Parallel finite-volume discrete Boltzmann method for inviscid compressible flows on unstructured grids

Lei Xu,[1,2] Rongliang Chen,[1,2,*] and Xiao-Chuan Cai[3,†]

[1] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China*
[2] *Shenzhen Key Laboratory for Exascale Engineering and Scientific Computing, Shenzhen 518055, China*
[3] *Department of Mathematics, University of Macau, Macau, China*

(Received 4 September 2020; revised 7 December 2020; accepted 11 January 2021; published 19 February 2021)

In this paper, a finite-volume discrete Boltzmann method based on a cell-centered scheme for inviscid compressible flows on unstructured grids is presented. In the new method, the equilibrium distribution functions are obtained from the circle function in two-dimensions (2D) and the spherical function in three-dimensions (3D). Moreover, the advective fluxes are evaluated by Roe's flux-difference splitting scheme, the gradients of the density and total energy distribution functions are computed with a least-squares method, and the Venkatakrishnan limiter is employed to prevent oscillations. To parallelize the method we use a graph-based partitioning approach that also guarantees the load balancing. The method is validated by seven benchmark problems: (a) a 2D flow pasting a bump, (b) a 2D Riemann problem, (c) a 2D flow passing the RAE2822 airfoil, (d) flows passing the NACA0012 airfoil, (e) 2D supersonic flows around a cylinder, (f) an explosion in a 3D box, and (g) a 3D flow around the ONERA M6 wing. The benchmark tests show that the results obtained by the proposed method match well with the published results, and the parallel numerical experiments show that the proposed parallel implementation has close to linear strong scalability, and parallel efficiencies of 95.31% and 94.56% are achieved for 2D and 3D problems on a supercomputer with up to 4800 processor cores, respectively.

DOI: 10.1103/PhysRevE.103.023306

I. INTRODUCTION

After three decades of development, the standard lattice Boltzmann method (SLBM), initially developed from lattice gas automata [1], has become an alternative method for simulating incompressible fluid flows. The major advantage and main limitation of SLBM are that the method relies on a uniform Cartesian grid. Because of the uniform Cartesian grid, SLBM with different collision models [2] can achieve second-order spatial-temporal accuracy, and its intrinsic parallelism makes it suitable for supercomputers with a large number of processor cores. However, SLBM requires a high-resolution grid near the body and the walls, which makes it not suitable for practical engineering problems with complex curved boundaries or strong local gradients. Therefore, many versions of LBM on nonuniform grids have been developed, such as curvilinear [3], structured [4], multiblock [5], locally refined, adaptive mesh refinement [6], unstructured grids [7–9], and overlapping grids [10,11]. In this work, we focus on unstructured grids in two and three dimensions.

For incompressible flows, there are several approaches for unstructured grids. For example, Misztal *et al.* analyzed the standard forward Euler method and the operator splitting method on unstructured grids and proved that their cell-vertex finite-volume method (FVM) yields the Navier-Stokes equations by means of Chapman-Enskog expansion [12]. Li and Luo proposed a cell-centered finite-volume discrete Boltzmann method (FVDBM) [13,14] with the multiple-relaxation-time collision model on arbitrary grids for the efficient treatment of complex geometries [15]. Later, they devised a fully implicit FVDBM on arbitrary grids, and the corresponding linear system is solved by the block lower-upper symmetric-Gauss-Seidel algorithm [16]. Chen and Schaefer proposed a simple unified Gdounov-type upwind approach that does not need a Riemann solver for the face flux calculation on an unstructured cell-centered triangular grid [17]. To increase the temporal accuracy of solution and reduce the computational time, they also proposed an implicit Bhatnagar-Gross-Krook (BGK) collision by a semi-Lagrangian approach [18]. Di Ilio and his coworkers merged SLBM with FVDBM for simulating the flow with sharp features on an overlapping grid system, consisting of a uniform lattice nodes and a coordinate-free lattice structure [10,11].

However, for compressible flows there are quite a few difficulties. In SLBM for incompressible flows, a Taylor-series expansion of the Maxwellian function in terms of the Mach number is used to obtain a lattice Boltzmann version of the polynomial form, which inevitably limits the range of the Mach number. To simulate compressible flows, several modifications have been proposed. Alexander *et al.* [19] and Qian [20] proposed a multispeed (MS) approach that is a straightforward extension of SLBM with a low Mach number and contains nonlinear error terms in the macroscopic equations after the Chapman-Enskog expansion. Chen *et al.* [21] found that the particle distribution functions should contain high-order terms to eliminate the nonlinear terms and obtain the correct momentum flux and the heat flux to

[*] Corresponding author: xccai@um.edu.mo
[†] Corresponding author: rl.chen@siat.ac.cn

derive the macroscopic equations. Following Chen's work, Watari and Tsutahara [22] proposed a new truncated form of the Maxwellian distribution function with excellent numerical stability and more accurate simulations. Yan *et al.* [23] proposed a 2D LBM with three energy levels on a square lattice for the compressible Euler equations. Shi *et al.* [24] constructed a D2Q9 lattice for the compressible Euler equation by allowing the particles to possess both kinetic and thermal energies, and they used Harten's minmod total variation diminishing (TVD) finite difference scheme to solve the discrete velocity Boltzmann equation (DVBE) [25]. To simulate high-speed compressible flows, Sun *et al.* [26,27] presented a locally adaptive LBM in which a simplified equilibrium distribution function is used to replace the Maxwellian distribution function and a very large particle-velocity set is incorporated to enable greater variation in the mean velocity. This model can efficiently handle flows over a wide range of Mach numbers and capture shock waves. However, the lattices can no longer stream from one node to another, and a reconstruction is needed to obtain the values on the nodes. He *et al.* first proposed the double distribution function (DDF) model by adding an internal energy distribution function [28], and this model involves complicated gradient terms about the macroscopic flow variables. Guo *et al.* proposed a decoupled DDF model by introducing a total energy distribution function to replace He's internal energy distribution function [29]. Li *et al.* proposed a coupled DDF model with a flexible specific-heat ratio and Prandtl number. In their method, a particle distribution function based on the D2Q12 lattice is used to recover the compressible continuity and momentum equations, and an energy distribution function is used to recover the compressible energy equation [30]. Qu *et al.* [31] developed a 2D model for inviscid compressible flows using the D2Q13L2 lattice velocity model. They replaced the Maxwellian distribution function with a circular function. A Lagrangian interpolation polynomial was proposed to construct the equilibrium distribution function, and the DVBE was solved by a second-order TVD FVM. Based on Qu's work, Li *et al.* [32] presented a potential energy DDF model that remains part of the MS approach. To simulate three-dimensional compressible flows, Li and He *et al.* extended the circular function in 2D to a spherical function in 3D [33]. Qiu *et al.* compared two 3D lattice velocity models of compressible flows with different particle distribution functions and total energy distribution functions [34]. Besides these, a DDF thermal LBM was proposed by Feng and his coworkers by expanding the distribution functions with the Grad's moment expansion approach in terms of Hermite polynomials [35,36]. Saadat *et al.* presented a DDF LBM on standard lattices discretizing phase space through direct matching of moments [37,38]. Recently, a novel class of shifted lattices is developed, which improves the operating range of lattice Boltzmann simulations [39,40]. In this paper, the DDF kinetic equations for compressible flows are used to obtain the particle and total energy distribution functions, and the D2Q13 and D3Q25 lattice velocity models are employed to discretize the circular function in 2D and spherical function in 3D, respectively. Then, a second-order FVM on unstrutured grids and a Runge-Kutta scheme are used to discretize the spatial and temporal terms, respectively.

Although a lot of research has been carried out on parallel SLBM algorithms in homogeneous [41–43] and heterogeneous [44,45] environments, there has not been any development of parallel methods for compressible flows on unstructured grids. In this paper, to reduce the compute time and solve large-scale problems, we introduce a scalable parallel FVDBM for compressible flows on unstructured grids, and the parallel performance of the algorithm is carefully studied on a supercomputer with up to 4800 processors. Using seven benchmark problems, we show that nearly ideal speedup can be achieved for both 2D and 3D problems.

The rest of the paper is organized as follows. In Sec. II, the DDF kinetic equations for inviscid compressible flows are described. Section III introduces a new cell-centered FVM on unstructured grid. A parallelization strategy for the algorithm is given in Sec. IV. Some computational results for seven benchmark problems are presented in Sec. V. The parallel performance of the algorithm is discussed in Sec. VI. Finally, brief conclusions are summarized in Sec. VII.

II. COUPLED DOUBLE-DISTRIBUTION FUNCTION FOR INVISCID COMPRESSIBLE FLOWS

The coupled DDF LBM has two discrete Boltzmann BGK equations for the particle and the total energy distribution functions [29,30,34]

$$\frac{\partial f_\alpha(\mathbf{x},t)}{\partial t}+(\mathbf{e}_\alpha\cdot\nabla)f_\alpha(\mathbf{x},t)=-\frac{1}{\tau_f}\big[f_\alpha(\mathbf{x},t)-f_\alpha^{\rm eq}(\mathbf{x},t)\big], \tag{1}$$

$$\begin{aligned}\frac{\partial h_\alpha(\mathbf{x},t)}{\partial t}&+(\mathbf{e}_\alpha\cdot\nabla)h_\alpha(\mathbf{x},t)\\ &=-\frac{1}{\tau_h}\big[h_\alpha(\mathbf{x},t)-h_\alpha^{\rm eq}(\mathbf{x},t)\big]\\ &\quad+\frac{1}{\tau_{hf}}(\mathbf{e}_\alpha\cdot\mathbf{u})\big[f_\alpha(\mathbf{x},t)-f_\alpha^{\rm eq}(\mathbf{x},t)\big],\end{aligned} \tag{2}$$

where f_α and h_α are the particle and total energy distribution functions at the point $\mathbf{x}$ and time t in the αth direction of the lattice velocity $\mathbf{e}_\alpha$, respectively. $f_\alpha^{\rm eq}$ and $h_\alpha^{\rm eq}$ are the corresponding equilibrium distribution functions. τ_f and τ_h are the particle and total energy relaxation times, respectively. $\mathbf{u}$ is the macroscopic flow velocity, and τ_{hf} is defined as

$$\tau_{hf}=\frac{\tau_h\tau_f}{\tau_f-\tau_h}.$$

The equilibrium distribution functions of the particle and total energy should satisfy the following velocity moment condition in order to recover the compressible Euler equations:

$$\sum_\alpha f_\alpha^{\rm eq}=\rho,$$

$$\sum_\alpha f_\alpha^{\rm eq}e_{\alpha i}=\rho u_i,$$

$$\sum_\alpha f_\alpha^{\rm eq}e_{\alpha i}e_{\alpha j}=\rho u_i u_j+p\delta_{ij},$$

$$\sum_\alpha f_\alpha^{\rm eq}e_{\alpha i}e_{\alpha j}e_{\alpha k}=\rho u_i u_j u_k+p(u_k\delta_{ij}+u_j\delta_{ik}+u_i\delta_{jk}),$$

$$\sum_\alpha h_\alpha^{\mathrm{eq}} = \rho E,$$
$$\sum_\alpha h_\alpha^{\mathrm{eq}} e_{\alpha i} = (\rho E + p)u_i,$$
$$\sum_\alpha h_\alpha^{\mathrm{eq}} e_{\alpha i} e_{\alpha j} = (\rho E + 2p)u_i u_i + p(E + RT)\delta_{ij},$$

where ρ is the density, $p = \rho RT$ is the pressure, R is the specific gas constant, T is the temperature, $E = \frac{bRT}{2} + \frac{u^2}{2}$ is the total energy, and $b = \frac{2}{\gamma - 1}$ is a constant calculated from the specific heat ratio γ. i, j, and k denote the components of the x, y, and z directions. δ_{ij}, δ_{ik}, and δ_{jk} are the Kronecker delta functions. The Prandtl number is defined by $\mathrm{Pr} = \frac{\tau_f}{\tau_h}$.

Because the equilibrium distribution functions are derived from the low-Mach-number expansion of the Maxwellian function, LBM has difficulty in simulating high-speed compressible flows. Qu *et al.* presented a scheme to construct a lattice Boltzmann model without the assumption of a low Mach number [46]. The Maxwellian function is replaced by a simple circular function in their scheme, and the equilibrium distribution functions are constructed by Lagrangian interpolation polynomials to assign the circular function to a D2Q13 lattice velocity model [see Fig. 1(a)]. Based on Qu's work, Li *et al.* extended the scheme and proposed a D3Q25 lattice velocity model [see Fig. 1(b)] for the 3D cases by replacing the Maxwellian function with a spherical function [33]. The discrete velocity model can be expressed as

$$\mathbf{e} = \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 & 2 & 0 & -2 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 & 0 & 2 & 0 & -2 \end{pmatrix} \times c$$

in 2D and

$$\mathbf{e} = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 2 & -2 \end{pmatrix} \times c$$

in 3D, where $c = \sqrt{dRT_c}$, T_c is the characteristic temperature, and d is the spatial dimension. The total energy equilibrium distribution function h_α^{eq} can be calculated by the particle distribution function f_α^{eq} with the following equation:

$$h_\alpha^{\mathrm{eq}} = [E + (\mathbf{e}_\alpha - \mathbf{u}) \cdot \mathbf{u}] f_\alpha^{\mathrm{eq}} + \varpi_\alpha \frac{p}{c^2} RT,$$

where

$$\varpi_\alpha = \begin{cases} 0, & \alpha = 0, \\ -\frac{1}{3}, & \alpha = 1\text{–}4, \\ \frac{1}{4}, & \alpha = 5\text{–}8, \\ \frac{1}{12}, & \alpha = 9\text{–}12, \end{cases}$$

in 2D [30] and

$$\varpi_\alpha = \begin{cases} 0, & \alpha = 0, \\ -\frac{5}{14}, & \alpha = 1\text{–}6, \\ \frac{1}{7}, & \alpha = 7\text{–}18, \\ \frac{1}{14}, & \alpha = 19\text{–}24, \end{cases}$$

in 3D [34]. The macroscopic variables (density ρ, velocity $\mathbf{u}$, total energy E, internal energy e, and pressure p) can be calculated by

$$\sum_\alpha f_\alpha = \rho,$$
$$\sum_\alpha f_\alpha \mathbf{e}_\alpha = \rho \mathbf{u},$$
$$\sum_\alpha h_\alpha = \rho E,$$
$$e = E - \frac{u^2}{2},$$
$$p = (\gamma - 1)\rho e. \tag{3}$$

III. A FINITE-VOLUME DISCRETIZATION OF THE DISCRETE VELOCITY BOLTZMANN BGK EQUATIONS ON UNSTRUCTURED GRIDS

In this section we first present the formulation of the finite-volume discrete Boltzmann scheme. The DVBE based on (1) and (2) can be rewritten as

$$\frac{\partial}{\partial t} g_\alpha(\mathbf{x}, t) + (\mathbf{e}_\alpha \cdot \nabla) g_\alpha(\mathbf{x}, t) = \Omega_{g\alpha}(\mathbf{x}, t), \tag{4}$$

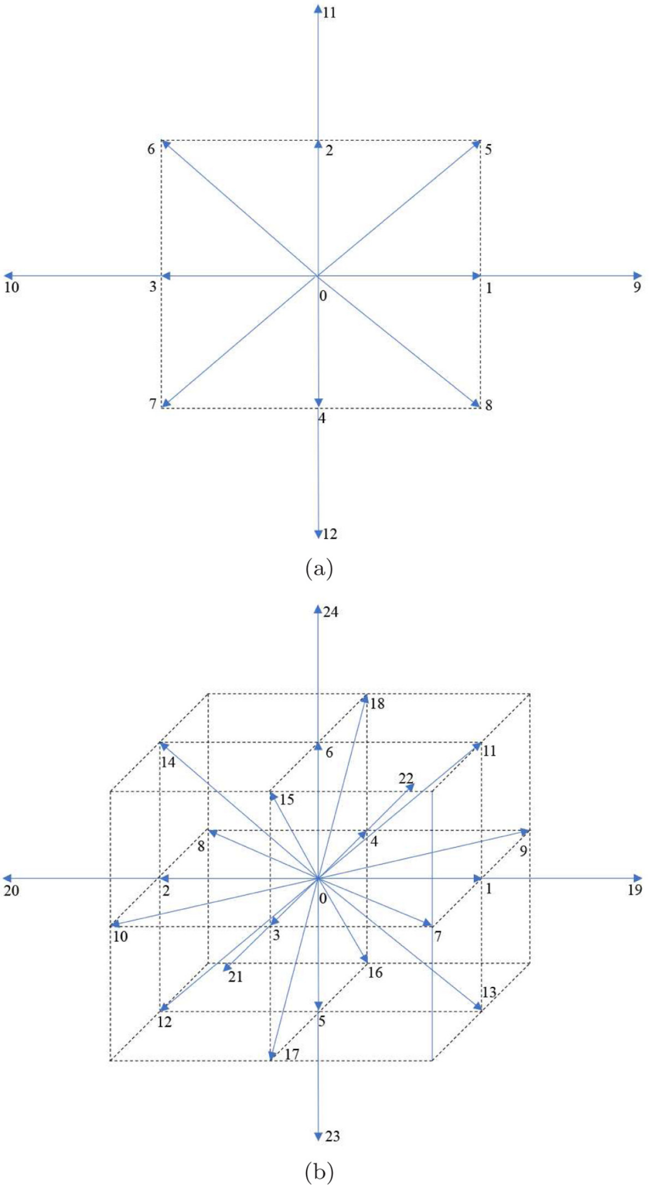
where $g = f, h$, $\Omega_{f\alpha} = -\frac{1}{\tau_f}(f_\alpha - f_\alpha^{\mathrm{eq}})$, and $\Omega_{h\alpha} = -\frac{1}{\tau_h}(h_\alpha - h_\alpha^{\mathrm{eq}}) + \frac{1}{\tau_{hf}}(\mathbf{e}_\alpha \cdot \mathbf{u})(f_\alpha - f_\alpha^{\mathrm{eq}})$.

We divide the fluid domain into many nonoverlapping polygons in 2D or polyhedrons in 3D, which are called control volumes (CVs) [47,48]. Figure 2 shows a schematic of the FVM with a cell-centered scheme. The geometric centroid i stores the distribution functions and macroscopic variables. The outward normal unit vector of the interface shared by cell i and j is $\mathbf{n}_{ij}$ (from cell i to cell j), whose face area is denoted as l_{ij}. Take the integration over cell i for both sides of (4), and then the Gauss theorem is used to transform the volume integrals of the advective term into boundary integrals, which yields

$$\frac{\partial}{\partial_t} \int_{V_i} g_\alpha(\mathbf{x}, t) dV + \oint_{\partial V_i} (\mathbf{e}_\alpha \cdot \mathbf{n}) g_\alpha(\mathbf{x}, t) dl = \int_{V_i} \Omega_{g\alpha} dV, \tag{5}$$

where V_i denotes the cell volume of cell i, ∂V_i is the boundary of cell i, and $\mathbf{n}$ is the outward unit vector, locally normal to ∂V_i. The distribution function at the centroid of the cell i is defined by $g_{\alpha i}(t)$, which is assumed to denote the cell-averaged value,

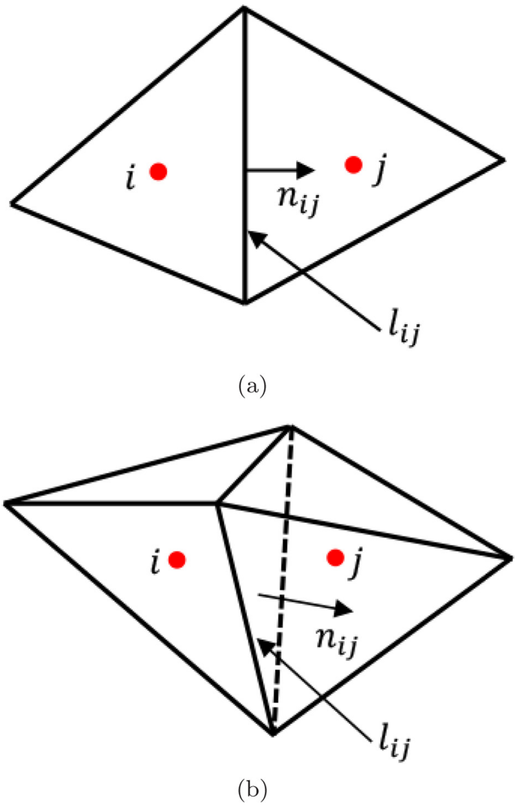
$$g_{\alpha i}(t) = \frac{1}{V_i} \int g_\alpha(\mathbf{x}, t) dV. \tag{6}$$



(a)

(b)

FIG. 1. Lattice velocity model in 2D and 3D. (a) D2Q13 and (b) D3Q25.



(a)

(b)

FIG. 2. Cell-centered scheme on unstructured grids in 2D (a) and 3D (b).

Equation (5) can be rewritten in a semidiscrete form as

$$\frac{d}{dt}g_{\alpha i}(t) = A_{\alpha i} + C_{\alpha i},$$

where

$$C_{\alpha i} = -\int_{V_i} \Omega_{g\alpha} dV, \tag{7}$$

and

$$A_{\alpha i} = -\frac{1}{V_i}\oint_{\partial V_i} G(g_{\alpha i})dl, \tag{8}$$

where $C_{\alpha i}$ and $A_{\alpha i}$ denote the collision and advection of particles, respectively. $G(g_\alpha) = (\mathbf{e}_\alpha \cdot \mathbf{n})g_\alpha$ is the flux density of g_α across the cell interface. Since the distribution function g_α is taken as the cell-averaged value in (6) and (7) can be calculated by

$$C_{\alpha i} = \int_{V_i} \Omega_{g\alpha}(g_\alpha)dV = \Omega_{g\alpha}(g_{\alpha i}). \tag{9}$$

Equation (8) can be discretized along the cell faces belonging to the cell boundary ∂V_i,

$$A_{\alpha i} = -\frac{1}{V_i}\sum_{j=1}^{Ns} G(g_{\alpha ij})l_{ij}, \tag{10}$$

in which $g_{\alpha ij}$ is the value at the cell interface. Based on the right-hand side of (10), the values of the cell interface need to be reconstructed within V_i, and the flux density G is computed along the cell interface. Roe's flux-difference splitting scheme is employed to evaluate the advective fluxes at the face of the CV from the left and right states by solving the Riemann problem [49]. This scheme can be expressed as

$$G(g_{\alpha ij}) = \tfrac{1}{2}\big[G\big(g_\alpha^L\big) + G\big(g_\alpha^R\big) - |\alpha^{(ij)}|\big(g_\alpha^R - g_\alpha^L\big)\big], \tag{11}$$

where the factor $|\alpha^{(ij)}|$ is the scaled characteristic speed, which is taken to be equal to the scaled microscopic velocity normal to the cell interface, $\mathbf{e}_\alpha \cdot \mathbf{n}_{ij}$; and g_α^L and g_α^R are the left and right distribution functions, respectively. The left and right states of the interface are reconstructed by a piecewise linear reconstruction, which assumes that the distribution function is piecewise linearly distributed over the CV. This can be written as

$$\begin{aligned} g_\alpha^L &= g_{\alpha i} + \Psi_{\alpha i}^g(\nabla g_{\alpha i} \cdot r_L),\\ g_\alpha^R &= g_{\alpha j} + \Psi_{\alpha j}^g(\nabla g_{\alpha j} \cdot r_R), \end{aligned} \tag{12}$$

where r_L and r_R point from the cell centroid to the face midpoint, $\nabla g_{\alpha i}$ and $\nabla g_{\alpha j}$ are the gradients of g_α at the cell centroid i and j, and $\Psi^g_{\alpha i}$ and $\Psi^g_{\alpha j}$ are limiter functions. The gradient of $g_{\alpha i}$ is computed by the inverse distance weighted least-squares method [50]. On an unstructured grid, the limiters are applied to reduce the gradients, which can prevent the generation of oscillations. The Venkatakrishnan limiter [51,52] is widely used due to its superior convergence properties. It reduces the reconstructed gradient $\nabla g_{\alpha i}$ by the factor [47]

$$\Psi_{\alpha i} = \min_j \begin{cases} \frac{1}{\Delta_2}\left[\frac{(\Delta^2_{1,\max}+\epsilon^2)\Delta_2+2\Delta^2_2\Delta_{1,\max}}{\Delta^2_{1,\max}+2\Delta^2_2+\Delta_{1,\max}\Delta_2+\epsilon^2}\right] & \text{if } \Delta_2 > 0, \\ \frac{1}{\Delta_2}\left[\frac{(\Delta^2_{1,\min}+\epsilon^2)\Delta_2+2\Delta^2_2\Delta_{1,\min}}{\Delta^2_{1,\min}+2\Delta^2_2+\Delta_{1,\min}\Delta_2+\epsilon^2}\right] & \text{if } \Delta_2 < 0, \\ 1 & \text{if } \Delta_2 = 0, \end{cases} \tag{13}$$

where

$$\Delta_2 = \nabla g_{\alpha i}\cdot r_L,$$
$$\Delta_{1,\max} = g_{\max} - g_{\alpha i},$$
$$\Delta_{1,\min} = g_{\min} - g_{\alpha i}.$$

Here $g_{\max}$ and $g_{\min}$ are the maximum and minimum values of all neighboring cells, including the cell itself.

In the rest of this section, we discuss a time marching method. The second-order Runge-Kutta method is employed to discretize the temporal term. Let $g^n_{\alpha i}$ and $g^{n+1}_{\alpha i}$ be the distribution functions at time levels t^n and t^{n+1}, respectively. The semidiscretization of (4) can be written as

$$g^{n+1}_{\alpha i} = g^n_{\alpha i} + \Delta t\left(\frac{1}{2}K^g_{\alpha i1} + \frac{1}{2}K^g_{\alpha i2}\right), \tag{14}$$

$$K^g_{\alpha i1} = A^g_{\alpha i} + C^g_{\alpha i},\, K^g_{\alpha i2} = \left(A^g_{\alpha i} + C^g_{\alpha i}\right)|_{g^*_{\alpha i}=g^n_{\alpha i}+\Delta t K^g_{\alpha i1}}, \tag{15}$$

where $\Delta t = t^{n+1} - t^n$ is the time-step size. The Courant-Friedrichs-Lewy (V_{CFL}) criterion is given by [7]

$$\Delta t = V_{\text{CFL}}\frac{V_{\min}}{(|\mathbf{e}|_{\max} + |\mathbf{u}|_{\max})\left(l^x_{\min} + l^y_{\min}\right)},$$

where $V_{\min}$ is the minimal cell volume in the domain, $|\mathbf{u}|_{\max}$ is the magnitude of the maximum macroscopic velocity at each time step in the computational domain, and $l^x_{\min}$ and $l^y_{\min}$ are the projected lengths of the minimal volume cell in the x and y directions, respectively. The value of the V_{CFL} criterion is set to be 0.4 in this paper.

For the boundary conditions, the ghost cell method [47,48] and the nonequilibrium extrapolation scheme [53] are adopted. Figure 3 shows a boundary cell ABC with its centroid C_i and the edge AB lies on the boundary. The ghost cell ABC' with centroid C'_i is the reflective image about the boundary edge AB. To deal with the subsonic and supersonic inlet and outlet boundaries, the values of the macroscopic variables ρ, $\mathbf{u}$ and E at the centroid C'_i are given on the boundries. For the Neumann boundary condition, the macroscopic variables at the centroid C'_i are assumed to be equal to the values at the centroid C_i. On the slip wall, the density ρ and the total energy E at the centroid C'_i are taken as the values at the centroid C_i, and the velocity at the centroid C'_i is calculated by reflecting the velocity vector at the centroid C_i, which can be written as

$$\mathbf{u}(C'_i) = \mathbf{u}(C_i) - 2(\mathbf{u}(C_i)\cdot\mathbf{n})\mathbf{n}.$$

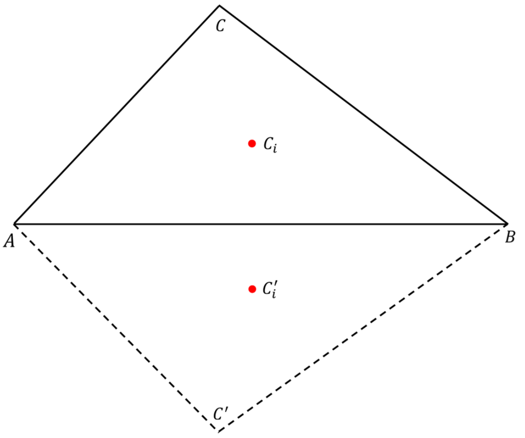


FIG. 3. Illustration of the ghost cell method.

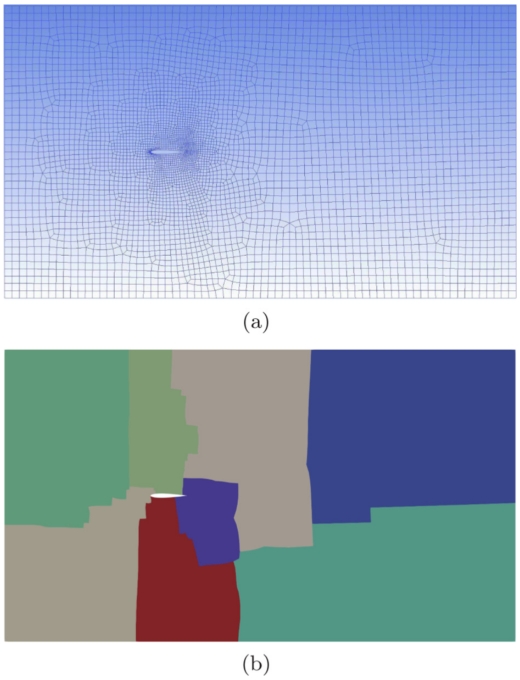
Both the particle distribution functions and the total energy distribution functions at the centroid C_i and C'_i can be divided into two parts: equilibrium part and nonequilibrium part

$$f_\alpha(C'_i) = f^{\text{eq}}_\alpha(C'_i) + f^{\text{neq}}_\alpha(C'_i),$$
$$h_\alpha(C'_i) = h^{\text{eq}}_\alpha(C'_i) + h^{\text{neq}}_\alpha(C'_i).$$

$f^{\text{eq}}_\alpha(C_i)$ and $h^{\text{eq}}_\alpha(C'_i)$ are obtained based on the macroscopic variables at the centroid C'_i and $f^{\text{neq}}_\alpha(C'_i)$ and $h^{\text{neq}}_\alpha(C'_i)$ are approximated by $f^{\text{neq}}_\alpha(C_i)$ and $h^{\text{neq}}_\alpha(C_i)$, respectively. Then $f_\alpha(C'_i)$ and $h_\alpha(C'_i)$ can be calculated by

$$f_\alpha(C'_i) = f^{\text{eq}}_\alpha(C'_i) + f^{\text{neq}}_\alpha(C_i), \tag{16}$$

$$h_\alpha(C'_i) = h^{\text{eq}}_\alpha(C'_i) + h^{\text{neq}}_\alpha(C_i). \tag{17}$$


(a)

(b)

FIG. 4. An example of the unstructured grid (a) for the flow simulation around the NACA0012 airfoil and its partition (b) for parallel computing. Different colors denote different subdomains.

Algorithm 1. Parallel FVDBM for compressible flows.

```
Input: The unstructured grid, initial macroscopic conditions, boundary conditions, the number of MPI processes;
Output: Simulation results;
Start the simulation:
1:  A processor reads the grid file;
2:  Decompose the grid parallelly using ParMETIS, and distribute the subdomains to the corresponding processors;
3:  Each processor generates the ghost cells, computes the grid information, and allocate buffers to send and receive data;
4:  Initialize the macroscopic variables $\rho$, $\mathbf{u}$, $E$, the distribution functions $g_{\alpha i}$;
5:  repeat
6:      Transfer the distribution functions of the outermost cells in a subdomain to the neighboring processors;
7:      Deal with the boundary conditions on (16) and (17);
8:      for all faces in the unstructured grid do
9:          Based on the least-squares method, compute the gradients $\nabla g_{\alpha i}$ and $\nabla g_{\alpha j}$ for the left cell $i$ and the right cell $j$;
10:         Calculate the limiters $\Psi_{\alpha i}$ and $\Psi_{\alpha j}$ based on (13);
11:         Reconstruct the left and right states $g_\alpha^L$ and $g_\alpha^R$ of the face with (12);
12:         Obtain the flux density $G(g_{\alpha ij})$ according to (11), and then update the advective flux $A_{\alpha i}^g$ and $A_{\alpha j}^g$;
13:     end for
14:     for all cells in the unstructured grid do
15:         Execute the collision term $C_{\alpha i}$ on the basis of (9);
16:         Update the macroscopic variables $\rho$, $\mathbf{u}$ and $E$ in accordance with (3);
17:     end for
18:     For all cells, calculate $K_{\alpha i1}^g$ and $g_{\alpha i}^*$ based on (15);
19:     Repeat 6–17 to compute $K_{\alpha i2}^g$ in (15);
20:     Update the distribution functions with (14);
21: Until the stopping conditions are satisfied;
22: Output the simulation results.
```

IV. A PARALLELIZATION STRATEGY FOR THE PROPOSED ALGORITHM

To parallelize the proposed algorithm, the unstructured grid must be decomposed into N subdomains (N is the number of processor cores of the parallel computer). In our work, ParMETIS [54] is used to partition the unstructured grid which is regarded as a graph whose vertices are cells of the grid. Figure 4 shows an example of the grid for the simulation of flows around the NACA0012 airfoil and its partition into eight subdomains using ParMETIS. Note that the shapes of the subdomains are different, but all the subdomains have nearly the same number of grid cells, and thus the computational load is well balanced. One of the reasons for the different shapes is that the grid is nonuniform with finer grid cells near the airfoil. In this paper, the algorithm is implemented using DMPlex in the open source package PETSc [55]. In the FVDBM, most operations are local except the calculation of the advective fluxes, which requires the distribution functions of the neighboring cells, each processor core has to send and receive the values of the distribution functions in the outermost cells. Moreover, before the calculation of advective fluxes, the processor must exchange the particle distribution functions and the total energy distribution functions with the neighboring processors, and a global synchronization is needed until the data exchange is finished. The framework of the parallel FVDBM for compressible flows is described in Algorithm1. For the steady compressible flows, the stopping conditions are

$$\frac{\|\mathbf{u}^{n+1}-\mathbf{u}^n\|_2}{\|\mathbf{u}^n\|_2} \leqslant 10^{-6}, \quad \frac{\|\mathbf{E}^{n+1}-\mathbf{E}^n\|_2}{\|\mathbf{E}^n\|_2} \leqslant 10^{-5},$$

where $\mathbf{u}^{n+1}$ and $\mathbf{E}^{n+1}$ refer to the macroscopic velocity and the total energy at the time t^{n+1}, respectively. To compute the norms in the stopping conditions, global communication and synchronization are necessary.

V. NUMERICAL SIMULATIONS

In this section, we discuss some benchmark problems to validate the proposed method. 7 benchmark problems are considered: (a) a flow passing a bump in a channel, (b) a 2D Riemann problem, (c) a flow passing the RAE2822 airfoil, (d) flows passing the NACA0012 airfoil, (e) supersonic flows around a cylinder, (f) an explosion in a 3D box, and (g) a flow around the ONERA M6 wing. In all the numerical experiments, the dimensionless form of the equations is used. There are three independent reference variables for the normalization including the reference density ρ_0, the reference length L_0 and the reference internal energy e_0. The other reference variables are defined as

$$u_0 = \sqrt{e_0}, \quad t_0 = \frac{L_0}{u_0},$$

where u_0 is the reference velocity and t_0 is the reference time [31]. The reference internal energy e_0 should be set to be slightly higher than the maximum specific stagnation internal energy to ensure that the circular function in 2D or spherical function in 3D is located inside the lattice to avoid extrapolation [31]. The specific-heat ratio and the Prandtl number are set to be 1.4 and 0.71, respectively.

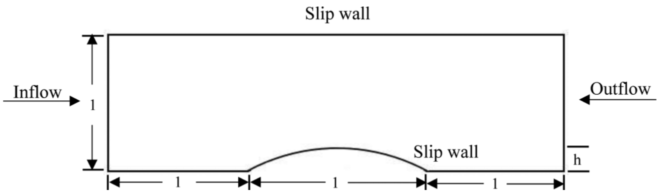


FIG. 5. The setup of the flow simulation in the channel with a circular bump, $h = 0.1$ is the height of the bump.

A. A flow passing a bump in a channel

In this experiment, a steady-state flow through the 2D GAMM channel is simulated [56] (see Fig. 5). The channel has a circular bump with 10% height. The inflow Mach number is $M_\infty = 0.675$, and a computational grid with 27 784 quadrangle cells is used. The top and bottom walls are slip adiabatic boundaries, the left boundary is the subsonic inlet, and the right boundary is the subsonic outlet.

The Mach number contours are shown in Fig. 6, and the Mach number profiles on the bottom wall are plotted in Fig. 7, which are in agreement with the results calculated by solving Euler equations with WENO scheme [57].

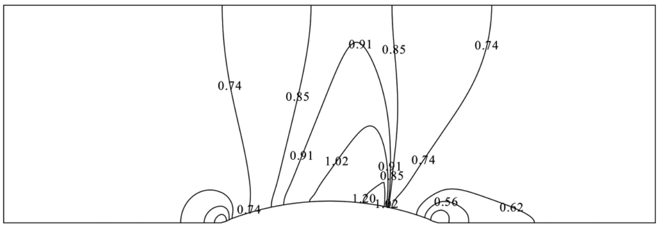


FIG. 6. Mach number contours ($M_\infty = 0.675$).

B. A two-dimensional Riemann problem

The 2D unsteady Riemann problem is a popular benchmark. It consists of a square domain of width 1 and height 1. A homogeneous Neumann boundary condition is imposed on four walls. In this experiment, the first configuration of the work of Kurganov and Tadmov is considered [58]. The initial conditions are given as

$$(\rho, u, v, p) = \begin{cases} (1, 0, 0, 1) & \text{if } 0.5 < x < 1 \text{ and } 0.5 < y < 1, \\ (0.5197, -0.7259, 0, 0.4) & \text{if } 0 < x \leqslant 0.5 \text{ and } 0.5 < y < 1, \\ (0.1072, -0.7259, -1.4045, 0.0439) & \text{if } 0 < x \leqslant 0.5 \text{ and } 0 < y \leqslant 0.5, \\ (0.2579, 0, -1.4045, 0, 0.15) & \text{otherwise.} \end{cases}$$

To simulate this flow, a grid with 11 603 nodes and 22 804 cells is used. The result is compared with the result obtained by the shock capturing scheme [58] on a 400 × 400 uniform grid. It can be seen in Fig. 8 that the shock of the density contours (top figure of Fig. 8) is well computed by the present method comparing with the bottom figure [58].

C. A flow passing the RAE2822 airfoil

In this experiment, a steady-state flow around the RAE2822 airfoil is studied. The free stream Mach number is $M_\infty = 0.75$, and the angle of attack is $\alpha = 3^\circ$. A grid with 99 380 nodes and 98 406 quadrangle cells is employed. As shown in Fig. 9, the outer boundary is approximately 30 times the chord length from the airfoil. A slip adiabatic wall boundary condition is imposed for the airfoil. The subsonic inlet and outlet conditions are given on the far-field boundary and outflow boundary, respectively. The reference length is set as the chord length. The pressure coefficient C_p can be calculated by

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty u_\infty}, \tag{18}$$

where ρ_∞, u_∞, and p_∞ are the density, velocity, and pressure of the flow at the far field. The pressure contours and pressure coefficient profiles are shown in Fig. 10 and Fig. 11, respectively. The squares in Fig. 11 are the results of Jameson's central scheme through the solution of Euler equations derived by Meister [59]. Figure 11 shows that the results obtained by our method agree well with the ones in Ref. [59], and the shock wave is captured correctly.

D. Flows passing the NACA0012 airfoil

Three simulations are performed in this experiment to compute the steady-state flows passing the NACA0012 airfoil with different configurations including (1) $M_\infty = 0.63, \alpha = 2^\circ$; (2) $M_\infty = 0.85, \alpha = 1^\circ$; and (3) $M_\infty = 0.8, \alpha = 1.25^\circ$. A grid

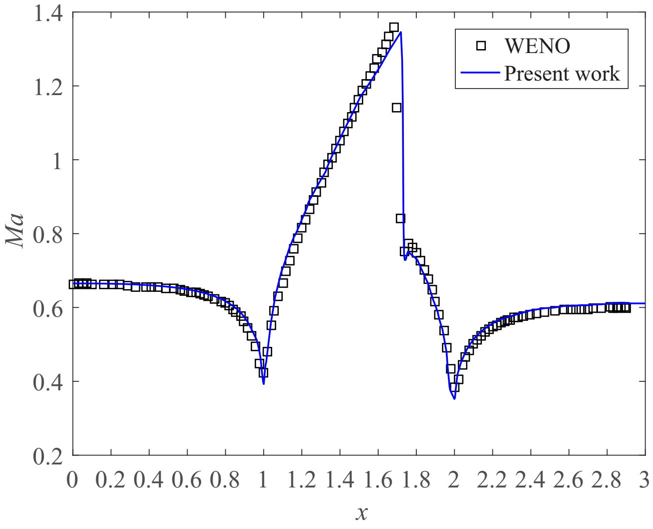
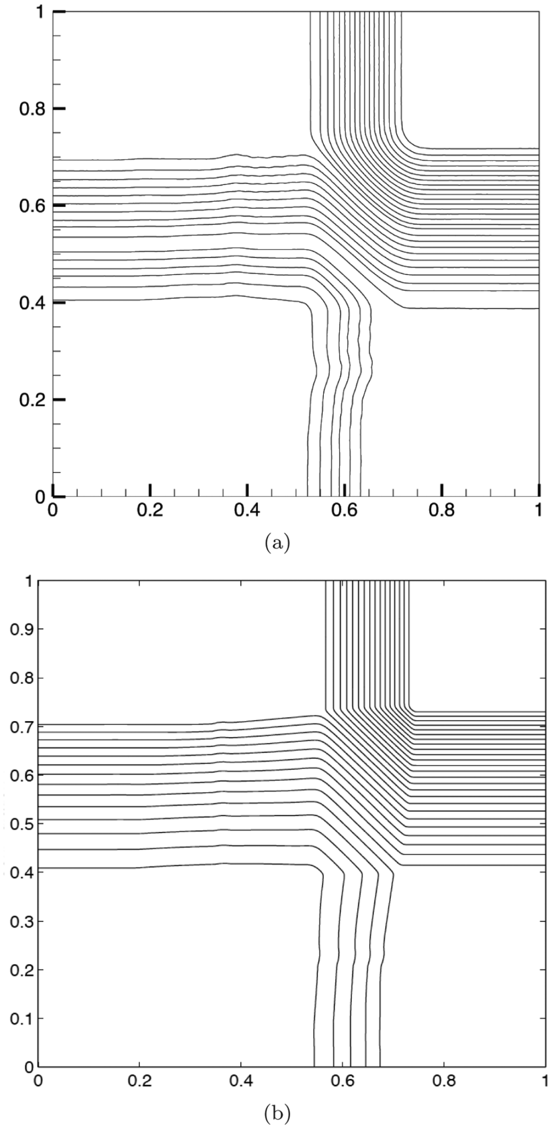


FIG. 7. Mach number along the top wall. WENO in the figure refers to the result from Ref. [57].



(a)

(b)

FIG. 8. Density contours of the 2D Riemann problem, $t = 0.2$. (a) our result, (b) result in Ref. [58].

with 104 385 nodes and 103 400 quadrangle cells is used. The boundary conditions are the same as the flow passing the RAE2822 airfoil. The pressure contours of the first case are shown in Fig. 12, and the comparison of the pressure coefficient profiles with the reference results are illustrated in Fig. 13. For the second case, the pressure contours are given in Fig. 14, and the comparison of the pressure coefficient profiles with the results computed by means of solving the Euler equations are shown in Fig. 15. Figure 16 and Fig. 17 show the pressure contours and the pressure coefficients of the third case, respectively. It can be observed that all three cases agree well with the results in Ref. [60].

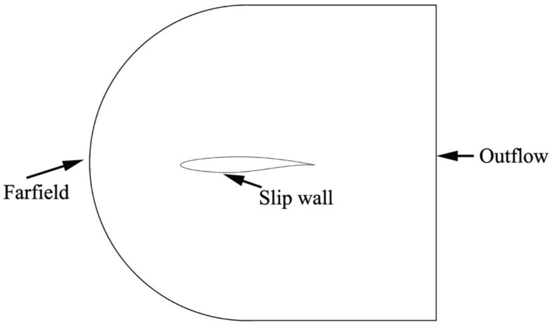


FIG. 9. The computational domain and boundary condition of a flow around the RAE2822 airfoil.

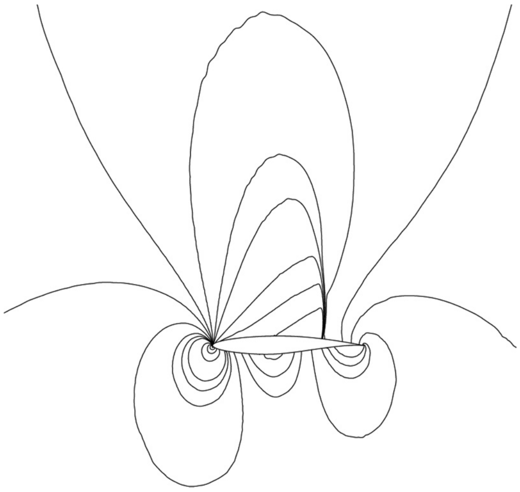


FIG. 10. Pressure contours of flow around the RAE2822 airfoil ($M_\infty = 0.75, \alpha = 3^\circ$).

E. Supersonic flows around a cylinder

Steady supersonic flows around a cylinder are computed in this experiment. Two Mach number are considered as $M_\infty = 3, 5$. A grid with 19 481 nodes and 19 200 cells is used, in which the boundary is generated by [61]

$$x = -[R_x - (R_x - 1)\xi]\cos(\theta(2\eta - 1)),$$
$$y = [R_y - (R_y - 1)\xi]\sin(\theta(2\eta - 1)),$$

where $R_x = 3$, $R_y = 6$, $\theta = \frac{5\pi}{12}$, $\xi \in [0, 1]$, and $\eta \in [0, 1]$. The slip and adiabatic boundary conditions are applied on

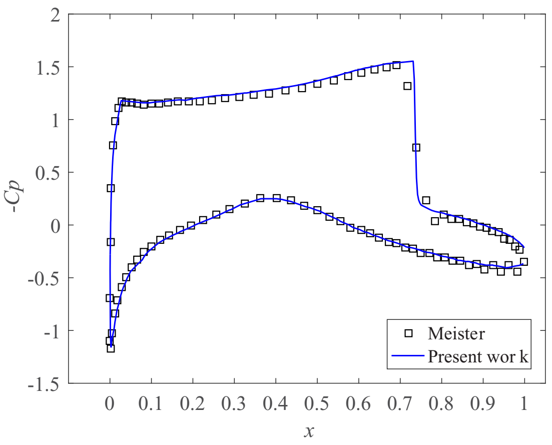


FIG. 11. C_p of the flow around the RAE2822 airfoil ($M_\infty = 0.75$, $\alpha = 3^\circ$). In the figure, Meister refers to the result published in Ref. [59].

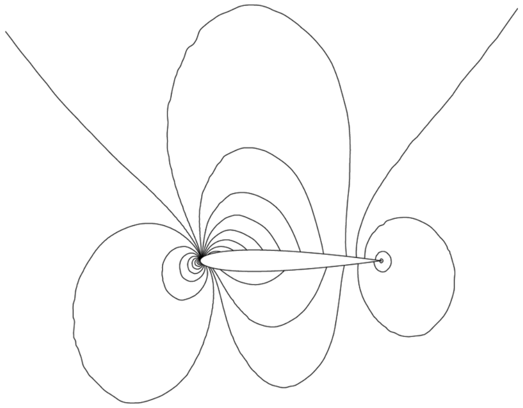


FIG. 12. Pressure contours of the flow around the NACA0012 airfoil ($M_\infty = 0.63$, $\alpha = 2^\circ$).

the cylinder wall ($\eta = 1$), at the supersonic inlet we assume $\eta = 0$, and the supersonic outlet boundary condition is applied at $\xi = 0$ and $\xi = 1$. The flows are initialized based on the free-stream state. The pressure contours for two different Mach numbers are plotted in Fig. 18, and the pressure profiles along the central line are shown in Fig. 19 and compared with the results obtained from a sixth-order compact-Roe scheme with an adaptive filter by Visbal and Gaitonde [62]. The shock wave is captured without spurious oscillations, and our pressure profiles agree well with the reference data.

F. An explosion in a 3D box

In this experiment, an unsteady explosion in a 3D enclosed box is considered. As shown in Fig. 20, the computational domain is [0, 1] × [0, 1] × [0, 1]. The initial variables are set as

$$(\rho, u, v, w, p) = \begin{cases} (5, 0, 0, 0, 5) & \text{if } \|\mathbf{x} - 0.4\|_2 \leqslant 0.3, \\ (1, 0, 0, 0, 1) & \text{others.} \end{cases}$$

We assume all six boundaries are reflective. A spherical shock wave expands in the enclosed box and interacts in a complex manner as time goes. A grid with 155 873 nodes and 896 277 tetrahedrons is employed in our test. Some density isosurfaces with $\rho = 1.8$ at $t = 0.25, 0.375, 0.5$ are shown in Fig. 21 and compared with the results reported in Ref. [63] using a block-structured adaptive mesh refinement method. The density contours at $z = 0.4$ and $t = 0.5$ are illustrated in Fig. 22. Good agreement is observed in the comparison of the density contours with the results in Ref. [63].

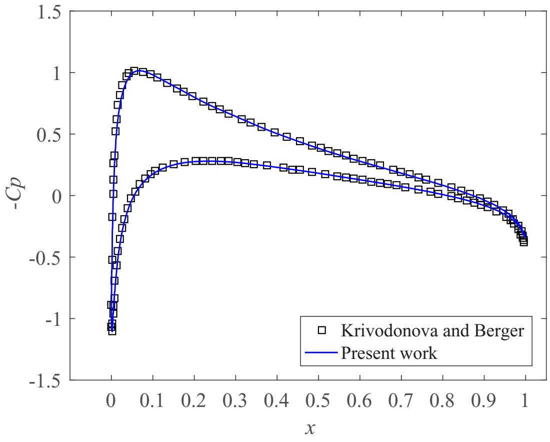


FIG. 13. C_p of the flow around the NACA0012 airfoil ($M_\infty = 0.63$, $\alpha = 2^\circ$).

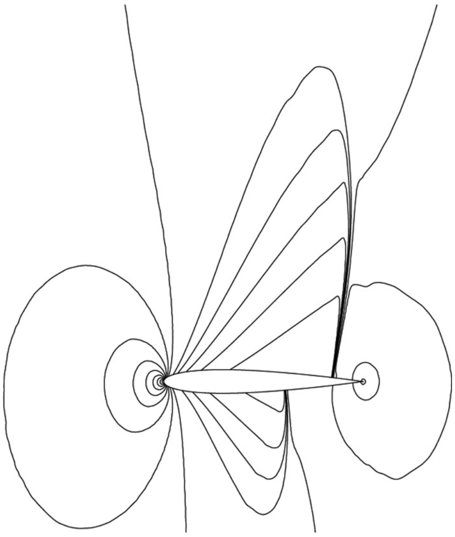


FIG. 14. Pressure contours of the flow around the NACA0012 airfoil ($M_\infty = 0.85$, $\alpha = 1^\circ$).

G. A flow around the ONERA M6 wing

To further understand the proposed method for 3D problems with a complex geometry, we consider a steady transonic flow around the ONERA M6 wing, in which same boundary conditions as the flow passing the RAE2822 airfoil are imposed. The free stream Mach number is set at 0.8395, and the angle of attack is taken as 3.06°. The geometry is provided in Ref. [64] and a grid with 144 117 nodes and 750 029 tetrahedrons is used for the simulation. Figure 23 shows the pressure

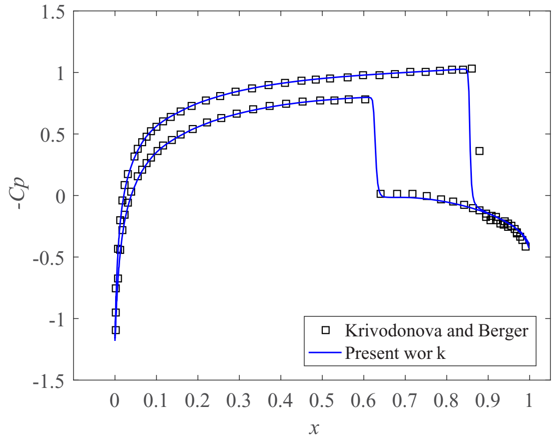


FIG. 15. C_p of the flow around the NACA0012 airfoil ($M_\infty = 0.85$, $\alpha = 1^\circ$).

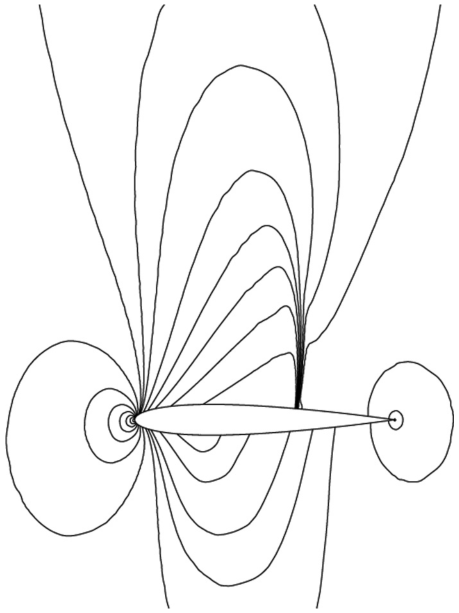


FIG. 16. Pressure contours of the flow around the NACA0012 airfoil ($M_\infty = 0.8, \alpha = 1.25^\circ$).

coefficients, which indicate that the numerical results agree well with the experimental measurements in Ref. [65].

VI. SCALABILITY TEST ON A SUPERCOMPUTER

In this section, we investigate the parallel scalability of the proposed method on a computer using up to 4800 processor cores. We consider two representative problems; one in 2D and one in 3D.

For the 2D flow around the NACA0012 airfoil, a grid with 137 523 200 quadrangular cells is used, while a grid with 55,362,624 tetrahedron cells is used for the 3D flow around the ONERA M6 wing. Table I shows the compute times for the first 100 time steps. Figure 24 and Fig. 25 illustrate the speedup and efficiency of both cases. In the figures, "Speedup," "Efficiency," and "Cores" refer to the speedup, parallel efficiency, and number of processor cores,

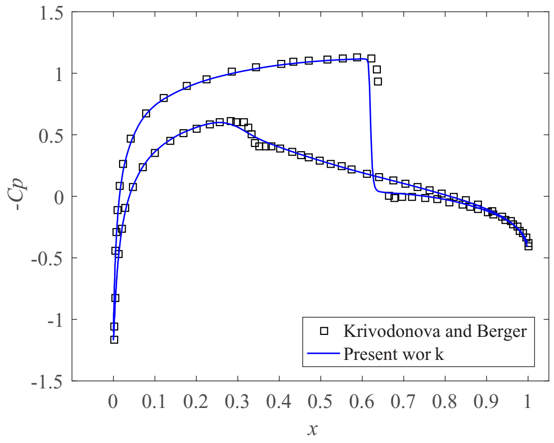


FIG. 17. C_p of the flow around the NACA0012 airfoil ($M_\infty = 0.8, \alpha = 1.25^\circ$).

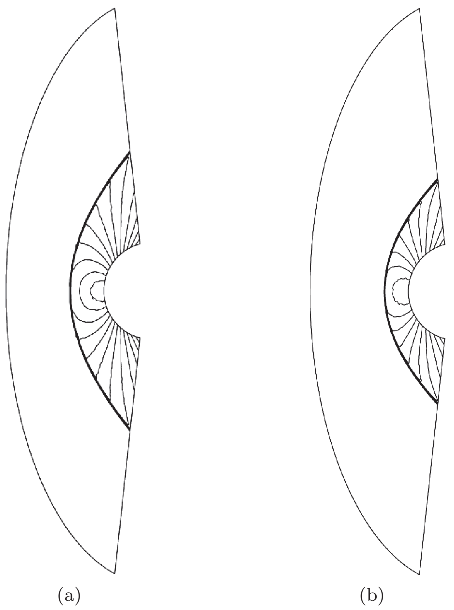


FIG. 18. Pressure contours of supersonic flow past a cylinder. (a) $M_\infty = 3$ and (b) $M_\infty = 5$.

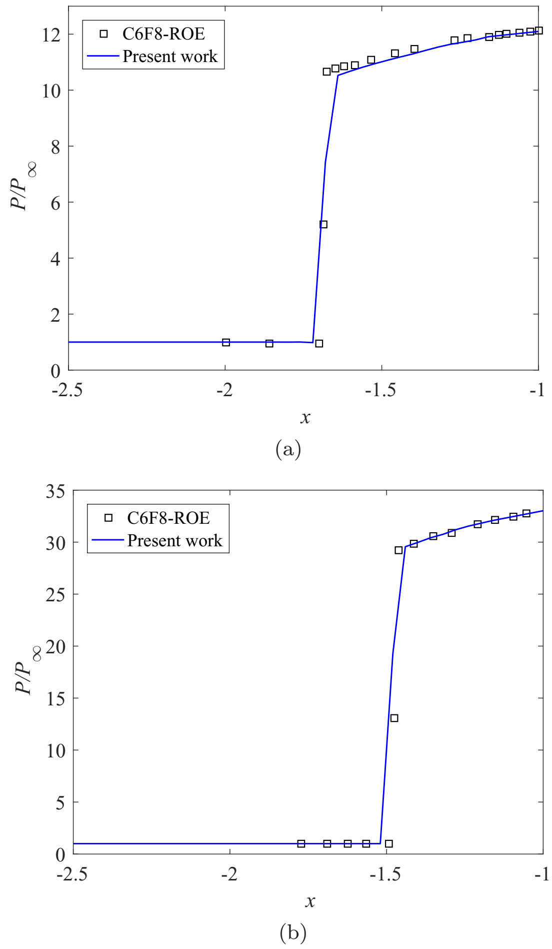


FIG. 19. Pressure coefficient profile along the central line. (a) $M_\infty = 3$ and (b) $M_\infty = 5$. C6F8-ROE refers to the results in Visbal and Gaitonde [62].

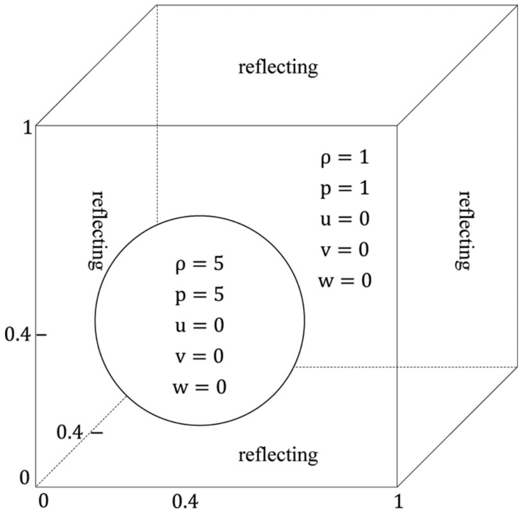


FIG. 20. Configuration of the explosion in a 3D box.

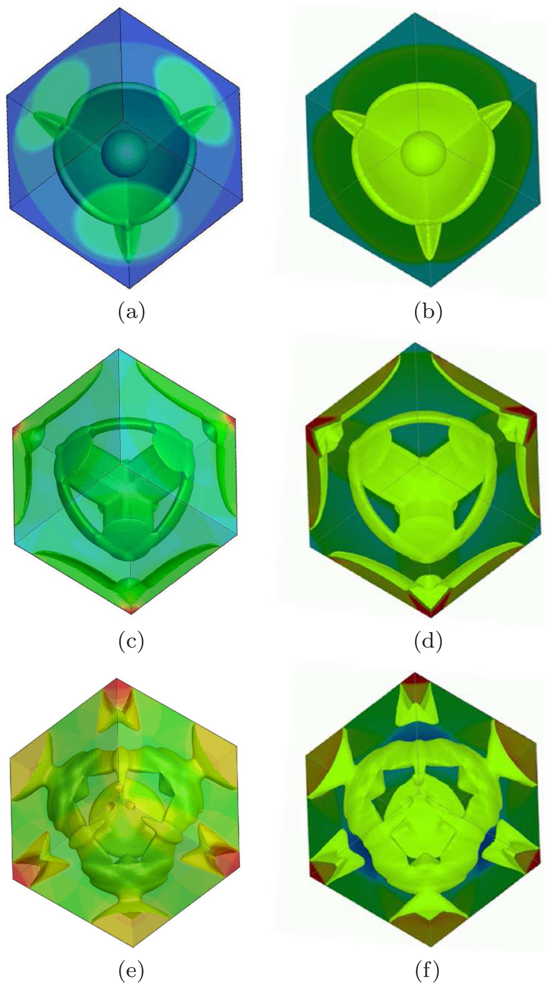


FIG. 21. Density isosurfaces of the explosion in a 3D box at (a) $t = 0.25$, (c) $t = 0.375$, and (e) $t = 0.5$. Panels (b), (d), and (f) are the corresponding results in Ref. [63].

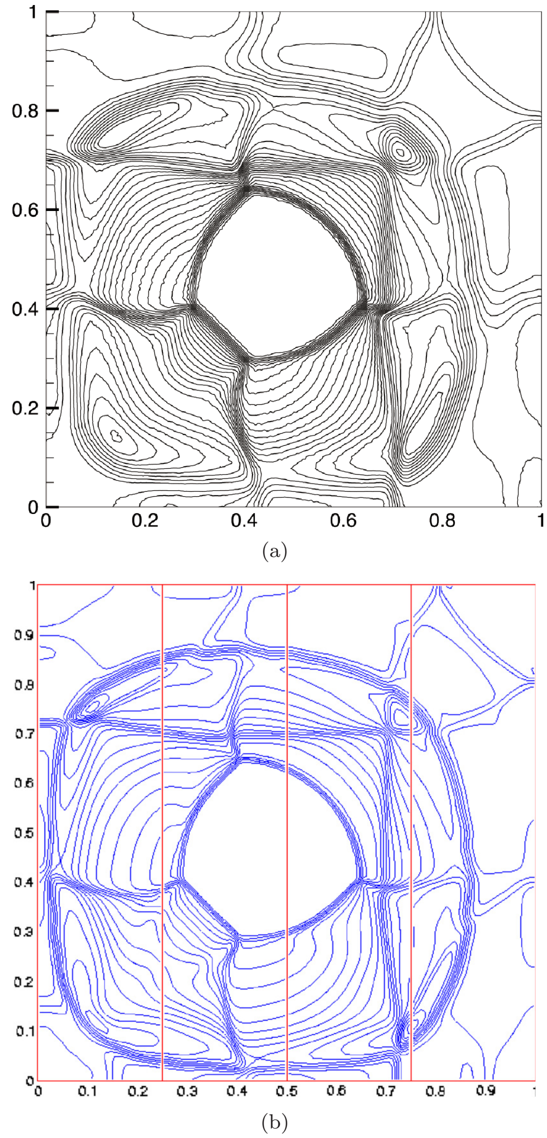
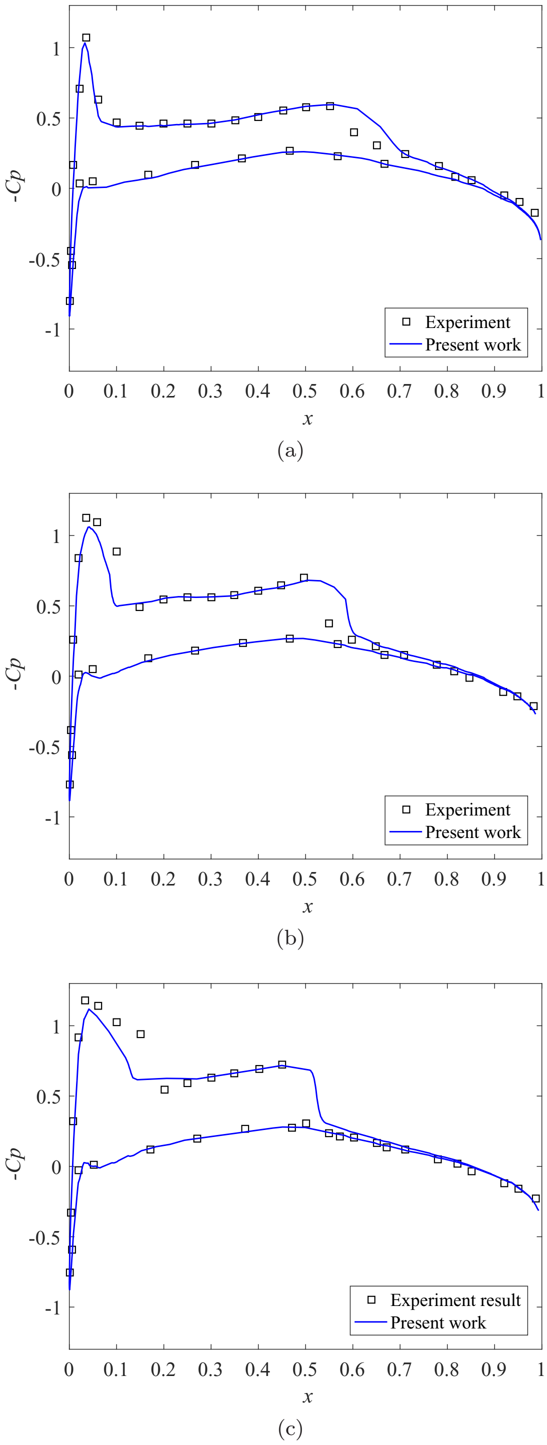


FIG. 22. Density contours of the explosion in a 3D box at $z = 0.4$ and $t = 0.5$. (a) Our result and (b) result in Ref. [63].

respectively. "Ideal" and "Present" refer to the ideal speedup and the speedup of the proposed method, respectively. Efficiencies of 95.31% and 94.56% are achieved for the 2D case and 3D case with up to 4800 processor cores, respectively.

VII. CONCLUDING REMARKS

In this work, a parallel FVDBM on 2D and 3D unstructured grids was introduced and studied for high speed inviscid compressible flows. The proposed method was carefully investigated for seven benchmark problems in both 2D and 3D, and the results show that the method can accurately capture the shocks in all cases including steady state and unsteady problems of subsonic, transonic, and supersonic flows. A close to linear scalability is observed on a parallel compter with thousands of processor cores, which implies that the method has the potential for high-fidelity simulations of compressible flows in complex computational domains.



(a)

(b)

(c)

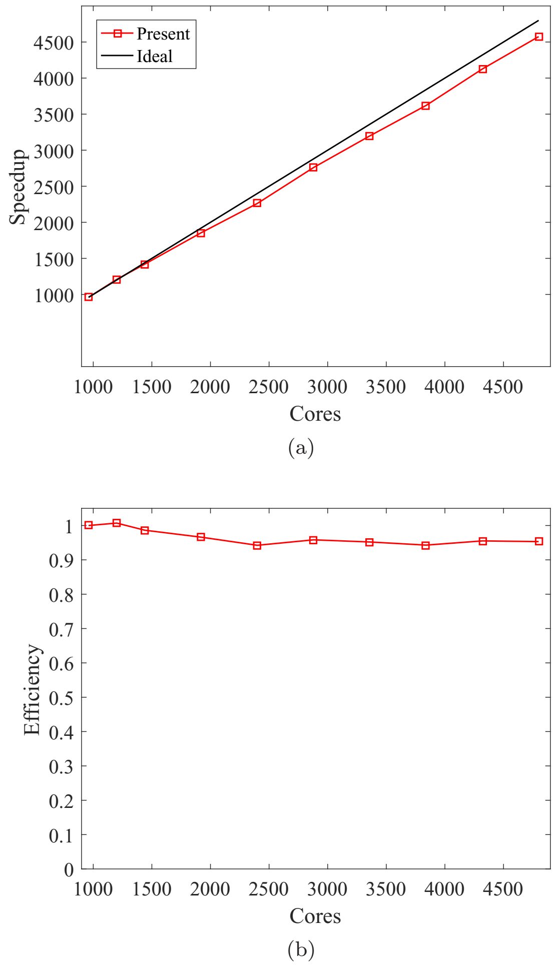
FIG. 23. Comparison of the pressure coefficient distribution at different sections on the OMERA M6 wing, η refers to the spanwise location, and the x directions are nondimensionalized by local chord. (a) $\eta = 0.2$, (b) $\eta = 0.44$, and (c) $\eta = 0.65$.

TABLE I. The strong scalability results for the compressible flows. “Cores,” “Time (2D),” and “Time (3D)” refer to the number of processor cores, the compute time in second of the first 100 time steps for the 2D case and that for the 3D case, respectively.

Cores	Time (2D)	Time (3D)
1200	118.49	119.26
2400	65.98	59.40
4800	32.62	31.15

FVDBM for inviscid compressible flows consists of three parts: a discrete lattice velocity model, equilibrium distribution functions, and coupled DDF evolution equations. In this work, D2Q13 and D3Q25 lattice velocity model are adopted for 2D and 3D problems, respectively. The corresponding equilibrium distribution functions are constructed based on circular and spherical functions. Based on our presented coupled DDF evolution equations, other discrete lattice velocity models and equilibrium distribution functions (such as Hermite polynomials [35], direct matching of moments [38] and shifted stencils [39,40]) for inviscid and viscid compressible flows are also be able to incorporate with our solver.

In (1) and (2), the nonlocality (the advective term) is linear and the nonlinearity (the collision term) is local. However, the transport term $\mathbf{\nabla} \cdot (\rho \mathbf{u}\mathbf{u})$ in compressible Euler equations is nonlocal and nonlinear at the same time, which



(a)

(b)

FIG. 24. The speedup (a) and parallel efficiency (b) of the proposed method for the 2D NACA0012 case. The grid has 137 523 200 quadrangular cells in this test.

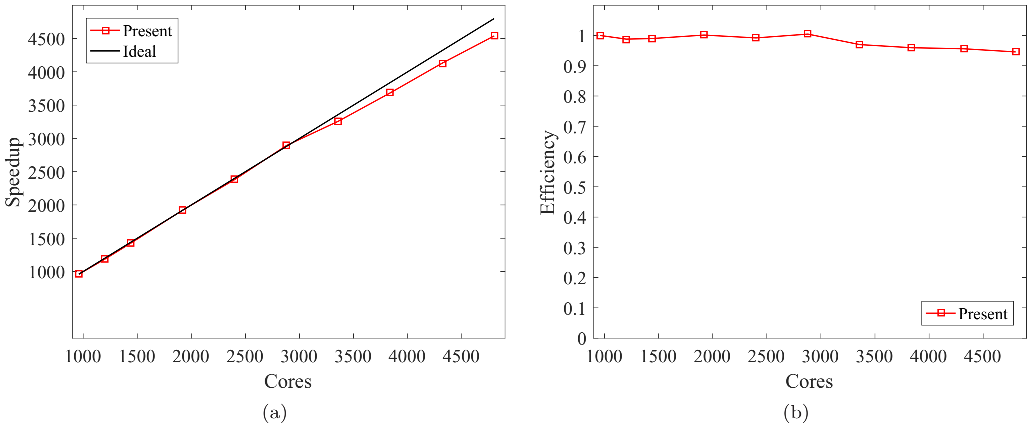


FIG. 25. The speedup (a) and parallel efficiency (b) of the proposed method for the 3D ONERA M6 case. The grid has 55 362 624 tetrahedron cells in this test.

becomes demanding in terms of numerical stability. However, compared to the classical FVM solvers for the compressible Euler equations, FVDBM involves more independent variables, which requires more memory and communication. Whether one method is better than the other is still an open question.

ACKNOWLEDGMENTS

This work was partially supported by the National Key R&D Program of China 2016YFB0200601, Shenzhen Grants No. ZDSYS201703031711426 and No. JCYJ20170818153840322, and NSF of China Grant No. 12071461.

[1] U. Frisch, B. Hasslacher, and Y. Pomeau, Phys. Rev. Lett. **56**, 1505 (1986).
[2] L.-S. Luo, W. Liao, X. Chen, Y. Peng, and W. Zhang, Phys. Rev. E **83**, 056710 (2011).
[3] S. Busuioc and V. E. Ambrus, Phys. Rev. E **99**, 033304 (2019).
[4] A. Ghaffarian and K. Hejranfar, Meccanica **54**, 1561 (2019).
[5] Y. Ma, Y. Wang, and M. Xie, Nucl. Sci. Eng. **193**, 1219 (2019).
[6] F. Loffler, Z. Cao, S. R. Brandt, and Z. Du, J. Comput. Sci. **16**, 79 (2016).
[7] D. V. Patil and K. N. Lakshmisha, J. Comput. Phys. **228**, 5262 (2009).
[8] G. Di Ilio, S. Ubertini, S. Succi, and G. Falcucci, J. Sci. Comput. **83**, 56 (2020).
[9] L. Xu and R. Chen, Int. J. Heat Mass Transf. **160**, 120156 (2020).
[10] G. Di Ilio, D. Chiappini, S. Ubertini, G. Bella, and S. Succi, Phys. Rev. E **95**, 013309 (2017).
[11] G. Di Ilio, D. Chiappini, S. Ubertini, G. Bella, and S. Succi, Comput. Fluids **166**, 200 (2018).
[12] M. K. Misztal, A. Hernandez-Garcia, R. Matin, H. O. Sorensen, and J. Mathiesen, J. Comput. Phys. **297**, 316 (2015).
[13] Y. Gan, A. Xu, G. Zhang, Y. Zhang, and S. Succi, Phys. Rev. E **97**, 053312 (2018).
[14] A. Xu, G. Zhang, Y. D. Zhang, P. Wang, and Y. J. Ying, Front. Phys. **13**, 135102 (2018).
[15] W. Li and L.-S. Luo, Commun. Comput. Phys. **20**, 301 (2016).
[16] W. Li and L.-S. Luo, J. Comput. Phys. **327**, 503 (2016).
[17] L. Chen and L. Schaefer, Comput. Math. Appl. **75**, 3105 (2018).
[18] L. Chen, S. Succi, X. Cai, and L. Schaefer, Phys. Rev. E **101**, 063301 (2020).
[19] F. J. Alexander, S. Chen, and J. D. Sterling, Phys. Rev. E **47**, R2249(R) (1993).
[20] Y. Qian, J. Sci. Comput. **8**, 231 (1993).
[21] Y. Chen, H. Ohashi, and M. Akiyama, Phys. Rev. E **50**, 2776 (1994).
[22] M. Watari and M. Tsutahara, Phys. Rev. E **67**, 036306 (2003).
[23] Y. Guangwu, C. Yaosong, and Hu Shouxin, Phys. Rev. E **59**, 454 (1999).
[24] W. Shi, W. Shyy, and R. Mei, Numer. Heat Tranf. B-Fundam. **40**, 1 (2001).
[25] C. B. Laney, *Computational Gasdynamics* (Cambridge University Press, Cambridge, UK, 1998).
[26] C. Sun, J. Comput. Phys. **161**, 70 (2000).
[27] C. Sun and A. T. Hsu, Phys. Rev. E **68**, 016303 (2003).
[28] X. He, S. Chen, and G. D. Doolen, J. Comput. Phys. **146**, 282 (1998).
[29] Z. Guo, C. Zheng, B. Shi, and T. S. Zhao, Phys. Rev. E **75**, 036704 (2007).
[30] Q. Li, Y. L. He, Y. Wang, and W. Q. Tao, Phys. Rev. E **76**, 056705 (2007).
[31] K. Qu, C. Shu, and Y. T. Chew, Phys. Rev. E **75**, 036706 (2007).
[32] K. Li and C. Zhong, Int. J. Numer. Methods Fluids **77**, 334 (2015).
[33] Q. Li, Y. L. He, Y. Wang, and G. H. Tang, Phys. Lett. A **373**, 2101 (2009).
[34] R.-F. Qiu, Y.-C. You, C.-X. Zhu, R.-Q. Chen, and J.-F. Zhu, Pramana **89**, 81 (2017).
[35] X. W. Shan, X. F. Yuan, and H. D. Chen, J. Fluid Mech. **550**, 413 (2006).
[36] Y. Feng, P. Sagaut, and W.-Q. Tao, Comput. Fluids **131**, 45 (2016).
[37] I. Karlin and P. Asinari, Physica A **389**, 1530 (2010).

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

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

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

[41] J. Huang, C. Yang, and X.-C. Cai, SIAM J. Sci. Comput. **37**, S291 (2015).

[42] J. Huang, C. Yang, and X.-C. Cai, SIAM J. Sci. Comput. **38**, A1701 (2016).

[43] S. Schmieschek, L. Shamardin, S. Frijters, T. Kruger, U. D. Schiller, J. Harting, and P. V. Coveney, Comput. Phys. Commun. **217**, 149 (2017).

[44] P. Hong, L. Huang, L. Lin, and C. Lin, Comput. Fluids **110**, 1 (2015).

[45] Y. Yu, K. Li, Y. Wang, and T. Deng, Comput. Fluids **110**, 114 (2015).

[46] K. Qu, *Development of Lattice Boltzmann Method for Compressible Flows*, Ph.D. thesis, National University of Singapore (2008).

[47] J. Blazek, *Computational Fluid Dynamics: Principles and Applications*, 3rd ed. (Elevier, Amsterdam, 2015).

[48] F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab* (Springer, Berlin, 2016).

[49] P. Roe, J. Comput. Phys. **43**, 357 (1981).

[50] M. Stiebler, J. Tolke, and M. Krafczyk, Comput. Fluids **60**, 814 (2006).

[51] V. Venkatakrishnan, *31st AIAA Aerospace Sciences Meeting & Exhibit*, Reno, NV, USA, AIAA (1993) pp. 93–0880.

[52] V. Venkatakrishnan, J. Comput. Phys. **118**, 120 (1995).

[53] Z. Guo, C. Zheng, and B. Shi, Chin. Phys. **11**, 366 (2002).

[54] G. Karypis and K. Schloegel, *ParMETIS: Parallel graph partitioning and sparse matrix ordering library version 4.0*, University of Minnesota (2013).

[55] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, PETSc Users Manual, Technical Report ANL-95/11-Revision 3.13, Argonne National Laboratory (2020).

[56] K. Gustavsson, Inviscid flow over a bump, http://www.nada.kth.se/katarina/NGSSC.html.

[57] J. Furst, Flow Turbul. Combust. **76**, 331 (2006).

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

[59] K. Qu, C. Shu, and Y. T. Chew, Adv. Appl. Math. Mech. **2**, 573 (2010).

[60] L. Krivodonova and M. Berger, J. Comput. Phys. **211**, 492 (2006).

[61] G.-S. Jiang and C.-W. Shu, J. Comput. Phys. **126**, 202 (1996).

[62] M. R. Visbal and D. V. Gaitonde, in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, AIAA 2005-1265 (1993).

[63] Amroc3d, Explosion in a 3D box, http://amroc.sourceforge.net/examples/euler/3d/html/box3d_n.htm.

[64] J. W. Slater, ONERA M6 wing, https://www.grc.nasa.gov/www/wind/valid/m6wing/m6wing01/m6wing01.html.

[65] V. Schmitt and F. Charpin, Pressure distributions on the ONERA-M6-wing at transonic Mach numbers, Technical Report 138, Experimental Data Base for Computer Program Assessment, Report of the Fluid Dynamics Panel Working Group 04, AGARD AR (1979).