of unessential heritages of the kinetic theory of gases.

Similarities between LBM and ACM [9] are sometimes reminded in the literature. It is well known indeed that the Chapman-Enskog expansion of the LBM updating rule delivers the governing equation of ACM: the artificial compressibility equations (ACE). The latter consist of the same momentum equations as the incompressible Navier-Stokes equations (INSE), in addition to an artificial continuity equation including pressure time derivative. ACE can be also recovered by the more systematic expansion such as the Hilbert method under diffusive scaling [10].

The lattice kinetic scheme (LKS) [11] (a variant of LBM) also shows similarities with ACM at the level of computer programming, despite the fact that the former deals with distribution functions of gas molecules, while the latter only with hydrodynamic (macroscopic) variables.

For a special value of the relaxation parameter in the LBM updating rule, an updated value of the distribution function depends only on the previous equilibrium function at an arbitrary mesh point in the stencil. Since equilibria are in turn function of macroscopic variables only, the LKS updating rule can be immediately recognized as a kind of finite difference scheme, acting on hydrodynamic variables. As a result, the moment system of LKS delivers a variant of ACM. Recently, taking advantage of the similarities between LBM and ACM, the latter was reformulated as a high order accurate numerical method (fourth order in space and second order in time) [12].

Motivated by the common belief that an important reason of success of the LBM (in particular MRT-LBM [13]) is its remarkable robustness for simulating the various complex flows, the stability of the revived ACM has been further enhanced [14].

In this paper, in an attempt of making ACM even more similar to LBM, we propose yet a new formulation of ACM referred to as link-wise ACM (LW-ACM) in the following text. For the sake of completeness, we summarize both the main features of the revived ACM [12, 14], still valid for the present LW-

ACM, and the additional advantages due to a link-wise formulation.

(1) ACM deals with macroscopic variables only, thus offering the opportunity of exploiting all the pre-existing finite-difference (FD) technologies: This is, for instance, a clear advantage when imposing inlet and outlet boundary conditions. On the contrary, LBM needs to account for nonhydrodynamic quantities (sometimes called *ghost* quantities) which, though may not have direct impact on the hydrodynamic behavior, they can still be responsible of numerical instabilities [15]. Unfortunately, owing to nonlinearities, there are no clear and general recipes yet, on how to optimally design ghost quantities with desired stability properties. As far as the popular compact stencils are concerned, such as D2Q9, D3Q15 and D3Q19 [16] with no special corrections, LBM ghost quantities remain numerical artifacts: Positive effects of such quantities for enhancing stability of usual FD schemes are still far from being clearly demonstrated. ACM fully overcomes this issue, focusing instead on the minimum set of information for incompressible fluid dynamics.

(2) Similarly to LBM, ACM posses the ability of computing transient solutions of incompressible Navier-Stokes equations (INSE), without resorting to a Poisson equation for pressure. The underlying idea, directly inspired by the asymptotic analysis of LBM schemes, is to multiply the pressure time derivative of artificial continuity equation by a mesh-dependent parameter. In this way, the numerical Mach number, which is a mere numerical artifact for INSE (rigorously valid in the limit of vanishing Mach number) is linked to the mesh spacing. Higher accuracy than LBM schemes can be also achieved by exploiting the asymptotic behavior of the solution of the artificial compressibility equations for small Mach numbers [12].

(3) ACM can use different meshing techniques. For example, it is possible to use simple lattice structures, namely Cartesian structured meshes, eventually recursively refined like those also used by LBM, or it can be even formulated in a finite-volume fashion including unstructured body-fitted meshes. In the latter case, the same comments discussed at the beginning of this section about the computational overhead for generating unstructured meshes hold as well. On the other hand, adopting simple lattice structures is not so straightforward as in LBM: The wall treatment depends on the dimensionality of the problem (namely the look-up table for discriminating wall boundary conditions in 2D is different from the one in 3D). Previous problem can be overcome by LBM thanks to the link-wise formulation. A "link" is a generic direction identified by a discrete velocity of the lattice and coincides with one of the characteristics along which advection is performed (consistently with the method of characteristics – MOC). Such a numerical scheme based on a finite set of links can cope with a complex boundaries by (a) identifying the intersections of each link with the wall and (b) updating the variable corresponding to such a link by a local rule. The local rule is always the same and the intersec-

tions can be computed once for all during pre-processing. The previous procedure easily applies to any orientations of the wall with respect to the lattice, as well as to any dimensions. In this paper, the above advantages of LBM are made available to ACM.

(4) ACM deals with the minimum number of fields describing incompressible fluid dynamics: D+1, where D is the physical dimension of the problem. On the other hand, LBM deals with discrete distribution functions f_i, which are as many as the lattice velocities Q. LBM has thus a memory overhead due to: D+1<Q. Between these two sets of variables, there is a simple connection: Local equilibria $f_i^{(e)}$ (Q variables) can be computed by means of macroscopic quantities only (D+1 variables). Introducing a larger set of variables $f_i^{(e)}$ may seem a redundant and useless artifact. However, this work aims at demonstrating that formulating ACM in terms of $f_i^{(e)}$ offers advantages as well. In particular, as far as the updating rule of the algorithm is similar to LBM, it is possible (eventually with minor changes) to take advantage of most of LBM technology. For example, link-wise ACM can also be formulated in terms of local equilibrium $f_i^{(e)}$ and this enables a convenient treatment of complex moving boundaries typical of the LBM (see next). In conclusion, link-wise ACM has two possible (and fully equivalent) formulations: (a) in terms of macroscopic variables like standard ACM (capable of exploiting pre-existing FD technology) and (b) in terms of local equilibrium (capable of exploiting pre-existing LBM technology).

This paper is organized in sections as follow. The link-wise artificial compressibility algorithm for incompressible isothermal fluid dynamics is introduced In Section 2.1, where some classical benchmarks are presented (isothermal Couette flow, generalized Green-Taylor vortex flow and Minion & Brown flow) as well. In Section 2.6 the link-wise wall boundary conditions are discussed, including moving and complex walls, and some numerical tests are presented (2D lid driven cavity flow, 3D diagonally driven cavity flow and Circular Couette flow). Finally, Section 3 reports some concluding remarks.

2 Link-wise Artificial Compressibility Method

2.1 The main algorithm: Link-wise formulation

The Boltzmann equation is the fundamental equation in kinetic theory of gases, describing time evolution of the distribution function of gas molecules as a function of time, space coordinates, and molecular velocity. The Bhatnagar-Gross-Krook (BGK) model equation inherits the main features of the original Boltzmann equation, with the fluid-dynamic description of the BGK solution

for small Knudsen numbers being much simpler to obtain. Hence, owing to a remarkably less demanding effort, it come advantageous the employment of the BGK equation at the heart of kinetic methods for solving INSE. A well known drawback of the BGK equation is that the recovered Prandtl number is unity, while the original Boltzmann equation yields a value near to 2/3. However, since most of the LBM schemes do not consider the energy equation, the issue of inaccurate thermal diffusivity can be often neglected. At the same time, it is allowed to employ the isothermal BGK with a constant collision frequency for this purpose [17]. Hence isothermal BGK will be our starting point in the following derivation. A crucial *ingredient* of any lattice Boltzmann scheme is a finite set of microscopic velocities, called lattice. The generic lattice velocity is identified by the subscript i, where $0 \leq i \leq Q-1$. The LBM simulates the time evolution of a weakly compressible gas flow in nearly continuum regime by solving a kinetic equation on the lattice and yields the solution of the incompressible Navier-Stokes equation as its leading order. Hence, the relaxation frequency in the BGK equation can be expressed as a function of the kinematic viscosity ν: In particular, the relaxation frequency is set equal to c^2/ν, where c is the lattice sound speed. The dimensionless form of the simplified BGK equation on a lattice takes the form

$$\frac{\partial f_i^{**}}{\partial \hat{t}} + \hat{\mathbf{v}}_i \cdot \hat{\nabla} f_i^{**} = \frac{c^2}{\nu}\left(f_i^{(e)} - f_i^{**}\right), \tag{1}$$

where $\hat{\mathbf{x}}$, $\hat{t}$, and $\hat{\mathbf{v}}_i$ are the (dimensionless) space coordinates, time, and molecular velocity components, respectively; f_i^{**} is the distribution function of gas molecules for the i-th velocity on the lattice; $f_i^{(e)}$ is the equilibrium distribution function. The distribution function f_i^{**} is defined at a discrete set [2] of spatial points $\hat{\mathbf{x}} = \mathbf{x}/\Delta x$, where Δx is the dimensionless mesh spacing, $\Delta x = \Delta x'/L$, with $\Delta x'$ the mesh spacing in physical units and L the characteristic length scale of the flow field. Similarly, time levels are defined as $\hat{t} = t/\Delta t$, where Δt is the dimensionless time step, $\Delta t = \Delta t'/(L/U)$, with $\Delta t'$ the time step in physical units and U a characteristic flow speed. The Q lattice velocities $\hat{\mathbf{v}}_i$ are defined according to the considered scheme [16]. All points $\hat{\mathbf{x}}$ form a regular lattice such that $\hat{\mathbf{x}} - \hat{\mathbf{v}}_i$ belongs to the lattice, regardless of $\hat{\mathbf{x}}$ and $\hat{\mathbf{v}}_i$.

The quantities $f_i^{(e)}$ are local functions of density $\rho = \sum_i f_i$ and momentum $\rho\mathbf{u} = \sum_i \hat{\mathbf{v}}_i f_i$ computed at $\hat{\mathbf{x}}$ and $\hat{t}$, namely $f_i^{(e)} = f_i^{(e)}(\rho, \mathbf{u})$ (see Appendix A for some examples). The quantities $f_i^{(e)}$ are designed in order to recover the incompressible isothermal fluid dynamics [16]. In particular, recovering incompressible Euler equations requires that $\sum_i f_i^{(e)} = \rho$ and $\sum_i \hat{\mathbf{v}}_i f_i^{(e)} = \rho\mathbf{u}$, i.e. conservation of hydrodynamic moments, and $\sum_i \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i f_i^{(e)} = \mathbf{\Pi}^{(e)} = \rho\mathbf{u}\mathbf{u} + p\,\mathbf{I}$,

[2] If not evident otherwise, we use "hat" notation for lattice quantities expressed by means of integer values and "prime" notation for quantities expressed in physical units.

with p function of density only (isothermal case): $p = c^2\,\rho$. Further constraints can be found by asymptotic analysis (see Appendix B for details). However consistency leaves some degrees of freedom in designing these functions, which can be used for improving stability (one possible strategy is discussed in Appendix C). In the following, it will be convenient to consider separately the odd parts of equilibria, namely the quantities $f_i^{(e,o)}$, defined as

$$f_i^{(e,o)}(\rho, \mathbf{u}) = \frac{1}{2}\left(f_i^{(e)}(\rho, \mathbf{u}) - f_i^{(e)}(\rho, -\mathbf{u})\right). \tag{2}$$

Let us suppose that $\nu \ll 1$: then it is possible to find an approximated solution of (1) by singular regular expansion, where:

$$f_i^{**} = f_i^{(e)} - \frac{\nu}{c^2}\,\hat{\mathbf{v}}_i \cdot \hat{\nabla} f_i^{(e)} + O(\nu^2). \tag{3}$$

Introducing the above approximation in the advection term of Eq. (1), it yields

$$\frac{\partial f_i^{**}}{\partial \hat{t}} = -\hat{\mathbf{v}}_i \cdot \hat{\nabla} f_i^{(e)} + \frac{\nu}{c^2}\,(\hat{\mathbf{v}}_i \cdot \hat{\nabla})^2 f_i^{(e)} + \frac{c^2}{\nu}\left(f_i^{(e)} - f_i^{**}\right) + O(\nu^2). \tag{4}$$

Here, the goal is to derive an algorithm formulated in terms of only hydrodynamic quantities, i.e. the statistical macroscopic moments of f_i^{**} corresponding to microscopic quantities conserved by the collisional operator (right hand side of Eq. (1)). In particular, the local equilibrium $f_i^{(e)}$ is defined such that it has the same hydrodynamic quantities of f_i^{**}. Hence, as far as the computation of the hydrodynamic quantities is concerned, the collisional operator in (4) is unessential. Removing the latter term determines a modification in the model equation, though there is no effect on the hydrodynamic quantities. Let us define a new model equation by removing the collisional term and neglecting terms $O(\nu^2)$ in (4), which can be re-formulated with respect to the new distribution function f_i^* as follows:

$$\frac{\partial f_i^*}{\partial \hat{t}} = -\hat{\mathbf{v}}_i \cdot \hat{\nabla} f_i^{(e)} + \frac{\nu}{c^2}\,(\hat{\mathbf{v}}_i \cdot \hat{\nabla})^2 f_i^{(e)}. \tag{5}$$

Clearly the previous one is the Fokker-Planck equation, with drift coefficient equal to 1 and diffusion coefficient equal to ν/c^2. The above model equation (5) can be recast in the equivalent form

$$\frac{\partial f_i^*}{\partial \hat{t}} = -\eta_2\left(\hat{\mathbf{v}}_i \cdot \hat{\nabla} f_i^{(e,e)} - \eta_4/\eta_2\,(\hat{\mathbf{v}}_i \cdot \hat{\nabla})^2 f_i^{(e,e)}\right) - \eta_1\left(\hat{\mathbf{v}}_i \cdot \hat{\nabla} f_i^{(e,o)} - \eta_3/\eta_1\,(\hat{\mathbf{v}}_i \cdot \hat{\nabla})^2 f_i^{(e,o)}\right), \tag{6}$$

where the odd part of the equilibrium distribution function is defined by (2), while the even part is $f_i^{(e,e)} = f_i^e - f_i^{(e,o)}$, $\eta_1 = \eta_2 = 1$ and $\eta_3 = \eta_4 = \nu/c^2$. Recalling that

$$\hat{\mathbf{v}}_i \cdot \hat{\nabla} f_i^{(e,o/e)} - \frac{1}{2}\,(\hat{\mathbf{v}}_i \cdot \hat{\nabla})^2 f_i^{(e,o/e)} \approx f_i^{(e,o/e)}(\hat{\mathbf{x}} - \hat{\mathbf{v}}_i, \hat{t}) - f_i^{(e,o/e)}(\hat{\mathbf{x}}, \hat{t}), \tag{7}$$

we modify once more the model equation by setting $\eta_1 = 2\,\nu/c^2$ and $\eta_4 = 1/2$ (while other parameters remain unchanged, namely $\eta_2 = 1$ and $\eta_3 = \nu/c^2$). By doing so, $\eta_4/\eta_2 = \eta_3/\eta_1 = 1/2$ which enables to use the approximation (7). By means of the above set of parameters, Eq. (6) becomes

$$\frac{\partial f_i^*}{\partial \hat{t}} = -\left(f_i^{(e,e)}(\hat{\mathbf{x}},\hat{t}) - f_i^{(e,e)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t})\right) - 2\,\frac{\nu}{c^2}\left(f_i^{(e,o)}(\hat{\mathbf{x}},\hat{t}) - f_i^{(e,o)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t})\right). \tag{8}$$

As common in LBM, we apply the forward Euler rule for approximating first order time derivatives:

$$f_i^*(\hat{\mathbf{x}},\hat{t}+1) = f_i^*(\hat{\mathbf{x}},\hat{t}) - \left(f_i^{(e,e)}(\hat{\mathbf{x}},\hat{t}) - f_i^{(e,e)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t})\right) - 2\,\frac{\nu}{c^2}\left(f_i^{(e,o)}(\hat{\mathbf{x}},\hat{t}) - f_i^{(e,o)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t})\right). \tag{9}$$

As far as the computation of hydrodynamic quantities is concerned, the first term of the right hand side in (9) can be substituted by $f_i^{(e)}(\hat{\mathbf{x}},\hat{t})$ (they have same hydrodynamic moments). The final model equation can thus be re-formulated in terms of the distribution function f_i as follows

$$f_i(\hat{\mathbf{x}},\hat{t}+1) = f_i^{(e)}(\hat{\mathbf{x}},\hat{t}) - \left(f_i^{(e,e)}(\hat{\mathbf{x}},\hat{t}) - f_i^{(e,e)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t})\right) - 2\,\frac{\nu}{c^2}\left(f_i^{(e,o)}(\hat{\mathbf{x}},\hat{t}) - f_i^{(e,o)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t})\right), \tag{10}$$

or equivalently

$$f_i(\hat{\mathbf{x}},\hat{t}+1) = f_i^{(e)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t}) + \left(1 - 2\,\frac{\nu}{c^2}\right)\left(f_i^{(e,o)}(\hat{\mathbf{x}},\hat{t}) - f_i^{(e,o)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t})\right). \tag{11}$$

In order to fix ideas, let us consider a local equilibrium such that $c^2 = 1/3$ (see Appendix A for details). Moreover, as common in LBM, the kinematic viscosity can be expressed in terms of the relaxation frequency ω of the numerical scheme (see Appendix B for details), namely

$$\nu = \frac{1}{3}\left(\frac{1}{\omega} - \frac{1}{2}\right). \tag{12}$$

Substituting Eq. (12) into Eq. (11) yields the Link-Wise re-formulation of the Artificial Compressibility Method (LW-ACM) for the incompressible isothermal fluid dynamics, expressed by the following system of algebraic equations

$$f_i(\hat{\mathbf{x}},\hat{t}+1) = f_i^{(e)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t}) + 2\left(\frac{\omega-1}{\omega}\right)\left(f_i^{(e,o)}(\hat{\mathbf{x}},\hat{t}) - f_i^{(e,o)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i,\hat{t})\right), \quad i = 0,\ldots,Q-1 \tag{13}$$

where Q is the number of lattice velocities.

A clear advantage of the above scheme is that all quantities appearing in Eqs. (13) and (2) only depends on known (equilibrium) functions at a mesh node and its close neighbors. This introduces a significant simplification in the

treatment of boundary conditions, which can be directly borrowed from finite-difference technology (see e.g. the isothermal Couette flow test case reported in Section 2.5.1).

Similarly to LBM, the algebraic equations (13) can be implemented in three subsequent steps ("pull" formulation), namely pre-combining, streaming and post-combining,

$$f_i^*(\hat{\mathbf{x}} - \hat{\mathbf{v}}_i, \hat{t}) = f_i^{(e)}(\hat{\mathbf{x}} - \hat{\mathbf{v}}_i, \hat{t}) - 2\left(\frac{\omega - 1}{\omega}\right) f_i^{(e,o)}(\hat{\mathbf{x}} - \hat{\mathbf{v}}_i, \hat{t}), \tag{14a}$$

$$f_i^{**}(\hat{\mathbf{x}}, \hat{t} + 1) = f_i^*(\hat{\mathbf{x}} - \hat{\mathbf{v}}_i, \hat{t}), \tag{14b}$$

$$f_i(\hat{\mathbf{x}}, \hat{t} + 1) = f_i^{**}(\hat{\mathbf{x}}, \hat{t} + 1) + 2\left(\frac{\omega - 1}{\omega}\right) f_i^{(e,o)}(\hat{\mathbf{x}}, \hat{t}). \tag{14c}$$

Pre- and post-combining are local processes involving arithmetic operators, whereas streaming alone takes care of data exchange among the nearest neighbors of an arbitrary cell.

The implementation strategy given by Eqs. (14) admits a straightforward inclusion of external forcing, by considering the additional step

$$f_i^{\text{force}}(\hat{\mathbf{x}}, \hat{t} + 1) = f_i(\hat{\mathbf{x}}, \hat{t} + 1) + f_i^{(e,o)}\left(\rho(\hat{\mathbf{x}}, \hat{t}), \mathbf{g}(\hat{\mathbf{x}}, \hat{t})\right), \tag{15}$$

where $\mathbf{g} = (g_x, g_y)^T$ is the external acceleration. The previous correction is local: Similarly to finite-difference schemes, the external forcing is applied to the point where it is supposed to act. The functions $f_i^{(e,o)}$ are used for convenience (they are already known), for ensuring that the force only applies to the momentum equations.

We notice that, the same simple procedure cannot be applied to the lattice Boltzmann method, because a correction to the distribution function may affect the dynamics of the higher order moments as well. Consistent treatment of the forcing typically involves some special (non-trivial) techniques [18]. Details on the effects due to the correction (15), by asymptotic analysis, are reported in the Appendix B. Imposing a given physical acceleration $\bar{\mathbf{g}}$ within a flow, requires the tuning of numerical acceleration $\mathbf{g}$, namely $\mathbf{g} = \epsilon^3\,\bar{\mathbf{g}}$ (in case of diffusive scaling). The same approach can be adopted to implement mass sources in the numerical scheme.

2.2 The main algorithm: Finite difference formulation

Since the right hand side of Eq. (13) only depends on the equilibrium condition, which is in turn a function of the macroscopic quantities, it is possible to provide a finite difference formula, expressed in terms of macroscopic quantities, which is fully equivalent to (13). As commonly done in the finite-difference

literature, we denote by $\{P\}$ the set of computational points surrounding a generic point P (otherwise stated, the generic computational stencil). All the quantities are intended computed at the generic time level $\hat{t}$, while the superscript "+" denotes a quantity at the next time level $\hat{t}+1$. The unknown quantities are given by the velocity components $\mathbf{u} = (u, v)^T$ and the pressure p. Hence the equivalent finite-difference formulas must provide a way to compute u_P^+, v_P^+ and p_P^+ namely

$$u_P^+ = f_u\left(u_{\{P\}}, v_{\{P\}}, p_{\{P\}}; \omega\right), \tag{16a}$$

$$v_P^+ = f_v\left(u_{\{P\}}, v_{\{P\}}, p_{\{P\}}; \omega\right), \tag{16b}$$

$$p_P^+ = f_p\left(u_{\{P\}}, v_{\{P\}}, p_{\{P\}}; \omega\right). \tag{16c}$$

See Appendix D for a complete example based on the D2Q9 lattice [16]. The same finite-difference counterpart can be found for the Lattice Kinetic Scheme (LKS) [11], recovered in case $\omega = 1$, but Eq. (13) is also valid for tunable ω and consequently tunable viscosity ν (in particular, for high Reynolds number flows). Moreover, the same derivation can be done for the FD-LKSν proposed in [17], but Eq. (13) is formulated only along a particular lattice link and hence it can also take advantage of most of LBM technology (which is link-wise).

(1) Availability of two alternative formulations of the same numerical scheme is very convenient. For example, it is possible to commute (even dynamically) between the formulation based on Eq. (13) and that based on Eqs. (16), depending on the best option in dealing with the local boundary conditions.

(2) Similarly to conventional ACM [17, 12, 14], the formulation based on Eqs. (16) can be improved by introducing a semi-implicit step for updating the pressure field. Essentially the step in Eq. (16c) can be substituted by

$$p_P^+ = f_p\left(u_{\{P\}}^+, v_{\{P\}}^+, p_{\{P\}}; \omega\right), \tag{17}$$

using the already updated velocity field.

(3) The finite-difference formulation allows to choose different relaxation parameters ω for different (macroscopic) equations. Let us define ω_u the relaxation frequency used in Eq. (16a) and (16b), while let us define $\omega_p \neq \omega_u$ the relaxation frequency used in Eq. (16c). Consequently two kinematic viscosities follow, namely $\nu = \nu(\omega)$ and $\nu_p = \nu(\omega_p)$, where the function $\nu = \nu(\omega)$ is given by Eq. (12). By introducing these relaxation frequencies in Eqs. (16) and applying Taylor expansion to such novel expressions (see Appendix B for details), the equivalent system of macroscopic equations can be recovered in the continuum limit ($\epsilon \to 0$). The momentum equation involves the kinematic viscosity ν (as previously), while the pseudo-compressibility term in the artificially compressible continuity equation, namely the first term in Eq. (B.16), becomes propor-

tional to ϵ^2/ν_p, namely

$$\frac{\epsilon^2}{6\rho_0\nu_p}\frac{\partial\bar{\rho}}{\partial t} + \nabla\cdot\bar{\mathbf{u}} = O(\epsilon^4/\nu_p^2). \tag{18}$$

For high Reynolds number flows, where $\nu \ll 1$, replacing ν with ν_p in the previous equation helps to improve the accuracy in recovering the incompressible limit of Navier-Stokes equations. In fact ν_p can be chosen larger than ν such that the diverge-free condition for the velocity field can be accurately satisfied even on moderately refined meshes, i.e. meshes with moderately small grid spacing ϵ.

2.3 Physical and lattice units

Table 1
Conversion table from lattice to physical units in case of diffusive scaling, i.e. $\Delta x = \epsilon$ and $\Delta t = \epsilon^2$ with $\epsilon = \Delta x/L = 1/N$ and N the number of mesh points along one axis. Quantities in lattice units are readily computed in the code, but they are mesh-dependent. Corresponding quantities in physical units are mesh-independent and can be computed during post-processing. In the text, in case of ambiguity, quantities in physical units are denoted by over-bar.

Quantity	Lattice units	Physical units
Pressure	$p = 1/3\sum_i f_i$	$\bar{p} = (p - p_0)/\epsilon^2$
Velocity	$\mathbf{u} = \sum_i \hat{\mathbf{v}}_i f_i / \sum_i f_i$	$\bar{\mathbf{u}} = \mathbf{u}/\epsilon$
Force by Eqs. (35, 36)	$\mathcal{F} = \sum_{i\in\mathsf{S}} \hat{\mathbf{v}}_i \mathbf{p}_i$	$\Delta\bar{\mathcal{F}} = (\mathcal{F} - \mathcal{F}_0)/\epsilon^2$
Torque by Eqs. (46, 47)	$\mathcal{T} = \sum_{i\in\mathsf{S}} (\hat{\mathbf{x}} - \hat{\mathbf{x}}_c) \times \mathbf{p}_i$	$\bar{\mathcal{T}} = \mathcal{T}$
Temperature	$T = \sum_i g_i$	$\bar{T} = T$

Let us assume $\Delta x = \epsilon$ and $\Delta t = \epsilon^2$ (diffusive scaling), with $\epsilon = \Delta x/L = 1/N$ and N the number of mesh points. As reported in the Appendix B, asymptotic analysis [10] of (13) and (2) shows that, in the limit of vanishing grid spacing, $\epsilon \ll 1$, the quantities $\bar{p} = (p - p_0)/\epsilon^2 = (p' - p'_0)/U^2$ and $\bar{\mathbf{u}} = \mathbf{u}/\epsilon = \mathbf{u}'/U$ satisfy the incompressible isothermal Navier-Stokes equations, with viscosity given by Eq. (12). The subscript 0 denotes mean value over the whole computational domain. Here, we stress that for a correct use of the proposed algorithm, it is important to consider a proper post-processing of the numerical results. To this respect, we notice that, for instance, the velocity field $\mathbf{u} = \sum_i \hat{\mathbf{v}}_i f_i / \sum_i f_i$ is mesh-dependent: $\mathbf{u} = \mathbf{u}(\epsilon)$, with $\mathbf{u}$ going to zero as the mesh spacing ϵ vanishes. As a result, $\mathbf{u}$ is not the proper choice for describing the velocity field of incompressible flows: To this aim, instead, the quantity $\bar{\mathbf{u}} = \mathbf{u}/\epsilon$ is

adopted due to mesh-independence. Similar considerations apply also to other fields. For consistency with the LBM literature, in this work, the units of quantities directly based on f_i are referred to *lattice units*, while units of the corresponding mesh-independent quantities are termed *physical units*.

For the sake of clarity, the complete set of formulas for converting units of all relevant variables are reported in Table 1. We stress that finite-difference formulation of the proposed method can be carried out directly in physical units, thus avoiding the above post-processing. Nevertheless, here we prefer to keep the above post-processing for consistency with the Lattice Boltzmann community.

Finally, for consistency, accurate solution of INSE requires $\epsilon^2/\nu \ll 1$, which is a criterion valid for the LBM as well (see Appendix B for details and the following discussion about the Minion & Brown flow in Section 2.5.3). On the other hand, for stability, numerical evidences suggest that LW-ACM is stable for $1 \leq \omega < 2$, which corresponds only to half of the stability range for the relaxation frequency of the LBM. The attempts to overcome this limitation by considering also the local equilibrium in the down-wind computational node along the same link identified by the lattice velocity $\hat{\mathbf{v}}_i$, i.e. $\hat{\mathbf{x}} + \hat{\mathbf{v}}_i$, did not succeed so far. The reason may be due to the fact that, as far as only hydrodynamics is concerned, stability requires "up-wind" schemes, which discretize hyperbolic partial differential equations by using differencing biased in the direction of lattice velocities. In particular, the quantity $f_i(\hat{\mathbf{x}}, \cdot)$ streams along the link i-th with the lattice velocity $\hat{\mathbf{v}}_i$ and hence the corresponding up-wind node is $\hat{\mathbf{x}} - \hat{\mathbf{v}}_i$. Remarkably, Eq. (13) determines the dynamics of the quantity $f_i(\hat{\mathbf{x}}, \cdot)$ by only (equilibrium-based) information collected in the generic node $\hat{\mathbf{x}}$ and in the up-wind counterpart $\hat{\mathbf{x}} - \hat{\mathbf{v}}_i$.

In order to understand the consequences of the reduced stability range, let us introduce the Reynolds number $\mathrm{Re} = LU/\nu'$, namely

$$\mathrm{Re} = \frac{\mathrm{Ma}/\mathrm{Kn}}{1/3\ (1/\omega - 1/2)}, \tag{19}$$

where $\mathrm{Kn} \equiv \Delta x$ is the numerical Knudsen number, which is the dimensionless grid spacing, and $\mathrm{Ma} = U/(\Delta x'/\Delta t')$ is the numerical Mach number, which is the parameter controlling the magnitude of the artificial compressibility (see Appendix B for details). The Reynolds number is dictated by the physical problem under investigation, while the right hand side of the previous expression involves only numerical parameters, namely ω, Ma and Kn. In particular, as done also in LBM, for high Reynolds number flows in LW-ACM, it is convenient to chose $\omega \lesssim 2$ and $\mathrm{Ma} \sim \mathrm{Kn}$. On the other hand, the reduced stability range, i.e. $1 \leq \omega < 2$, induces a slight change in the parameters of LW-ACM for low Reynolds number flows. In fact, for soft-matter systems and/or in micro-scale flows solved by LW-ACM, it is convenient to chose $\omega \gtrsim 1$ and

Ma $\ll$ Kn (fine time stepping). However it is important to point out that this drawback is a consequence of the link-wise formulation constraint: No such limitation appears in the original ACM [12, 14].

2.4 Optimized computer implementation

In the following, we discuss a few strategies useful for reducing the computational time, thanks to an optimized implementation of the algorithm (14) and (16). Similarly to LBM, the performance characteristics of single-processor implementations depends on the effect of different data layouts [19] (multi-processor optimization strategies are not considered in the present work, see Ref. [20]).

First of all, the streaming step should not overwrite data required for updating neighboring sites. A usual way to work around the resulting data dependencies owing to the propagation step is the use of two arrays (one for the current and the other for the next time step), and toggling between them [19]. It is also possible (and even more efficient) using a single array, with proper ordering of the sequence of streamed lattice directions, though this may become cumbersome when dealing with wall boundary conditions. The best data layout requires that the distribution functions of the current cell are contiguously located in memory (e.g. by using the first index in Fortran, which addresses consecutive memory locations due to column major order [19]).

Secondly, to reduce the memory traffic, it is important that pre-combining, streaming and post-combining are executed in a single loop and not independently of each other in separate loops or routines, similarly to LBM [19]. This goal can be easily accomplished by reformulating Eqs. (14) in term of f_i^*: In fact, the hydrodynamic moments of f_i^* are not exactly equal to the hydrodynamic quantities, but the former are known functions of the latter. Hence it is convenient to compute directly f_i^*, which are ready to be streamed, and to extract the hydrodynamic quantities from f_i^*. This simplifies the implementation of a single updating loop through all computational sites at each time step.

Finally, it is important to reduce as much as possible the number of floating point operations and memory accesses per updated site. In LBM, the D3Q19 lattice (see Appendix A for details) with BGK collision operator requires about 180-200 floating point operations per cell and time step as well as reading 19 floating point values and writing to 19 different memory locations [19]. Roughly half of the floating point operations and half of the memory accesses are required by the D2Q9 lattice.

Let us considering the D2Q9 lattice and the LBM-style formulation of LW-

ACM, as dictated by Eqs. (14), with the optimization tricks reviewed above. Such an implementation of LW-ACM requires 115 floating point operations (+28% compared to BGK-LBM) and 26 memory accesses (+44%) per cell. However, in the following, only the computational performances of LW-ACM in the FD formulation are further investigated, and its superior capabilities are demonstrated (compared to BGK-LBM). From a computational point

Table 2
Performance test of the link-wise ACM (by FD formulation) vs. BGK-LBM, based on the Minion & Brown flow [22] with $\mathrm{Re} = 10,000$ in the time range $t \in [0, 1]$, solved by a mesh with 512×512 nodes/sites and performing 12,800 iterations ($\mathrm{Ma} = 0.04$). Both codes are serial and use double precision. The considered workstation has Intel® Core™ i7-920 (Bloomfield, 4 physical cores, 8MB L3) with clock rate 2.67GHz (due to TurboMode™ actually running at 2.80 GHz) and 12 GB of DDR3 memory (1333 MHz). The used Fortran compiler is Intel®version 11.1up8 (optimization level option "-O3") and the operative system is Ubuntu Linux i10.04 LTS (64 bit). The million fluid lattice cell updates per second (MLUPS) are reported for both methods.

Elementary stencil	Link-wise ACM by FD formulation	BGK-LBM
# of additions/subtractions	80	70
# of multiplications	60	40
# of floating point operations	140	110
# of actual data (t)	27	9
# of updated data ($t+1$)	3	9
external size of the stencil	3×3×3	3×3×9
MLUPS	29.43	27.28

of view, it may appear that formulas (16) are not suitable for an efficient implementation, since they involve many floating point operations. However, because they are derived from Eq. (13), it is possible to simplify them using (by-hand) common subexpression elimination (CSE) [20]. See Appendix D for a complete example based on the D2Q9 lattice [16]. Moreover, the same implementation tricks discussed above can be properly applied here.

First of all, the memory storage required by link-wise ACM is exactly one third of that of BGK-LBM (only hydrodynamic variables are needed). At each time step, it is enough to go through all computational cells/sites once and this can be done straightforwardly, because updating formulas are already expressed in terms of hydrodynamic variables. Finally, for locating the macroscopic quantities (p, u, v) contiguously in memory, it is possible to collect them in a single array and to use the first index for addressing them. This

leads to an optimized FD-style implementation of Eqs. (16). On the D2Q9 lattice, a comparison between the FD-style implementation of link-wise ACM and BGK-LBM is reported in Table 2. Link-wise ACM requires more floating point operations but less memory accesses than BGK-LBM.

For clarity, let us analyze the updating process at each time. The external size of the stencil of LW-ACM is smaller than the one of LBM (3×3×3 instead of 3×3×9 respectively, where 3×3 is due to the D2Q9 lattice, 3 is the number of hydrodynamic quantities and 9 the number of discrete distribution functions). During the generic updating process, if the cache is large enough to hold 3 "lines" (or 3 planes in 3D) of the computational domain, then updating the hydrodynamic quantities in a cell requires the loading of only the actual values of a further cell in the cache (3 loads). In the worst case, updating the hydrodynamic quantities in a cell requires the loading of the actual values of three additional cells in the cache: This amounts to 9 loads from actual array (3 physical quantities from the current, previous and next "line"). In any case, 3 write-allocate transfers from main memory to cache and 3 stores to get the updated array from cache back to main memory are always required. This leads to 9–15 in total per nodal updates.

On the other hand, BGK-LBM requires 9 loads from the actual array (discrete distribution function), 9 write-allocate transfers and 9 stores to the updated array, leading to 27 memory transfers. For BGK-LBM, there is no reuse of data from cache because every discrete distribution function is only used once. As the number of memory transfers usually affects the performance more than the number of floating point operations, the performance of link-wise ACM is superior than that of BGK-LBM. Some performance data are reported in Table 2. FD-style implementation of link-wise ACM was able to achieve 29.43 million fluid lattice cell updates per second (MLUPS), which is the standard way to measure the performance of LBM implementations [19]. For the previous test, this value is roughly 8% faster than BGK-LBM.

Remark: In our opinion, there is still room for improvement according to the performance model (based on assuming either infinitely fast memory or infinitely fast compute units). For example, the numerical code for solving the 2D lid driven cavity test case achieved 32.3 MLUPS and this was essential for simulating very high Reynolds number flows. Importantly, with an efficient implementation, this algorithm may be one of the few which is compute-bound and not memory-bound. The latter observation is of particular interest for General-Purpose computing on Graphics Processing Units (GPGPU).

For sake of simplicity, numerical tests with simple boundary conditions are discussed first. Here, by simple boundary conditions, we mean either finite-difference boundary conditions (isothermal Couette flow) or periodic (generalized Green-Taylor vortex flow and Minion & Brown flow). More general boundary conditions taking advantage of the LBM technology will be discussed in Section 2.6.

2.5.1 Isothermal Couette flow

In this section, we consider the plane Couette flow where a viscous fluid is confined in a gap between two parallel plates, with the one moving in its own plane with respect to the other. Here, two configurations are simulated by the present LW-ACM method on several meshes: Couette flow without wall injection (referred to as Test 1), and Couette flow with wall injection (referred to as Test 2). In the latter configuration (Test 2) fluid is injected from the bottom wall into the gap and extracted from the top wall with a constant orthogonal velocity $\bar{v}_0$. At the stationary condition, the above configurations admit the following exact solutions:

$$\bar{u}(y) = \frac{1}{2}\hat{u}_L\left(1+\frac{y}{L}\right) + \bar{u}_0\left(1-\frac{y^2}{L^2}\right), \qquad \text{(Test 1)}, \tag{20}$$

$$\bar{u}(y) = \quad \bar{u}_L\left(\frac{\exp(y\,\mathrm{Re}/L)-1}{\exp(\mathrm{Re})-1}\right), \qquad \text{(Test 2)}, \tag{21}$$

where the Reynolds number Re is the main control parameter, L is a characteristic length depending on the considered test and ν is the kinematic viscosity.

For Test 1, L is half the gap height, while $\bar{u}_0$ represents a velocity based on the imposed pressure gradient $\nabla\bar{p}$:

$$\bar{u}_0 = \frac{L^2\nabla\bar{p}}{2\rho\nu},$$

$\bar{u}_L$ is the velocity of the top wall $\bar{u}_L = \bar{u}(y = L)$ and ρ is the density. The bottom wall is assumed stationary: $\bar{u}(y = -L) = 0$.

In case of wall injection (Test 2), L is the gap height, the Reynolds number Re in (20) is defined on the basis of the injection velocity $\bar{v}_0$, namely $\mathrm{Re} = \bar{v}_0 L/\nu$. Velocities at the top and bottom walls are: $\bar{u}(y = L) = \bar{u}_L$ and $\bar{u}(y = 0) = 0$, respectively. In all simulations, no-slip boundary conditions are applied along the wall by simply imposing the local equilibrium with the desired velocity, while periodic boundary conditions are adopted at the inlet and outlet.

Table 3
Convergence analysis for Couette flow without (Test 1) and with (Test 2) wall injection in case of both diffusive and acoustic scaling.

$\Delta t \propto \Delta x^2$ (diffusive scaling)				
			Error $L^2[\bar{u}]$	
$\epsilon \equiv \Delta x$	$\mathrm{Ma} \propto \Delta t/\Delta x$	$\nu \propto \mathrm{Re}^{-1}$	Test 1	Test 2
1/10	3.0×10^{-2}	3.0×10^{-2}	1.74×10^{-3}	4.59×10^{-4}
1/20	1.5×10^{-2}	3.0×10^{-2}	4.49×10^{-4}	1.21×10^{-4}
1/40	7.5×10^{-3}	3.0×10^{-2}	1.20×10^{-4}	3.11×10^{-5}
$\Delta t \propto \Delta x$ (acoustic scaling)				
			Error $L^2[\bar{u}]$	
$\epsilon \equiv \Delta x$	$\mathrm{Ma} \propto \Delta t/\Delta x = \mathrm{const.}$	$\nu \propto \mathrm{Re}^{-1}$	Test 1	Test 2
1/10	3.0×10^{-1}	3.0×10^{-3}	4.27×10^{-2}	1.05×10^{-2}
1/20	3.0×10^{-1}	1.5×10^{-3}	2.76×10^{-2}	5.19×10^{-3}
1/40	3.0×10^{-1}	7.5×10^{-4}	1.54×10^{-2}	2.66×10^{-3}

First of all, diffusive scaling is considered: This strategy consists in scaling the velocity field on different meshes, keeping fixed the relaxation frequency (see Appendix B for details). This scaling ensures second order convergence in the accuracy, as reported in upper part of Table 3, where the L^2 norm of deviation of numerical results from exact solution (20) is shown. In addition, acoustic scaling is considered. This strategy consists in tuning the relaxation frequency on different meshes in order to keep constant the computed velocity field (see Appendix B for details). This scaling ensures first order convergence in the accuracy, as reported in the lower part of Table 3.

2.5.2 Generalized Green-Taylor vortex flow

In this section, some numerical results of the Taylor-Green vortex flow are reported. The latter problem is widely employed as a benchmark for various incompressible Navier-Stokes solvers, owing to the existence of a simple analytical solution. The original problem is characterized by the exponential decay in time due to viscous dissipation. However, here the original problem is modified such that it becomes periodic in time by the introduction of a proper external acceleration $\bar{\mathbf{g}} = (\bar{g}_x, \bar{g}_y)^T$ to Eq. (B.12), where

$$\begin{aligned} \bar{g}_x(t,x,y) &= \sin(x - \bar{u}_0 t)\cos(y - \bar{v}_0 t)(2\nu\cos t - \sin t), \\ \bar{g}_y(t,x,y) &= -\cos(x - \bar{u}_0 t)\sin(y - \bar{v}_0 t)(2\nu\cos t - \sin t), \end{aligned} \tag{22}$$

with $\bar{u}_0$ and $\bar{v}_0$ being constants aiming at preventing that the advection term balances with the pressure gradient. The modified problem admits the following analytical solution

$$\begin{aligned}
\bar{u}(t,x,y) &= \bar{u}_0 + \sin(x-\bar{u}_0 t)\cos(y-\bar{v}_0 t)\cos t, \\
\bar{v}(t,x,y) &= \bar{v}_0 - \cos(x-\bar{u}_0 t)\sin(y-\bar{v}_0 t)\cos t, \\
\bar{p}(t,x,y) &= \frac{1}{4}[\cos 2(x-\bar{u}_0 t) + \cos 2(y-\bar{v}_0 t)]\cos^2 t.
\end{aligned} \tag{23}$$

We solve the above modified problem numerically in the domain $\Omega = [0 \leq x \leq 2\pi] \times [0 \leq y \leq 2\pi]$ using periodic boundary condition, and $(\bar{u}_0, \bar{v}_0) = (0.3, 0.6)$. The kinematic viscosity is $\nu = 0.1$. Diffusive scaling is adopted for all simulations, namely $\mathrm{Ma} = 2\pi\,\mathrm{Kn}$ or equivalently $\Delta t = \Delta x^2$ (see Appendix B for details).

The L^1 error data are reported in Table 4 at $t = 60$ for link-wise ACM, link-wise ACM with semi-implicit formulation (see the end of Section 2.1 where this formulation is presented), standard (second-order) ACM and multiple-relaxation-time (MRT) LBM. The standard ACM is described in Ref. [12]. In the MRT-LBM [13, 21], the consistent treatment of forcing is based on Ref. [18]. The time step employed in the LBM computation is the same as the one of ACM, i.e. $\mathrm{Ma} = 2\pi\,\mathrm{Kn}$ or equivalently $\Delta t = \epsilon^2$, and the tuning parameters of MRT are $s_1 = s_4 = s_6 (= {\tau_\rho}^{-1} = {\tau_j}^{-1}) = 0$, $s_2 (= {\tau_e}^{-1}) = 1.63$, $s_3 (= {\tau_\epsilon}^{-1}) = 1.14$, $s_5 = s_7 (= {\tau_q}^{-1}) = 1.92$ (see Refs. [21, 18]). All methods show nearly second-order convergence. Link-wise ACM shows larger numerical errors than ACM and MRT-LBM. However it must be stressed that the implementation of forcing in link-wise ACM is straightforward, while the consistent treatment of forcing in LBM is much more complicated [18]. Moreover in link-wise ACM, spatial operators (gradient and Laplace operator) do not need to be discretized individually (unlike ACM) but it is enough to discretize along the generic lattice direction: This makes much easier the treatment of three-dimensional cases.

2.5.3 Minion & Brown flow

LW-ACM and LBM are characterized by different values of artificial compressibility, leading to a different robustness with respect to under-resolved simulations. By referring to the pseudo-continuity equation for LW-ACM (see Eq. (B.16) in the Appendix B), it follows that accurate solution of divergence-free condition for the velocity field requires: $\epsilon^2/\nu \ll 1$. This is even a more severe condition than the one of LBM, in case of high Reynolds number flows. In the following, we further explore this issue, by means of the Minion & Brown flow.

Table 4
The L^1 norm of the error versus $\epsilon \equiv \Delta x \equiv \mathrm{Ma}/2\pi$ at $t = 60$ in the problem of the generalized Taylor-Green vortex problem for $\nu = 0.1$.

Link-wise ACM			
$\epsilon \equiv \Delta x$	Error $L^1[\bar{u}]$	Error $L^1[\bar{v}]$	Error $L^1[\bar{p}]$
$\pi/16$	1.69262×10^{-2}	1.96356×10^{-2}	3.45590×10^{-2}
$\pi/32$	4.38763×10^{-3}	4.70793×10^{-3}	7.91104×10^{-3}
$\pi/64$	1.41952×10^{-3}	1.50330×10^{-3}	2.21013×10^{-3}
Link-wise ACM (semi-implicit, see Section 2.1)			
$\epsilon \equiv \Delta x$	Error $L^1[\bar{u}]$	Error $L^1[\bar{v}]$	Error $L^1[\bar{p}]$
$\pi/16$	1.70353×10^{-2}	2.02203×10^{-2}	3.09165×10^{-2}
$\pi/32$	4.41348×10^{-3}	4.70412×10^{-3}	6.11065×10^{-3}
$\pi/64$	1.41906×10^{-3}	1.50227×10^{-3}	1.73786×10^{-3}
ACM			
$\epsilon \equiv \Delta x$	Error $L^1[\bar{u}]$	Error $L^1[\bar{v}]$	Error $L^1[\bar{p}]$
$\pi/16$	8.03750×10^{-3}	1.00313×10^{-2}	9.70682×10^{-3}
$\pi/32$	1.92844×10^{-3}	2.47186×10^{-3}	1.93893×10^{-3}
$\pi/64$	5.64682×10^{-4}	6.61285×10^{-4}	4.34911×10^{-4}
MRT-LBM			
$\epsilon \equiv \Delta x$	Error $L^1[\bar{u}]$	Error $L^1[\bar{v}]$	Error $L^1[\bar{p}]$
$\pi/16$	7.67964×10^{-3}	8.90307×10^{-3}	1.67526×10^{-2}
$\pi/32$	1.84928×10^{-3}	2.17164×10^{-3}	3.63345×10^{-3}
$\pi/64$	6.12810×10^{-4}	6.81060×10^{-4}	1.11633×10^{-3}

Minion & Brown [22] studied the performance of various numerical schemes for under-resolved simulations of the 2D incompressible NavierStokes equations. The relevance of this flow for testing robustness and accuracy of LBM schemes was first pointed out by Dellar [23]. Minion & Brown considered initial conditions corresponding to the perturbed shear layer

$$\bar{u}(t,x,y) = \begin{cases} \tanh(k(y-1/4)), & y \le 1/2, \\ \tanh(k(3/4-y)), & y > 1/2, \end{cases} \tag{24}$$

$$\bar{v}(t,x,y) = \delta \sin(2\pi(x+1/4)), \tag{25}$$

in the periodic domain $\Omega = [0 \leq x \leq 1] \times [0 \leq y \leq 1]$. The parameter k controls the shear layer width, while δ the magnitude of the initial perturbation. The shear layer is expected to roll up due to Kelvin-Helmholtz instability. With $k = 80$, $\delta = 0.05$, and Reynolds number $\mathrm{Re} = 1/\nu = 10000$, the thinning shear layer between the two rolling up vortices becomes under-resolved on a 128×128 grid. Minion & Brown [22] found that conventional numerical schemes using centered differences became unstable for this under-resolved flow, whereas "robust" or "upwind" schemes that actively suppress grid-scale oscillations all produce two spurious secondary vortices at the thinnest points of the two shear layers at $t = 1$. Dellar [23] found that, even though it is stable on the previous mesh, two spurious vortices are generated by the BGK LBM scheme based on the D2Q9 lattice and with unmodified bulk viscosity. The same author proposed a way to increase the bulk viscosity for overcoming this problem, and verified that the same result can be achieved by using MRT-LBM. Along the same idea, alternative formulations for modifying the dissipation of the LBM schemes have been proposed [24, 25].

Link-wise ACM is stable for the previous test, but the velocity field is not accurate enough due to numerical viscosity. For $\epsilon = 1/128$ and $\nu = 1/10000$, indeed, the term that multiply the pressure time derivative in the pseudo-continuity equation (Eq. (B.16) in the Appendix B), namely ϵ^2/ν, reaches the value of 0.61, thus preventing an accurate fulfillment of the diverge-free condition for the velocity field. In the following, two improvements are worked out for circumventing the above problem.

First, we apply the strategy discussed at the end of Section 2.2, introducing a numerical fictitious viscosity $\nu_p = 170\,\nu$, with the corresponding relaxation frequency ω_p computed by means of (12). The relaxation frequency ω_p is used in the macroscopic updating rule (16c). Hence, the pseudo-compressibility term in the continuity equation, namely the first term in Eq. (B.16), becomes proportional to $\epsilon^2/\nu_p = 0.0036 \ll \epsilon^2/\nu$ and this improves the quality of the solution. The vorticity at $t = 1$ is reported in Figure 1, where the velocity field looks much better, but the two spurious secondary vortices at the thinnest points of the two shear layers are still present (similarly to BGK-LBM).

The second improvement follows the idea to increase the bulk viscosity ξ, as suggested by Dellar [15]. As discussed in Appendix C, instead of the standard $f_i^{(e)}$ (see Eq. (A.1) in the Appendix A) in Eq. (13), we consider a modified set of functions, namely $f_i^{(e*)} = f_i^{(e*)}(\gamma)$, where γ is a free parameter related to the bulk viscosity ξ: $\xi = 2\rho_0\nu\,(1 + 2\,\gamma)$. This strategy enables to increase the bulk viscosity and represents a valuable example, showing how to use moments of the updated distribution function for the local computation of derivatives (see the Appendix C for details). The above argument is a further confirmation that the present LW-ACM can easily incorporate technologies originally developed for LBM.

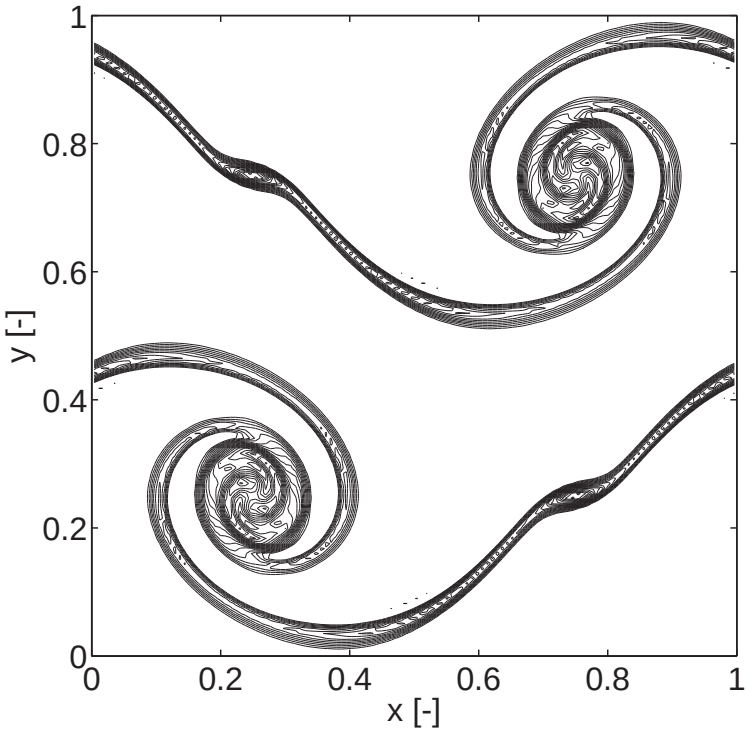

Fig. 1. Contours of vorticity at $t = 1$ from the link-wise ACM with the first improvement ($\nu_p = 170\,\nu$, see Section 2.1) on a 128×128 grid with $\mathrm{Ma} = 0.04$ and $\mathrm{Re} = 10000$. See also Fig. 8 in [22].

As a concluding remark, it is worth to point out that, even though LW-ACM can solve this under-resolved test case by means of previously discussed improvements, the average and peak values of the velocity field divergence remain slightly larger than those computed by MRT-LBM. For sake of completeness, optimized ACM [14] yields much smaller values of velocity field divergence than those of MRT-LBM. However, it is important to keep in mind that the Minion & Brown flow is a very severe test due to the small initial perturbation δ, which realizes a very sharp boundary layer. Hence this test is a multi-scale problem and some of the regularity assumptions used in deriving the numerical schemes may not hold completely.

2.6 Link-wise wall boundary conditions

The link-wise formulation of the proposed method offers significant advantages when dealing with wall boundary conditions. First, let us consider simple structured boundaries, where walls are aligned along the mesh (for general cases, the reader can refer to the next section). In particular, let us suppose that walls are located halfway between two consecutive nodes in an ideal step-wise geometry. Instead of using Eqs. (14), it proves convenient focusing on the quantities f_i^* (after pre-combining) streaming out of a single point $\mathbf{x}$ ("push"

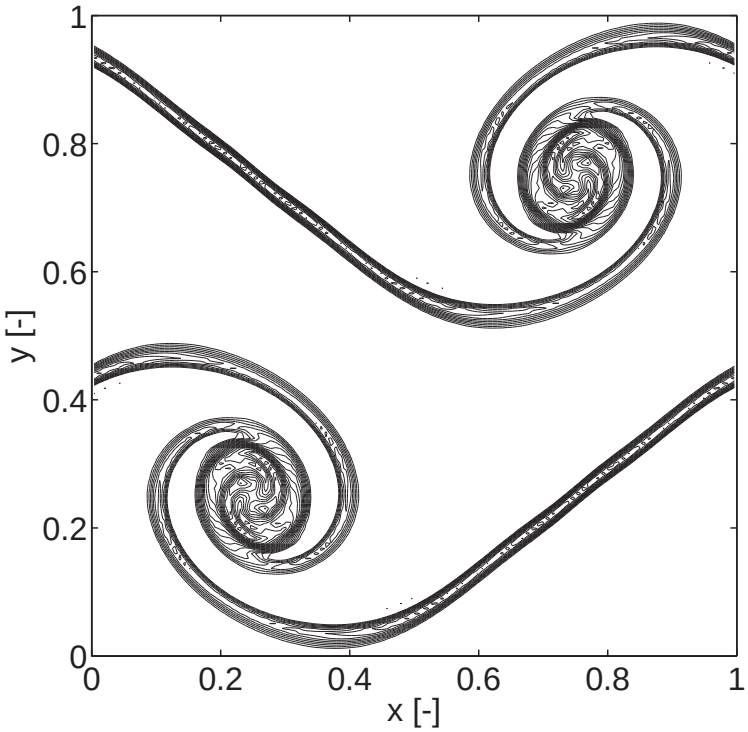

Fig. 2. Contours of vorticity at $t = 1$ from the link-wise ACM with the both suggested improvements ($\nu_p = 170\,\nu$, see Section 2.1, and $\gamma = 0.4$, see Section C) on a 128×128 grid with $\mathrm{Ma} = 0.04$ and $\mathrm{Re} = 10000$.

formulation)

$$f_i^*(\hat{\mathbf{x}}, \hat{t}) = f_i^{(e)}(\hat{\mathbf{x}}, \hat{t}) - 2\left(\frac{\omega - 1}{\omega}\right) f_i^{(e,o)}(\hat{\mathbf{x}}, \hat{t}). \tag{26}$$

Let us suppose that $\hat{\mathbf{x}}$ is a fluid node close to a complex wall boundary at rest such that $\hat{\mathbf{x}}+\hat{\mathbf{v}}_i$ is a wall node. In an ideal step-wise geometry, the wall location is assumed halfway between $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}} + \hat{\mathbf{v}}_i$. Hence, during the streaming step, the information stored in the discrete distribution function pointing towards the wall are reflected along the same link by the wall (according to the popular *bounce-back* rule), namely

$$f_{BB(i)}^{**}(\hat{\mathbf{x}}, \hat{t} + 1) = f_i^*(\hat{\mathbf{x}}, \hat{t}), \tag{27}$$

where $BB(i)$ is the bounce-back operator giving the lattice link opposite to i-th. Finally the post-combining step can be performed in the usual way, namely

$$f_{BB(i)}(\hat{\mathbf{x}}, \hat{t} + 1) = f_{BB(i)}^{**}(\hat{\mathbf{x}}, \hat{t} + 1) + 2\left(\frac{\omega - 1}{\omega}\right) f_{BB(i)}^{(e,o)}(\hat{\mathbf{x}}, \hat{t}). \tag{28}$$

Considering that $\hat{\mathbf{v}}_i = -\hat{\mathbf{v}}_{BB(i)}$ and $f_i^{(e,o)} = -f_{BB(i)}^{(e,o)}$, the combination of previous steps yields

$$f_{BB(i)}(\hat{\mathbf{x}}, \hat{t} + 1) = f_i^{(e)}(\hat{\mathbf{x}}, \hat{t}) + \left(2 - \frac{2}{\omega}\right) 2 f_{BB(i)}^{(e,o)}(\hat{\mathbf{x}}, \hat{t}). \tag{29}$$

In case of moving complex boundary with velocity $\mathbf{u}_w$, the procedure reported in [26] (here reformulated in terms of $f^{(e,o)}_{BB(i)}$) suggests the inclusion of the additional term

$$\delta f_{BB(i)}(\rho_0, \mathbf{u}_w) = 2 f^{(e,o)}_{BB(i)}(\rho_0, \mathbf{u}_w), \tag{30}$$

where $f^{(e,o)}_{BB(i)}$ is given by Eq. (2) and ρ_0 is the averaged density over the whole computational domain (see Appendix B for details). For sake of simplicity, let us consider the diffusive scaling: $\Delta x = \epsilon \ll 1$, $\Delta t = \epsilon^2$. Hence, the incompressible limit implies $\rho(\hat{\mathbf{x}}) = \rho_0 + O(\epsilon^2)$. Moreover, the point $\hat{\mathbf{x}}$ is only $\epsilon/2$ away from the moving wall. Combining the previous conditions, the following approximation holds $\mathbf{u}(\hat{\mathbf{x}}) = \mathbf{u}_w + O(\epsilon^2)$ and

$$f^{(e,o)}_{BB(i)}(\hat{\mathbf{x}}, t) = f^{(e,o)}_{BB(i)}(\rho_0, \mathbf{u}_w) + O(\epsilon^2),$$

meaning that the rightmost term of Eq. (29) produces a similar effect to the correction (30). Hence, the suggested correction for LBM will be multiplied by a scaling factor (complement to one of the factor multiplying the last term in Eq. (29)) in link-wise ACM, namely

$$f^{w}_{BB(i)}(\hat{\mathbf{x}}, \hat{t}+1) = f_{BB(i)}(\hat{\mathbf{x}}, \hat{t}+1) + \left(\frac{2}{\omega} - 1\right) \delta f_{BB(i)}(\rho_0, \mathbf{u}_w), \tag{31}$$

where $f^{w}_{BB(i)}$ is the proper boundary condition in case of moving boundary.

Link-wise formulation is also very useful in computing hydrodynamical forces acting on bodies. A popular approach in LBM literature is based on the so-called momentum exchange algorithm (MEA), originally proposed in [27] and lately improved in [28] (see [29] for a complete discussion). In LBM, at every time step, the momentum

$$\mathbf{p}_i = \hat{\mathbf{v}}_i f_i^{\text{post}}(\hat{\mathbf{x}}, \hat{t}) - \hat{\mathbf{v}}_{BB(i)} f_{BB(i)}(\hat{\mathbf{x}}, \hat{t}+1) = \hat{\mathbf{v}}_i \left[f_i^{\text{post}}(\hat{\mathbf{x}}, \hat{t}) + f_{BB(i)}(\hat{\mathbf{x}}, \hat{t}+1) \right], \tag{32}$$

is transferred from the fluid to the solid body (f_i^{post} is the post-collisional distribution function). For sake of simplicity, here we restrict ourselves on bodies at rest. In link-wise ACM, the quantity f_i^{post} is purposely defined in order to get rid of the last term in the post-combining step, namely

$$f_i^{\text{post}}(\hat{\mathbf{x}}, \hat{t}) = f_i^{*}(\hat{\mathbf{x}}, \hat{t}) - 2\left(\frac{\omega - 1}{\omega}\right) f^{(e,o)}_{BB(i)}(\hat{\mathbf{x}}, \hat{t}). \tag{33}$$

Introducing the previous definition in Eq. (32) yields

$$\mathbf{p}_i = \hat{\mathbf{v}}_i \left[f_i^{*}(\hat{\mathbf{x}}, \hat{t}) + f^{**}_{BB(i)}(\hat{\mathbf{x}}, \hat{t}+1) \right]. \tag{34}$$

The previous expression for link-wise ACM is general. In case of step-wise geometries (as those considered in this section), the expression (27) holds and Eq. (34) can be recast as: $\mathbf{p}_i = 2\hat{\mathbf{v}}_i f_i^{*}(\hat{\mathbf{x}}, \hat{t})$. The force exerted on a body is

computed by a summation of the contributions (34) over all the boundary links surrounding its surface:

$$\mathcal{F} = \sum_{i \in \mathsf{S}} \mathbf{p}_i, \tag{35}$$

where S is the set of links starting from all surrounding nodes intersecting the body.

Conversion of the force (35) from lattice units to physical units requires substraction of the hydrostatic component $\mathcal{F}_0$ generated by the averaged density field ρ_0 (see also the Appendix B for details). Computing the hydrostatic force on partial boundaries of a body by Eq. (35 and 34) can be accomplished after exclusion of the velocity-dependent components of f_i^* and $f_{BB(i)}^{**}$. If the force is computed on the entire body surface, the hydrostatic force is null. The remaining quantity $\mathcal{F} - \mathcal{F}_0$ scales as the second order tensor: in case of diffusive scaling $\mathcal{F} - \mathcal{F}_0 \sim \epsilon^2$. However the number of points in the set S increases proportionally to $1/\epsilon$. Consequently the following scaling holds (see also Table 1)

$$\Delta\bar{\mathcal{F}} = (\mathcal{F} - \mathcal{F}_0)/\epsilon^2 = \left(-\mathcal{F}_0/\epsilon + \sum_{i \in \mathsf{S}(1/\epsilon)} \mathbf{p}_i \right) / \epsilon. \tag{36}$$

2.6.1 2D lid driven cavity flow

In this section, the effective enhanced stability of the present LW-ACM method is tested by means of the classical two-dimensional (2D) lid driven cavity problem (see also [30, 31, 32]). Such a benchmark has been considered owing to a singularity of the pressure in the lid corners, which makes it a severe test for robustness of numerical schemes, especially starting from moderately high Reynolds numbers. In all simulations, we consider a square domain $(x, y) \in [0, L] \times [0, L]$, with $L = 1$. Such a domain is discretized by uniform collocated grid with $N \times N$ points. The boundaries are located half-cell away from the computational nodes. Let us denote $\hat{\mathbf{x}}_b$ the generic boundary computational node. In all inner nodes ($\hat{\mathbf{x}} \neq \hat{\mathbf{x}}_b$), Eq. (13) holds for any lattice velocity $\hat{\mathbf{v}}_i$. In an arbitrary boundary node $\hat{\mathbf{x}}_b$, Eq. (13) holds for any lattice velocity $\hat{\mathbf{v}}_i$ such that $\hat{\mathbf{x}}_b + \hat{\mathbf{v}}_i$ is still a computational node. In case $\hat{\mathbf{x}}_b + \hat{\mathbf{v}}_i$ falls out of the computational grid, the boundary condition (31) is applied, with $\mathbf{u}_w$ being the boundary velocity (imposed half-cell away from the boundary node $\hat{\mathbf{x}}_b$). In the following numerical simulations, $\mathbf{u}_w = (u_L, 0)^T$ at the lid wall, where u_L is the lid velocity, and $\mathbf{u}_w = \mathbf{0}$ for all the other walls. At the lid corners, the lid velocity is imposed, while for other corners the boundary conditions (31) are adopted.

In Figs. 3 and 4, numerical results corresponding to several grids and methods are reported for Reynolds number $\mathrm{Re} = u_L L/\nu = 5000$. In this case, it is

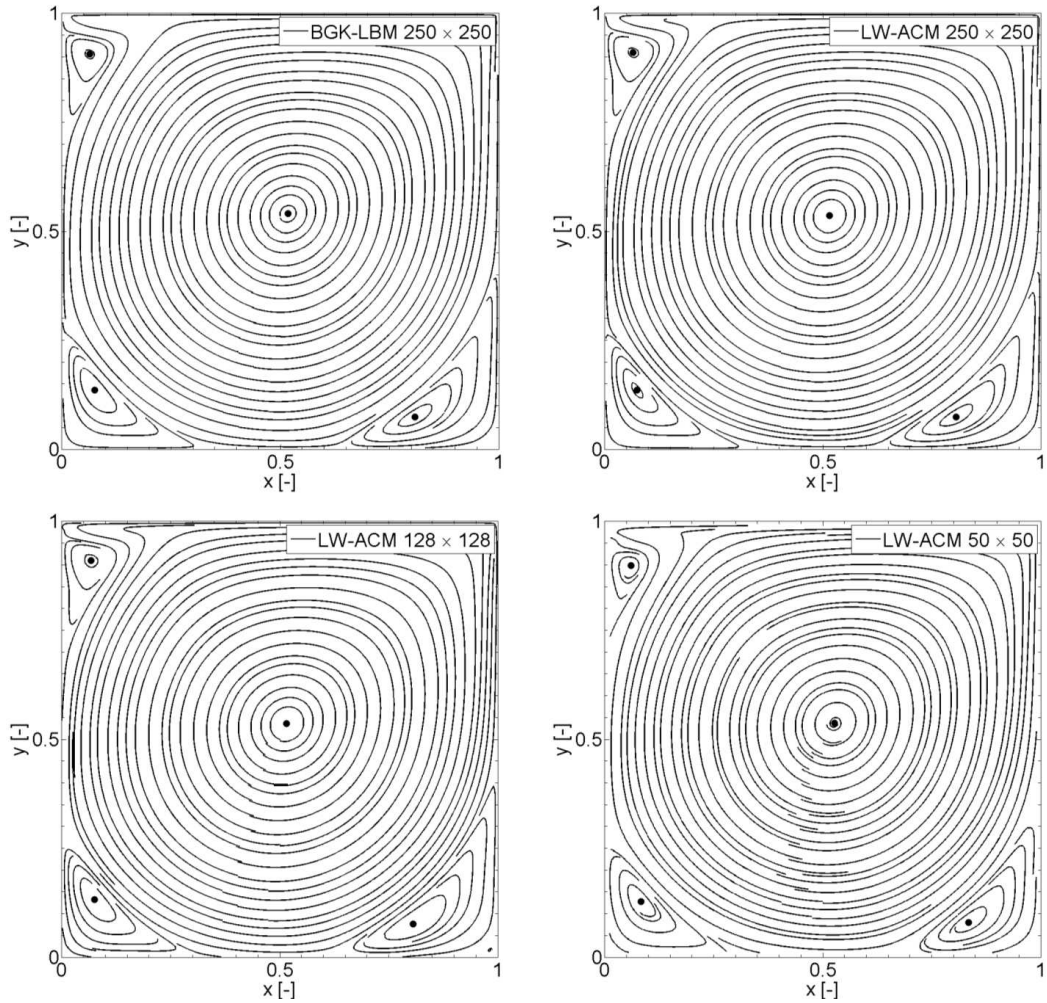

Fig. 3. Streamlines for the lid driven cavity flow with Re = 5000. Different numerical methods and different meshes are compared. The location of the four minima of the stream-function is denoted by filled circles.

known that the flow is characterized by four main vortexes, whose actual *centers* can be found by searching for local extrema of the stream function ψ defined as:

$$u = \frac{\partial \psi}{\partial y}, \qquad v = -\frac{\partial \psi}{\partial x}, \tag{37}$$

with u and v being the horizontal and vertical component of the velocity field, respectively.

For standard Lattice Boltzmann method [16] with BGK collisional operator and $D2Q9$ lattice, the coarsest grid which ensures numerical stability was found to be 250×250. On the other hand, the present LW-ACM method can be safely adopted with 125×125 grid, and reasonably accurate results are found as reported in Figs. 3 and 4. In addition, LW-ACM shows stability even with 50×50 grid (1/25 the total number of nodes needed by BGK-LBM method), where LW-ACM is still able to describe the main features of the 2D

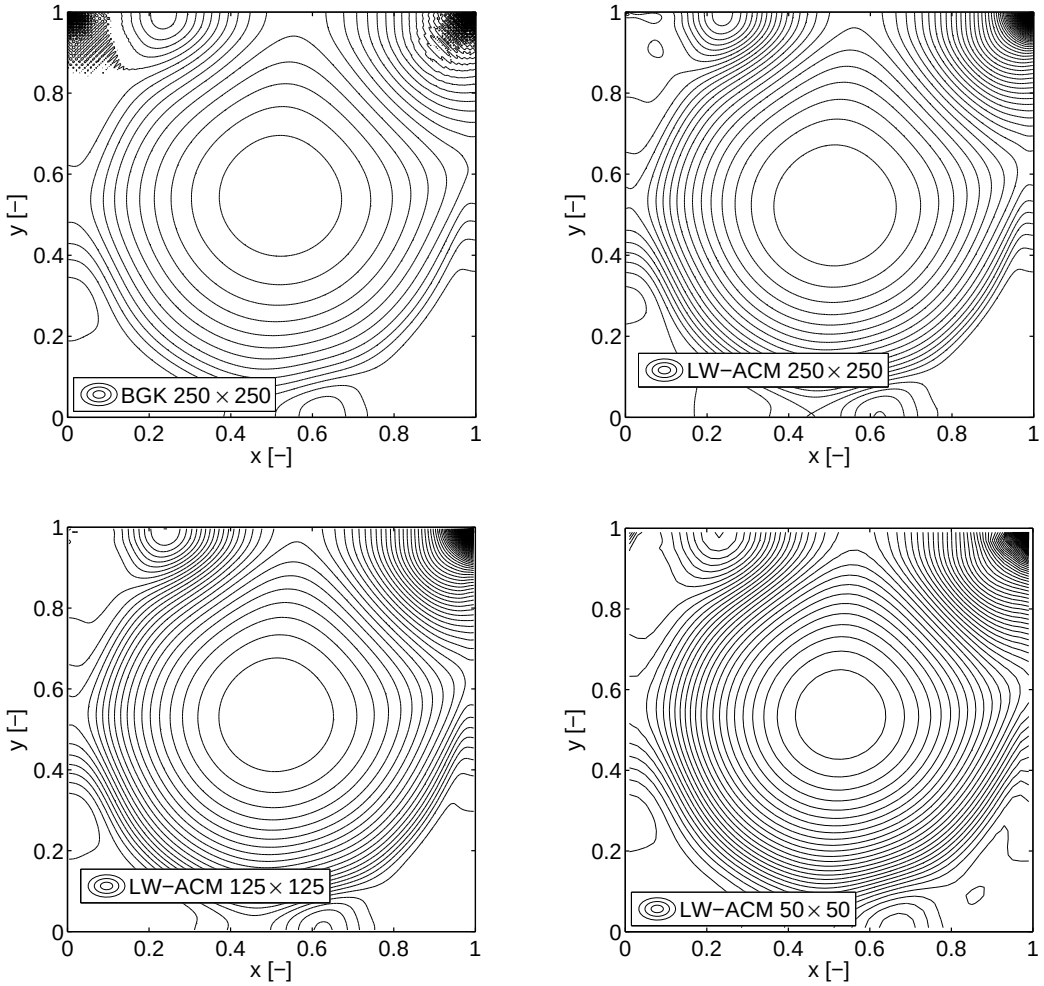

Fig. 4. Pressure contours for the lid driven cavity flow with Re = 5000. Different numerical methods and different meshes are compared.

cavity flow at Re = 5000.

In Fig. 4 the pressure contours for the same test are reported. As visible in the upper-left part of Fig. 4 the BGK solution shows some *checkerboard* pressure distribution at the top corners of the cavity. The mesh resolution is still enough to overcome the checkerboard instability mechanism: however this comes at the price of a very large computational domain (larger than 250×250). On the other hand, no such a problem was noticed with LW-ACM, even for quite coarse grids (down to 50×50). The absence of spurious oscillations in the numerical solutions by the artificial compressibility method (ACM) for this test case has been already pointed out [12]. Hence, the previous numerical evidences demonstrate that also the present link-wise formulation of ACM inherits the same feature.

It is worth stressing that numerical stability on coarse grids, yet with poor

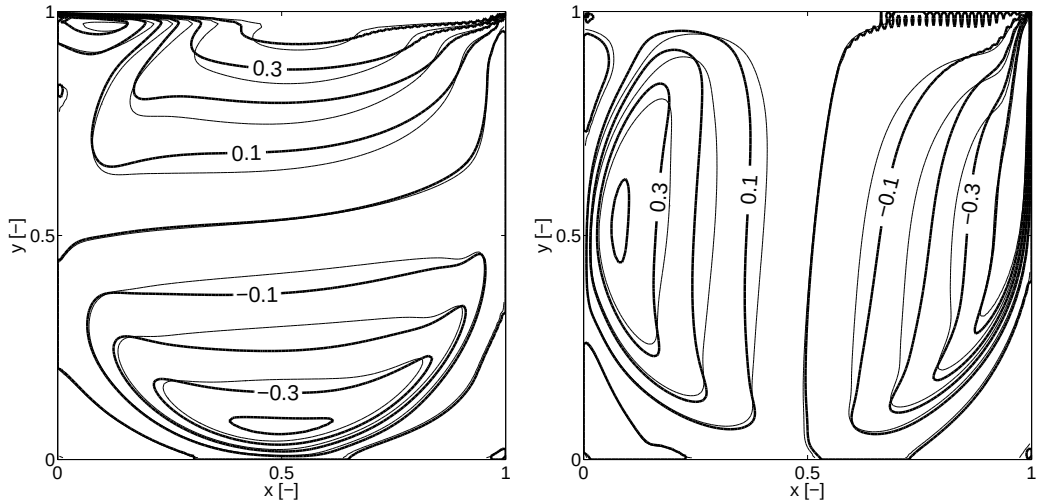

Fig. 5. Comparison of the velocity field for the lid driven cavity flow at Re = 5000 with 128×128 grids: horizontal component u and vertical component v are reported on the left and right hand side, respectively. Thin and bold lines denote the present LW-ACM (with $\nu_p = \nu$) and optimized ACM solution [12], respectively. The latter method is based on a second order accurate scheme in time, and fourth order accurate scheme in space (bulk fluid).

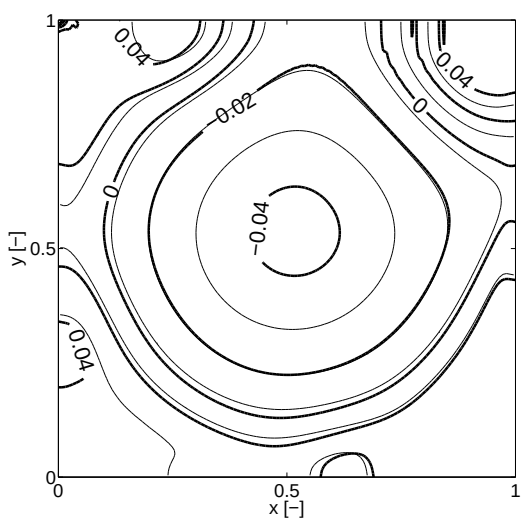

Fig. 6. Comparison of the pressure field for the lid driven cavity flow at Re = 5000 with 128×128 grids. Thin and bold lines denote the present LW-ACM (with $\nu_p = \nu$) and optimized ACM solution [12], respectively. The latter method is based on a second order accurate scheme in time, and fourth order accurate scheme in space (bulk fluid).

accuracy, is a highly desirable feature in several engineering problems, e.g. where a loose grid resolution of details of no interest (for the overall flow phenomena) should not prevent global convergence of the code.

Clearly the numerical schemes should be compared in terms of the actual

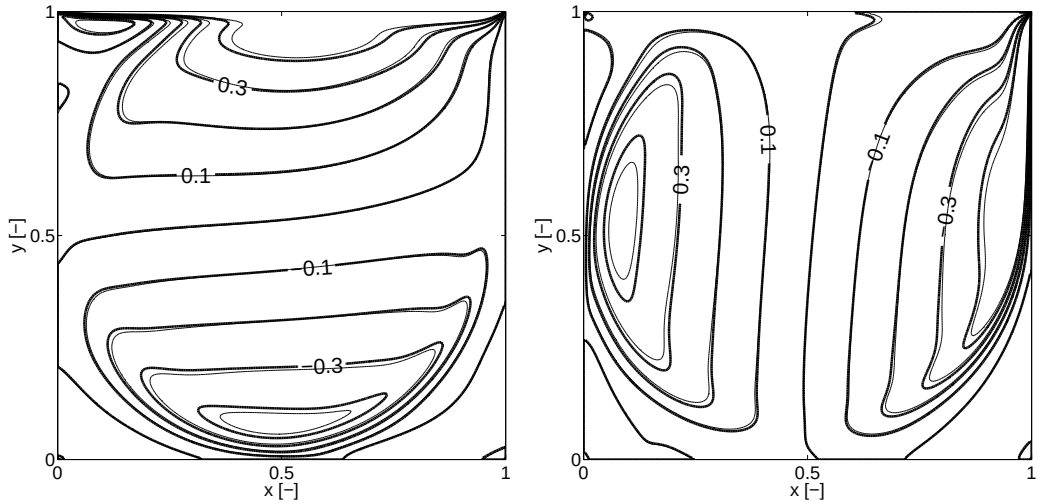

Fig. 7. Comparison of the velocity field for the lid driven cavity flow at Re = 5000 with 256×256 grids: horizontal component u and vertical component v are reported on the left and right hand side, respectively. Thin and bold lines denote the present LW-ACM (with $\nu_p = \nu$) and ACM solution [12], respectively. The latter method is based on a second order accurate scheme in time, and fourth order accurate scheme in space.

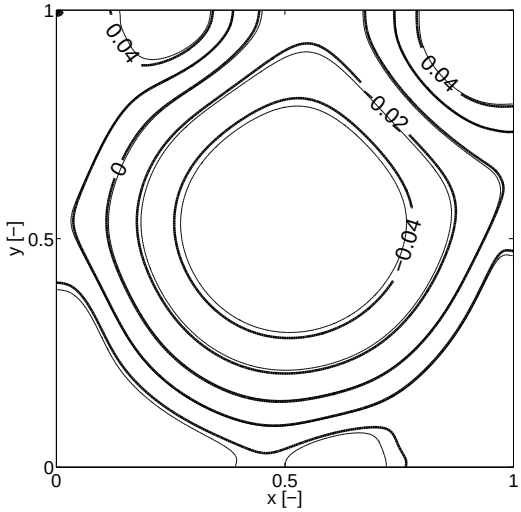

Fig. 8. Comparison of the pressure field for the lid driven cavity flow at Re = 5000 with 256×256 grids. Thin and bold lines denote the present LW-ACM (with $\nu_p = \nu$) and ACM solution [12], respectively. The latter method is based on a second order accurate scheme in time, and fourth order accurate scheme in space.

accuracy as well. From the very beginning, it is worth to point out that all considered methods (i.e. BGK-LBM, MRT-LBM, ACM, LW-ACM) are based on artificial compressibility and even steady state solutions depend on the numerical Mach number (in particular, the pressure gradients depend on the Mach number, as well as the number of time steps). In particular, reducing the numerical Mach number improves the quality of the results. See Appendix

B for details. Hence a fair comparison among different methods requires using the same numerical Mach number.

The flow fields of a 2D lid-driven cavity problem with Re = 5000 and 128×128 grid, as predicted by the optimized ACM method [12] and the present LW-ACM (with Mach number Ma = 0.2 and $\nu_p = \nu$), have been compared. Here optimized ACM means that (a) high wave numbers are damped for the suppression of the checkerboard instability and (b) the Richardson extrapolation in the Mach number (except around top singular corners) is employed [12]. As reported in Figs. 5 and 6, LW-ACM shows both a smoother and more accurate behavior (see also Table 5), beside a remarkably simpler implementation.

Moreover, in Figs. 7 and 8, we report the flow fields of a 2D lid-driven cavity problem with Re = 5000 and 256×256 grid as predicted by the optimized ACM [12] and the present LW-ACM (with Mach number Ma = 0.2 and $\nu_p = \nu$), where a small mismatch between the two solutions is observed. This time the optimized ACM method is able to reproduce the reference results [30] with excellent accuracy (despite the use of a much coarser grid: 256^2 vs. 2048^2), although minor pressure oscillations are still visible at the lid corners and this affects, e.g., the prediction of entrophy (see Eq. (38) and Table 5). Differences between ACM and LW-ACM are mainly due to: 1)different accuracy of the two schemes (second and first order accuracy in time for ACM and the present LW-ACM respectively, whereas fourth and second order accuracy in space for ACM and the present LW-ACM respectively), and 2) slightly different treatment of boundaries. ACM imposes boundary conditions on the computational nodes, while in LW-ACM, analogously to LBM, the wall boundary conditions are imposed half cell away from the computational node. This allows one to avoid singularities, which appear in the top corners for this test case. Despite all this, it is fair to say that the agreement between the above two solutions is quite good.

In Figs. 9 and 10, the flow field computed by the LW-ACM with Re = 5000 and 256×256 grid is compared to a reference solution from the literature [30], where a state of the art code based on finite differences is used with a remarkably fine grid (2048×2048). Despite a significant disparity in the number of computational nodes (LW-ACM makes use of 1/64 the nodes adopted for the reference solution), an excellent agreement is found. It is worth stressing that, the above problem was also simulated by the multiple relaxation time lattice Boltzmann method (MRT-LBM) with 256×256 grid. Comparison between MRT-LBM and the reference solution is also very good, however the issue of spurious pressure oscillations in the upper-left corner of the cavity could not be avoided. Further comparisons between the LW-ACM and other methods are proposed in Table 5. Here, coordinates of the primary vortex center (x_p, y_p), coordinates of the lower-right vortex center (x_{lr}, y_{lr}), total kinetic energy E and enstrophy Z are reported for several schemes and grids, where the latter

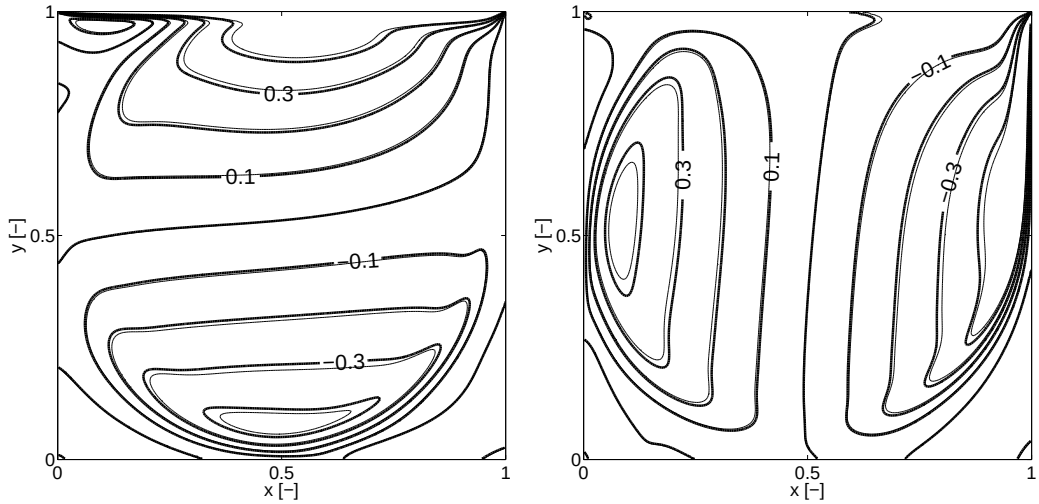

Fig. 9. Comparison of the velocity field for the lid driven cavity flow at Re = 5000: horizontal component u and vertical component v are reported on the left and right hand side, respectively. Thin and bold lines denote the present LW-ACM (256×256 grid, with $\nu_p = \nu$) and the reference solution [30] (2048×2048 grid), respectively.

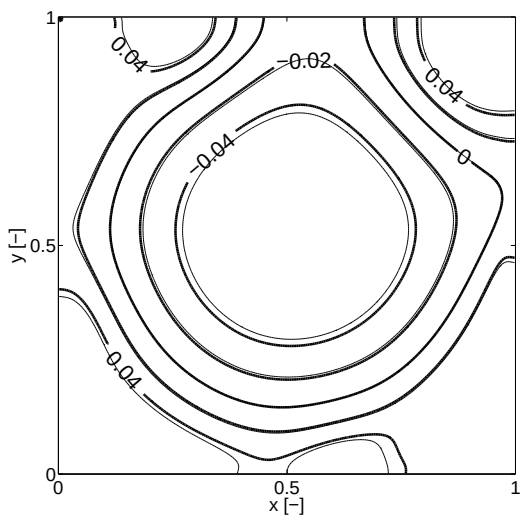

Fig. 10. Comparison of the pressure field for the lid driven cavity flow at Re = 5000. Thin and bold lines denote the present LW-ACM (256×256 grid) and the reference solution [30] (2048×2048 grid), respectively.

two quantities are computed as follows:

$$
\begin{aligned}
E &= \frac{1}{2}\int_{\Omega} \|\mathbf{u}\|^2 \, d\Omega \approx \frac{1}{2}\Delta x \Delta y \sum_{i,j} \left(u_{i,j}^2 + v_{i,j}^2 \right), \\
Z &= \frac{1}{2}\int_{\Omega} \|\omega\|^2 \, d\Omega \approx \frac{1}{2}\Delta x \Delta y \sum_{i,j} \omega_{i,j}^2,
\end{aligned}
\tag{38}
$$

with $\omega = \partial_x v - \partial_y u$, Δx and Δy being the vorticity and the grid spacings respectively. In our study, we notice that one consequence of spurious pressure

oscillations in the solution of classical lattice Boltzmann schemes is that both BGK-LBM and MRT-LBM show remarkable inaccuracy in recovering the enstrophy value predicted by the reference [30], whereas the present LW-ACM overcomes the above issue.

Finally, based on Figs. 3, 4, 5, 6, 7, 8, 9, 10, on comparisons of local and global quantities proposed in Table 5, we can conclude that LW-ACM represents an excellent alternative in terms of simplicity, stability and accuracy.

Table 5
2D lid driven cavity flow at Re = 5000: Comparison between the present LW-ACM (with Mach number Ma = 0.2, $\nu = \nu_p$) and alternative solvers for INSE from literature [12, 30, 13]. In artificial compressibility methods (LW-ACM, BGK-LBM, MRT-LBM), even steady state solutions depend on the numerical Mach number: Hence a fair comparison among different methods requires using the same numerical Mach number (Ma = 0.2 in this case). †This is the optimized version of the ACM method proposed in [12] where (a) high wave numbers are damped for the suppression of the checkerboard instability and (b) the Richardson extrapolation in the Mach number (except around top singular corners) is employed. *Owing to both the accuracy of the scheme and the size of meshes adopted, these results are considered as a reference for the present study. However, since enstrophy Z for 2D lid-driven cavity goes to infinity as the grid spacing goes to zero [30], a meaningful comparison for Z is among similar grids.

Scheme	Grid	(x_p, y_p)	(x_{lr}, y_{lr})	Energy	Enstrophy
Present	128×128	$(0.51652, 0.53754)$	$(0.81081, 0.079079)$	0.039845	29.247
[12]	128×128	$(0.52052, 0.53954)$	$(0.82883, 0.071071)$	0.027430	41.249
[12]†	128×128	$(0.51652, 0.53854)$	$(0.80981, 0.072072)$	0.038371	37.704
BGK-LBM	128×128	unstable	unstable	unstable	unstable
MRT-LBM	128×128	$(0.51652, 0.53554)$	$(0.80881, 0.075075)$	0.043600	37.404
[30]*	128×128	$(0.51562, 0.53906)$	$(0.80469, 0.070313)$	0.043566	30.861
Present	256×256	$(0.51552, 0.53554)$	$(0.80581, 0.074074)$	0.044391	34.821
[12]	256×256	$(0.51652, 0.53654)$	$(0.80881, 0.072072)$	0.040896	43.198
[12]†	256×256	$(0.51451, 0.53654)$	$(0.80380, 0.072072)$	0.048114	42.290
BGK-LBM	250×250	$(0.51752, 0.54054)$	$(0.80781, 0.074074)$	0.041614	40.455
MRT-LBM	256×256	$(0.51552, 0.53554)$	$(0.80681, 0.074074)$	0.045222	40.833
[30]*	256×256	$(0.51562, 0.53516)$	$(0.80859, 0.074219)$	0.046204	34.368
[30]*	2048×2048	$(0.51465, 0.53516)$	$(0.80566, 0.073242)$	0.047290	40.261

Remark-In this study, towards the end of making an extensive comparison

among state of the art INSE solvers, simulations are performed by different methods and grids as reported in Table 5. Since boundaries may be located differently for different methods (e.g. unlike ACM [12] where boundaries coincide with computational nodes, in LW-ACM boundaries are half cell away from computational nodes), upon convergence, all the fields are first interpolated (by cubic spline interpolation) on a shifted grid (same size as the one used for fluid dynamic computations) having the boundaries located on the computational nodes. The values of global kinetic energy and enstrophy given by Eq. (38) are based on the latter shifted grids.

Moreover, if one performs the calculation of streamlines and vortex locations on the basis of the same nodes of the fluid dynamic grid, the final accuracy will depends on both the accuracy of the numerical solution and the grid itself. Hence, results on coarse grids are penalized twice. Therefore, towards the end of computing the coordinates (x_p, y_p) and (x_{lr}, y_{lr}), all the hydrodynamic fields (as computed by the several schemes and different meshes) are first interpolated (by cubic spline interpolation) on a larger mesh, and thereafter the stream function and the vortex locations are computed. The latter post-processing procedure is composed of the following subsequent steps: 1) interpolation of the results on fixed fine grid (1000^2 in Table 5); 2) computation of the stream function and its local extrema denoting the vortex centers.

2.6.2 3D diagonally driven cavity flow

One of the main advantage of the proposed link-wise formulation of ACM consists in its independence on the space dimensionality (as far as the considered equilibrium satisfies the constraints required by the target equations: see Appendix B for details). As a result, the extension of LW-ACM to three-dimensional flows is straightforward. In the following calculations, the D3Q19 lattice [16] will be used: even though that is not a Hermitian lattice (such as D3Q27), the former lattice allows to satisfy the constraints required by the Navier-Stokes equations (in particular Eq. (B.10)).

Here, we have chosen the three-dimensional (3D) diagonally lid-driven cavity flow, which is a classical benchmark for numerical solvers of the incompressible Navier-Stokes equations (see also [33, 34, 35]). The cavity is a cubic box with unit edge as schematically sketched in Fig. 11. The boundary condition at the top plane $(x, y, 1)$ is $\mathbf{u}_L = (\sqrt{2}, \sqrt{2}, 0)/2$ so that $u_L = \|\mathbf{u}_L\| = 1$, whereas the remaining five walls are subject to no-slip boundary conditions. The computational domain is discretized by a uniform collocated grid with N^3 nodes, with boundaries located half-cell away from the computational nodes. Towards the end of making a comparison with data from literature, calculations have been performed by the LW-ACM at two Reynolds numbers studied in [33, 35] and [34] (Re = 700, Re = 2000), and two grids: $N = 48$ and $N = 60$. Let us de-

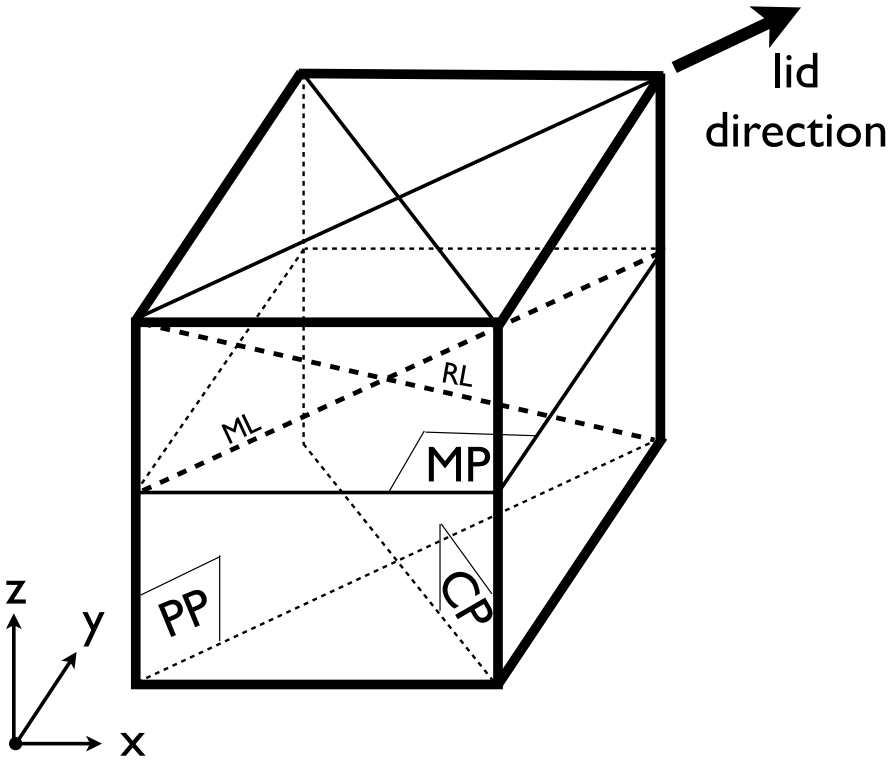

Fig. 11. Cavity flow with the lid moving along its diagonal. Velocity components along axes x, y and z are denoted u, v and w respectively.

note $\hat{\mathbf{x}}_w$ the generic boundary computational node. In all inner computational nodes ($\hat{\mathbf{x}} \neq \hat{\mathbf{x}}_w$), Eq. (13) holds for any lattice velocity $\hat{\mathbf{v}}_i$.

In this test case, numerical stability is significantly affected by boundary conditions. For that reason, in Ref. [34], Authors suggest to use equilibrium-based boundary conditions for the sliding wall at relatively high Reynolds numbers on small computational grids. Nevertheless, as pointed out [34], this implementation imposes an incorrect constant pressure at the boundary, with the momentum transfer significantly weakened in the direction perpendicular to the lid. Moreover, in Ref. [34], the "node" bounce-back boundary conditions are applied to the remaining five walls for imposing no-slip boundary conditions. Although such an approach reduces oscillations caused by the parity invariance and thus enhances the numerical stability, the several simplifications discussed above were necessary to simulate the 3D cavity flow with Re = 2000, $D3Q15$ lattice and 52^3 grid.

On the contrary, the present LW-ACM method does not need to resort to the above simplifications any longer. At an arbitrary boundary node $\hat{\mathbf{x}}_w$ Eq. (31) holds, with $\mathbf{u}_w$ being the boundary velocity (imposed half-cell away from

the boundary computational node $\hat{\mathbf{x}}_w$). This increases the accuracy in treating the boundaries (compared to [34]) and, most importantly, it makes problems in three-dimensions just a straightforward extension of the ones in two-dimensions (see previous section).

In Figs. 12, 13 and 14 we report a comparison between the velocity fields (in the MP, CP and PP planes of Fig. 11) by both the commercial code FLUENT (non-uniform 68^3 grid) [33, 35], here considered as a reference, and the present LW-ACM (60^3 uniform grid) at Reynolds Re = 700: All the flow structures are correctly reproduced by LW-ACM. Due to a lack of local quantities in Ref. [33] for Re = 2000, we solved the latter case by the commercial code FLUENT on our own and we included these results as well in the following comparison. It is important to point out (as reported in Ref. [33]) that commercial code FLUENT did not converge with uniform mesh, i.e. without wall mesh refinement. On the other hand, no such problem was found by LW-ACM. Based on a comparison of local quantities in Fig. 15 for Re = 700, in Fig. 16 for Re = 2000 and parallel/perpendicular global momenta $M_\parallel$, $M_\perp$:

$$\begin{aligned} M_\parallel &= \frac{1}{2}\int_V (u+w)^2 dV \approx \frac{1}{2}\Delta x \Delta y \Delta z \sum_{i,j,k} \left(u_{i,j,k} + w_{i,j,k}\right)^2 , \\ M_\perp &= \frac{1}{2}\int_V (u-w)^2 dV \approx \frac{1}{2}\Delta x \Delta y \Delta z \sum_{i,j,k} \left(u_{i,j,k} - w_{i,j,k}\right)^2 \end{aligned} \tag{39}$$

reported in Table 6, we can conclude that the present LW-ACM is indeed able to recover the reference solution with significant accuracy.

In Fig. 17, results of the 3D lid-driven cavity flow at higher Reynolds number, Re = 2000, are reported. It is worth stressing that, here all the main structures of the flow are correctly described by LW-ACM even with grids coarser than the one adopted in the reference solution [33, 35]. We stress that, describing secondary vortexes in this case is known to be a severe test for numerical schemes (in particular catching top-left and bottom-right secondary vortexes).

For the sake of completeness, we also notice as the Reynolds number increases larger deviations of the LW-ACM solution from the reference are observed in terms of the parallel/perpendicular global momenta $M_\parallel$, $M_\perp$ (see Table 6).

Finally, in Fig. 18 the numerical results obtained by both LW-ACM and the MRT-LBM [34] are shown. These two simulations are not perfectly comparable each other. In fact, Authors in Ref. [34] were forced by stability issues to implement some simplifications when dealing with the boundary conditions (mainly, equilibrium-based boundary conditions for imposing the lid velocity and "node" bounce-back for the no-slip boundary conditions). Those simplifications reduce the accuracy with regards to that recovered by Eq. (31).

Other minor difference is that [34] and the present study were obtained by the

D3Q15 lattice with 52^3 grid, and D3Q19 lattice with 48^3 grid respectively. We notice that, in this case, MRT-LBM makes use of a larger number of degrees of freedom compared to LW-ACM: $52^3 \times 15 > 48^3 \times 19$. In spite of this, it is quite clear by Fig. 18 that the pressure field recovered by the present LW-ACM is remarkably smoother than the one obtained by MRT-LBM. More specifically, a crucial difference is that LW-ACM predicts smooth pressure increase at the top-right corner, while the MRT-LBM results are affected by oscillations around the imposed constant pressure at the top plane.

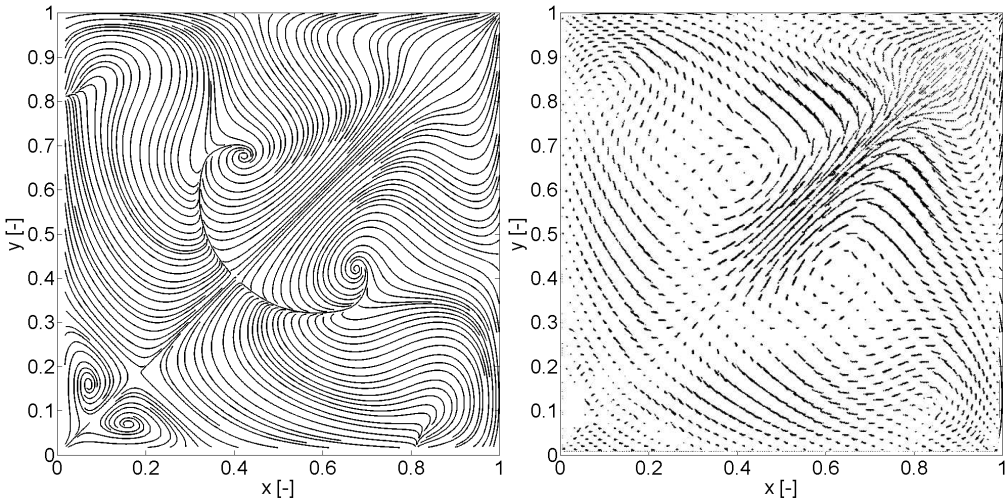

Fig. 12. Flow in the middle plane $z = 0.5$ of the 3D diagonally driven cavity (MP plane in Fig. 11) at Re = 700. Comparison between the LW-ACM with 60^3 grid (left) and a reference solution [33] obtained by the commercial code FLUENT (right) with 68^3 total number of grid nodes.

Table 6
3D diagonally driven cavity: Volume integral of momentum flux. Comparisons are carried out between the present LW-ACM method and the reference solution in [33] adopting 60^3 and 68^3 total number of grid nodes, respectively.

	Present, Re = 700	[33, 35], Re = 700	Present, Re = 2000	[33, 35], Re = 2000
$\int_V M_{\parallel}$	0.203×10^{-1}	0.216×10^{-1}	0.134×10^{-1}	0.163×10^{-1}
$\int_V M_{\perp}$	0.232×10^{-2}	0.283×10^{-2}	0.174×10^{-2}	0.239×10^{-2}

2.6.3 Circular Couette flow

Dealing with moving complex boundaries is very important in many applications: for example, particle suspensions, granular flows and active (bio-)agents immersed in the flow. In these cases, the essential issue is to reduce as much

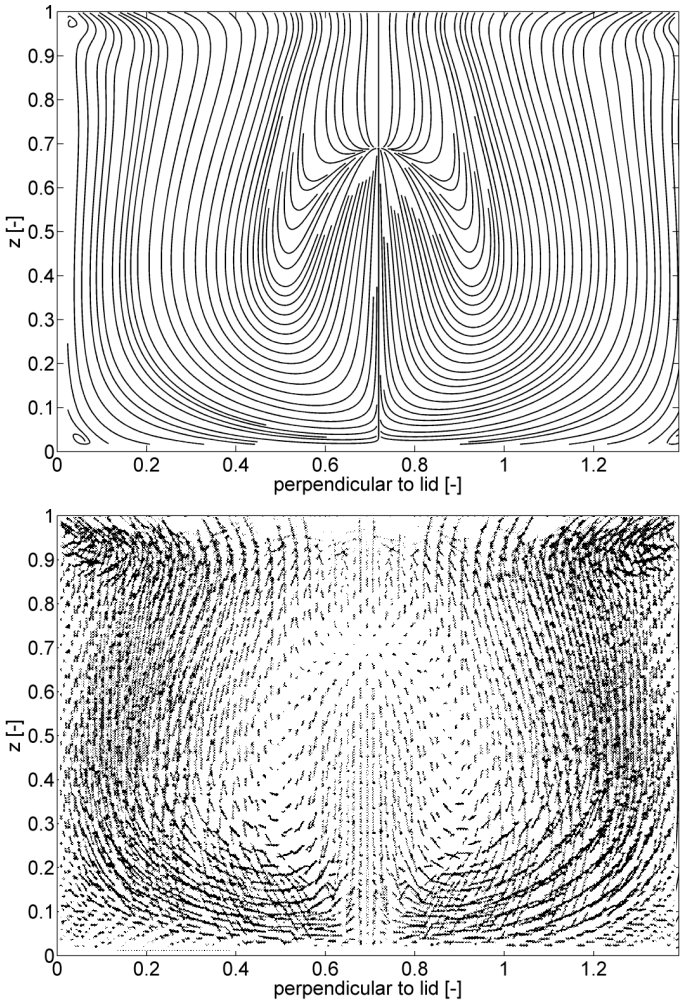

Fig. 13. Flow in the plane perpendicular to the direction of the lid (CP plane in Fig. 11) at Re = 700. Comparison between the LW-ACM with 60^3 Cartesian grid (top) and a reference solution [33] obtained by the commercial code FLUENT (bottom) with 68^3 total number of grid nodes. At the centerline, a stagnation point is observed at $z = 0.68$ (present), and $z = 0.74$ (FLUENT).

as possible the computational demand by avoiding re-meshing every time that the considered objects move in the flow. Taking into account moving objects is also complicated by the need of re-initializing the portions of the flow field which are filled again by the fluid after the motion of the objects. The latter feature is neglected here, because it is a general issue, not peculiar of the link-wise methods.

First of all, we extend the wall boundary treatment discussed in the previous sections. Let us suppose that $\hat{\mathbf{x}}$ is a fluid node close to a complex wall boundary at rest such that $\hat{\mathbf{x}}+\hat{\mathbf{v}}_i$ is a wall node. Let us focus on the intersection between the i-th lattice link and the wall. The distance between the latter intersection and the fluid node, divided by the mesh spacing Δx, gives the normalized distance $0 \leq q \leq 1$ (above we considered only the case: $q = 1/2$). In this case,

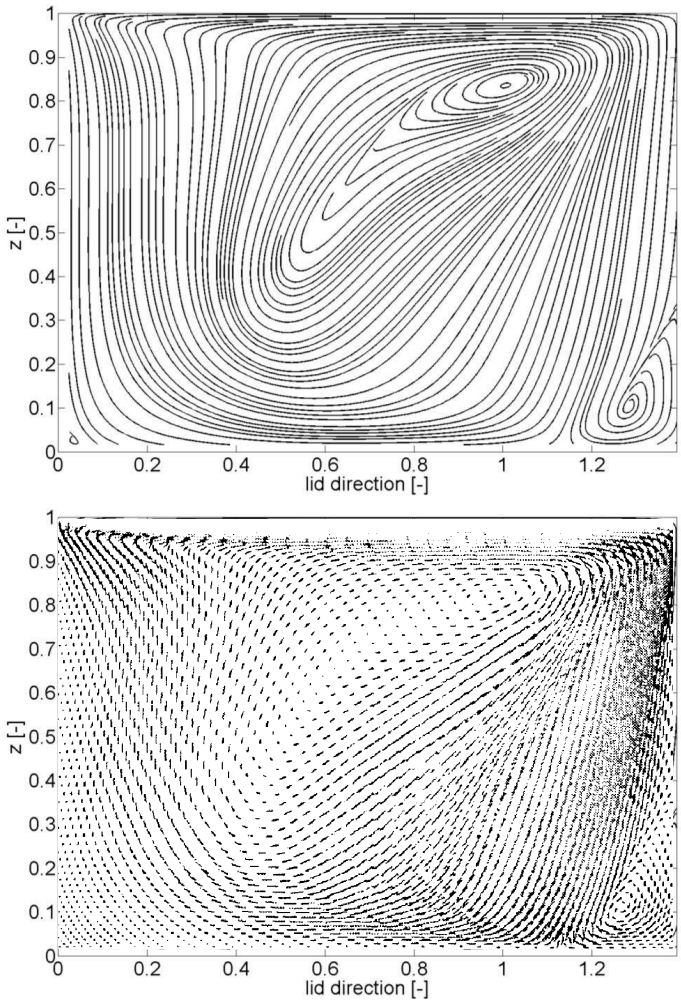

Fig. 14. Flow in the plane parallel to the direction of the lid (PP plane in Fig. 11) at Re = 700. Comparison between the LW-ACM with 60^3 Cartesian grid (top) and a reference solution [33] obtained by the commercial code FLUENT (bottom) with 68^3 total number of grid nodes.

the streaming step can be performed following the same procedure provided for LBM in [26] (instead of using Eq. (28)), namely

$$f^{**}_{BB(i)}(\hat{\mathbf{x}}, \hat{t}+1) = \begin{cases} 2q f^*_i(\hat{\mathbf{x}}, \hat{t}) + (1-2q) f^*_i(\hat{\mathbf{x}} - \epsilon\, \hat{\mathbf{v}}_i, \hat{t}), \; q < 1/2, \\ \frac{1}{2q} f^*_i(\hat{\mathbf{x}}, \hat{t}) + \left(1 - \frac{1}{2q}\right) f^*_{BB(i)}(\hat{\mathbf{x}}, \hat{t}), \quad q \geq 1/2, \end{cases}$$

where $BB(i)$ is the bounce-back operator giving the lattice link opposite to i-th. Finally the post-combining step can be performed in the usual way, namely by means of Eq. (28).

In case of moving complex boundary with velocity $\mathbf{u}_w$, the procedure reported

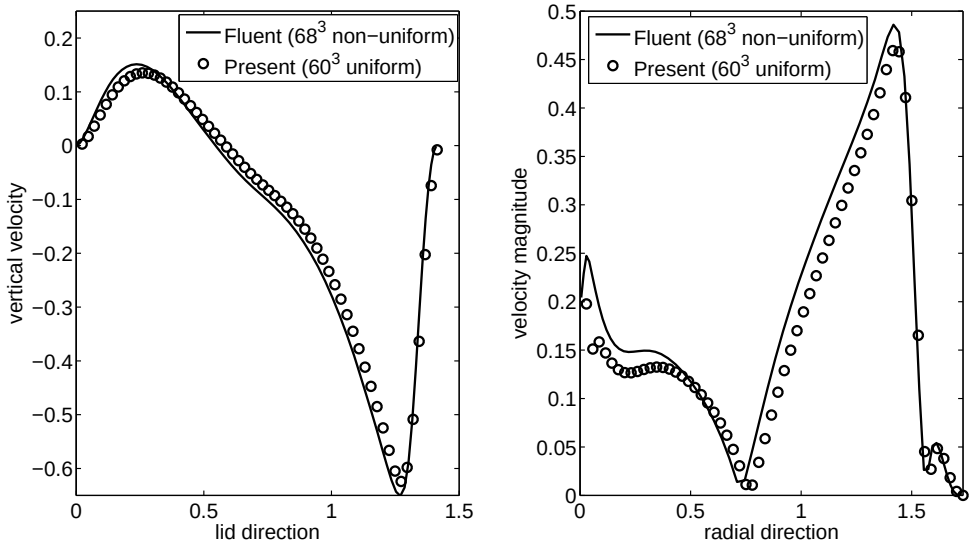

Fig. 15. Velocity profile along the line ML (left) and the line RL (right) at Re = 700 (see also Fig. 11). Comparison between the present LW-ACM with uniform 60^3 Cartesian grid and a reference solution (consistent with [33]) obtained by the commercial code FLUENT with non-uniform 68^3 total number of grid nodes.

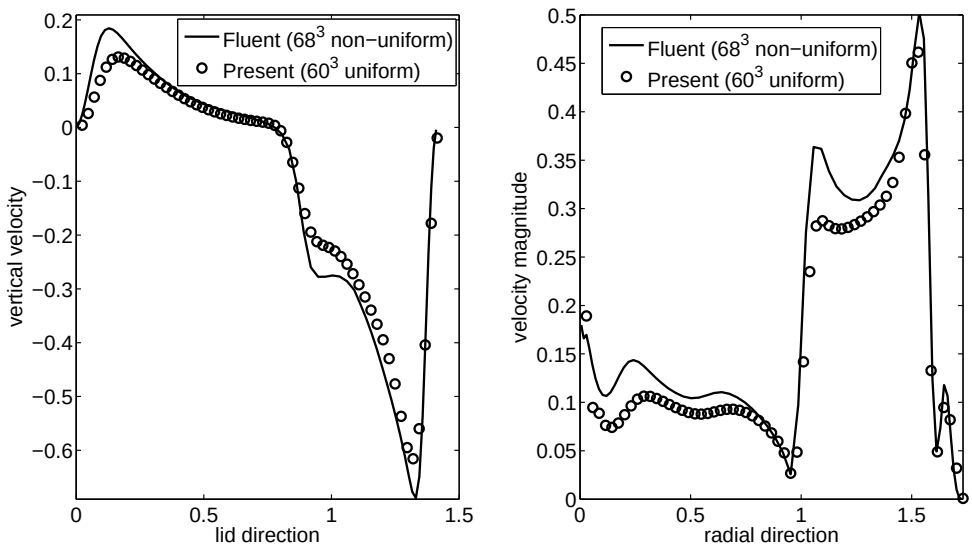

Fig. 16. Velocity profile along the line ML (left) and the line RL (right) at Re = 2000 (see also Fig. 11). Comparison between the present LW-ACM with uniform 60^3 Cartesian grid and a reference solution obtained by the commercial code FLUENT with non-uniform 68^3 total number of grid nodes.

in [26] suggests to consider an additional term, namely

$$\delta f_{BB(i)}(\rho_0, \mathbf{u}_w) = \begin{cases} 2 f^{(e,o)}_{BB(i)}(\rho_0, \mathbf{u}_w), \ q < 1/2, \\ \frac{1}{q} f^{(e,o)}_{BB(i)}(\rho_0, \mathbf{u}_w), \ q \geq 1/2, \end{cases} \tag{40}$$

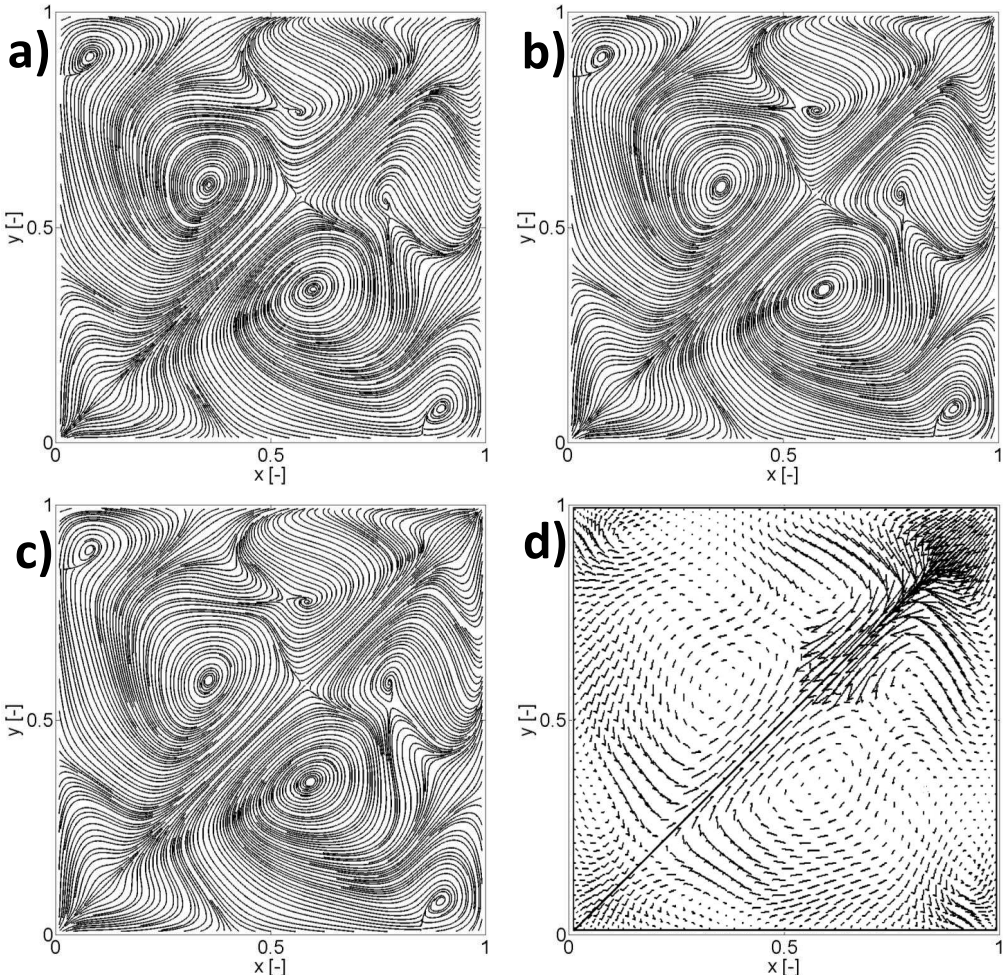

Fig. 17. Flow in the middle plane $z = 0.5$ of the 3D diagonally driven cavity (MP plane in Fig. 11) at Re = 2000. a) LW-ACM with 48^3 uniform grid. b) LW-ACM with 60^3 uniform grid. c) LW-ACM with 68^3 uniform grid. d) Reference solution by the commercial code FLUENT with 68^3 non-uniform grid [35].

where $f^{(e,o)}_{BB(i)}$ is given by Eq. (2) and ρ_0 is the average value of the density over the whole computational domain (see Appendix B for details). Similarly to what we did in the previous sections, in case of diffusive scaling, the suggested correction for LBM will be multiplied by a scaling factor in link-wise ACM. Hence $f^{w}_{BB(i)}$ given by Eq. (31) is the proper boundary condition in case of moving complex boundary.

In this section, numerical results are reported for the circular Couette flow, where a viscous fluid is confined in the gap between two concentric rotating cylinders. In our study, we assume the inner cylinder (with radius r_i) at rest while the outer cylinder (with radius r_e) rotates at a constant angular velocity $1/r_e$. The latter flow admits the following exact solution:

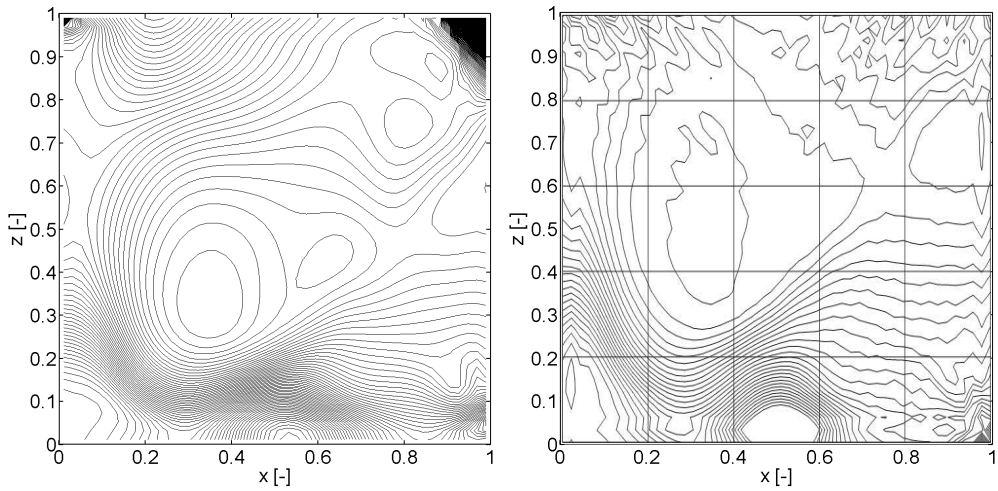

Fig. 18. Pressure contours for the diagonally driven cavity flow for Re = 2000 at the lateral mid-plane ($y = 0.5$): Solution obtained by present LW-ACM with 48^3 uniform grid and $D3Q19$ lattice (left), and MRT-LBM method with 52^3 uniform grid and $D3Q15$ lattice [34] (right).

Table 7
The L^1 norm of the error versus $\epsilon \equiv \Delta x \equiv \mathrm{Ma}$ at $t = 20$ in the problem of the circular Couette flow for $\nu = 0.07$.

Link-wise ACM			
$\epsilon \equiv \Delta x$	Error $L^1[\bar{u}_\theta]$	Error $L^1[\bar{p}]$	Error $L^1[\lvert\bar{\mathcal{T}}\rvert]$
1/20	1.53627×10^{-3}	8.81208×10^{-4}	4.20562×10^{-4}
1/40	3.50537×10^{-4}	3.58432×10^{-4}	1.98687×10^{-4}
1/80	1.77257×10^{-4}	1.97650×10^{-4}	9.13289×10^{-5}
1/160	3.42570×10^{-5}	6.16474×10^{-5}	3.66160×10^{-5}
MRT-LBM			
$\epsilon \equiv \Delta x$	Error $L^1[\bar{u}_\theta]$	Error $L^1[\bar{p}]$	Error $L^1[\lvert\bar{\mathcal{T}}\rvert]$
1/20	4.66795×10^{-3}	2.52316×10^{-3}	7.58091×10^{-5}
1/40	1.52864×10^{-3}	8.47929×10^{-4}	1.39351×10^{-4}
1/80	3.08607×10^{-4}	2.99584×10^{-4}	6.98541×10^{-5}
1/160	7.99695×10^{-5}	1.09817×10^{-4}	3.39760×10^{-5}

$$\bar{u}(t,r,\theta) = -C\left(\frac{r}{r_i} - \frac{r_i}{r}\right)\sin(\theta), \tag{41}$$

$$\bar{v}(t,r,\theta) = C\left(\frac{r}{r_i} - \frac{r_i}{r}\right)\cos(\theta), \tag{42}$$

$$\bar{p}(t,r,\theta) = \bar{p}_i + C^2 \ln\left(\frac{r_i^2}{r^2}\right) - \frac{C^2}{2}\left(\frac{r_i^2}{r^2} - \frac{r^2}{r_i^2}\right), \tag{43}$$

$$\bar{\mathcal{T}} = 4\pi C\,\nu\, r_i, \tag{44}$$

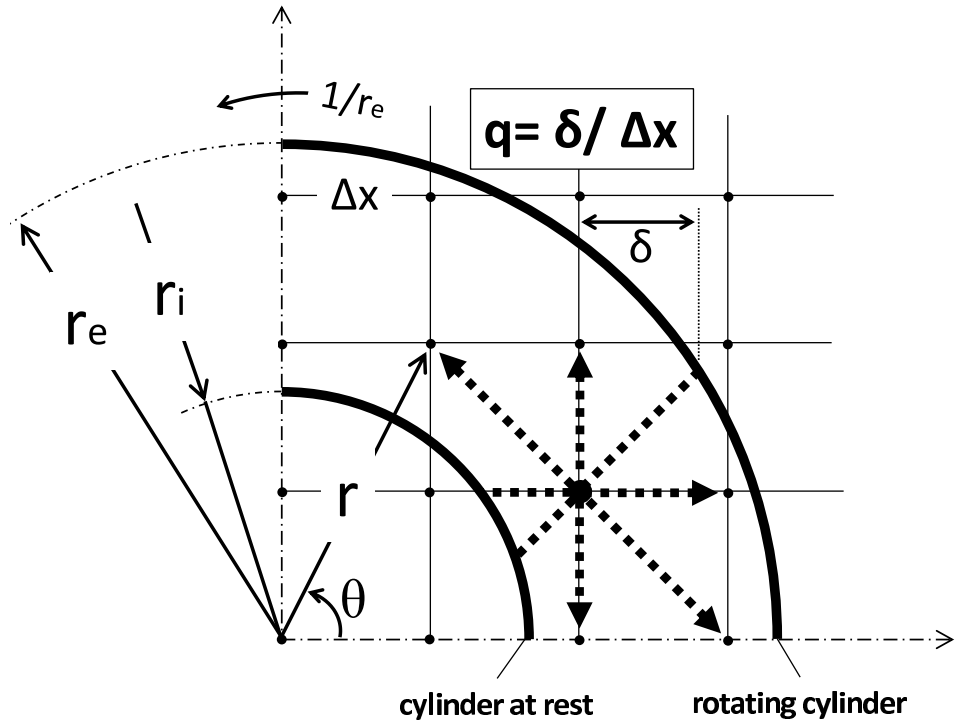

Fig. 19. Set-up of the circular Couette flow, where one quarter of the domain is reported. For the sake of clarity, a significantly coarse grid is represented.

where

$$C = \frac{1}{r_e/r_i - r_i/r_e}. \tag{45}$$

Here, $\bar{u}$, $\bar{v}$, $\bar{p}$ and $\bar{\mathcal{T}}$ denote horizontal velocity, vertical velocity, pressure and the torque on the inner cylinder respectively, with θ being the angle between radial direction and the horizontal axis. A schematic representation of this setup is reported in Figure 19. Diffusive scaling is considered for this test case, where the velocity field is scaled on meshes with different sizes, keeping fixed the relaxation frequency (see Appendix B for details). The latter scaling ensures second order convergence in the accuracy, as reported in Table 7.

Moreover, the torque exerted by the fluid on the inner cylinder is computed. To this end, Eq. (34) is applied in combination with the boundary conditions provided by (40). The computation of $\bar{\mathcal{T}}$ was finally performed by a summation of the contributions (34) over all the boundary links around its surface, namely

$$\mathcal{T} = \sum_{i \in \mathsf{S}} (\hat{\mathbf{x}} - \hat{\mathbf{x}}_c) \times \mathbf{p}_i, \tag{46}$$

where $\hat{\mathbf{x}}_c$ is the center of the cylinders and S is the set of links starting from all nodes surrounding the body and intersecting the body itself. Similarly to what has been done for scaling the force exerted on a body, the above torque (assumed acting on the whole inner cylinder) must be converted from lattice

units to physical units (see Table 1 for details). More specifically, the formula is the following

$$\bar{\mathcal{T}} = \sum_{i \in \mathsf{S}(1/\epsilon)} (\hat{\mathbf{x}} - \hat{\mathbf{x}}_c) \times \mathbf{p}_i, \tag{47}$$

because $|\hat{\mathbf{x}} - \hat{\mathbf{x}}_c| \sim 1/\epsilon$ and this automatically takes into account the force scaling reported in Eq. (36).

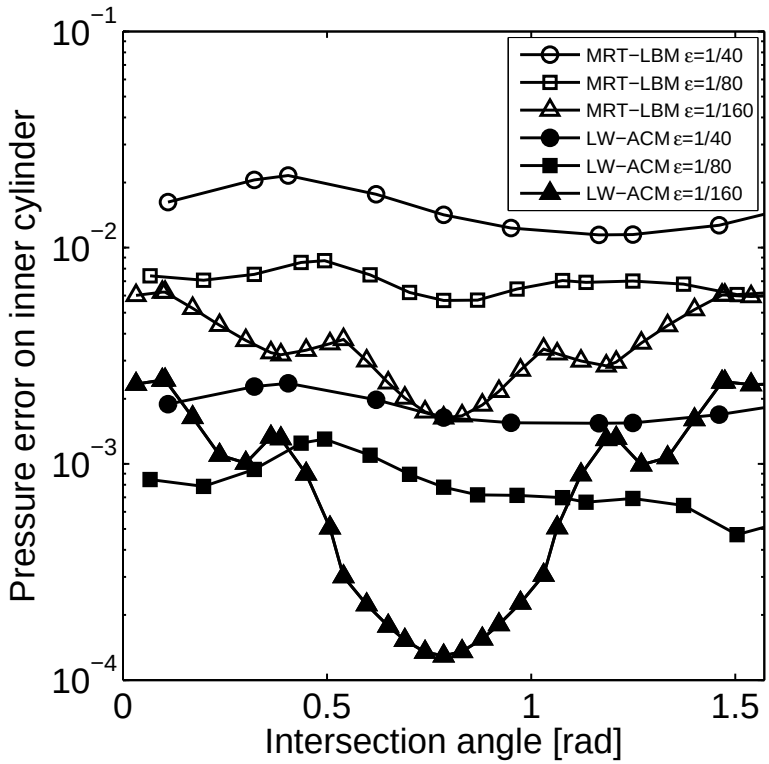

Fig. 20. Circular Couette flow: Pressure error (absolute value of the discrepancy with respect to the analytical solution) of the pressure solution interpolated on the inner cylinder surface as a function of the intersection angle (in the interval $\theta \in [0, \pi/2]$), between the versor normal to the inner cylinder (radial direction) and the horizontal axis of the Cartesian grid.

In Table 7, the numerical results for the circular Couette flow are reported. As expected, the numerical solution in terms of velocity and pressure shows almost second order convergence rate. On the other hand, the modified MEA for link-wise ACM in computing $\left|\bar{\mathcal{T}}\right|$ shows first order convergence rate (similarly to the original MEA for LBM). The torque is a global quantity and hence it may hide some error compensation. In order to report a quantitative and detailed assessment of the local stresses, we consider the local wall shear stress on the inner cylinder surface, namely

$$\bar{\tau}_i = \nu \left.\frac{d\bar{w}}{dr}\right|_i = \frac{\bar{\mathcal{T}}}{2\pi\, r_i^2}, \tag{48}$$

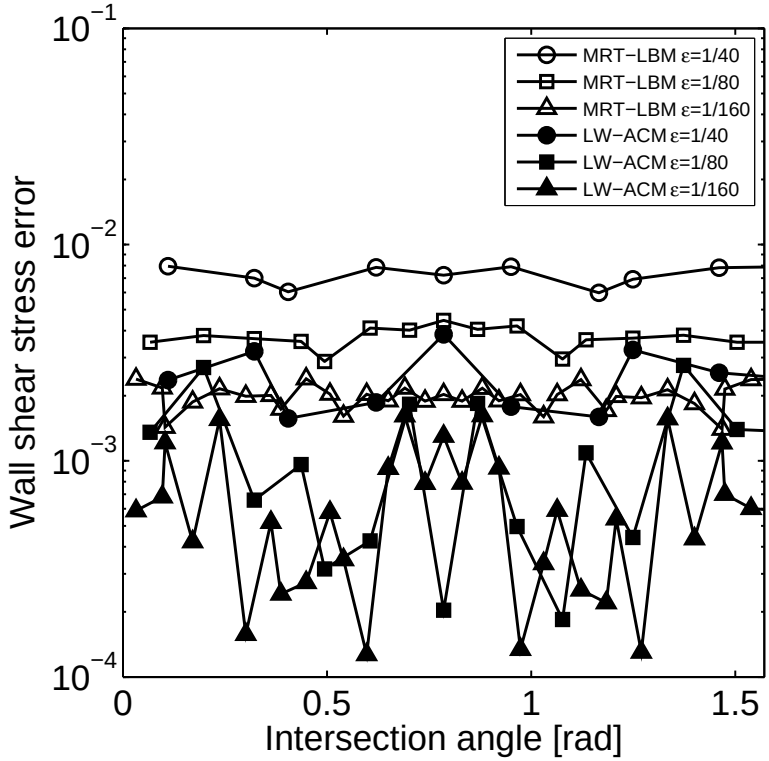

Fig. 21. Circular Couette flow: Wall shear stress error (absolute value of the discrepancy with regards to the analytical solution) calculated by means of the local solution gradient on the inner cylinder surface as a function of the intersection angle (in the interval $\theta \in [0, \pi/2]$), between the versor normal to the inner cylinder (radial direction) and the horizontal axis of the Cartesian grid.

where $\bar{w} = \sqrt{\bar{u}^2 + \bar{v}^2}$. In particular, in Figs. 20 and 21 the pressure and the wall shear stress absolute error are reported respectively, as a function of the intersection angle (in the interval $\theta \in [0, \pi/2]$) between the versor normal to the inner cylinder (radial direction) and the horizontal axis of the Cartesian grid. In both cases, the errors are computed by the absolute value of the discrepancy between the computed quantities and the analytical solutions. Different meshes are considered, namely $\epsilon = 1/40$, $1/80$ and $1/160$. The numerical results show that, for both pressure values and wall shear stresses, the LW-ACM produces numerical results which are systematically more accurate than those by LBM, even though the two methods are using the same boundary conditions. On finer meshes, the numerical errors are more scattered due to amplification by the logarithmic scale.

In this section, we reported evidences about the possibility to use boundary conditions, originally formulated for LBM, also in the context of LW-ACM. The drawback is that these boundary conditions are less intuitive than those commonly used in FD and this is particularly strident for LW-ACM, which has a FD formulation indeed. As a concluding remark, it is important to recall that

the formulation of the boundary condition is very transparent in the original ACM, e.g. no slip wall velocity and zero pressure gradient normal to the wall. Moreover, by taking into account the curvature correctly (e.g. by finite-volume ACM using body-fitted cell system), the accuracy is dramatically improved. The reason is that, in link-wise ACM and in LBM, the boundary condition for curved wall is based on one dimensional interpolation, but this method is not very accurate for computing stresses (depending on local spatial derivatives). Hence for accurate solving boundary layers, the finite-volume ACM using body-fitted cell system is preferable. However, in the present paper, we used link-wise boundary conditions because they are extremely simple to be generalized in three dimensions and they have potential when dealing with moving complex objects (e.g. particles).

3 Conclusions

In the present work, a novel method for low Mach number fluid dynamic simulations is proposed, taking inspiration from the best features of both the Lattice Boltzmann Method (LBM) and more classical computational fluid dynamic (CFD) techniques such as the Artificial Compressibility Method (ACM). The main advantage is the possibility of exploiting well established technologies originally developed for LBM and classical CFD, with special emphasis on finite differences (at least in the present paper), at the cost of minor changes. For instance, like LBM, it is possible to use simple Cartesian structured meshes, eventually recursively refined in the vicinity of solid walls, and there is no need of solving Poisson equations for pressure. On the other hand, any boundary condition designed for finite difference schemes can be easily included.

As far as solving incompressible Navier-Stokes equations - INSE - (by minimal amount of unknowns) is the only concern, the pseudo-kinetic heritages of LBM represent a severe limitation to several aspects such as designing flexible boundary conditions, introducing tunable forcing terms and analyzing consistency of the numerical scheme (asymptotics). On the contrary, the suggested method has no such pseudo-kinetic heritages. Or in other words, following the standard LBM nomenclature, the present LW-ACM requires no high-order moments beyond hydrodynamics (often referred to as *ghost moments*) and no kinetic expansion such as Chapman-Enskog, Hilbert, van Kampen. Like finite difference schemes, only standard Taylor expansion is needed for analyzing consistency. Beside the above aspects, numerical evidences reported in this work suggest that LW-ACM represents an excellent alternative in terms of simplicity, stability and accuracy. Hence, in this framework (solving INSE by minimal amount of unknowns), the utility of high-order moments is questionable.

Finally, preliminary efforts towards optimal implementations have shown that LW-ACM is capable of similar computational speed as optimized (BGK-) LBM. In addition, the memory demand is significantly smaller than (BGK-) LBM. In our opinion, there is still room for improvement according to the performance model (based on assuming either infinitely fast memory or infinitely fast compute units). Importantly, with an efficient implementation, this algorithm may be one of the few which is compute-bound and not memory-bound. The latter observation is of particular interest for General-Purpose computing on Graphics Processing Units (GPGPU).

Acknowledgments

The authors are grateful to Professor Charles-Henri Bruneau of Université Bordeaux I, for providing us the numerical data of lid-driven cavity flow. PA would like to thank Dr. Paul J. Dellar for pointing out the Minion & Brown flow test case, Dr. Martin Geier for stressing the importance of having techniques able to compute hydrodynamical stresses locally, Dr. Salvador Izquierdo for critical discussions about LBM boundary conditions. Moreover PA would like to thank Dr. Thomas Zeiser for suggestions about efficient coding and for measuring the performance reported in Table 2 (based on elementary codes developed by PA). Finally, PA and AFdR would like to acknowledge the support of the national PRIN 2008 project "Microscopic modeling and degradation analysis of the membrane electrode assembly (MEA) in high temperature PEM fuel cells" 2010-2012.

A Appendix: Equilibrium distribution functions

The quantities $f_i^{(e)}$ are designed in order to recover the incompressible isothermal fluid dynamics. For sake of completeness, we report here the explicit expressions of the equilibrium functions for some popular lattices.

The D2Q9 lattice [16], suitable for two dimensional problems ($D = 2$), consists of the following discrete velocities ($Q = 9$): $\hat{\mathbf{v}}_0 = (0, 0)$, $\hat{\mathbf{v}}_i = (\pm 1, 0)$ and $(0, \pm 1)$, for $i = 1$–4, and $\hat{\mathbf{v}}_i = (\pm 1, \pm 1)$, for $i = 5$–8, where the i-th equilibrium distribution function $f_i^{(e)}$ reads

$$f_i^{(e)} = w_i \rho \left[1 + 3\hat{\mathbf{v}}_i \cdot \mathbf{u} + \frac{9}{2} \left(\hat{\mathbf{v}}_i \cdot \mathbf{u} \right)^2 - \frac{3}{2} \mathbf{u}^2 \right], \tag{A.1}$$

with ρ the fluid density, and w_i the weights

$$w_i = \begin{cases} 4/9 & i = 0, \\ 1/9 & i = 1\text{–}4, \\ 1/36 & i = 5\text{–}8. \end{cases} \tag{A.2}$$

More explicitly, the complete set of equilibria takes the form:

$$f^{(e)} = \begin{bmatrix} 4/9\,\rho - 2/3\,\rho u^2 - 2/3\,\rho v^2, \\ 1/9\,\rho + 1/3\,\rho u + 1/3\,\rho u^2 - 1/6\,\rho v^2, \\ 1/9\,\rho + 1/3\,\rho v + 1/3\,\rho v^2 - 1/6\,\rho u^2, \\ 1/9\,\rho - 1/3\,\rho u + 1/3\,\rho u^2 - 1/6\,\rho v^2, \\ 1/9\,\rho - 1/3\,\rho v + 1/3\,\rho v^2 - 1/6\,\rho u^2, \\ 1/36\,\rho + 1/12\,\rho(u+v) + 1/8\,\rho(u+v)^2 - 1/24\,\rho(u^2+v^2), \\ 1/36\,\rho - 1/12\,\rho(u-v) + 1/8\,\rho(-u+v)^2 - 1/24\,\rho(u^2+v^2), \\ 1/36\,\rho - 1/12\,\rho(u+v) + 1/8\,\rho(-u-v)^2 - 1/24\,\rho(u^2+v^2), \\ 1/36\,\rho + 1/12\,\rho(u-v) + 1/8\,\rho(u-v)^2 - 1/24\,\rho(u^2+v^2) \end{bmatrix},$$

where u and v are the velocity components, i.e. $(u, v)^T = \mathbf{u}$, with pressure being $p = \rho/3$. The above equations (A.1) and (A.3) can be generalized as follows [36]

$$f^{(g)}\left(\Pi_{xx}, \Pi_{yy}\right) = \begin{bmatrix} \rho\left(1 - \Pi_{xx}\right)\left(1 - \Pi_{yy}\right), \\ \rho\left(\Pi_{xx} + u\right)\left(1 - \Pi_{yy}\right)/2, \\ \rho\left(\Pi_{xx} - u\right)\left(1 - \Pi_{yy}\right)/2, \\ \rho\left(1 - \Pi_{xx}\right)\left(\Pi_{yy} + v\right)/2, \\ \rho\left(1 - \Pi_{xx}\right)\left(\Pi_{yy} - v\right)/2, \\ \rho\left(\Pi_{xx} + u\right)\left(\Pi_{yy} + v\right)/4, \\ \rho\left(\Pi_{xx} - u\right)\left(\Pi_{yy} + v\right)/4, \\ \rho\left(\Pi_{xx} - u\right)\left(\Pi_{yy} - v\right)/4, \\ \rho\left(\Pi_{xx} + u\right)\left(\Pi_{yy} - v\right)/4, \end{bmatrix} \tag{A.3}$$

with the equation (A.1) being a special case of (A.3): If one assumes $\Pi_{xx} = 1/3 + u^2$ and $\Pi_{yy} = 1/3 + v^2$, then $f^{(g)}\left(1/3 + u^2, 1/3 + v^2\right) = f^{(e)}$ (if third

order terms with respect to velocity components are neglected). However, it is possible to introduce more involved functions depending on additional parameters. For instance, a quasi-equilibrium function which is useful for tuning bulk viscosity of both lattice Boltzmann and link-wise ACM schemes can be expressed as

$$f^{(qe)}\left(\rho, \mathbf{u}, \mathrm{Tr}\right) = f^{(g)}\left(\frac{\mathrm{Tr} + u^2 - v^2}{2}, \frac{\mathrm{Tr} - u^2 + v^2}{2}\right), \tag{A.4}$$

where Tr is an additional tunable parameter (usually corresponding to the trace of the second order tensor $\mathbf{\Pi} = \sum_i \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i f_i$ normalized by density (see also the Appendix C).

The D3Q19 lattice, which is suitable for three dimensional problems ($D = 3$), consists of the following discrete velocities ($Q = 19$): $\hat{\mathbf{v}}_0 = (0, 0, 0)$; $\hat{\mathbf{v}}_i = (\pm 1, 0, 0)$ and $(0, \pm 1, 0)$ and $(0, 0, \pm 1)$, for $i = 1$–6; $\hat{\mathbf{v}}_i = (\pm 1, \pm 1, 0)$ and $(\pm 1, 0, \pm 1)$ and $(0, \pm 1, \pm 1)$, for $i = 7$–18. Here, the the i-th function $f_i^{(e)}$ is formally identical to (A.1), with the following weights

$$w_i = \begin{cases} 1/3 & i = 0, \\ 1/18 & i = 1\text{–}6, \\ 1/36 & i = 7\text{–}18. \end{cases} \tag{A.5}$$

B Appendix: Asympthotic analysis

In (13), with $\hat{\mathbf{x}} = \mathbf{x}/\Delta x$ and $\hat{t} = t/\Delta t$, both Δt and Δx must approach zero, though it is not clear the value of the limit: $\lim_{\Delta x \to 0} \Delta t/\Delta x$. In order to clarify this point, we apply Taylor expansion to (13)

$$\begin{aligned} f_i(t + \Delta t) = {} & f_i^{(e)} - \Delta x\, \hat{\mathbf{v}}_i \cdot \nabla f_i^{(e)} + \frac{\Delta x^2}{2} (\hat{\mathbf{v}}_i \cdot \nabla)^2 f_i^{(e)} - \frac{\Delta x^3}{6} (\hat{\mathbf{v}}_i \cdot \nabla)^3 f_i^{(e)} + \dots \\ & + \left(2 - \frac{2}{\omega}\right) \left(\Delta x\, \hat{\mathbf{v}}_i \cdot \nabla f_i^{(e,o)} - \frac{\Delta x^2}{2} (\hat{\mathbf{v}}_i \cdot \nabla)^2 f_i^{(e,o)} + \frac{\Delta x^3}{6} (\hat{\mathbf{v}}_i \cdot \nabla)^3 f_i^{(e,o)}\right) + \dots, \end{aligned} \tag{B.1}$$

where all the quantities are computed in the same point $\mathbf{x}$ and hence this is no more explicitly reported. A summation of Eqs. (B.1) over i yields:

$$\frac{\partial \rho}{\partial t} + \left(\frac{2}{\omega} - 1\right) \frac{\Delta x}{\Delta t} \nabla \cdot (\rho \mathbf{u}) + \frac{1}{6} \left(\frac{2}{\omega} - 1\right) \frac{\Delta x^3}{\Delta t} (\nabla \cdot)^3 \mathsf{Q}^{(e)} = \frac{\Delta x^2}{2\,\Delta t} (\nabla \cdot)^2 \mathbf{\Pi}^{(e)} + O\left(\frac{\Delta x^4}{\Delta t} (\nabla \cdot)^2 \mathbf{\Pi}^{(e)}\right), \quad \text{(B.2)}$$

with $\mathsf{Q}^{(e)} = \sum_i \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i f_i^{(e)}$. In the rightmost term of the above expression (B.2), we rely upon the fact that spatial derivatives of the fourth-order moments have the same growth rate of spatial derivatives of the second-order moments. Moreover, the diverge is raised to a power which is selected for consistency with the units of other terms. This is unessential, as far as the order of magnitude of the term is concerned. Multiplying (B.1) by $\hat{\mathbf{v}}_i$ and summing over i, it yields

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \frac{\Delta x}{\Delta t} \nabla \cdot \mathbf{\Pi}^{(e)} = \frac{\Delta x^2}{\Delta t} \left(\frac{1}{\omega} - \frac{1}{2}\right) (\nabla \cdot)^2 \mathsf{Q}^{(e)} + O\left(\frac{\Delta x^3}{\Delta t} \nabla \cdot \mathbf{\Pi}^{(e)}\right). \quad \text{(B.3)}$$

Similar considerations as Eq. (B.2) apply to the rightmost term in the above equation (B.3). Concerning the relationship between Δt and Δx, (B.3) suggests two possible strategies:

$$\Delta t \propto \Delta x, \qquad \text{acoustic scaling,} \quad \text{(B.4a)}$$

$$\Delta t \propto \Delta x^2, \qquad \text{diffusive scaling.} \quad \text{(B.4b)}$$

Sometimes, the dimensionless mesh spacing $\Delta x = \Delta x'/L$ is referred to as spacing ($\Delta x \equiv h$) in the literature on finite difference method or even numerical Knudsen number ($\Delta x \equiv \mathrm{Kn}$) in the Lattice Boltzmann literature. Similarly, it is possible to introduce a numerical Mach number: $\mathrm{Ma} = U/(\Delta x'/\Delta t')$. In this way, the dimensionless time step $\Delta t = \Delta t'/(L/U)$ can be expressed as $\Delta t \equiv \mathrm{Kn}\,\mathrm{Ma}$. Hence, the acoustic scaling corresponds to constant Ma, while the diffusive scaling to $\mathrm{Ma} \propto \Delta x$.

Regardless of the adopted strategy, the numerical scheme must converge towards the physical solution of the incompressible Navier-Stokes equations. The physical solution is identified by the Reynolds number, which is the reciprocal of the factor multiplying the second-order spatial derivatives in Eq. (B.3), namely

$$\frac{\Delta x^2}{\Delta t} \left(\frac{1}{\omega} - \frac{1}{2}\right) \propto \frac{1}{\mathrm{Re}}. \quad \text{(B.5)}$$

Hence, according to (B.5), in acoustic scaling ω needs to be tuned in order to guarantee a constant Reynolds number, while in diffusive scaling a fixed ω already ensures a constant Reynolds number. Moreover, in acoustic scaling, the smaller Δx the smaller ω for keeping fixed the Reynolds number. The latter case can be problematic in the present LW-ACM due to the heuristic stability domain, $1 \leq \omega < 2$, thus diffusive scaling is generally preferable.

Similarly to standard LBM, in LW-ACM, the definition of Δt and Δx is implicit (and this is sometimes a source of confusion). In fact, the end-user

can select both Kn $\propto 1/N$ (with N the number of mesh points along the flow characteristic length) and Ma by a scaling factor for the velocity field. Hence, the acoustic scaling requires the tuning of the relaxation frequency ω on different meshes, keeping constant the computed velocity field. On the other hand, the diffusive scaling corresponds to a scaling of the velocity field (see also the section below) on different meshes, keeping fixed the relaxation frequency.

B.1 Diffusive scaling

Substituting $\Delta x = \epsilon \ll 1$ and $\Delta t = \epsilon^2$ into (B.3), it yields

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \frac{1}{\epsilon}\nabla \cdot \mathbf{\Pi}^{(e)} = \left(\frac{1}{\omega} - \frac{1}{2}\right)(\nabla\cdot)^2 \mathsf{Q}^{(e)} + O\left(\epsilon \nabla \cdot \mathbf{\Pi}^{(e)}\right). \tag{B.6}$$

Due to the presence of a mesh-dependent parameter in (B.6), upon convergence, all moments need to be scaled via the following post-process:

$$\bar{\mathbf{u}} = (\mathbf{u} - \mathbf{u}_0)/\epsilon, \qquad \bar{\mathbf{\Pi}}^{(e)} = (\mathbf{\Pi}^{(e)} - \mathbf{\Pi}_0)/\epsilon^2, \qquad \bar{\mathsf{Q}}^{(e)} = (\mathsf{Q}^{(e)} - \mathbf{Q}_0)/\epsilon, \tag{B.7}$$

where $\mathbf{u}_0$, $\mathbf{\Pi}_0$ and $\mathbf{Q}_0$ are constants, with the odd quantities $\mathbf{u}_0$ and $\mathbf{Q}_0$ equal to zero. On the contrary, the even quantity $\mathbf{\Pi}_0 = p_0\mathbf{I}$, where p_0 is the average pressure over the whole computational domain (the average even moment $\mathbf{\Pi}^{(e)}$ over the whole computational domain depends mainly on the amount of mass and only slightly on the flow field).

As a result, the normalized pressure field is defined as $\bar{p} = (p - p_0)/\epsilon^2$ and the normalized density field as $\bar{\rho} = (\rho - \rho_0)/\epsilon^2$ or equivalently $\rho = \rho_0 + \epsilon^2\,\bar{\rho}$. Upon substitution of the latter quantities into (B.2), it follows

$$\nabla \cdot \bar{\mathbf{u}} = O(\epsilon^2). \tag{B.8}$$

Recalling that $\mathbf{\Pi}^{(e)} = \rho\mathbf{u}\mathbf{u} + p\,\mathbf{I}$ and introducing the scaled quantities, we obtain

$$\rho_0 \frac{\partial \bar{\mathbf{u}}}{\partial t} + \rho_0 \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \nabla \bar{p} = \left(\frac{1}{\omega} - \frac{1}{2}\right)(\nabla\cdot)^2 \bar{\mathsf{Q}}^{(e)} + O(\epsilon^2). \tag{B.9}$$

In order to recover the incompressible isothermal fluid dynamics, the quantities $f_i^{(e)}$ are designed [16] such that

$$Q_{ijk}^{(e)} = \frac{\rho}{3}\left(u_i\delta_{jk} + u_j\delta_{ik} + u_k\delta_{ij}\right). \tag{B.10}$$

Consequently

$$\nabla \cdot \nabla \cdot \bar{\mathsf{Q}}^{(e)} = \frac{\rho_0}{3}\nabla^2\bar{\mathbf{u}} + \frac{2\,\rho_0}{3}\nabla\,\nabla \cdot \bar{\mathbf{u}} + O(\epsilon^2) = \frac{\rho_0}{3}\nabla^2\bar{\mathbf{u}} + O(\epsilon^2), \tag{B.11}$$

where the last result is due to Eq. (B.8). Introducing the previous assumption in Eq. (B.9), it yields

$$\boxed{\frac{\partial \bar{\mathbf{u}}}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \frac{1}{\rho_0} \nabla \bar{p} = \nu \nabla^2 \bar{\mathbf{u}} + O(\epsilon^2).} \tag{B.12}$$

where

$$\nu = \frac{1}{3}\left(\frac{1}{\omega} - \frac{1}{2}\right). \tag{B.13}$$

Introducing the previous expression into Eq. (B.2), it yields

$$\frac{\epsilon^2}{\rho_0}\frac{\partial \bar{\rho}}{\partial t} + 6\nu \nabla \cdot \bar{\mathbf{u}} = \frac{\epsilon^2}{2} \nabla \cdot \left(\bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \frac{1}{\rho_0} \nabla \bar{p}\right) + O(\epsilon^4). \tag{B.14}$$

Taking into account the new definition given by (B.13), Eq. (B.8) should be expressed more rigorously as $\nabla \cdot \bar{\mathbf{u}} = O(\epsilon^2/\nu)$. Combining the latter equation and Eq. (B.12), it follows

$$\nabla \cdot \left(\bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \frac{1}{\rho_0} \nabla \bar{p}\right) = O(\epsilon^2/\nu). \tag{B.15}$$

Introducing the above expression into Eq. (B.14), it yields

$$\boxed{\frac{\epsilon^2}{6\rho_0 \nu}\frac{\partial \bar{\rho}}{\partial t} + \nabla \cdot \bar{\mathbf{u}} = O(\epsilon^4/\nu^2).} \tag{B.16}$$

The divergence-free condition for the velocity field requires that $\epsilon^2/\nu \ll 1$, which is consistent with (B.12) as well.

B.2 Acoustic scaling

Assuming $\Delta x = \epsilon \ll 1$ and $\Delta t = \epsilon$, Eq. (B.3) yields

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot \mathbf{\Pi}^{(e)} = \epsilon \left(\frac{1}{\omega} - \frac{1}{2}\right) (\nabla \cdot)^2 \mathsf{Q}^{(e)} + O(\epsilon^2). \tag{B.17}$$

This time, there is no need to scale all the moments and a proper tuning of ω is sufficient instead (see below). Leaving the moments unscaled and substituting the above assumptions in Eq. (B.2), we obtain

$$\frac{\partial \rho}{\partial t} + \left(\frac{2}{\omega} - 1\right) \nabla \cdot (\rho \mathbf{u}) = O(\epsilon). \tag{B.18}$$

The above equation (B.18) proves that acoustic scaling should not be used in link-wise ACM, when dealing with transient simulations, while for steady

state flows we get:

$$\nabla \cdot (\rho \mathbf{u}) = O(\epsilon/\nu), \tag{B.19}$$

showing that, if the density (pressure) gradients are small (consistently with the incompressible limit), the above equation provides indeed an accurate divergence-free velocity field. However, in acoustic (unlike diffusive scaling) the compressibility error cannot be reduced by mesh refinement. Taking into account Eq. (B.10) and Eq. (B.19), we get

$$\nabla \cdot \nabla \cdot \mathsf{Q}^{(e)} = \frac{1}{3}\nabla^2(\rho \mathbf{u}) + O(\epsilon/\nu), \tag{B.20}$$

hence

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla p = \bar{\nu}\nabla^2(\rho \mathbf{u}) + O(\epsilon), \tag{B.21}$$

where $\bar{\nu} = \epsilon\,\nu$. If the density (and pressure) gradients are small, the solution to the system formed by (B.19) and (B.21) provides a reasonable approximation of the Navier-Stokes solution in the incompressible limit. However, the latter system does not asymptotically converge towards the incompressible Navier-Stokes solution, as the mesh get finer and finer (at least, as far as the discretization error is smaller than the compressibility error). This is the main reason why the diffusive scaling is preferred in this work.

B.3 Forcing

Let us consider the forcing step described by Eq. (15), with $\Delta x = \epsilon \ll 1$, $\Delta t = \epsilon^{\beta+1}$ (or equivalently $\mathrm{Ma} = \mathrm{Kn}^{\beta}$) where β is a free parameter ($\beta = 0$ and $\beta = 1$ denote acoustic and diffusive scaling, respectively). The correction due to (15) leads to an additional term in (B.3):

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \frac{\Delta x}{\Delta t}\nabla \cdot \mathbf{\Pi}^{(e)} = \frac{\Delta x^2}{\Delta t}\left(\frac{1}{\omega} - \frac{1}{2}\right)(\nabla\cdot)^2\mathsf{Q}^{(e)} + O\left(\frac{\Delta x^3}{\Delta t}\nabla \cdot \mathbf{\Pi}^{(e)}\right) + \frac{1}{\Delta t}\,\rho \mathbf{g}. \tag{B.22}$$

From the definition $\Delta t \equiv \mathrm{Kn}\,\mathrm{Ma}$ and $\mathbf{u} = \mathrm{Ma}\,\bar{\mathbf{u}}$ (see the previous section), the proper scaling for the forcing term follows:

$$\bar{\mathbf{g}} = \frac{1}{\mathrm{Kn}\,\mathrm{Ma}^2}\,\mathbf{g} = \frac{1}{\epsilon^{2\beta+1}}\,\mathbf{g}. \tag{B.23}$$

A certain physical acceleration $\bar{\mathbf{g}}$ (fixed for a given problem), can be imposed in the numerical code through the mesh-dependent acceleration $\mathbf{g} = \epsilon^{2\beta+1}\,\bar{\mathbf{g}}$.

Under the assumption of negligible viscous heating and conservation of internal energy, in the incompressible limit, the temperature field is governed by an advection-diffusion equation. Let us call T the normalized temperature field such that $T \leq 1$ in all the domain of interest, with the quantities $f_i^{(e)}$ defined such that $p = \rho T/3$.

There were a number of suggestions aiming at enabling thermal fluid-dynamic simulations with the lattice Boltzmann method [37]. Among the most interesting ones, we remind: (a) Increase of the number of velocities and inclusion of higher-order nonlinear terms (in flow velocity) in the equilibrium distribution functions; (b) inclusion of finite difference corrections aiming at the fulfillment of energy conservation on standard lattices [38] and, (c) use of two sets of distribution functions for particle number density (f_i), and energy density (g_i), doubling the number of discrete velocities. Even though the first and the second approaches are preferable from the theoretical point of view, the last one is characterized by a much simpler implementation. When dealing with the incompressible limit, the pressure field is characterized by small variations. However, cases may be experienced where large temperature and density gradients compensate each other. If both pressure gradients and temperature gradients are small, a weak coupling between fluid dynamic and energy equations is realized and the simplified approach (c) discussed above can be adopted. This approach will be further extended below in the framework of the present LW-ACM. Extensions of the approach (b) in the framework of LW-ACM are currently under investigation as well.

The LW-ACM approach for (weak) thermal fluid dynamic simulations makes use of the following system of algebraic equations

$$g_i(\hat{\mathbf{x}}, \hat{t}+1) = g_i^{(e)}(\hat{\mathbf{x}}-\hat{\mathbf{v}}_i, \hat{t}) + 2\left(\frac{\omega_t - 1}{\omega_t}\right)\left(g_i^{(e,e)}(\hat{\mathbf{x}}, \hat{t}) - g_i^{(e,e)}(\hat{\mathbf{x}} - \hat{\mathbf{v}}_i, \hat{t})\right), \quad \text{(B.24)}$$

for $i = 0, \ldots, Q_t - 1$ (the number of lattice velocities for solving the thermal field can be different from that used for solving the velocity field, i.e. $Q_t \neq Q$). The quantities $g_i^{(e)}$ are local functions of $T = \sum_i g_i$ and $T\mathbf{u} = \sum_i \hat{\mathbf{v}}_i g_i$ at the same point $\hat{\mathbf{x}}$ and time $\hat{t}$. The quantities $g_i^{(e)}$ are designed in order to recover the advection-diffusion equation, according to the constraints discussed below. In particular, recovering the advection-diffusion equation requires : $\sum_i g_i^{(e)} = T$ and $\sum_i \hat{\mathbf{v}}_i g_i^{(e)} = T\mathbf{u}$, i.e. the conservation of hydrodynamic moments, and $\sum_i \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i g_i^{(e)} = \mathbf{\Pi}_t^{(e)} = \mathbf{u}\mathbf{u} + T/3\,\mathbf{I}$. On the other hand, the even parts of equilibria $g_i^{(e,e)}$ are defined as

$$g_i^{(e,e)}(T, \mathbf{u}) = \frac{1}{2}\left(g_i^{(e)}(T, \mathbf{u}) + g_i^{(e)}(T, -\mathbf{u})\right). \quad \text{(B.25)}$$

Eq. (B.24) is formulated in terms of $\hat{\mathbf{x}} = \mathbf{x}/\Delta x$ and $\hat{t} = t/\Delta t$. In the following, let us consider the diffusive scaling, namely $\Delta x = \epsilon \ll 1$ and $\Delta t = \epsilon^2$ (see the Appendix B for further details on diffusive scaling). Taylor expansion of Eq. (B.24) yields

$$g_i(t+\epsilon^2) = g_i^{(e)} - \epsilon\, \hat{\mathbf{v}}_i \cdot \nabla g_i^{(e)} + \frac{\epsilon^2}{2}(\hat{\mathbf{v}}_i \cdot \nabla)^2 g_i^{(e)} + \left(2 - \frac{2}{\omega_t}\right)\left(\epsilon\, \hat{\mathbf{v}}_i \cdot \nabla g_i^{(e,e)} - \frac{\epsilon^2}{2}(\hat{\mathbf{v}}_i \cdot \nabla)^2 g_i^{(e,e)}\right) + \ldots, \tag{B.26}$$

where all the quantities are computed in the same point $\mathbf{x}$ and time t and hence this is no more explicitly reported. Summation over i of the equations (B.26) yields

$$\frac{\partial T}{\partial t} + \frac{1}{\epsilon}\nabla \cdot (T\mathbf{u}) + \frac{\epsilon}{6}(\nabla\cdot)^3 \mathbf{Q}_t^{(e)} = \left(\frac{1}{\omega_t} - \frac{1}{2}\right)(\nabla\cdot)^2 \mathbf{\Pi}_t^{(e)} + O\left(\epsilon^2 (\nabla\cdot)^2 \mathbf{\Pi}_t^{(e)}\right), \tag{B.27}$$

with $\mathbf{Q}_t^{(e)} = \sum_i \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i g_i^{(e)}$. In the rightmost term of the expression (B.27), we rely upon the fact that spatial derivatives of the fourth-order moments have the same growth rate of spatial derivatives of the second-order moments. Moreover, in the same term, the diverge is raised to a power which is selected for consistency with the units of other terms. This is unessential, as far as the order of magnitude of the term is concerned.

Similarly to the fluid dynamic equations (see the Appendix B), we apply the following post-processing for scaling all moments (arbitrary constants have been already omitted):

$$\bar{\mathbf{u}} = \mathbf{u}/\epsilon, \qquad \bar{\mathbf{\Pi}}_t^{(e)} = \mathbf{\Pi}_t^{(e)}, \qquad \bar{\mathsf{Q}}_t^{(e)} = \mathsf{Q}_t^{(e)}/\epsilon. \tag{B.28}$$

One possibility for automatic implementation of the latter scaling is to assume

$$(\bar{\mathsf{Q}}_t^{(e)})_{ijk} = \frac{1}{3}\left(\bar{u}_i \delta_{jk} + \bar{u}_j \delta_{ik} + \bar{u}_k \delta_{ij}\right), \tag{B.29}$$

similarly to Eq. (B.10), because thus all terms in $\mathsf{Q}_t^{(e)}$ become proportional to the velocity field (and they are automatically scaled by means of the first of the above scalings).

Moreover, taking into account the second scaling and the definition of $\mathbf{\Pi}_t^{(e)}$ reported in the main text, we conclude that $\bar{\mathbf{\Pi}}_t^{(e)} = \epsilon^2 \bar{\mathbf{u}}\bar{\mathbf{u}} + T/3\,\mathbf{I}$, where T does not need to be scaled (or equivalently $\bar{T} = T$). Substituting the previous scalings into (B.26), and taking into account (B.8), we obtain

$$\boxed{\frac{\partial \bar{T}}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \bar{T} = \alpha \nabla^2 \bar{T} + O(\epsilon^2),} \tag{B.30}$$

where

$$\alpha = \frac{1}{3}\left(\frac{1}{\omega_t} - \frac{1}{2}\right). \tag{B.31}$$

Table B.1
Convergence analysis for thermal Couette flow in case of diffusive scaling, with Prandtl number $\mathrm{Pr} = 0.71$.

Link-wise ACM					
$\epsilon \equiv \Delta x$	$\mathrm{Ma} \propto \Delta t/\Delta x$	$\nu \propto \mathrm{Re}^{-1}$	$\alpha \propto \mathrm{Pe}^{-1}$	Error $L^2[\bar{u}]$	Error $L^2[T]$
1/5	1.11×10^{-2}	5.55×10^{-2}	1.11×10^{-2}	3.10×10^{-3}	1.61×10^{-6}
1/10	5.55×10^{-2}	5.55×10^{-2}	1.11×10^{-2}	8.25×10^{-4}	4.28×10^{-7}
1/20	2.78×10^{-3}	5.55×10^{-2}	1.11×10^{-2}	2.12×10^{-4}	1.01×10^{-7}

Here, a few numerical results are reported for the thermal Couette problem, which is realized by confining a viscous fluid in a gap between two parallel plates. Assuming that the one plate (hot wall located at $y = 0$ and with temperature $\bar{T}_N$) moves in its own plane, whereas the other (cold wall located at $y = 2L$ and with temperature $\bar{T}_S$) is at rest, the controlling parameters are the Prandtl number $\mathrm{Pr} = \nu/\alpha$ (measuring the momentum diffusivity: ν, to heat diffusivity: α) and the Eckert number $\mathrm{Ec} = \bar{u}^2/c_v\Delta\bar{T}$ (measuring the kinetic energy: $\rho\bar{u}^2/2$, to internal energy: $\rho c_v \Delta\bar{T}$). Thermal Couette flow admits the following analytical solution of the temperature field:

$$\bar{T}(y) = \bar{T}_S + \frac{y}{2L}\Delta\bar{T} + \frac{\mathrm{Br}\Delta\bar{T}}{2}\frac{y}{2L}\left(1 - \frac{y}{2L}\right), \tag{B.32}$$

with $\Delta\bar{T} = \left(\bar{T}_N - \bar{T}_S\right)$, and the Brinkman number $\mathrm{Br} = \mathrm{Pr}\,\mathrm{Ec}$.

Diffusive scaling was considered in our simulations, where the velocity field is scaled on meshes with different sizes, keeping fixed the relaxation frequency (see the Appendix B for details). Some preliminary numerical results are reported in Table B.1 for $\mathrm{Pr} = \nu/\alpha = 0.71$.

C Appendix: Computing derivatives locally

LW-ACM allows a straightforward local computation of spatial derivatives. In this respect, a good example is provided by the following strategy for tuning bulk viscosity. Since the LW-ACM (like LBM and ACM) is an artificial compressibility scheme, bulk viscosity can be regarded as a free parameter (if the incompressible limit is the only concern).

One possible strategy for tuning the bulk viscosity is described below. Instead of standard $f_i^{(e)}$ (see Eq. (A.1) in Appendix A) in Eq. (13), we consider a modified set of functions, namely $f_i^{(e*)}$, defined as

$$f_i^{(e*)} = f_i^{(qe)} \left(\rho, \mathbf{u}, \mathrm{Tr}^{(e)} + \gamma \left(\mathrm{Tr}^{(+)} - \mathrm{Tr}^{(e)}\right)\right), \tag{C.1}$$

where $f^{(qe)}(\rho, \mathbf{u}, \mathrm{Tr})$ is given by Eq. (C.1), γ is a free parameter, $\mathrm{Tr}^{(e)}$ is the trace of the tensor $\mathbf{\Pi}^{(e)}$ normalized by the density and $\mathrm{Tr}^{(+)}$ is the trace of the tensor $\mathbf{\Pi}^{(+)} = \sum_i \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i f_i(t + \Delta t)$ normalized again by the density. Let us consider the two dimensional case ($D = 2$): by definition, $\mathrm{Tr}^{(e)} = 2/3 + \mathbf{u}^2$. Substituting the latter into Eq. (C.1) and taking into account the definition of $f_i^{(qe)}$ given by Eq. (A.4), it is possible to compute the second order tensor of $f_i^{(e*)}$, namely

$$\mathbf{\Pi}^{(e*)} = \mathbf{\Pi}^{(e)} + \frac{\gamma}{2} \left(\mathrm{Tr}^{(+)} - \mathrm{Tr}^{(e)}\right) \mathbf{I}. \tag{C.2}$$

At the leading order, the modified equilibrium differs from the standard one only due to even moments.

In order to simplify the last term of the Eq. (C.2), by Eqs. (B.1), we compute the following quantities

$$\mathbf{\Pi}^{(+)} = \mathbf{\Pi}^{(e)} - 6\nu\, \Delta x\, \nabla \cdot \mathsf{Q}^{(e)} + O\left(\Delta x^2 (\nabla \cdot)^2 \mathbf{\Pi}^{(e)}\right), \tag{C.3}$$

where $\mathbf{\Pi}^{(+)} = \sum_i \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i f_i(t + \Delta t)$. Recalling the definition in (B.10), it yields

$$\mathbf{\Pi}^{(+)} = \mathbf{\Pi}^{(e)} - 2\nu\, \Delta x \left(\nabla(\rho \mathbf{u}) + \nabla(\rho \mathbf{u})^T + \nabla \cdot (\rho \mathbf{u})\, \mathbf{I}\right) + O\left(\Delta x^2 (\nabla \cdot)^2 \mathbf{\Pi}^{(e)}\right). \tag{C.4}$$

with its trace taking the form:

$$\mathrm{Tr}^{(+)} = \mathrm{Tr}^{(e)} - 8\nu\, \Delta x\, \nabla \cdot (\rho \mathbf{u}) + O\left(\Delta x^2 (\nabla \cdot)^2 \mathbf{\Pi}^{(e)}\right). \tag{C.5}$$

Considering the diffusive scaling, namely $\Delta x = \epsilon \ll 1$ and $\Delta t = \epsilon^2$ (see Appendix B for further details about the diffusive scaling), the previous expression can be recast as:

$$\bar{\mathrm{Tr}}^{(+)} = \bar{\mathrm{Tr}}^{(e)} - 8\rho_0 \nu\, \nabla \cdot \bar{\mathbf{u}} + O(\epsilon^2). \tag{C.6}$$

Introducing the previous expression into Eq. (C.2) and applying the scaling to the remaining terms, it reads:

$$\bar{\mathbf{\Pi}}^{(e*)} = \bar{\mathbf{\Pi}}^{(e)} - 4\rho_0 \nu\, \gamma\, \nabla \cdot \bar{\mathbf{u}}\, \mathbf{I} + O(\epsilon^2). \tag{C.7}$$

Substituting $\bar{\mathbf{\Pi}}^{(e*)}$ instead of $\bar{\mathbf{\Pi}}^{(e)}$ into Eq. (B.9) and taking into account Eq. (B.11), it yields

$$\rho_0 \frac{\partial \bar{\mathbf{u}}}{\partial t} + \rho_0 \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \nabla \bar{p} = \nu\, \nabla^2 \bar{\mathbf{u}} + \xi\, \nabla\, \nabla \cdot \bar{\mathbf{u}} + O(\epsilon^2), \tag{C.8}$$

where $\xi = 2\rho_0\nu\,(1+2\,\gamma)$ is related to the bulk viscosity. The previous equation is consistent with Eq. (B.12), because the gradient of the divergence of the velocity field is as large as the leading error (hence it is not spoiling the consistency). However, the range $\gamma \geq 0$ is usually beneficial to numerical stability. This strategy enables to increase the bulk viscosity by using the updated distribution function for computing locally all required derivatives (involved in the divergence of the velocity field).

D Appendix: Equivalent finite-difference formulas

Here, we provide some finite-difference formulas fully equivalent to Eq. (13) for a chosen lattice. Let us consider the popular D2Q9 lattice [16] for two dimensional problems ($D = 2$), and consisting of nine discrete velocities ($Q = 9$). The quantities $f_i^{(e)}$ can be explicitly defined for this lattice [16]. They allow to recover the incompressible Euler equations (with $p = \rho/3$), and they are consistent with the property given by (B.11), which is essential for recovering Navier-Stokes equations. The numerical algorithm is fully defined, upon substitution of $f_i^{(e)}$ into the Eq. (13).

According to the finite-difference literature, we define the generic computational stencil by means of cardinal directions. The generic point P with a position vector $\hat{\mathbf{x}} = (n, m)^T$ (the superscript T denotes transposition) is identified by a pair of integers n and m. By means of the subscripts E and W, we denote the neighboring points $(n \pm 1, m)^T$, respectively. Similarly, by means of the subscripts N and S, we mean the neighboring points $(n, m \pm 1)^T$, respectively. Two types of subscripts may be used concurrently for identifying the diagonal points.

Concerning time levels, if not otherwise stated, all quantities are intended as computed at the generic time level $\hat{t}$, with the superscript "+" meaning a quantity at the new time level $\hat{t} + 1$. The unknown quantities are given by the velocity components $\mathbf{u} = (u, v)^T$ and the pressure p (the density for this model is given by $\rho = 3\,p$). Hence the equivalent finite-difference formulas must provide a way to compute u_P^+, v_P^+ and p_P^+.

Applying the definitions of hydrodynamic quantities to Eq. (13), it follows:

$$
\begin{aligned}
p_P^+ u_P^+ = & p_E(-6u_E^2 + 6u_E + 3v_E^2 - 2)/18 + p_W(6u_W^2 + 6u_W - 3v_W^2 + 2)/18 \\
& -p_{NE}(3u_{NE}^2 + 9u_{NE}v_{NE} - 3u_{NE} + 3v_{NE}^2 - 3v_{NE} + 1)/36 \\
& +p_{NW}(3u_{NW}^2 - 9u_{NW}v_{NW} + 3u_{NW} + 3v_{NW}^2 - 3v_{NW} + 1)/36 \\
& -p_{SE}(3u_{SE}^2 - 9u_{SE}v_{SE} - 3u_{SE} + 3v_{SE}^2 + 3v_{SE} + 1)/36 \\
& +p_{SW}(3u_{SW}^2 + 9u_{SW}v_{SW} + 3u_{SW} + 3v_{SW}^2 + 3v_{SW} + 1)/36 \\
& +2/3(1/\omega - 1)\,[p_E u_E - 3p_P u_P + p_W u_W \\
& +(p_{NE}u_{NE} + p_{NW}u_{NW} + p_{SE}u_{SE} + p_{SW}u_{SW})/4 \\
& +(p_{NE}v_{NE} - p_{NW}v_{NW} - p_{SE}v_{SE} + p_{SW}v_{SW})/4]\,,
\end{aligned} \tag{D.1}
$$

$$
\begin{aligned}
p_P^+ v_P^+ = & p_N(3u_N^2 - 6v_N^2 + 6v_N - 2)/18 + p_S(-3u_S^2 + 6v_S^2 + 6v_S + 2)/18 \\
& -p_{NE}(3u_{NE}^2 + 9u_{NE}v_{NE} - 3u_{NE} + 3v_{NE}^2 - 3v_{NE} + 1)/36 \\
& -p_{NW}(3u_{NW}^2 - 9u_{NW}v_{NW} + 3u_{NW} + 3v_{NW}^2 - 3v_{NW} + 1)/36 \\
& +p_{SE}(3u_{SE}^2 - 9u_{SE}v_{SE} - 3u_{SE} + 3v_{SE}^2 + 3v_{SE} + 1)/36 \\
& +p_{SW}(3u_{SW}^2 + 9u_{SW}v_{SW} + 3u_{SW} + 3v_{SW}^2 + 3v_{SW} + 1)/36 \\
& 2/3(1/\omega - 1)\,[p_N v_N - 3p_P v_P + p_S v_S \\
& +(p_{NE}u_{NE} - p_{NW}u_{NW} - p_{SE}u_{SE} + p_{SW}u_{SW})/4 \\
& +(p_{NE}v_{NE} + p_{NW}v_{NW} + p_{SE}v_{SE} + p_{SW}v_{SW})/4]\,.
\end{aligned} \tag{D.2}
$$

$$
\begin{aligned}
p_P^+ = & -2p_P(3u_P^2 + 3v_P^2 - 2)/9 \\
& -p_E(-6u_E^2 + 6u_E + 3v_E^2 - 2)/18 + p_W(6u_W^2 + 6u_W - 3v_W^2 + 2)/18 \\
& -p_N(3u_N^2 - 6v_N^2 + 6v_N - 2)/18 + p_S(-3u_S^2 + 6v_S^2 + 6v_S + 2)/18 \\
& +p_{NE}(3u_{NE}^2 + 9u_{NE}v_{NE} - 3u_{NE} + 3v_{NE}^2 - 3v_{NE} + 1)/36 \\
& +p_{NW}(3u_{NW}^2 - 9u_{NW}v_{NW} + 3u_{NW} + 3v_{NW}^2 - 3v_{NW} + 1)/36 \\
& +p_{SE}(3u_{SE}^2 - 9u_{SE}v_{SE} - 3u_{SE} + 3v_{SE}^2 + 3v_{SE} + 1)/36 \\
& +p_{SW}(3u_{SW}^2 + 9u_{SW}v_{SW} + 3u_{SW} + 3v_{SW}^2 + 3v_{SW} + 1)/36 \\
& +2/3(1/\omega - 1)\,[-p_E u_E + p_S v_S + p_W u_W - p_N v_N \\
& +(p_{SW}u_{SW} + p_{NW}u_{NW} - p_{NE}u_{NE} - p_{SE}u_{SE})/4 \\
& +(p_{SW}v_{SW} - p_{NW}v_{NW} - p_{NE}v_{NE} + p_{SE}v_{SE})/4]\,,
\end{aligned} \tag{D.3}
$$

It is important to stress that the above expressions (D.1), (D.2), (D.3) for the considered lattice model are fully equivalent to (13) up to machine precision, since no asymptotic analysis is requested in their derivation.

From a computational perspective, optimal implementation requires that the number of floating point operations are reduced as much as possible by common subexpression elimination (CSE) [20]. Moreover, for locating the macroscopic quantities (p, u, v) contiguously in the memory, it is possible to collect them in a single array and to use the first index for addressing them, namely $\mathsf{M}(1:3, 1:\mathsf{N_x}, 1:\mathsf{N_y})$ where $\mathsf{N_x} \times \mathsf{N_y}$ is the generic mesh. This leads to an

optimized FD-style implementation. First of all, let us compute the following auxiliary quantities

$$
\begin{aligned}
pu_P &= \mathsf{M}(1,\mathsf{i},\mathsf{j})\,\mathsf{M}(2,\mathsf{i},\mathsf{j}), \qquad \varphi u_P = 2\,pu_P,\\
pv_P &= \mathsf{M}(1,\mathsf{i},\mathsf{j})\,\mathsf{M}(3,\mathsf{i},\mathsf{j}), \qquad \varphi v_P = 2\,pv_P,\\
pv_N &= \mathsf{M}(1,\mathsf{i},\mathsf{j}+1)\,\mathsf{M}(3,\mathsf{i},\mathsf{j}+1), \qquad pv_S = \mathsf{M}(1,\mathsf{i},\mathsf{j}-1)\,\mathsf{M}(3,\mathsf{i},\mathsf{j}-1),\\
pu_E &= \mathsf{M}(1,\mathsf{i}+1,\mathsf{j})\,\mathsf{M}(2,\mathsf{i}+1,\mathsf{j}), \qquad pu_W = \mathsf{M}(1,\mathsf{i}-1,\mathsf{j})\,\mathsf{M}(2,\mathsf{i}-1,\mathsf{j}),\\
pu_{NE} &= \mathsf{M}(1,\mathsf{i}+1,\mathsf{j}+1)\,\mathsf{M}(2,\mathsf{i}+1,\mathsf{j}+1),\\
pu_{NW} &= \mathsf{M}(1,\mathsf{i}-1,\mathsf{j}+1)\,\mathsf{M}(2,\mathsf{i}-1,\mathsf{j}+1),\\
pu_{SE} &= \mathsf{M}(1,\mathsf{i}+1,\mathsf{j}-1)\,\mathsf{M}(2,\mathsf{i}+1,\mathsf{j}-1),\\
pu_{SW} &= \mathsf{M}(1,\mathsf{i}-1,\mathsf{j}-1)\,\mathsf{M}(2,\mathsf{i}-1,\mathsf{j}-1),\\
pv_{NE} &= \mathsf{M}(1,\mathsf{i}+1,\mathsf{j}+1)\,\mathsf{M}(3,\mathsf{i}+1,\mathsf{j}+1),\\
pv_{NW} &= \mathsf{M}(1,\mathsf{i}-1,\mathsf{j}+1)\,\mathsf{M}(3,\mathsf{i}-1,\mathsf{j}+1),\\
pv_{SE} &= \mathsf{M}(1,\mathsf{i}+1,\mathsf{j}-1)\,\mathsf{M}(3,\mathsf{i}+1,\mathsf{j}-1),\\
pv_{SW} &= \mathsf{M}(1,\mathsf{i}-1,\mathsf{j}-1)\,\mathsf{M}(3,\mathsf{i}-1,\mathsf{j}-1),
\end{aligned}
\tag{D.4}
$$

and

$$
\begin{aligned}
\phi_1 &= pu_E\,(-\mathsf{M}(2,\mathsf{i}+1,\mathsf{j})+1) + \mathsf{M}(1,\mathsf{i}+1,\mathsf{j})\,(\mathsf{M}(3,\mathsf{i}+1,\mathsf{j})^2/2 - r_{13}),\\
\phi_2 &= pu_W\,(\mathsf{M}(2,\mathsf{i}-1,\mathsf{j})+1) - \mathsf{M}(1,\mathsf{i}-1,\mathsf{j})\,(\mathsf{M}(3,\mathsf{i}-1,\mathsf{j})^2/2 - r_{13}),\\
\phi_3 &= pv_N\,(-\mathsf{M}(3,\mathsf{i},\mathsf{j}+1)+1) + \mathsf{M}(1,\mathsf{i},\mathsf{j}+1)\,(\mathsf{M}(2,\mathsf{i},\mathsf{j}+1)^2/2 - r_{13}),\\
\phi_4 &= pv_S\,(\mathsf{M}(3,\mathsf{i},\mathsf{j}-1)+1) - \mathsf{M}(1,\mathsf{i},\mathsf{j}-1)\,(\mathsf{M}(2,\mathsf{i},\mathsf{j}-1)^2/2 - r_{13}),\\
\phi_5 &= pu_{NE}\,(\mathsf{M}(2,\mathsf{i}+1,\mathsf{j}+1) + 3\,\mathsf{M}(3,\mathsf{i}+1,\mathsf{j}+1) - 1) + \ldots\\
&\qquad \cdots + pv_{NE}\,(\mathsf{M}(3,\mathsf{i}+1,\mathsf{j}+1) - 1) + \mathsf{M}(1,\mathsf{i}+1,\mathsf{j}+1)\,r_{13},\\
\phi_6 &= pu_{NW}\,(\mathsf{M}(2,\mathsf{i}-1,\mathsf{j}+1) - 3\,\mathsf{M}(3,\mathsf{i}-1,\mathsf{j}+1) + 1) + \ldots\\
&\qquad \cdots + pv_{NW}\,(\mathsf{M}(3,\mathsf{i}-1,\mathsf{j}+1) - 1) + \mathsf{M}(1,\mathsf{i}-1,\mathsf{j}+1)\,r_{13},\\
\phi_7 &= pu_{SE}\,(\mathsf{M}(2,\mathsf{i}+1,\mathsf{j}-1) - 3\,\mathsf{M}(3,\mathsf{i}+1,\mathsf{j}-1) - 1) + \ldots\\
&\qquad \cdots + pv_{SE}\,(\mathsf{M}(3,\mathsf{i}+1,\mathsf{j}-1) + 1) + \mathsf{M}(1,\mathsf{i}+1,\mathsf{j}-1)\,r_{13},\\
\phi_8 &= pu_{SW}\,(\mathsf{M}(2,\mathsf{i}-1,\mathsf{j}-1) + 3\,\mathsf{M}(3,\mathsf{i}-1,\mathsf{j}-1) + 1) + \ldots\\
&\qquad \cdots + pv_{SW}\,(\mathsf{M}(3,\mathsf{i}-1,\mathsf{j}-1) + 1) + \mathsf{M}(1,\mathsf{i}-1,\mathsf{j}-1)\,r_{13},
\end{aligned}
\tag{D.5}
$$

where $r_{13} = 1/3$. By means of the quantities (D.4) and (D.5), it is possible to compute the following additional quantities

$$
\begin{aligned}
\phi_{Pp} &= pu_P + pv_P, \qquad \phi_{Pm} = pu_P - pv_P,\\
\phi_{NEp} &= pu_{NE} + pv_{NE}, \qquad \phi_{SWp} = pu_{SW} + pv_{SW},\\
\phi_{NWm} &= pu_{NW} - pv_{NW}, \qquad \phi_{SEm} = pu_{SE} - pv_{SE},\\
\Phi_{NE} &= \phi_{NEp} - \phi_{Pp}, \qquad \Phi_{SW} = \phi_{Pp} - \phi_{SWp},\\
\Phi_{NW} &= \phi_{NWm} - \phi_{Pm}, \qquad \Phi_{SE} = \phi_{Pm} - \phi_{SEm},\\
\Phi_{NWSE} &= \Phi_{NW} - \Phi_{SE}, \qquad \Phi_{NESW} = \Phi_{NE} - \Phi_{SW}.
\end{aligned}
\tag{D.6}
$$

Finally, auxiliary quantities are used in computing the updating formulas:

$$
\begin{aligned}
\rho_P^+ &= \mathsf{M}(1,\mathsf{i},\mathsf{j})\, r_{43} - \varphi u_P\, \mathsf{M}(2,\mathsf{i},\mathsf{j}) - \varphi v_P\, \mathsf{M}(3,\mathsf{i},\mathsf{j}) + \phi_2 - \phi_1 + \phi_4 - \phi_3 + \dots \\
&\quad \dots + (\phi_5 + \phi_6 + \phi_7 + \phi_8)/4 - b_4\,(\Phi_{NW} + \Phi_{SE} - \Phi_{NE} - \Phi_{SW}) + \dots \\
&\quad \dots + b\,(pu_E - pu_W + pv_N - pv_S), \\
p_P^+ &\equiv \mathsf{M}^+(1,\mathsf{i},\mathsf{j}) = \rho_P^+/3, \qquad \text{(D.7)} \\
u_P^+ &\equiv \mathsf{M}^+(2,\mathsf{i},\mathsf{j}) = (\phi_1 + \phi_2 + (\phi_6 - \phi_5 - \phi_7 + \phi_8)/4 + \dots \\
&\quad \dots - a\,(pu_E - \varphi u_P + pu_W) - b_4\,(\Phi_{NESW} + \Phi_{NWSE}))/\rho_P^+, \qquad \text{(D.8)} \\
v_P^+ &\equiv \mathsf{M}^+(3,\mathsf{i},\mathsf{j}) = (\phi_3 + \phi_4 + (\phi_7 + \phi_8 - \phi_5 - \phi_6)/4 + \dots \\
&\quad \dots - a\,(pv_N - \varphi v_P + pv_S) - a_4\,(\Phi_{NESW} - \Phi_{NWSE}))/\rho_P^+, \qquad \text{(D.9)}
\end{aligned}
$$

where $r_{43} = 4/3$, $b = 2 - 2/\omega$, $a_4 = a/4$, $b = 2 - 2/\omega_p$ and $b_4 = b/4$. The previous optimized formulas are consistent with Eqs. (D.1-D.3) in case $\omega_p = \omega$, $b = a$ and $b_4 = a_4$. Kinematic viscosity is controlled via the parameter ω according to (12), whereas the parameter ω_p is responsible of the artificial compressibility. In particular, if $\omega_p \neq \omega$, the first term in Eq. (B.16) becomes proportional to ϵ^2/ν_p, where ν_p is the value obtained by using ω_p in Eq. (12). Without additional computational costs, a proper choice of $\nu_p > \nu$ allows to reduce the compressibility error in case of under-resolved simulations.

References

[1] V.D. Liseikin. Grid Generation Methods. *Series: Scientific Computation*, Springer, 2nd ed., 2010.

[2] M.P. Kirkpatrick, S.W. Armfield and J.H. Kent. A representation of curved boundaries for the solution of the NavierStokes equations on a staggered three-dimensional Cartesian grid. *J. Comput. Phys.*, 184:136, 2003.

[3] D.P. Young, R.G. Melvin, M.B. Bieterman, F.T. Johnson, S.S. Samant and J.E. Bussoletti. A locally refined rectangular grid finite element method: application to computational fluid dynamics and computational physics. *J. Comput. Phys.*, 92:166, 1991.

[4] D.D. Zeeuw and K.G. Powell. An adaptively refine Cartesian mesh solver for the Euler equation. *J. Comput. Phys.*, 104:5668, 1993.

[5] T. Ye, R. Mittal, H.S. Udaykumar and W. Shyyy. An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries. *J. Comput. Phys.*, 156:209240, 1999.

[6] D.M. Ingram, D.M. Causon and C.G. Mingham. Developments in Cartesian cut cell methods. *Math. Comp. Simul.*, 61:561572, 2003.

[7] X.L. Luo, Z.L. Gu, K.B. Lei, S. Wang and K. Kase. A three-dimensional Cartesian cut cell method for incompressible viscous flow with irregular domains. *Int. J. Numer. Meth. Fluids*, in press , 2011.

[8] S. Succi. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. *Oxford University Press*, New York, 2001.

[9] A.J. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.*, 2:12-26, 1976.

[10] M. Junk, A. Klar, L.-S. Luo. Asymptotic analysis of the lattice Boltzmann equation. *J. Comput. Phys.*, 210(2):676-704, 2005.

[11] T. Inamuro. A lattice kinetic scheme for incompressible viscous flows with heat transfer. *Proc. R. Soc. Lond. A*, 360: 210(1792):477-484, 2002.

[12] T. Ohwada, P. Asinari. Artificial compressibility method revisited: Asymptotic numerical method for incompressible NavierStokes equations. *J. Comp. Physics*, 229:16981723, 2010.

[13] D. d'Humières. Generalized lattice Boltzmann equations. in: B.D. Shizgal, D.P. Weaver (Eds.), *Rarefied Gas Dynamics: Theory and Simulations, in: Prog. Astronaut. Aeronaut.*, 159:450, 1992.

[14] T. Ohwada, P. Asinari and D. Yabusaki. Artificial compressibility method and lattice Boltzmann method: Similarities and differences *Comput. Math. Appl.*, in press, 2011.

[15] P.J. Dellar. Nonhydrodynamic modes and a priori construction of shallow water lattice Boltzmann equations. *Physical Review E*, 65:036309, 2002.

[16] Y.H. Qian, D. d'Humières, P. Lallemand. Lattice BGK models for the NavierStokes equation. *Europhys. Lett.*, 17:479484, 1992.

[17] P. Asinari, T. Ohwada. Connection between kinetic methods for fluid-dynamic equations and macroscopic finite-difference schemes. *Comput. Math. Appl.*, 58: 841-861, 2009.

[18] Z. Guo, C. Zheng, and B. Shi. Lattice Boltzmann equation with multiple effective relaxation times for gaseous microscale flow. *Physical Review E*, 77:036707, 2008.

[19] G. Wellein, T. Zeiser, S. Donath and G. Hager. On the single processor performance of simple lattice Boltzmann kernels. *Comp. Fluids*, 35: 910-919, 2006.

[20] G. Hager and G. Wellein. G. Wellein, T. Zeiser, S. Donath and G. Hager. Introduction to High Performance Computing for Scientists and Engineers. *CRC Press*, 2010.

[21] P. Lallemand, L.-S. Luo. Theory of the lattice Boltzmann method: dispersion, dissipation, isotropy, Galilean invariance, and stability. *Physical Review E*, 61:6546, 2000.

[22] M.L. Minion and D.L. Brown. Performance of underresolved two-dimensional incompressible flow simulations II. *J. Comput. Phys.*, 138:734, 1997.

[23] P.J. Dellar. Bulk and shear viscosities in lattice Boltzmann equations. *Physical Review E*, 64:031203, 2001.

[24] R.A. Brownlee, A.N. Gorban and J. Levesley. Nonequilibrium entropy limiters in lattice Boltzmann methods. *Physica A: Statistical Mechanics and its Applications*, 387:385-406, 2008.

[25] P. Asinari and I.V. Karlin. Quasiequilibrium lattice Boltzmann models

with tunable bulk viscosity for enhancing stability. *Physical Review E*, 81:016702, 2010.

[26] M. Bouzidi, M. Firdaouss and P. Lallemand. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of fluids*, 13(11), 2001.

[27] A.J.C. Ladd. Numerical simulations of particular suspensions via a discretized Boltzmann equation - Part 1. *J. Fluid Mech.*, 271:285-310, 1994.

[28] E. Lorenz, A. Caiazzo and A.G. Hoekstra Corrected momentum exchange method for lattice Boltzmann simulations of suspension flow. *Phys. Review E*, 79:036705, 2009.

[29] A. Caiazzo, *Ph.D. thesis*, Scuola Normale Superiore, Pisa, Italy, and Technische Universit at Kaiserslautern, Germany, 2007.

[30] C.-H. Bruneau, M. Saad. The 2D lid-driven cavity problem revisited. *Computers & Fluids*, 35:326-348, 2006.

[31] M. Sahin, R.G. Owens. A novel fully-implicit finite volume method applied to the lid-driven cavity problem. Part I and II. *Int J Numer Methods Fluids*, 42(1), 2003.

[32] L.S. Luo, W. Liao, X. Chen, Y. Peng, and W. Zhang. Numerics of the lattice Boltzmann method: Effects of collision models on the lattice Boltzmann simulations. *Phys. Rev. E*, 83:056710, 2011.

[33] A. Povitsky. High-incidence 3-D Lid-driven Cavity Flow. *American Institute of Aeronautics and Astronautics*, 01-2847:1–7, 2001.

[34] D. d'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo. Multiple-relaxation-time lattice Boltzmann models in three dimensions. *Phil. Trans. R. Soc. Lond. A*, 360:437-451, 2002.

[35] A. Povitsky. Three-dimensional flow in cavity at yaw. *Nonlinear Analysis*, 63:e1573–e1584, 2005.

[36] P. Asinari, I.V. Karlin. Generalized Maxwell state and H theorem for computing fluid flows using the lattice Boltzmann method. *Physical Review E*, 79:036703, 2009.

[37] P. Lallemand, L.-S. Luo. Theory of the lattice Boltzmann method: Acoustic and thermal properties in two and three dimensions. *Physical Review E*, 68:036706, 2003.

[38] N.I. Prasianakis, I.V. Karlin. Lattice Boltzmann method for thermal flow simulation on standard lattices. *Physical Review E*, 76:016702, 2007.