High-order semi-Lagrangian kinetic scheme for compressible turbulence

Dominik Wilde [1,2,*] Andreas Krämer [3] Dirk Reith [2,4] and Holger Foysi [1]

[1]*Department of Mechanical Engineering, University of Siegen, Paul-Bonatz-Straße 9-11, 57076 Siegen-Weidenau, Germany*
[2]*Institute of Technology, Resource and Energy-efficient Engineering (TREE), Bonn-Rhein-Sieg University of Applied Sciences, Grantham-Allee 20, 53757 Sankt Augustin, Germany*
[3]*Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 6, 14195 Berlin, Germany*
[4]*Fraunhofer Institute for Algorithms and Scientific Computing (SCAI), Schloss Birlinghoven, 53754 Sankt Augustin, Germany*

Turbulent compressible flows are traditionally simulated using explicit time integrators applied to discretized versions of the Navier-Stokes equations. However, the associated Courant-Friedrichs-Lewy condition severely restricts the maximum time-step size. Exploiting the Lagrangian nature of the Boltzmann equation's material derivative, we now introduce a feasible three-dimensional semi-Lagrangian lattice Boltzmann method (SLLBM), which circumvents this restriction. While many lattice Boltzmann methods for compressible flows were restricted to two dimensions due to the enormous number of discrete velocities in three dimensions, the SLLBM uses only 45 discrete velocities. Based on compressible Taylor-Green vortex simulations we show that the new method accurately captures shocks or shocklets as well as turbulence in 3D without utilizing additional filtering or stabilizing techniques other than the filtering introduced by the interpolation, even when the time-step sizes are up to two orders of magnitude larger compared to simulations in the literature. Our new method therefore enables researchers to study compressible turbulent flows by a fully explicit scheme, whose range of admissible time-step sizes is dictated by physics rather than spatial discretization.

I. INTRODUCTION

One major challenge in fluid dynamics is the study of compressible turbulent flows, involving intrinsic as well as variable density compressibility effects [1–7]. Applications range from aviation [8] or astrophysics [9] to the investigation of canonical flows like boundary layers [10], channel flow [11–13], mixing layers [14–18], jets and aeroacoustics [19–21], or shock-turbulence interaction [22], to mention a few, considering the vast literature available. These flows feature both solenoidal and dilatational structures, which constantly interact and possibly cause shock waves [5,23].

Numerical simulations have become an indispensable tool to understand their physics, and many studies exploring compressible turbulent flows have been conducted using high-order compact finite difference, optimized dispersion-relation preserving schemes [19,24–30] for the spatial derivatives, often combined with low-dispersion-dissipation Runge-Kutta schemes for time-integration [19,31,32]. Although these methods provide accurate results, the time steps are generally small [33], because of the methods' Eulerian time derivatives, which describe how the variables of interest pass through fixed locations in the field. Thus, the admissible time-step sizes are tightly linked to spatial resolution. This issue is for many discretizations linked to the Courant-Friedrichs-Lewy (CFL) condition,

$$c\delta_t/\delta_x < \text{CFL}_{\text{max}}, \qquad (1)$$

using linear stability theory, relating a characteristic velocity c to the spatial and temporal discretization intervals

δ_x and δ_t, respectively (see Ref. [34], for example). Though implicit time integration schemes often provide larger stability domains, their application can be unfeasible for transient problems due to their computational cost. Explicit time integration schemes with scheme-specific CFL_{max}, by contrast, enforce small time steps δ_t for high-flow velocities, typically occurring in many high-speed compressible flows. Another obvious way to circumvent the CFL condition in Eq. (1) is to incorporate Lagrangian time derivatives, which track the motion of the variables of interest moving through the domain.

In practice, Semi-Lagrangian (SL) schemes are used instead, which provide a viable alternative to the discretization of Eulerian time derivatives. SL schemes discretize the Lagrangian solution by tracking the trajectories back in time. The prefix "semi" indicates that the trajectories' end points usually do not coincide with the simulation grid points, which requires application of an appropriate interpolation scheme. SL methods were successfully incorporated in algorithms solving the Navier-Stokes equations [35], although tracking of the fluid trajectories was often found to be cumbersome, introducing additional errors [36]. The major advantage when using SL schemes in kinetic theory is that the trajectories are *linear*, resulting in cancellation of the tracking error. Consequently, SL schemes were both applied to the Vlasov equation [37–39] and to the Bhatnagar-Gross-Krook (BGK)-Boltzmann equation [40,41]. Recently, we introduced the semi-Lagrangian lattice Boltzmann method (SLLBM) [42,43] for compressible flows [44], which solves the lattice Boltzmann equation using a high-order SL streaming step.

---

*wilde.aerospace@gmail.com

In this article, we explore the capabilities of the SLLBM for three-dimensional compressible flows. Furthermore, we demonstrate that the SLLBM remains stable for time-step sizes that exceed typical CFL constraints of Eulerian solvers by orders of magnitude. To yield a lean scheme, the SLLBM is combined with state-of-the-art cubature rules for the velocity discretization [45–47]. This combination proves capable of modeling compressible turbulence with time steps that are at least one order of magnitude larger than in standard Eulerian methods and *decouple* the spatial from the temporal discretization.

Background

We start with a critical look at the more specialized lattice Boltzmann method (LBM) [48]. Despite the successes of the standard LBM in the computation of multiphase [49], particle-laden [50], thermal [51], or turbulent flows [52,53], compressible LBM [54] were overlooked for a relatively long time, but regained attraction during the last decade [55–68]. Let us recall the force-free BGK-Boltzmann equation

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \boldsymbol{\nabla} f = -\frac{1}{\lambda}(f - f^{\text{eq}}), \tag{2}$$

with the continuous distribution function f, the equilibrium distribution function f^{eq}, the particle velocity $\boldsymbol{\xi}$, and the relaxation time λ. To discretize Eq. (2), the original LBM is based on three key principles. First, the equilibrium distribution function f^{eq} is polynomially expanded into a series of Hermite polynomials $\mathcal{H}^{(n)}$, with expansion coefficients being the equilibrium moments $\mathbf{a}_{\text{eq}}^{(n)}$ [69],

$$f^{\text{eq}} \approx \omega(\boldsymbol{\xi}) \sum_{n=0}^{N} \frac{1}{n!} \boldsymbol{a}_{\text{eq}}^{(n)} : \boldsymbol{\mathcal{H}}^{(n)}, \tag{3}$$

where N is the expansion order and $\omega(\boldsymbol{\xi})$ the weight function. Since $\boldsymbol{a}_{\text{eq}}^{(n)}$ and $\boldsymbol{\mathcal{H}}^{(n)}$ are symmetric tensors of rank n, the product involves contraction to all D^n scalar components, depending on dimension D. Second, a Gauß-Hermite quadrature is applied to the unbounded velocity space of the Boltzmann equation, leading to discrete particle velocity sets [69]. The moments are then found by the quadrature

$$\boldsymbol{a}^{(n)} = \int_{\mathbb{R}^D} \omega(\boldsymbol{\xi}) \frac{f}{\omega(\boldsymbol{\xi})} \boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi} = \sum_{i=0}^{Q-1} f_i \boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi}_i), \tag{4}$$

with the weighted discrete distribution functions $f_i = w_i f(\boldsymbol{\xi}_i)/\omega(\boldsymbol{\xi}_i)$. The combination of Q discrete particle velocities $\boldsymbol{\xi}_i$ and weights w_i, the velocity set, is usually derived by the Gauß-product rule applied to a one-dimensional Gauß-Hermite quadrature. Third, the discrete Boltzmann equation is integrated along characteristics with an inherent Lagrangian discretization of the Boltzmann equation's material derivative to obtain a stable numerical scheme and second-order temporal convergence [70].

Unfortunately, the LBM in its original formulation is mainly restricted to Cartesian grids and velocity discretizations that match the regular lattices. The customary "D2Q9" based on second-order expansion in Eq. (3) is plagued

by a cubic error being proportional to the Mach number. Consequently, compressible simulations either demand correction terms that nihilate the errors and restore Galilean invariance [64,71], hybrid LBMs with separate temperature fields [72,73], or higher-order discretizations of Eqs. (3) and (4). Yet the resulting high-order velocity sets are only compact when the abscissae may reside off-lattice, and utilization of such velocity sets therefore requires an efficient off-lattice Boltzmann solver. Previous Eulerian off-lattice Boltzmann schemes [74–79], like finite difference or finite volume LBM, would be suited in principle. However, they sacrifice the Lagrangian time integration along characteristics. Moreover, their time step is severely restricted by a CFL condition of the streaming step, Eq. (1), with respect to the fast discrete particle velocities.

In contrast, the SLLBM preserves all of the aforementioned key principles of the LBM but it also decisively extends its capabilities. Here, the SLLBM is in the spirit of a work by Pavlo *et al.* for thermal flows who used an interpolation-based LBM with a D2Q17 velocity set [80]. In previous works [42–44] we have shown that a high-order interpolation increases the spatial order of the method and reduces net mass fluctuations to an acceptable level. Also, we have demonstrated the unconditional stability of the advection step, when incorporating Gauß-Lobatto-Chebyshev nodes for the interpolation up to third order, and that the stability is practically not affected even with fourth order. The flexibility in terms of meshing and velocity sets encouraged us to search for efficient quadrature rules solving the weight function. This research led us to long-established cubature rules [45–47], i.e., multivariate quadratures, which are often used in Kalman filters, e.g., in Ref. [81].

II. METHODOLOGY

A. Compressible semi-Lagrangian LBM

The compressible SLLBM uses the established lattice Boltzmann equation with the BGK collision operator [42]

$$h_i(\mathbf{x} + \delta_t \boldsymbol{\xi}_i, t + \delta_t) = h_i(\mathbf{x}, t) - \frac{1}{\tau} \big[h_i(\mathbf{x}, t) - h_i^{\text{eq}}(\mathbf{x}, t) \big]. \tag{5}$$

Here, h_i denotes either f_i or the second distribution function g_i related to the variable heat capacity ratio γ. The shifted dimensionless relaxation parameter $\tau = \lambda/\delta_t + 0.5 = \mu/(P\delta_t) + 0.5$ depends on dynamic viscosity μ, and pressure $P = \rho T$ with density ρ and temperature T. The discrete equilibrium distribution function f_i^{eq} is

$$f_i^{\text{eq},N}(\mathbf{x}, t) = w_i \sum_{n=0}^{N} \frac{1}{n!} \boldsymbol{a}_{eq}^{(n)}(\mathbf{x}, t) : \boldsymbol{\mathcal{H}}_i^{(n)}, \tag{6}$$

and $g^{\text{eq}} = (2C_v - D)T f_i^{\text{eq}}$, with heat capacity at constant volume C_v, and number of dimensions D. Both $\boldsymbol{a}_{eq}^{(n)}$ and $\boldsymbol{\mathcal{H}}_i^{(n)} := \boldsymbol{\mathcal{H}}^{(n)}(\boldsymbol{\xi}_i)$ are listed in Appendices A and B, respectively. To adjust the heat conductivity, a quasiequilibrium approach [82] is applied to Eq. (5), for more details see Appendix D.
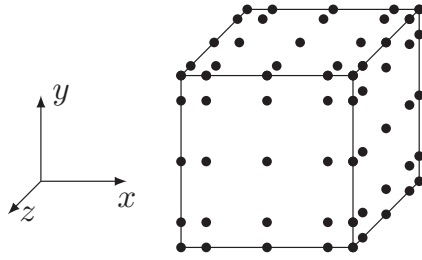
FIG. 1. Support points of a three-dimensional reference cell with Gauß-Lobatto-Chebyshev points of order $p = 4$. Interior and rear points are not shown.

Density ρ, momentum ρu, and energy E are determined by

$$\rho = \sum_{i=0}^{Q-1} f_i, \quad \rho \mathbf{u} = \sum_{i=0}^{Q-1} f_i \boldsymbol{\xi}_i, \quad 2\rho E = \sum_{i=0}^{Q-1} (f_i |\boldsymbol{\xi}_i|^2 + g_i). \quad (7)$$

Appendix C demonstrates the connection of the present SLLBM model with the macroscopic equations using a Chapman-Enskog analysis [83]. Note that the analysis leaves out the interpolation, which would lead to source terms in the mass, momentum and energy equations due to the error introduced by the interpolation procedure. The combination of an integration along characteristics of the streaming step and a Crank-Nicolson scheme for the collision step incorporates a second-order temporal error [84], whose order can be increased by multistep schemes [85]. In standard LBMs, the particle velocities $\boldsymbol{\xi}_i$ in Eq. (5) are designed to end up on one of the neighboring nodes, and the time-step size is invariably set to unity for the same purpose. By contrast, the SLLBM's particle distribution functions are still integrated along characteristics, but the departure points may reside offside the grid, i.e., they are off-lattice. To recover the off-lattice values an interpolation is needed. While several interpolation strategies are possible, we chose a cell-oriented approach, which means that once a departure point is identified, the degrees of freedom points $\hat{h}_{i\Xi j}$ in the enclosing cell are used for the interpolation

$$h_i(\mathbf{x}, t) = \sum_{j=1}^{N_j} \hat{h}_{i\Xi j}(t) \psi_{\Xi j}(\mathbf{x}) \quad (8)$$

in cell Ξ, with the number of support points per cell N_j and with the basis functions $\psi_{\Xi j}$. A three-dimensional reference cell with polynomial order $p = 4$ is shown in Fig. 1. For each of the N support points in the simulation, there are Q particle velocities, i.e., there are $N \cdot Q$ departure points to be identified. Therefore, at the beginning of the simulation the path from each support point to the corresponding Q departure points is tracked through all adjacent cells. Then a sparse matrix $\boldsymbol{\Psi}$ stores the shape function values $\psi_{\Xi j}$ belonging to the departure point's position; the algorithm is presented in Ref. [43]. The actual streaming step is expressed as a matrix-vector multiplication

$$\boldsymbol{h}_i(t + \delta_t) = \boldsymbol{\Psi}_i \boldsymbol{h}_i(t), \quad (9)$$

whereas the collision step remains local. For the realization we have wrapped our semi-Lagrangian lattice Boltzmann
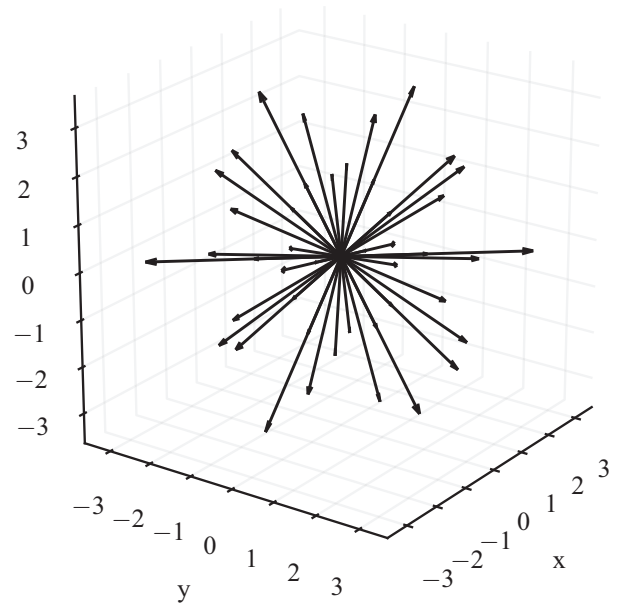

FIG. 2. Three-dimensional D3Q45 velocity set with 45 abscissae, derived by a degree-nine cubature rule originally by Konyaev [89].

solver *NATriuM* [43] around the finite element software package *deal.ii* [86] for mesh, cell, and matrix organization.

B. Cubature-based velocity sets

The discretization of the velocity space is a key principle for any simulation with the lattice Boltzmann method. If a quadrature is to be applied, it must be suited to integrate the weight function $\exp(-x^2)$, and it has to be of ninth degree to enable compressible flow simulations [87]. A prominent method to derive two- and three-dimensional velocity sets is the Gauß-product rule applied to a one-dimensional quadrature. Application to the one-dimensional degree-nine Gauß-Hermite quadrature delivers a two-dimensional D2Q25 off-lattice velocity set with 25 abscissae, which we used for previous work [44]. Due to its structure, this velocity set is infeasible for standard on-lattice streaming but perfectly suited for the SLLBM.

For the simulations in this work we used velocity sets derived by cubature rules [45,46], exhibiting the same degree of precision but consisting of fewer support points to lower computational cost. In two dimensions we employed the degree-nine D2Q19 velocity set with 19 abscissae [88] that we have presented in recent work [47]. In three dimensions, the Gauß-product rule led to noncompetitive 125 abscissae for a three-dimensional degree-nine D3Q125 velocity set. Therefore, we successfully identified and implemented a degree-nine D3Q45 velocity set with only 45 components, which fulfills all requirements in terms of symmetry. The D3Q45 was derived by a cubature rule after Konyaev [89] and has recently been listed in the supplemental material of a cubature article by van Zandt [90]. The resulting discrete velocities are shown in Fig. 2, whereas weights and abscissae of the D2Q19 and D3Q45 are listed in Ref. [47].
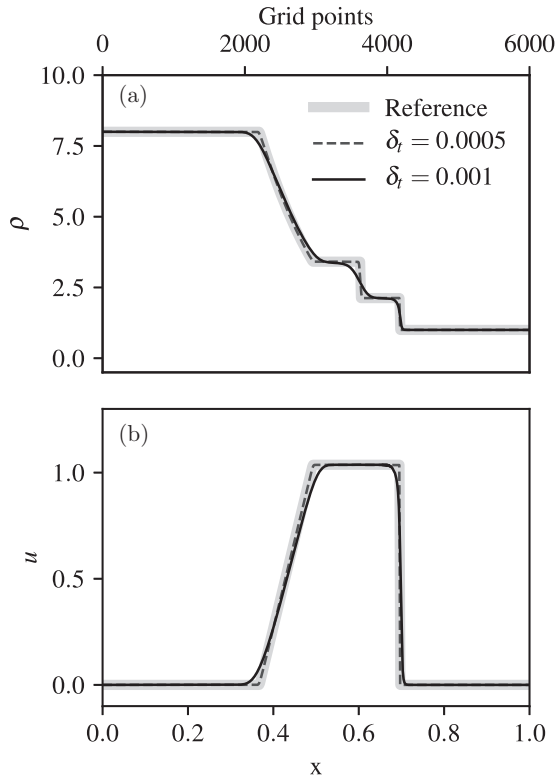
FIG. 4. Exemplary departure point location for the Sod shock tube configuration in this article with 2000 cells. The path of the longest abscissa of the 2D velocity set D2Q19 is shown. Starting from the current cell, the abscissa's path linearly traverses six contiguous cells and locates the departure point in the seventh cell. A third-order polynomial interpolation using the cell-local Gauß-Lobatto-Chebyshev support points is applied to reconstruct the distribution function value.

FIG. 3. Density (a) and velocity (b) of Sod shock tube simulation using SLLBM with 2000 cells at time $t = 0.1$, which is reached after only 100 time steps ($\delta_t = 0.001$), despite the fine spatial resolution of 6000 grid points. For a smaller time-step size and lower viscosity the reference is perfectly matched.

III. RESULTS

We test the proposed method through three test cases. The first two test cases, a temporally underresolved Sod shock tube and a two-dimensional Riemann problem, demonstrate the effect of large time-step sizes on the simulations. The third test case is the compressible three-dimensional Taylor-Green vortex at different Reynolds and Mach numbers to present the capability of SLLBM to simulate both turbulent and shocked three-dimensional flows.

A. Temporally underresolved Sod shock tube

The Sod shock tube illustrates the large range of time-step sizes accessible to the SLLBM. The domain $x \in [0, 1]$ was divided into two regions at $x = 0.5$ with initially

$$\rho_0 = 8, u_0 = 0, P_0 = 10, \quad \rho_1 = 1, u_1 = 0, P_1 = 1,$$

and a viscosity of $\mu = 7 \times 10^{-4}$. The domain $x \in [0, 1]$ was discretized using 2000 cells at third polynomial order, i.e., 6000 grid points in x direction. Despite this fine spatial resolution, which we only chose for demonstration purposes, the time-step size was set to $\delta_t = 0.001$, such that the solution at t = 0.1 was reached by performing only 100 time steps. Figure 3 shows that the SLLBM accurately traces the shock front despite the extremely large time step. For a smaller time-step size of $\delta_t = 0.0005$ and viscosity of $\mu = 10^{-5}$ the simulation results perfectly matched the inviscid reference

solution. To get an impression of the time-step size of other solvers, we repeated the simulations with the larger viscosity $\mu = 7 \times 10^{-4}$ using the same collision process and velocity discretization, but this time applying a spectral-element discontinuous Galerkin solver for the streaming step [43,77]. This solver also features high-order solutions, but requires a dedicated time integrator and the time-step size is bounded by the CFL condition of the streaming step (1). The simulation produced nearly identical results as the SLLBM with $\delta_t = 0.001$ (therefore not shown), but required $\delta_t = 0.000086$ to be numerically stable, i.e., 1154 time steps with an explicit exponential time integrator [91] and $\delta_t = 0.00005$, i.e., 2200 time steps with the more common fourth-order Runge-Kutta method.

As the distance of the departure points from the active nodes is proportional to the time-step size [Eq. (5)], the departure points of the SLLBM were located up to seven cells away. The trajectory is shown in Fig. 4 for an exemplary departure point. It is obvious that the CFL restriction of explicit Eulerian solvers prohibits the exchange of information crossing multiple cells during the streaming step. This property is of special interest in the case of simulations with body-fitted meshes, where the spatial extent of the smallest cells usually dictates the time-step size of the whole simulation. As opposed to Eulerian solvers, the maximum stable time-step size in the SLLBM is proportional only to the viscosity and not dictated by the mesh size. On top of that, when doubling the number of cells the number of SLLBM time steps can still be kept constant, whereas it inevitably doubles for the explicit discontinuous Galerkin solver. Finally, Fig. 5 confirms that the SLLBM is also capable to stably simulate the shock tube with a lower resolution of 100 cells, polynomial order $p = 4$ and 400 grid points, without additional numerical dissipation measures. Here, the time-step size was set to $\delta_t = 0.005$ and $\delta_t = 0.0005$ with viscosities $\mu = 0.002$ and $\mu = 0.0002$, respectively.

B. 2D Riemann study of time-step size effects

Case 12 of the two-dimensional Riemann problems was intensively studied by Lax [92] as well as Kurganov and Tadmor [93]. In one of our last works [44] we already showed that the SLLBM is capable to resolve the density contours of this test case with good visual agreement to the references. This time, similar to the shock tube test case in Sec. III, we examined the effect of the time-step size onto this test case.
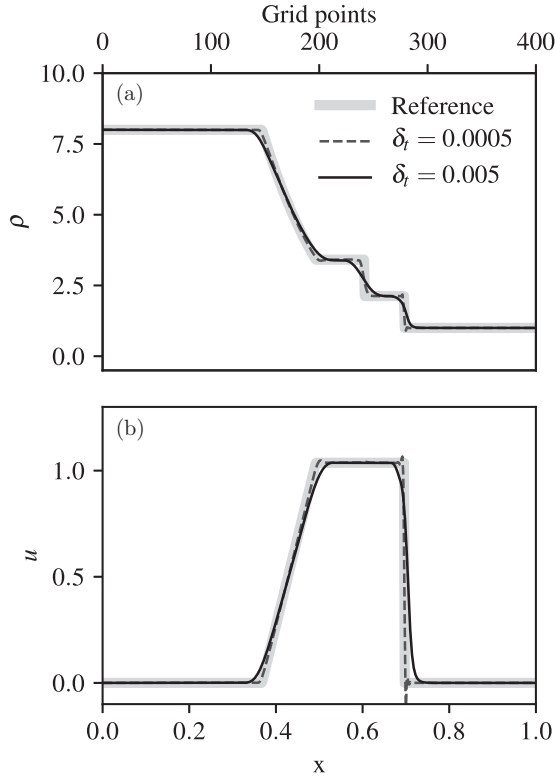
FIG. 5. (a) Density and (b) velocity of Sod shock tube simulation using SLLBM with 100 cells at time $t = 0.1$, which is reached after 20 time steps ($\delta_t = 0.005$), at a more reasonable resolution of 400 grid points. Simulations with smaller time-step sizes and viscosity match the reference better, despite slight overshoots visible near the shock fronts.

The initial conditions of the four quadrants are [79]

$$(\rho, u_x, u_y, P) = \begin{cases} (0.5313, 0, 0, 0.4), & x > 0, \quad y > 0, \\ (1, 0.7276, 0, 1), & x \leqslant 0, \quad y > 0, \\ (0.8, 0, 0, 1), & x \leqslant 0, \quad y \leqslant 0, \\ (1, 0, 0.7276, 1), & x > 0, \quad y \leqslant 0. \end{cases}$$
$$\text{(10)}$$

The heat capacity ratio was $\gamma = 1.4$, the number of cells was $N_{\Xi} = 128^2$ with polynomial order $p = 4$, i.e., $N = 512^2$ grid points. The simulation end time was $t_{\text{end}} = 0.25$. Figure 6 shows the density contours in the interval $\rho \in [0.412, 1.753]$, beginning with a small time-step size and low viscosity in simulation [Fig. 6(a)]. Each of the figures also lists the time-step size and viscosity. When increasing the time-step size, the viscosity also needs to be increased to ensure stable simulations as depicted for simulation [Fig. 6(b)]. The effect of a maladjusted time-step size is shown in simulation [Fig. 6(c)], where oscillations occur in the top right corner near the shock fronts. Finally, simulation [Fig. 6(d)] depicts the density contours for a approximately 50 times larger time-step size than for the case in Fig. 6(a). In this case, the shock fronts are widened, comparable to the observations in Sec. III A.

C. Compressible three-dimensional Taylor-Green vortex

To show that the SLLBM captures the intricate interactions between turbulent and compressible features including

shocklets, the compressible three-dimensional Taylor-Green vortex flow was simulated. On the triply periodic domain $S = [0, 2\pi]^3$, the initial conditions with constant temperature initial condition (CTIC) are

$$u_1(x_1, x_2, x_3, t = 0) = \sin x_1 \cos x_2 \cos x_3, \quad \text{(11)}$$

$$u_2(x_1, x_2, x_3, t = 0) = -\cos x_1 \sin x_2 \cos x_3, \quad \text{(12)}$$

$$u_3(x_1, x_2, x_3, t = 0) = 0, \quad \text{(13)}$$

$$\rho(x_1, x_2, x_3, t = 0) = 1 + \frac{\mathcal{C}}{16}[\cos(2x_1) \quad \text{(14)}$$

$$+ \cos(2x_2)][\cos(2x_3 + 2)], \quad \text{(15)}$$

$$T(x_1, x_2, x_3, t = 0) = 1, \quad \text{(16)}$$

with velocities **u**, density ρ, Mach number Ma, and temperature T. The numerator $\mathcal{C}$ differs between the cases in this work with $\mathcal{C} = 1$ for Re $= 100$ and Re $= 400$ as well as $\mathcal{C} = \gamma \text{Ma}^2$ for Re $= 1600$. The Reynolds number is defined as $Re = 1/\nu_\infty$, where the subscript ∞ denotes the value at $T = 1$. The Prandtl number is Pr $= 0.7$; the heat capacity ratio is $\gamma = 1.4$. The dynamic viscosity $\mu = \nu \rho$ obeys the Sutherland law

$$\mu = \frac{1.4042 T^{1.5}}{T + 0.40417} \mu_\infty \quad \text{(17)}$$

as proposed by the reference works [94,95].

In comparison to forced or decaying isotropic turbulence, this test case enables a deterministic initialization and thus an easier and more objective comparison.

1. Reynolds number Re = 400

Peng and Yang thoroughly studied the compressible Taylor-Green vortex at Reynolds number Re $= 400$ [94]. The original work used a compact eighth-order finite difference scheme [24] to discretize the convective terms in the Navier-Stokes equation in combination with a seventh-order weighted essentially nonoscillatory (WENO) scheme. The present compressible SLLBM uses fourth-order polynomials for the equilibrium [Eq. (3)] and for the interpolation [Eq. (8)], *without utilizing additional filtering or stabilizing* techniques other than the filtering introduced by the interpolation. Moreover, to satisfy the CFL condition, the original work applied a time-step size of $\delta_t = 0.0005$, whereas the SLLBM was capable to utilize time-step sizes two orders of magnitude larger: $\delta_t = 0.017$ for Mach number Ma $= 0.5$, $\delta_t = 0.033$ for Ma $= 1.0$, $\delta_t = 0.018$ for Ma $= 1.5$, and $\delta_t = 0.012$ for Ma $= 2.0$. Note that the lower time-step size for Ma $= 0.5$ compared to Ma $= 1.0$ is due to the current implementation of the SLLBM with only a single layer of ghost cells in each MPI process. The spatial resolution was $N_{\text{points}} = 256^3$, (i.e., $N_{\Xi} = 64^3$), whereas the reference operated with 512^3 grid points.

Figure 7 depicts the kinetic energy over time for all Mach numbers, which accurately follows the reference solution despite the coarse temporal discretization. Next, Fig. 8 depicts the solenoidal dissipation defined as

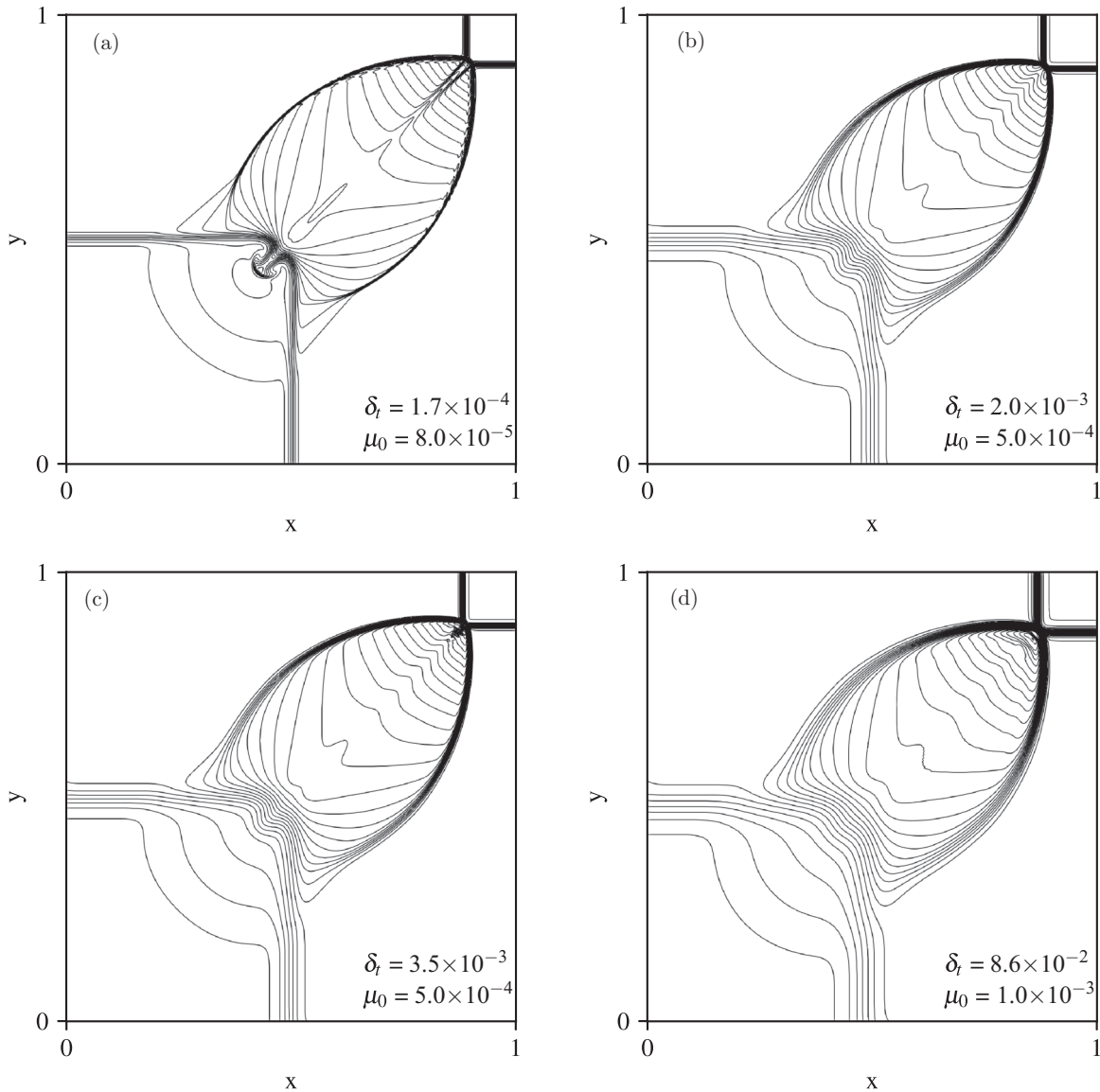$$\varepsilon^s = \frac{1}{\rho_{\text{ref}} \text{Re}} \int_S \mu (\nabla \times \mathbf{u})^2 \, d^3 \mathbf{x}. \quad \text{(18)}$$

FIG. 6. Density contours of the 2D Riemann simulations using 512×512 grid points. The time-step size of simulation (d) is approximately 50 times larger than the time-step size of simulation (a), but the viscosity differs by a factor of approximately 12. As a general rule, lower viscosities demand smaller time-step sizes. When choosing the time-step size too large for a given viscosity, the simulations tend to become unstable near the shock fronts, as depicted in panel (c). By contrast, simulation (b) remains stable due to a 57% reduction of the time-step size with equal prescribed viscosity.

All Mach numbers quite accurately matched the reference. The effect of changing the time step to similar levels as in the above mentioned reference can be seen for the smaller Mach numbers Ma = 0.5 and Ma = 1.0. The small deviation from the reference seen here is reduced by decreasing the time-step size to similar levels, here $\delta_t = 0.003$ in both cases.

The dilatational dissipation

$$\varepsilon^c = \frac{4}{3\rho_{\text{ref}}\text{Re}} \int_S \mu(\boldsymbol{\nabla} \cdot \mathbf{u})^2 \, d^3\mathbf{x} \tag{19}$$

is a measure for pressure work in the simulation. Figure 9 shows that the dilatational effects are strong in the beginning at small simulation times, surmounting the solenoidal dissipation. The comparison to the reference shows a slight deviation

from the reference and mild oscillations near the peak values around $t = 2.5$.

2. Reynolds number Re = 1600

The Taylor-Green vortex at Reynolds number Re = 400 shows strong dilatational effects, but the transition to fully developed turbulence requires higher Reynolds numbers. Therefore, another recent work by Lusher and Sandham examined this test case at Reynolds number Re = 1600 up to Mach number Ma = 1.25 [95]. In their study, the authors compared high-order finite difference schemes equipped by WENO or targeted essentially nonoscillatory (TENO) schemes of different orders. The present work re-examines the Mach numbers Ma = 1.0 and Ma = 1.25 up to $t = 20$,
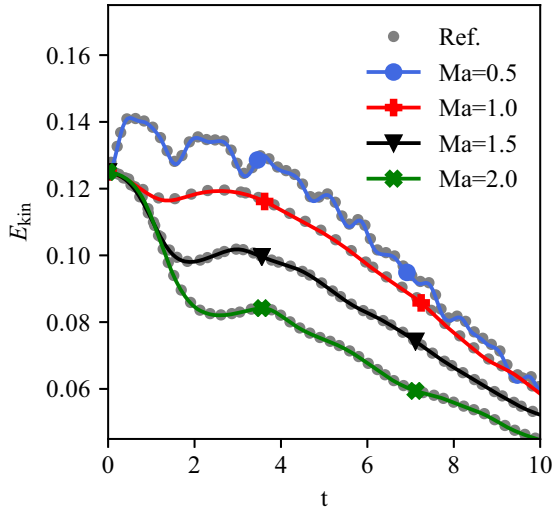
FIG. 7. Kinetic energy over time of the compressible 3D Taylor-Green vortex at Reynolds number Re = 400. Simulations by the SLLBM with the D3Q45 velocity set shown in Fig. 2. Reference from Ref. [94].

with time-step sizes $\delta_t = 0.010$, and $\delta_t = 0.009$, respectively. These time-step sizes were still 20 times larger than those in the reference [95]. The resolution was $N_{points} = 256^3$, whereas the reference used 512^3 grid points. Fig. 10 demonstrates that, just as in the case Re = 400, the kinetic energy over time is excellently reproduced for both Mach numbers. Figure 11 shows the solenoidal dissipation in comparison to the reference with even better results for the higher Mach number Ma = 1.25. The good agreement confirms the little numerical dissipation introduced by the SLLBM at polynomial order $p = 4$ in turbulent flows at transonic Mach numbers. By
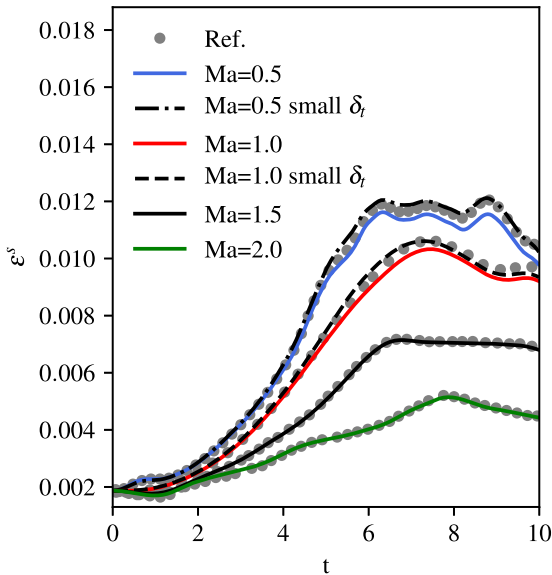

FIG. 8. Solenoidal dissipation ε^s of the compressible 3D Taylor-Green vortex at Reynolds number Re = 400. Small time-step sizes at Ma = 0.5 and Ma = 1.0 with 5.5 times and 11.1 times smaller times step sizes, respectively. The reduction induced a slightly better agreement with Ref. [94].
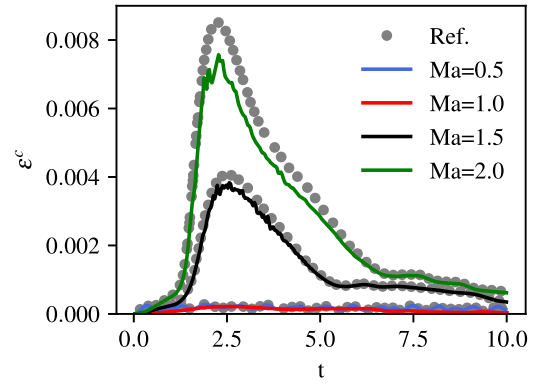

FIG. 9. Dilatational dissipation ε^c of the compressible 3D Taylor-Green vortex at Reynolds number Re = 400. The time-step sizes are identical to the large δ_t simulations in Fig. 8.

contrast, the interpolation order $p = 2$ worsens the solution despite the identical resolution of $N_{points} = 256^3$. Next, we investigated the dilatational dissipation. During the early phase of the simulation the large vortex structures begin to entangle. This entanglement leads to strong moving shock-like structures or turbulent shocklets with strong negative dilatation [95] and local numerical oscillations of the macroscopic variables due to an underresolution of these shocklets. A slice of the density field illustrating the shocklets, indicated by large jumps in the governing variables, is shown in Fig. 12. The size of the jumps agreed well with that obtained via classical Rankine-Hugoniot jump conditions. Additionally, during the early phase, the Mach numbers can be higher than the initially prescribed Mach numbers. Moreover, to compute Eq. (19), we made use of the gradients of the interpolation polynomials, which—in contrast to the distribution functions—are not continuous over the cells. It is likely that this approach to compute the dilatational dissipation faces issues at strong shocklets. This might explain the oscillations for Ma = 1.25 in Fig. 13, which depicts the dilatational
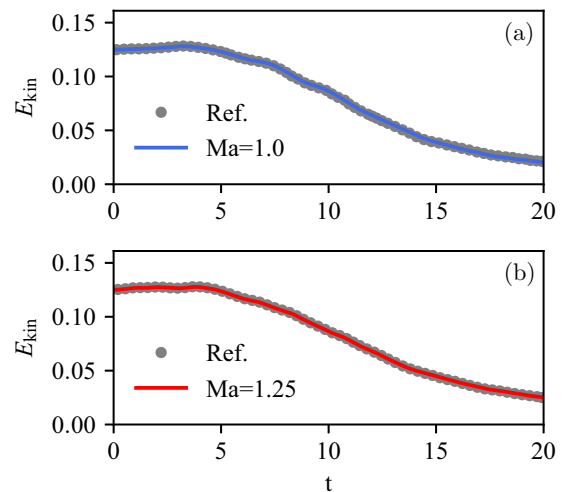

FIG. 10. Kinetic energy over time of the compressible 3D Taylor-Green vortex at Reynolds number Re = 1600 for Mach numbers (a) Ma = 1.0 and (b) Ma = 1.25. Reference from Ref. [95].
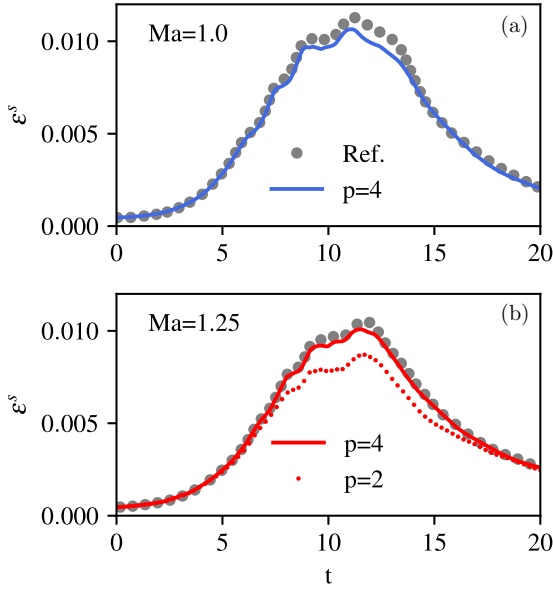
FIG. 11. Solenoidal dissipation ε^s of the compressible 3D Taylor-Green vortex at Reynolds number Re = 1600 for Mach numbers (a) Ma = 1.0 and (b) Ma = 1.25. Note the differences between second-order polynomials $p = 2$ and fourth-order polynomials $p = 4$.

dissipation. Despite these deviations from the reference in the beginning, the SLLBM was able to reproduce the dilatational dissipation well for the rest of the simulation, as shown in Fig. 13, without using additional stabilizing measures like filtering or artificial diffusivity.

3. Reynolds number Re = 100 for higher Mach numbers

As a last test case, Fig. 14 shows that for Re = 100 we were able to stably simulate the given test case for Mach numbers Ma = 2.5 and Ma = 3.0. The Knudsen number for
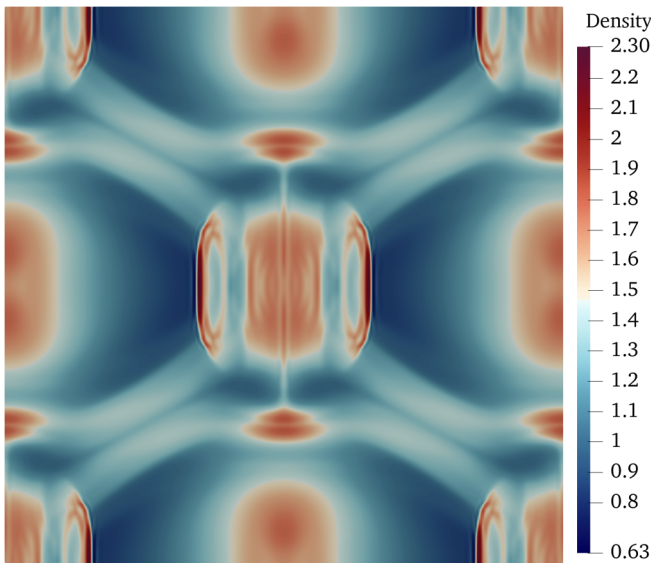

FIG. 12. Density of the three-dimensional Taylor-Green vortex with Re = 1600 and Mach number Ma = 1.25 at time $t = 3.34$. The x-z plane is shown at position $y = \pi$.
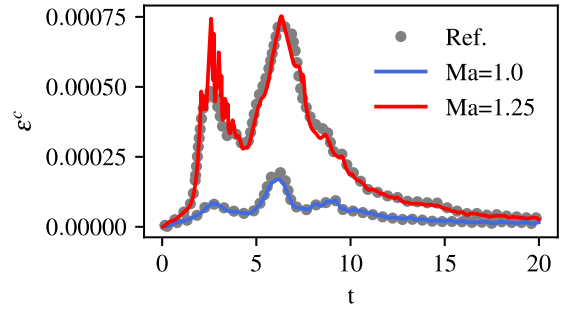

FIG. 13. Dilatational dissipation ε^c of the compressible 3D Taylor-Green vortex at Reynolds number Re = 1600.

this configuration was of order Kn = $\mathcal{O}(\text{Ma}/\text{Re}) = \mathcal{O}(10^{-2})$. Due to this Knudsen number at the upper end of the continuum flow scale, the SLLBM was once more able to use large times steps $\delta_t = 0.015$ and $\delta_t = 0.03$, respectively. Although only shown at low Reynolds numbers, these stable simulations indicate the principal capability of the SLLBM to perform stable simulations at high Mach numbers.

IV. DISCUSSION

As demonstrated by the numerical experiments, the SLLBM is able to simulate complex two- and three-dimensional compressible flows with shocks and shocklets. There are two main arguments in favor of the method. First, the method works with large and adjustable time-step sizes. Second, cubature rules provide lean velocity sets resulting in an efficient scheme. The D3Q45 used in this work is, to the authors knowledge, the smallest known degree-nine velocity set and key to reduce errors caused by interpolation.

The SLLBM's streaming-step is not subject to a CFL condition, as already shown in past works [42] and as confirmed by the shock tube simulations in this work. In contrast to the usual spatial filtering, the independence of the CFL condition opens the field of temporal large eddy simulations (TLES) with fine spatial, but coarse temporal resolutions [96,97]. Since no dedicated filter operation has been used, we classify the scheme as some sort of temporal implicit large eddy simulation (TILES). Despite the temporally coarse resolutions, most relevant flow features during the simulations were preserved, or they were recovered by scaling the time-step size
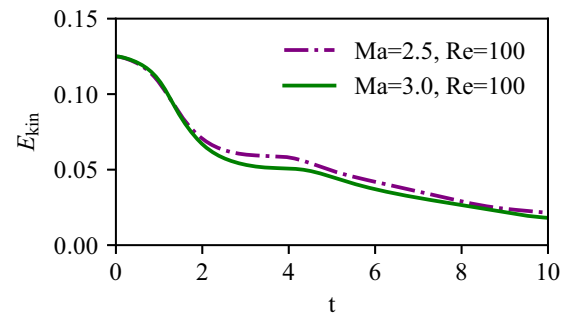

FIG. 14. Kinetic energy over time of the compressible 3D Taylor-Green vortex at Reynolds number Re = 100 for Mach numbers Ma = 2.5 and Ma = 3.0.

down: a key feature of the SLLBM that on-lattice Boltzmann schemes often miss.

Although the streaming step is not subject to a CFL condition, the interplay of collision operator and advection may still cause instabilities at small relaxation times and large time-step sizes δ_t as pointed out by various works, e.g., by Siebert *et al.* [98]. However, this limitation is—in contrast to the classical CFL condition—largely independent from the spatial discretization interval δ_x. That said, the stability regions of the BGK collision operator combined with the D3Q45 and fourth-order equilibria (at high Mach numbers and adjustable time-step sizes) are yet unknown. Nevertheless, we observed that the SLLBM produces good results without using additional filtering or stabilizing techniques. To accomplish simulations with even higher Reynolds and Mach numbers with large time-step sizes, the application of such techniques to the SLLBM might be beneficial, though.

A. Comparison to other LBM solvers

Let us now compare the SLLBM for three-dimensional compressible flows to other LBM solvers, with distinction between on-lattice solvers and off-lattice solvers. Since we discussed many aspects in recent works [44,47], we mainly restrict the discussion to compressible LBM solvers with applications to three-dimensional turbulent flows.

To perform a time step, off-lattice Boltzmann methods typically require a special treatment of the distribution function values such as the discretization by finite difference schemes [99–101] or finite volume schemes [79] and the application of a dedicated time integrator, e.g., a Runge-Kutta scheme. As an example for these Eulerian time integration schemes, Chen *et al.* presented compressible decaying three-dimensional isotropic turbulence simulations obtained by a modified discrete unified gas kinetic scheme (DUGKS), which is essentially a finite volume LBM with second-order spatial and temporal accuracy [102]. Also, the authors made use of a high-order Gaussian quadrature to discretize the velocity space, but they relied on a decisively larger D3Q77 velocity set with identical quadrature degree as the D3Q45. Like all explicit Eulerian time integration schemes, DUGKS is subject to the CFL condition Eq. (1).

A second category of off-lattice schemes are interpolation-based or semi-Lagrangian implementations. Pavlo *et al.* pioneered in using second-order interpolations to incorporate high-order velocity sets for simulations of thermal flows [80,103]. Renowned semi-Lagrangian implementations are the Particles on Demand method [61], which uses dynamically shifted velocity sets to reach high Mach numbers, or the SLLBM for compressible flows on unstructured meshes by Saadat *et al.* [65]. Unlike the present method, the authors of the latter approach used a D2Q9 velocity set and computed correction terms by exploiting the gradients of the distribution functions, which practically come at low costs when using interpolation polynomials. To the best of our knowledge, these methods have not been applied to complex three-dimensional flows, except for a spherical Riemann problem presented by Zakirov *et al.* using a D3Q125 velocity set [104]. Additionally, one decisive feature of the present method is the organization

of support points by cells, enabling three-dimensional high-order solutions.

On-lattice Boltzmann solvers have the Lagrangian integration along characteristics in common with interpolation-based schemes, albeit constrained by a strong coupling of space and time discretization. This explains, the rather large time-step sizes of on-lattice Boltzmann methods with increasing Mach number [47,105]. On-lattice Boltzmann methods generally exhibit second-order accuracy in space and time, but they are not as fiercely concerned by numerical diffusion as low-order off-lattice Boltzmann methods. In general, the computational costs of on-lattice Boltzmann methods' streaming step are low. On the downside, they often suffer from large velocity sets. For example, Frapolli *et al.* [55,57,106] were able to simulate various three-dimensional compressible flows, including isotropic decaying turbulence simulations or the flow around an Onera M6 wing, by the entropic lattice Boltzmann method. In many ways, their works were groundbreaking for the development of compressible LBMs. However, the authors used a velocity set with 343 discrete velocities and the weights were temperature-dependent, which limits the temperature range of the method. In addition, in many on-lattice Boltzmann frameworks the time-step size is not adjustable due to the linking of spatial and temporal discretization [105]. An exception to this are hybrid lattice Boltzmann methods, which solve the temperature field separately [58,62,72,107]. In this case, by adapting the speed of sound, the time-step size can be varied. Hybrid LBMs enable compact on-lattice velocity discretizations like the degree-seven D3Q39 velocity used by the work of Nie *et al.* [72] published a decade ago, and the method has been applied to complex flows as presented by Fares and Wessels [73]. In contrast to hybrid LBMs the present SLLBM solves the temperature equation by a lattice Boltzmann formulation with two separate distribution functions using a degree-nine D3Q45 off-lattice velocity set.

By using numerical equilibria instead of the polynomial equilibrium in this work, Latt *et al.* were able to simulate a three-dimensional flow around a sphere [66] by using only 39 discrete velocities, as similarly proposed by Frapolli [105]. Simulations of complex turbulent flows by this method are not available, yet. Recently, Saadat *et al.* [108] proposed an on-lattice Boltzmann model with a regular D3Q27 velocity set to perform three-dimensional decaying compressible isotropic turbulence simulations. Despite the low degree of the velocity set, this model proved capable to simulate shocked flows up to Mach numbers of Ma 1.5. This is achieved by correcting the error-prone high-order moments by expressions obtained from applying finite differences to the macroscopic variables.

In summary, the number of compressible LBM solvers for three-dimensional flows is still limited for both on-lattice and off-lattice Boltzmann schemes, which indicates the value of the presented SLLBM framework to set a pattern for future research.

B. Numerical efficiency

The numerical efficiency of the SLLBM depends on the implementation of the collision and the streaming step. The former requires at each support point the calculation of the discrete equilibrium function values given by Eq. (6).

Once the density, velocities, and temperature are gained from the distribution function values, the determination of the equilibrium, however, is well parallelizable. Note, that the tensors in Eq. (6) are symmetric, so that many entries need to be computed only once per time step and support point, e.g., $a_{xxxy,\text{eq}}$ is equal to $a_{yxxx,\text{eq}}$ and all other index permutations. In addition, the entries of the Hermite tensors $\mathcal{H}_i$ are constant for a given velocity set and only need to be determined once for the whole simulation.

Another decisive factor of the simulation performance is the streaming step, which requires a "triquartic" interpolation, i.e., interpolation polynomials of order $p = 4$ in three dimensions. The reference cell is shown in Fig. 1. Generally speaking, interpolations are costlier than the node-to-node streaming step of the standard LBM. To interpolate one distribution function value at a given departure point, all $(p + 1)^D = 125$ support points of the cell are processed with the shape function values of the interpolation that are stored in the Q matrices Ψ_i. These matrices are the cornerstone for the whole simulation. When accounting to equally shaped cells, like in all simulations in the present work, the size of the matrices can be significantly reduced, since the matrices are identical over nearly all cells (boundary cells excepted). However, for irregular or distorted three-dimensional meshes, the size of the matrices grows quickly, rendering the streaming step memory-bound. Therefore, matrix-free implementations [109] appear to be an attractive extension for SLLBM implementations. In matrix-free versions the shape function values will be computed afresh in each time step, which is potentially faster for today's high-performance computing clusters. In addition, the mass losses, still bothering interpolation LBM schemes, might be solved by applying conservative SL formulations [110].

At first glance, the second distribution function appears to be a severe limitation of the method due to the additional memory accesses needed and due to the computational effort. On closer inspection, however, it turns out that the overhead is kept in reasonable limits since the computation-intensive equilibrium distribution g_i^{eq} linearly depends on f_i^{eq} and the shape function values can be used for both f_i and g_i. Still, all aforementioned issues pay off in light of the large time-step sizes of semi-Lagrangian implementations. A major way to further reduce the computational cost is the reduction of discrete velocities in the Gaussian quadrature. The research for even compacter cubature rules is still ongoing [47], so that future degree-nine velocity sets will possibly be even more efficient.

Upcoming work will also incorporate boundary conditions into the SLLBM for compressible flows. Although boundary conditions usually pose problems to LBM solvers equipped with high-order velocity sets, the present algorithm relaxes this constraint: at the beginning of each simulation, the SLLBM departure points of the distribution function values, i.e., the values that have to be interpolated, need to be tracked. To this end, the SLLBM algorithm follows the trajectories of the particle velocities through all adjacent cells until the distance $\boldsymbol{\xi}_i \delta_t$ is covered. The advantage of this procedure is that intermediate boundaries will be detected en route. In the case of simple halfway bounce back boundaries, the particle trajectory is reversed and the remaining distance is covered
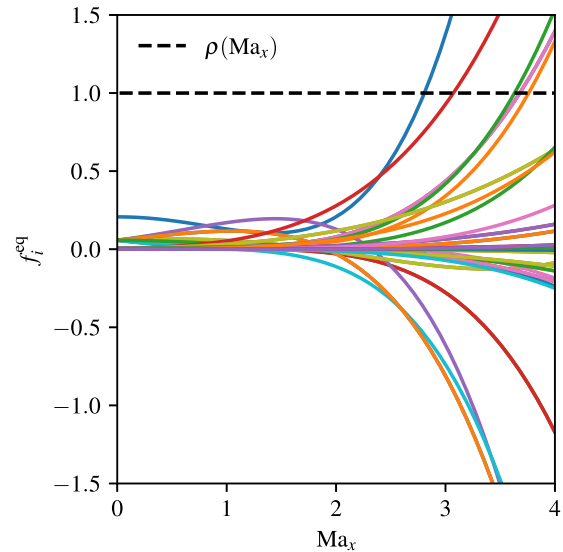

FIG. 15. All 45 equilibrium distribution functions f_i^{eq} of the D3Q45 velocity set over Mach numbers ranging from 0 to 4 in x direction. Density ρ and relative temperature θ were constantly set to unity; Mach numbers in both other directions were set to zero.

towards the opposite direction. Di Ilio *et al.* presented a good description of this procedure in Fig. 2 of Ref. [111]. Equilibrium boundaries work quite similarly: when the equilibrium boundary is hit, the distribution function will be set to equilibrium during the simulation.

One last remark regarding the equilibrium distribution function determined via Eq. (6). In the past the Hermite expansion-based equilibrium was deemed causing instabilities due to negative distribution function values at high Mach numbers [105]. Figure 15 depicts the equilibrium function values f_i^{eq} over the Mach number Ma_x in x direction. The figure manifests that most of the discrete velocities significantly diverge for Mach numbers larger than $\text{Ma} > 2.0$ toward either ∞ or $-\infty$. However, our simulations showed that simulations even up to $\text{Ma} = 3.0$ remained stable. This observation indicates that negative distribution function values are not per se a source of instabilities, since the values' sole role is to encode the moments of different orders. For the first moments this encoding can also be seen in Fig. 15: despite the increasing first-order moment of the momentum, the "zeroth-order" moment of the density remains $\rho = 1.0$. To ensure this, negative values are a consequence at high Mach numbers due to the discretization of the velocity space via Gaussian quadratures.

V. CONCLUSION

The SLLBM for three-dimensional compressible flows is a viable alternative to other solvers. The SLLBM allows very large time-step sizes not restricted by the customary CFL condition $\delta_t \leqslant \text{CFL}_{\max} \delta_x / |\boldsymbol{\xi}|_{\max}$, which would otherwise limit the maximum time-step size depending on the fastest particle velocity $|\boldsymbol{\xi}|_{\max}$. Instead, it is the interplay of collision and streaming which possibly causes instabilities depending on the flow. Although the presented SLLBM is a fourth-order spatial method and accurately captures shocks as well as turbulence, no stabilization or filtering in addition to the

interpolation were required for the presented test cases. Due to these unique features, the cubature-based fully explicit SLLBM enables researchers to perform compressible turbulence simulations, in which the admissible time-step sizes are largely decoupled from the spatial discretization, opening a new field of affordable simulations for compressible turbulent flows.

ACKNOWLEDGMENTS

APPENDIX A: EQUILIBRIUM MOMENTS

The equilibrium moments up to fourth-order read

$$a_{eq}^{(0)} = \rho, \tag{A1a}$$

$$a_{\alpha,eq}^{(1)} = \rho u_\alpha, \tag{A1b}$$

$$a_{\alpha\beta,eq}^{(2)} = \Pi_{\alpha\beta}^{\mathrm{eq}} = \rho[u_\alpha u_\beta + T_0(\theta - 1)\delta_{\alpha\beta}], \tag{A1c}$$

$$\begin{aligned} a_{\alpha\beta\gamma,eq}^{(3)} &= \mathcal{Q}_{\alpha\beta\gamma}^{\mathrm{eq}} \\ &= \rho[u_\alpha u_\beta u_\gamma + T_0(\theta - 1)(\delta_{\alpha\beta}u_\gamma \\ &\quad + \delta_{\alpha\gamma}u_\beta + \delta_{\beta\gamma}u_\alpha)], \end{aligned} \tag{A1d}$$

$$\begin{aligned} a_{\alpha\beta\gamma\delta,eq}^{(4)} &= \mathcal{R}_{\alpha\beta\gamma\delta}^{\mathrm{eq}} \\ &= \rho[u_\alpha u_\beta u_\gamma u_\delta + T_0(\theta - 1)((\delta_{\alpha\beta}\delta_{\gamma\delta} \\ &\quad + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma})T_0(\theta - 1) \\ &\quad + \delta_{\alpha\beta}u_\gamma u_\delta + \delta_{\alpha\gamma}u_\beta u_\delta + \delta_{\alpha\delta}u_\beta u_\gamma \\ &\quad + \delta_{\beta\gamma}u_\alpha u_\delta + \delta_{\beta\delta}u_\alpha u_\gamma + \delta_{\gamma\delta}u_\alpha u_\beta)], \end{aligned} \tag{A1e}$$

where $\theta = T/T_0$ is the relative temperature.

APPENDIX B: HERMITE TENSORS

The scaled Hermite tensors depending on the lattice speed of sound c_s and with $\hat{\xi}_i = \xi_i/c_s$ up to fourth order read

$$\mathcal{H}_i^{(0)} = 1, \quad \mathcal{H}_{i\alpha}^{(1)} = \frac{\hat{\xi}_{i\alpha}}{c_s}, \quad \mathcal{H}_{i\alpha\beta}^{(2)} = \frac{\hat{\xi}_{i\alpha}\hat{\xi}_{i\beta} - \delta_{\alpha\beta}}{c_s^2},$$

$$\mathcal{H}_{i\alpha\beta\gamma}^{(3)} = \frac{\hat{\xi}_{i\alpha}\hat{\xi}_{i\beta}\hat{\xi}_{i\gamma} - (\hat{\xi}_{i\alpha}\delta_{\beta\gamma} + \hat{\xi}_{i\beta}\delta_{\alpha\gamma} + \hat{\xi}_{i\gamma}\delta_{\alpha\beta})}{c_s^3},$$

$$\mathcal{H}_{i\alpha\beta\gamma\delta}^{(4)} = \frac{\hat{\xi}_{i\alpha}\hat{\xi}_{i\beta}\hat{\xi}_{i\gamma}\hat{\xi}_{i\delta} - \mathcal{T}_i + (\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma})}{c_s^4},$$

with

$$\begin{aligned} \mathcal{T}_i &= \hat{\xi}_{i\alpha}\hat{\xi}_{i\beta}\delta_{\gamma\delta} + \hat{\xi}_{i\alpha}\hat{\xi}_{i\gamma}\delta_{\beta\delta} + \hat{\xi}_{i\alpha}\hat{\xi}_{i\delta}\delta_{\beta\gamma} \\ &\quad + \hat{\xi}_{i\beta}\hat{\xi}_{i\gamma}\delta_{\alpha\delta} + \hat{\xi}_{i\beta}\hat{\xi}_{i\delta}\delta_{\alpha\gamma} + \hat{\xi}_{i\gamma}\hat{\xi}_{i\delta}\delta_{\alpha\beta}. \end{aligned}$$

APPENDIX C: CHAPMAN-ENSKOG ANALYSIS

This Appendix shows the approximation of the compressible Navier-Stokes equations, when applying a Chapman-Enskog analysis to the SLLBM model. To that end, a second-order multiscale expansion is applied to the temporal derivatives

$$\partial_\alpha = \epsilon\partial_\alpha, \tag{C1}$$

$$\partial_t = \epsilon\delta_t^{(1)} + \epsilon^2\delta_t^{(2)}, \tag{C2}$$

where ϵ is a smallness parameter usually identified as the Knudsen number [83] and the discrete distribution functions h_i,

$$h_i = h_i^{(0)} + \epsilon h_i^{(1)} + \epsilon^2 h_i^{(2)}. \tag{C3}$$

These expansion terms are applied to a second-order Taylor expansion of Eq. (5),

$$\left(\epsilon D_i + \frac{\epsilon^2}{2}D_i D_i\right)h_i = -\frac{1}{\tau}(h_i - h_i^{\mathrm{eq}}), \tag{C4}$$

with the material derivative [112]

$$\epsilon D_i = \epsilon D_i^{(1)} + \epsilon^2 D_i^{(2)} = \epsilon\partial_t^{(1)} + \epsilon\,\xi_{i\alpha}\partial_\alpha + \epsilon^2\partial_t^{(2)}. \tag{C5}$$

Next, Eq. (C3) is applied to Eq. (C4) and the terms of same order are collected. The zeroth-order terms of order ϵ^0 are

$$0 = -\frac{1}{\tau}(h^{(0)} - h^{\mathrm{eq}}), \tag{C6}$$

with the trivial relation

$$h_i^{(0)} = h_i^{\mathrm{eq}}. \tag{C7}$$

When applying the moment relations of Eq. (7) to the last Equation, we need to differ between the moments of f_i and g_i, i.e.,

$$\sum_i f_i^{(0)}\{1, \xi_{i\alpha}\} = \sum_i f_i^{\mathrm{eq}}\{1, \xi_{i\alpha}\} \tag{C8}$$

and

$$\sum_i \left(f_i^{(0)}|\boldsymbol{\xi}_i|^2 + g_i^{(0)}\right) = \sum_i \left(f_i^{\mathrm{eq}}|\boldsymbol{\xi}_i|^2 + g_i^{\mathrm{eq}}\right), \tag{C9}$$

showing mass, momentum, and energy conservation during the collision step, which also implies that

$$\sum_i f_i^{(1,2,\dots)}\{1, \xi_{i\alpha}\} = 0, \tag{C10}$$

$$\sum_i \left(f_i^{(1,2,\dots)}|\boldsymbol{\xi}_i|^2 + g_i^{(1,2,\dots)}\right) = 0. \tag{C11}$$

As a second step, the order ϵ^1 terms are collected,

$$D_i^{(1)}h_i^{(0)} = -\frac{1}{\tau}h_i^{(1)}. \tag{C12}$$

This time, the relations of Eq. (7) yield for the zeroth moment

$$D_i^{(1)}\rho = \partial_t^{(1)}\rho + \partial_\alpha(\rho u_\alpha) = 0, \tag{C13}$$

for the first moments with the help of Eq. (A1c) and $P = \rho(T - T_0)$,

$$\partial_t^{(1)}(\rho u_\alpha) + \partial_\beta \Pi_{\alpha\beta}^{(0)} = 0,$$

$$\partial_t^{(1)}(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) + \partial_\alpha(P)\delta_{\alpha\beta} = 0, \tag{C14}$$

and for the total energy

$$\partial_t^{(1)}(\rho E) + \partial_\alpha(u_\alpha \rho E) = -P\partial_\alpha u_\alpha, \tag{C15}$$

or

$$\partial_t^{(1)} T = -u_\alpha \partial_\alpha T - \frac{P}{\rho C_v} \partial u_\alpha \tag{C16}$$

for the temperature.

These are the Euler equations; to derive the Navier-Stokes equations, the terms of order ϵ^2 also need to be gathered. This leads to

$$D_i^{(2)} h_i^{(0)} + D_i^{(1)} h_i^{(1)} + \frac{1}{2} D_i^{(1)} D_i^{(1)} h_i^{(0)} = -\frac{1}{\tau} h_i^{(2)}. \tag{C17}$$

Equation (C12) simplifies the last Equation's derivatives,

$$D_i^{(2)} h_i^{(0)} + \left(1 - \frac{1}{2\tau}\right) D_i^{(1)} h_i^{(1)} = -\frac{1}{\tau} h_i^{(2)}. \tag{C18}$$

Due to vanishing terms, the zeroth moment of Eq. (C18) is

$$\partial_t^{(2)} \rho = 0. \tag{C19}$$

Then, the slightly more complicated part begins with the first moments of Eq. (C18),

$$\partial_t^{(2)}(\rho u_\alpha) = -\left(1 - \frac{1}{2\tau}\right) \partial_\beta \Pi_{\alpha\beta}^{(1)}. \tag{C20}$$

To express the nonequilibrium moment $\Pi_{\alpha\beta}^{(1)}$ by equilibrium moments, we derive the second-order moment of Eq. (C12), which is

$$\partial_t^{(1)} \Pi_{\alpha\beta}^{(0)} + \partial_\gamma \mathcal{Q}_{\alpha\beta\gamma}^{(0)} = -\frac{1}{\tau} \Pi_{\alpha\beta}^{(1)} \tag{C21}$$

and then complements Eq. (C20)

$$\partial_t^{(2)}(\rho u_\alpha) = \left(\tau - \frac{1}{2}\right) \partial_\beta \left(\partial_t^{(1)} \Pi_{\alpha\beta}^{(0)} + \partial_\gamma \mathcal{Q}_{\alpha\beta\gamma}^{(0)}\right). \tag{C22}$$

The equilibrium moments can be explicitly specified by Eqs. (A1c) and (A1d). By applying the product rule and by

using the Euler equations for mass Eq. (C13) and momentum Eq. (C14) and temperature Eq. (C16) to replace the time derivatives, one obtains

$$\partial_t^{(1)} \Pi_{\alpha\beta}^{(0)} = -\partial_\gamma(\rho u_\alpha u_\beta u_\gamma) - \frac{P}{C_v} \partial_\gamma u_\gamma \delta_{\alpha\beta}$$
$$- \partial_\gamma(P u_\gamma)\delta_{\alpha\beta} - u_\alpha \partial_\beta P - u_\beta \partial_\alpha P \tag{C23}$$

and

$$\partial_\gamma \mathcal{Q}_{\alpha\beta\gamma}^{(0)} = \partial_\gamma(\rho u_\alpha u_\beta u_\gamma) + P\partial_\beta u_\alpha + P\partial_\alpha u_\beta$$
$$+ \partial_\gamma(P u_\gamma)\delta_{\alpha\beta} + u_\alpha \partial_\beta P + u_\beta \partial_\alpha P. \tag{C24}$$

This turns Eq. (C22) into

$$\partial_t^{(2)}(\rho u_\alpha) = \left(\tau - \frac{1}{2}\right) \partial_\beta \left[P(\partial_\alpha u_\beta + \partial_\beta u_\alpha) - \frac{P}{C_v} \partial_\gamma u_\gamma \delta_{\alpha\beta}\right]. \tag{C25}$$

As a last step, the total energy is determined

$$\partial_t^{(2)}(\rho E) = -\left(1 - \frac{1}{2\tau}\right) \partial_\beta q_\beta^{(1)}, \tag{C26}$$

where

$$q_\beta^{(1)} = \sum_i \frac{1}{2} |\boldsymbol{\xi}_i|^2 \xi_{i\beta} f_i^{(1)} \tag{C27}$$

is a contracted variant of $\mathcal{Q}_{\alpha\beta\gamma}^{(1)}$ detailed in Eq. (A1d). Similar to Eq. (C20), this vector can be expressed by equilibrium counterparts

$$q_\beta^{(1)} = -\tau\left(\partial_t^{(0)} q_\beta^{(0)} + \partial_\gamma r_{\beta\gamma}^{(0)}\right), \tag{C28}$$

with

$$r_{\beta\gamma}^{(0)} = \sum_i \frac{1}{2} |\boldsymbol{\xi}_i|^2 \xi_{i\beta} \xi_{i\gamma} f_i^{(0)}. \tag{C29}$$

Resembling q_β, the tensor $r_{\beta\gamma}$ is the contracted variant of $\mathcal{R}_{\alpha\beta\gamma\delta}$ detailed in Eq. (A1e). Again, by a number of replacements, one obtains [113]

$$q_\beta^{(1)} = -\tau P \left[(1 + C_v)\partial_\beta T + \partial_\beta \left(u_\gamma[\partial_\beta u_\gamma + \partial_\gamma u_\beta] - u_\gamma \frac{\partial_\delta u_\delta}{C_v}\delta_{\beta\gamma}\right)\right]. \tag{C30}$$

This term now complements Eq. (C26). Finally, by summing up all contributions of orders ϵ^0, ϵ^1, and ϵ^2, the compressible Navier-Stokes equations are derived

$$\partial_t \rho + \partial_\alpha(\rho u_\alpha) = 0, \tag{C31}$$

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) = \partial_\beta(\sigma_{\alpha\beta}), \tag{C32}$$

$$\partial_t(\rho E) + \partial(\rho E u_\beta) = \partial_\beta(\kappa \partial_\beta T) + \partial_\beta(u_\gamma \sigma_{\beta\gamma}), \tag{C33}$$

with dynamic viscosity $\mu = \tau P$, thermal conductivity $\kappa = \tau P(C_v + 1) = \tau P C_p$, since the heat capacity at constant

pressure is defined as $C_p = C_v + 1$. The stress tensor denotes

$$\sigma_{\alpha\beta} = \mu(\partial_\alpha u_\beta + \partial_\beta u_\alpha) - P\delta_{\alpha\beta} - \frac{\mu}{C_v} \partial_\gamma u_\gamma \delta_{\alpha\beta}. \tag{C34}$$

If the effect of the interpolation was taken into account during the analysis, Eqs. (C31)–(C33) would contain a source term, whose derivation has been left out here for brevity. Since the dynamic viscosity depends on the local pressure P, the relaxation parameter has to be pressure-dependent, i.e., $\tau = \mu/(P\delta_t) + 0.5$. From Eq. (C34) the bulk viscosity, with respect to the notation in Eq. (C32), can be identified as $\mu_b = \mu/C_v$.

Note that the presented derivation links the thermal conductivity to the dynamic viscosity. In our approach, we used

the quasiequilibrium approach to adjust the Prandtl number $\Pr = (\mu C_p)/\kappa$. For the corresponding extension of the Chapman-Enskog analysis, we refer to Ref. [56].

APPENDIX D: QUASIEQUILIBRIUM APPROACH FOR VARIABLE PRANDTL NUMBER

To obtain a variable Prandtl number $\Pr$, the following equation replaces Eq. (5):

$$
\begin{aligned}
h_i(\mathbf{x} + \delta_t \boldsymbol{\xi}_i, t + \delta_t) = {} & h_i(x, t) - \frac{1}{\tau}\left[h_i(x, t) - h_i^{\mathrm{eq}}(x\xi_i, t)\right] \\
& + \left(\frac{1}{\tau} - \frac{1}{\tau_{\mathrm{Pr}}}\right) h_i^*(x, t),
\end{aligned} \tag{D1}
$$

with $\tau_{\mathrm{Pr}} = (\tau - 0.5)/\Pr + 0.5$. The quasinonequilibrium h_i^* is thereby obtained by first computing the centered heat flux tensor

$$
\bar{Q}_{\alpha\beta\gamma} = \sum_{i=1}^{n} f_i(\xi_{i\alpha} - u_\alpha)(\xi_{i\beta} - u_\beta)(\xi_{i\gamma} - u_\gamma). \tag{D2}
$$

Then, the nonequilibrium part $\bar{Q}_{\alpha\beta\gamma}^{\mathrm{neq}} = \bar{Q}_{\alpha\beta\gamma} - \bar{Q}_{\alpha\beta\gamma}^{\mathrm{eq}}$ is applied to the Hermite tensor $\mathcal{H}_{i\alpha\beta\gamma}^{(3)}$,

$$
h_i^* = w_i \bar{Q}_{\alpha\beta\gamma}^{\mathrm{neq}} : \mathcal{H}_{i\alpha\beta\gamma}^{(3)}, \tag{D3}
$$

via full contraction of indices.

[1] S. K. Lele, Annu. Rev. Fluid Mech. **26**, 211 (1994).

[2] P. Chassaing, R. Antonia, F. Anselmet, L. Joly, and S. Sarkar, *Variable Density Fluid Turbulence* (Kluwer Academic Publishers, Boston, MA, 2002).

[3] L. Veynante and D. Vervisch, Prog. Energy Combust. Sci. **30**, 193 (2002).

[4] A. J. Smits and J. P. Dussauge, *Turbulent Shear Layers in Supersonic Flow*, 2nd ed. (Springer, Berlin, 2006).

[5] E. Garnier, N. Adams, and P. Sagaut, *LES for Compressible Flows* (Springer, Berlin, 2009).

[6] S. Pirozzoli, Annu. Rev. Fluid Mech. **43**, 163 (2011).

[7] T. B. Gatski and J.-P. Bonnet, *Compressibility, Turbulence and High Speed Flow* (Academic Press, San Diego, CA, 2013).

[8] R. Sharman and T. Lane, *Aviation Turbulence: Processes, Detection, Prediction* (Springer, Berlin, 2016), pp. 1–523.

[9] M. M. M. Low and R. S. Klessen, Rev. Mod. Phys. **76**, 125 (2004).

[10] H. H. Fernholz and P. J. Finley, *A Critical Compilation of Compressible Turbulent Boundary Layer Data*, Vol. 223 (AGARDograph, Neuilly sur Seine, France, 1976).

[11] P. Huang, G. Coleman, and P. Bradshaw, J. Fluid Mech. **305**, 185 (1995).

[12] H. Foysi, S. Sarkar, and R. Friedrich, J. Fluid Mech. **509**, 207 (2004).

[13] S. Ghosh, H. Foysi, and R. Friedrich, J. Fluid Mech. **648**, 155 (2010).

[14] P. Bradshaw, Ann. Rev. Fluid Mech. **9**, 33 (1977).

[15] D. W. Bogdanoff, AIAA J. **21**, 926 (1983).

[16] A. W. Vreman, N. D. Sandham, and K. H. Luo, J. Fluid Mech. **320**, 235 (1996).

[17] H. Foysi and S. Sarkar, Theor. Comput. Fluid Dyn. **24**, 565 (2010).

[18] K. V. Matsuno and S. K. Lele, AIAA 2020-0573, in *Proceedings of the AIAA SciTech 2020 Forum* (AIAA, Orlando, FL, 2020).

[19] T. Colonius and S. Lele, Prog. Aerosp. Sci. **40**, 345 (2004).

[20] D. Bodony and S. Lele, AIAA, AIAA-2006-0468, in *Proceedings of the AIAA SciTech 2006 Forum* (AIAA, Orlando, FL, 2006).

[21] H. Foysi, M. Mellado, and S. Sarkar, Int. J. Heat Fluid Flow **31**, 307 (2010).

[22] S. K. Lele and J. Larsson, J. Phys. Conf. Ser. **180**, 012032 (2009).

[23] S. Lee, S. K. Lele, and P. Moin, Phys. Fluids A **3**, 657 (1991).

[24] S. K. Lele, J. Comput. Phys. **103**, 16 (1992).

[25] C. K. Tam and J. C. Webb, J. Comput. Phys. **107**, 262 (1993).

[26] N. A. Adams and K. Shariff, J. Comput. Phys. **127**, 27 (1996).

[27] J. A. Ekaterinaris, J. Comput. Phys. **156**, 272 (1999).

[28] T. K. Sengupta, G. Ganeriwal, and S. De, J. Comput. Phys. **192**, 677 (2003).

[29] C. Bogey and C. Bailly, J. Comput. Phys. **194**, 194 (2004).

[30] J. Wang, L. P. Wang, Z. Xiao, Y. Shi, and S. Chen, J. Comput. Phys. **229**, 5257 (2010).

[31] F. Hu, M. Hussaini, and J. Manthey, J. Comput. Phys. **124**, 177 (1996).

[32] J. Berland, C. Bogey, and C. Bailly, Comput. Fluids **35**, 1459 (2006).

[33] N. Kwatra, J. Su, J. T. Grétarsson, and R. Fedkiw, J. Comput. Phys. **228**, 4146 (2009).

[34] B. Müller, Math. Models Methods Appl. Sci. **12**, 139 (1990).

[35] D. Xiu and G. E. Karniadakis, J. Comput. Phys. **172**, 658 (2001).

[36] E. Celledoni, B. K. Kometa, and O. Verdier, J. Sci. Comput. **66**, 91 (2016).

[37] E. Sonnendrücker, J. Roche, P. Bertrand, and A. Ghizzo, J. Comput. Phys. **149**, 201 (1999).

[38] N. Crouseilles, M. Mehrenberger, and E. Sonnendrücker, J. Comput. Phys. **229**, 1927 (2010).

[39] J. M. Qiu and A. Christlieb, J. Comput. Phys. **229**, 1130 (2010).

[40] G. Russo, P. Santagati, and S. B. Yun, SIAM J. Num. Anal. **50**, 1111 (2012).

[41] M. Groppi, G. Russo, and G. Stracquadanio, arXiv:1411.7929.

[42] A. Krämer, K. Küllmer, D. Reith, W. Joppich, and H. Foysi, Phys. Rev. E **95**, 023305 (2017).

[43] A. Krämer, D. Wilde, K. Küllmer, D. Reith, H. Foysi, and W. Joppich, Comput. Math. Appl. **79**, 34 (2020).

[44] D. Wilde, A. Krämer, D. Reith, and H. Foysi, Phys. Rev. E **101**, 053306 (2020).

[45] A. H. Stroud, Math. Comput. **27**, 437 (1973).

[46] R. Cools, J. Complexity **19**, 445 (2003).

[47] D. Wilde, A. Krämer, M. Bedrunka, D. Reith, and H. Foysi, J. Comput. Sci. **51**, 101355 (2021).

[48] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, *The Lattice Boltzmann Method: Principles and Practice* (Springer Nature, Berlin, 2017).

[49] L. Chen, Q. Kang, Y. Mu, Y. L. He, and W. Q. Tao, Int. J. Heat Mass Transf. **76**, 210 (2014).

[50] L. P. Wang, C. Peng, Z. Guo, and Z. Yu, Comput. Fluids **124**, 226 (2016).

[51] Y. Peng, C. Shu, and Y. T. Chew, Phys. Rev. E **68**, 026701 (2003).

[52] M. Geier, M. Schönherr, A. Pasquali, and M. Krafczyk, Comput. Math. Appl. **70**, 507 (2015).

[53] B. Dorschner, F. Bösch, S. S. Chikatamarla, K. Boulouchos, and I. V. Karlin, J. Fluid Mech. **801**, 623 (2016).

[54] F. J. Alexander, H. Chen, S. Chen, and G. D. Doolen, Phys. Rev. A **46**, 1967 (1992).

[55] N. Frapolli, S. S. Chikatamarla, and I. V. Karlin, Phys. Rev. E **92**, 061301(R) (2015).

[56] N. Frapolli, S. S. Chikatamarla, and I. V. Karlin, Phys. Rev. Lett. **117**, 010604 (2016).

[57] N. Frapolli, S. S. Chikatamarla, and I. V. Karlin, Phys. Rev. E **93**, 063302 (2016).

[58] Y. Feng, P. Sagaut, and W. Q. Tao, Comput. Fluids **131**, 45 (2016).

[59] C. Coreixas, G. Wissocq, G. Puigt, J. F. Boussuge, and P. Sagaut, Phys. Rev. E **96**, 033306 (2017).

[60] M. Atif, M. Namburi, and S. Ansumali, Phys. Rev. E **98**, 053311 (2018).

[61] B. Dorschner, F. Bösch, and I. V. Karlin, Phys. Rev. Lett. **121**, 130602 (2018).

[62] Y. Feng, P. Boivin, J. Jacob, and P. Sagaut, J. Comput. Phys. **394**, 82 (2019).

[63] S. A. Hosseini, C. Coreixas, N. Darabiha, and D. Thévenin, Phys. Rev. E **100**, 063301 (2019).

[64] M. H. Saadat, F. Bösch, and I. V. Karlin, Phys. Rev. E **99**, 013306 (2019).

[65] M. H. Saadat, F. Bösch, and I. V. Karlin, Phys. Rev. E **101**, 023311 (2020).

[66] J. Latt, C. Coreixas, J. Beny, and A. Parmigiani, Philos. Trans. R. Soc., A **378**, 20190559 (2020).

[67] G. Farag, S. Zhao, T. Coratger, P. Boivin, G. Chiavassa, and P. Sagaut, Phys. Fluids **32**, 066106 (2020).

[68] C. Coreixas and J. Latt, Phys. Fluids **32**, 116102 (2020).

[69] X. Shan and X. He, Phys. Rev. Lett. **80**, 65 (1998).

[70] X. He, S. Chen, and G. D. Doolen, J. Comput. Phys. **146**, 282 (1998).

[71] N. I. Prasianakis and I. V. Karlin, Phys. Rev. E **76**, 016702 (2007).

[72] X. Nie, X. Shan, and H. Chen, in *Proceedings of the 47th AIAA Aerosp. Sci. Meet. Incl. New Horizons Forum Aerosp. Expo.* (AIAA, Orlando, FL, 2009).

[73] E. Fares, M. Wessels, Y. Li, P. Gopalakrishnan, R. Zhang, C. Sun, N. Gopalaswamy, P. Roberts, J. Hoch, and H. Chen, in *Proceedings of the 52nd AIAA Aerosp. Sci. Meet. AIAA Sci. Technol. Forum Expo. (SciTech'14)* (AIAA, National Harbor, MD, 2014).

[74] A. Bardow, I. V. Karlin, and A. A. Gusev, Europhys. Lett. **75**, 434 (2006).

[75] T. Lee and C.-L. L. Lin, J. Comput. Phys. **185**, 445 (2003).

[76] T. Kataoka and M. Tsutahara, Phys. Rev. E **69**, 035701(R) (2004).

[77] M. Min and T. Lee, J. Comput. Phys. **230**, 245 (2011).

[78] Z. Guo, K. Xu, and R. Wang, Phys. Rev. E **88**, 033305 (2013).

[79] Z. Guo, R. Wang, and K. Xu, Phys. Rev. E **91**, 033313 (2015).

[80] P. Pavlo, G. Vahala, and L. Vahala, Phys. Rev. Lett. **80**, 3960 (1998).

[81] I. Arasaratnam and S. Haykin, IEEE Trans. Autom. Control **54**, 1254 (2009).

[82] C. Thantanapally, S. Singh, D. V. Patil, S. Succi, and S. Ansumali, Int. J. Mod. Phys. C **24**, 1340004 (2013).

[83] S. Chapman and T. G. Cowling, *The Mathematical Theory of Nonuniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases* (Cambridge University Press, Cambridge, UK, 1970).

[84] P. J. Dellar, Comput. Math. Appl. **65**, 129 (2013).

[85] D. Wilde, A. Krämer, K. Küllmer, H. Foysi, and D. Reith, Int. J. Numer. Methods Fluids **90**, 156 (2019).

[86] D. Arndt, W. Bangerth, B. Blais, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, U. Köcher, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Proell, K. Simon, B. Turcksin, D. Wells, and J. Zhang, J. Num. Math. (2019).

[87] X. Shan, X. F. Yuan, and H. Chen, J. Fluid Mech. **550**, 413 (2006).

[88] A. Haegemans and R. Piessens, SIAM J. Num. Anal. **14**, 492 (1977).

[89] S. Konyaev, in *Doklady Akademii Nauk*, Vol. 233 (Russ. Acad. of Sci., 1977), pp. 784–787.

[90] J. R. Van Zandt, ETNA Electron. Trans. Numer. Anal. **51**, 219 (2019).

[91] K. C. Uga, M. Min, T. Lee, and P. F. Fischer, Comput. Math. Appl. **65**, 239 (2013).

[92] P. Lax and X. Liu, SIAM J. Sci. Comput. **19**, 319 (1998).

[93] A. Kurganov and E. Tadmor, Numer. Methods Partial Differ. Equations **18**, 584 (2002).

[94] N. Peng and Y. Yang, Phys. Rev. Fluids **3**, 013401 (2018).

[95] D. J. Lusher and N. D. Sandham, AIAA J. **59**, 533 (2021).

[96] C. Pruett, Theor. Comput. Fluid Dyn. **22**, 275 (2008).

[97] D. Oberle, C. D. Pruett, and P. Jenny, Phys. Fluids **32**, 065112 (2020).

[98] D. N. Siebert, L. A. Hegele, Jr., and P. C. Philippi, Phys. Rev. E **77**, 026707 (2008).

[99] C. Sun and A. T. Hsu, Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top. **68**, 016303 (2003).

[100] M. Watari and M. Tsutahara, Physica A **364**, 129 (2006).

[101] K. Noah, F. S. Lien, and E. Yee, Int. J. Numer. Methods Fluids **93**, 927 (2021).

[102] T. Chen, X. Wen, L. P. Wang, Z. Guo, J. Wang, and S. Chen, Phys. Fluids **32**, 125104 (2020).

[103] P. Pavlo, G. Vahala, and L. Vahala, J. Stat. Phys. **107**, 499 (2002).

[104] A. V. Zakirov, B. A. Korneev, V. D. Levchenko, and A. Y. Perepelkina, Keldysh Inst. Prepr. 1 (2019).

[105] N. Frapolli, Entropic lattice Boltzmann models for thermal and compressible flows, Ph.D. thesis, ETH Zürich, Switzerland, 2017.

[106] N. Frapolli, S. Chikatamarla, and I. Karlin, Entropy **22**, 370 (2020).

[107] F. Renard, Y. Feng, J.-F. Boussuge, and P. Sagaut, Comput. Fluids **219**, 104867 (2021).

[108] M. H. Saadat, S. A. Hosseini, B. Dorschner, and I. V. Karlin, Phys. Fluids **33**, 046104 (2021).

[109] M. Kronbichler and K. Kormann, ACM Trans. Math. Software **45**, 03590 (2019).

[110] S. Y. Cho, S. Boscarino, G. Russo, and S. B. Yun, J. Comput. Phys. **432**, 110159 (2021).

[111] G. Di Ilio, B. Dorschner, G. Bella, S. Succi, and I. V. Karlin, J. Fluid Mech. **849**, 35 (2018).

[112] K. Küllmer, Pseudopotentialbasierte Lattice-Boltzmann-Verfahren für Mehrphasenströmungen, Ph.D. thesis, Universität Siegen, Germany, 2019.

[113] C. Coreixas, High-order extension of the recursive regularized lattice Boltzmann method, Ph.D. thesis, INP Toulouse, France, 2018.