AIMS Mathematics, 8(4): 8829–8846.
DOI:10.3934/math.2023443
Received: 10 October 2022
Revised: 09 January 2023
Accepted: 12 January 2023
Published: 08 February 2023

http://www.aimspress.com/journal/Math

Research article

A discrete unified gas kinetic scheme on unstructured grids for viscid compressible flows and its parallel algorithm

Lei Xu, Zhengzheng Yan* and Rongliang Chen*

Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

* **Correspondence:** Email: zz.yan@siat.ac.cn, rl.chen@siat.ac.cn.

Abstract: In this paper, we present a discrete unified gas kinetic scheme (DUGKS) on unstructured grids for high-speed viscid compressible flows on the basis of double distribution function (the density and the total energy distribution functions) Boltzmann-BGK equations. In the DUGKS, the discrete equilibrium distribution functions are constructed based on a D2Q17 circular function. In order to accelerate the simulation, we also illustrate a corresponding parallel algorithm. The DUGKS is validated by two benchmark problems, i.e., flows around the NACA0012 airfoil and flows past a circular cylinder with the Mach numbers range from 0.5 to 2.5. Good agreements with the referenced results are observed from the numerical results. The results of parallel test indicate that the DUGKS is highly parallel scalable, in which the parallel efficiency achieves 93.88% on a supercomputer using up to 4800 processors. The proposed method can be utilized for high-resolution numerical simulation of complex and high Mach number flows.

Keywords: discrete unified gas kinetic scheme; unstructured grid; parallel algorithm; viscid compressible flow; Boltzmann equation
Mathematics Subject Classification: 26A33, 34B10, 34B15

1. Introduction

Compressible flows exist in many practical engineering, such as aerophysics, astrophysics, aeroacoustics, combustion, explosion physics, etc, which is generally illustrated by the Navier-Stokes equations (NSEs) [1–4]. In the conventional computational fluid dynamics (CFD), the discrete method, for example, finite difference (FD) [5], finite volume (FV) [6], finite element methods [7, 8] are employed to discretize the NSEs. Instead of solving compressible NSEs, the kinetic schemes, such as the lattice Boltzmann method (LBM) [9], the gas kinetic scheme (GKS) [10], the lattice Boltzmann flux solver [11], the discrete velocity or discrete ordinate methods [12] and the (discrete) unified gas kinetic scheme [13,14], are derived from the Boltzmann equation to evolve the gas distribution function

and calculate the conservative variables by the gas distribution function.

The conventional LBM is a special FD discretization of the Boltzmann equation, in which the fluid domain is discretized into lattices coinciding with the uniform Cartesian grids, and the motion of particles is described by the evolution of particle distribution functions named as "collision" and "stream". The macroscopic variables are calculated by accumulating the particle distribution functions. It has been an alternative method to simulate low-speed incompressible flows. Owing to the restriction of uniform Cartesian grid, the conventional LBM is difficult for simulating the flow problems with strong local gradients or defined on geometrically complex domains. In recent years, a few off-lattice Boltzmann methods were proposed to avoid the Cartesian grid restriction, for example, finite difference LBMs [15–17], finite element LBMs [18, 19], FV-LBMs [20–22], and so on. In the conventional LBM, the equilibrium distribution function (EDF) is obtained by expanded the Maxwellian function by Taylor-series in terms of the Mach number, which can be only used for low Mach number incompressible flows. To simulate high Mach number compressible flows, an alternative method is to replace the Maxwellian function with a simple function avoiding the taylor-series expansion. Sun et al. proposed a locally adaptive LBM, in whic a simplified EDF (δ function) is used to substitute the Maxwellian function [23]. A D2Q13L2 lattice model for inviscid high-speed compressible flows is presented by Qu et al. Instead of the Maxwellian function, a circular function is employed. The EDF is constructred by a polynomial and a FV method with second-order total variation diminishing formular is employed to solve the lattice Boltzmann equation [24]. A LBM for high-speed viscid compressible flows based on D2Q17 circular function is proposed by Qu et al. [25, 26], in which the first to fourth order constraints of the density equilibrium function in the heat flux and the energy dynamics are considered [27]. Based on previous works, we presented a FV-LBM on unstructured grids for high Mach number compressible flows [22, 28].

Recently, the discrete unified gas kinetic scheme (DUGKS), a FV-based mesoscopic CFD method, has attracted more and more attention. The numerical flux in the DUGKS is constructed by the distribution function at a half time-step, in which a numerical characteristic solution of the Boltzmann equation is used. It has been widely studied in multiscale flows [29, 30], thermal incompressible flows [31, 32], thermal compressible flows [33], moving boundaries [34], gas mixture flows [35], and nonlinear convection diffusion equations [36]. Lately, a simplified DUGKS for incompressible flows is proposed, in which a whole time-step is applied to constructed the numerical flux at the cell interface [37], which is applied in this paper. Subsequently, they devised a simplified DUGKS for compressible flows [38], in which, the D2Q13 circular distribution function is used to substitute the Maxwellian function. In this paper, the D2Q17 circular function is employed instead of D2Q13 circular function in order to consider the first to fourth order constraints of distribution function.

In this paper, in order to finish the simulation in a short time, a parallel version of the proposed DUGKS on unstructured grids is implemented. In this paper, an open source package ParMETIS is used to partition the unstructured grid into irregular subdomains [39], which ensure the minimum of communication among subdomains and promises the load balance among processor cores. The parallel algorithm implemented by PETSc (portable, extensible toolkit for scientific computation) [40] is presented, and parallel performance is investigated on a supercomputer.

The remainder of the paper is organized as follows. DUGKS for viscid compressible flows as well as the parallel algorithm is presented in Section 2. Next, numerical results and the parallel performance are given in Section 3. Finally, some conclusions are made in Section 4.

2. Parallel DUGKS for viscid compressible flows

2.1. Kinetic model equations

The Boltzmann-BGK model for viscid compressible flows includes the density and the total energy distribution function. For the density distribution function, the discrete Boltzmann-BGK equation is written as

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla f = -\frac{1}{\tau_f}\left(f - f^{eq}\right), \tag{2.1}$$

where f represents the density distribution function with velocity $\boldsymbol{\xi}$ at the position $\mathbf{x}$ and time t, μ the viscosity, p the pressure, $\tau_f = \mu/p$ the particle relaxation time, and f^{eq} the local Maxwellian function expressed as [27, 41]

$$f^{eq} = \frac{\rho}{2(2\pi RT)^{D/2}} \exp\left(-\frac{(\boldsymbol{\xi} - \mathbf{u})^2}{2RT}\right),$$

where ρ denotes the density, T the temperature, $\mathbf{u}$ the fluid velocity, R the gas constant and D the spatial dimension.

To recover the compressible Navier-Stokes equations, for the total energy distribution function, the governing equation is introduced in the following form

$$\frac{\partial h}{\partial t} + \boldsymbol{\xi} \cdot \nabla h = -\frac{1}{\tau_h}\left[h - h^{eq}\right] + \frac{1}{\tau_{hf}}(\boldsymbol{\xi} \cdot \mathbf{u})\left(f - f^{eq}\right), \tag{2.2}$$

where h represents the total EDF, τ_h the total energy relaxation time, $\tau_{hf} = \tau_f \tau_h/(\tau_f - \tau_h)$, the Prandtl number Pr is computed by τ_f/τ_h, the total energy EDF h^{eq} is written as

$$h^{eq} = \frac{\rho(\boldsymbol{\xi}^2 + (b - D)RT)}{2(2\pi RT)^{D/2}} \exp\left(-\frac{(\boldsymbol{\xi} - \mathbf{u})^2}{2RT}\right),$$

where b is associated with the specific-heat ratio γ by $\gamma = (b + 2)/b$. The macroscopic variables are the moments of the distribution function,

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho E \end{pmatrix} = \begin{pmatrix} \int f d\boldsymbol{\xi} \\ \int \boldsymbol{\xi} f d\boldsymbol{\xi} \\ \int h d\boldsymbol{\xi} \end{pmatrix},$$

where $E = \mathbf{u}^2/2 + e$ denotes the total energy, e the internal energy. f^{eq} and h^{eq} have to meet the

following velocity moment condition to recover the compressible NSEs:

$$\begin{cases} \int f^{eq} d\boldsymbol{\xi} = \rho, \\ \int f^{eq} \xi_i d\boldsymbol{\xi} = \rho u_i, \\ \int f^{eq} \xi_i \xi_j d\boldsymbol{\xi} = \rho u_i u_j + p\delta_{ij}, \\ \int f^{eq} \xi_i \xi_j \xi_k d\boldsymbol{\xi} = \rho u_i u_j u_k + p(u_k \delta_{ij} + u_j \delta_{ik} + u_i \delta_{jk}), \\ \int f^{eq} \boldsymbol{\xi}^2 \xi_i \xi_j d\boldsymbol{\xi} = \rho u^2 u_i u_j + p\left[(D+2)RT\delta_{ij} + (D+4)u_i u_j + u^2 \delta_{ij}\right], \\ \int h^{eq} d\boldsymbol{\xi} = \rho E, \\ \int h^{eq} \xi_i d\boldsymbol{\xi} = (\rho E + p)u_i, \\ \int h^{eq} \xi_i \xi_j d\boldsymbol{\xi} = (\rho E + 2p)u_i u_i + p(E + RT)\delta_{ij}, \end{cases} \tag{2.3}$$

where $p = (\gamma - 1)\rho e$ represents the pressure, δ_{ij}, δ_{ik} and δ_{jk} the Kronecker delta functions, i, j, and k the components along the x, y and z directions.

To simulate the high Mach number compressible flows, the local Maxwellian EDF f^{eq} is substituted by a circular function. All the needed statistical relations in (2.3) has to be satisfied with the circular function. A D2Q13 circular function is introduced into DUGKS for high Mach number compressible flows by Zhong et al. [38]. In fact, f^{eq} deduced from the D2Q13 circular function satisfies all the 7 equations in (2.3) except the 5th one. Note that A D2Q17 circular function is also proposed by Qu et al. in [26], which fulfills all the equations in (2.3). The D2Q17 circular function can be written as

$$f^{eq} = \begin{cases} \dfrac{\rho}{4\pi c} & \text{if } \|\boldsymbol{\xi} - \mathbf{u}\| = c, \\ \dfrac{\rho}{2} & \text{if } \boldsymbol{\xi} = \mathbf{u}, \end{cases} \tag{2.4}$$

where $c = 2\sqrt{RT}$ denotes the radius of the circle. To discrete the continuous EDF, a fourth-order 17-node Lagrangian interpolating polynomial called assigning function is constructed, which has the following form

$$\begin{aligned} \phi_p(x, y) \;=\; & a_{p,0} + a_{p,1}x + a_{p,2}y + a_{p,3}x^2 + a_{p,4}xy + a_{p,5}y^2 + a_{p,6}x^3 + a_{p,7}x^2 y + a_{p,8}xy^2 \\ & + a_{p,9}y^3 + a_{p,10}x^4 + a_{p,11}x^3 y + a_{p,12}x^2 y^2 + a_{p,13}xy^3 + a_{p,14}y^4 + a_{p,15}x^5 + a_{p,16}y^5. \end{aligned} \tag{2.5}$$

Here $p = 0, 1, 2, \cdots, 16$. The corresponding discrete velocity shown in Figure 1 is expressed by

$$\begin{pmatrix} 0 & \frac{2}{3} & -\frac{2}{3} & -\frac{2}{3} & \frac{2}{3} & \frac{2}{3} & \frac{4}{3} & -\frac{2}{3} & -\frac{4}{3} & -\frac{2}{3} & -\frac{4}{3} & \frac{2}{3} & \frac{4}{3} & 2 & 0 & -2 & 0 \\ 0 & \frac{2}{3} & \frac{2}{3} & -\frac{2}{3} & -\frac{2}{3} & \frac{4}{3} & \frac{2}{3} & \frac{4}{3} & \frac{2}{3} & -\frac{4}{3} & -\frac{2}{3} & -\frac{4}{3} & -\frac{2}{3} & 0 & 2 & 0 & -2 \end{pmatrix}$$

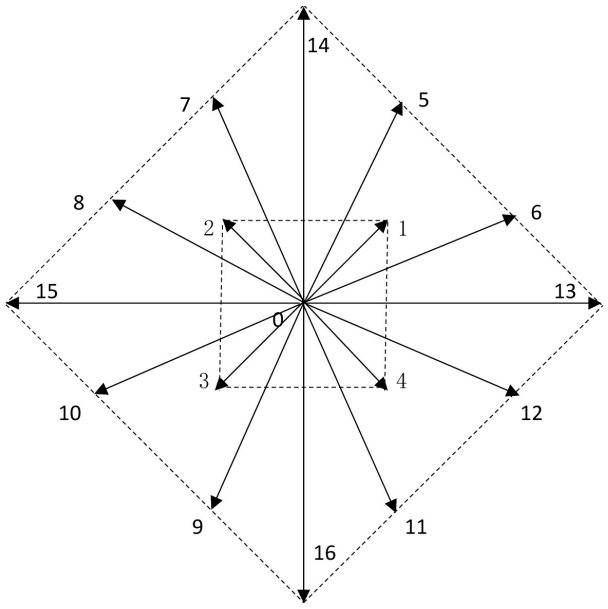


Figure 1. D2Q17 discrete velocity model.

Owing to the δ property of the Lagrangian plynomial, we get

$$\phi_p(\xi_{x,q}, \xi_{y,q}) = \delta_{p,q}, \tag{2.6}$$

where $q = 0, 1, \cdots, 16$, $\xi_{x,q}$ and $\xi_{y,q}$ represent the x and y components of the particle velocity in the qth direction. The coefficient $a_{p,q}$ can be determined by substituting the discrete velocity into (2.6). The density EDF is then calculated by

$$f_q^{eq} = \frac{\rho}{4\pi}\int_0^{2\pi} \phi_q(u_1 + c\cos\theta, u_2 + c\sin\theta)d\theta + \frac{\rho}{2}\phi_q(u_1, u_2). \tag{2.7}$$

Finally, we have

$$\begin{aligned}
f_0^{eq} &= \rho(\frac{69}{256}c^4 + \frac{69}{64}c^2u_1^2 + \frac{69}{64}c^2u_2^2 + \frac{3}{8}u_1^4 + \frac{33}{16}u_1^2u_2^2 + \frac{3}{8}u_2^4 - \frac{7}{8}c^2 - \frac{7}{4}u_1^2 - \frac{7}{4}u_2^2 + 1),\\
f_1^{eq} &= \rho(\frac{27}{40}u_1 + \frac{27}{40}u_2 + \frac{9}{16}u_1^2 - \frac{27}{64}u_1^3 + \frac{9}{16}u_2^2 - \frac{27}{64}u_2^3 - \frac{9}{64}u_1^4 + \frac{81}{1280}u_2^5 - \frac{9}{64}u_2^4 + \frac{81}{1280}u_1^5 - \frac{117}{1024}c^4\\
&\quad + \frac{9}{32}c^2 - \frac{63}{64}u_1^2u_2^2 - \frac{99}{256}c^2u_2 - \frac{27}{64}u_1u_2^3 - \frac{117}{256}c^2u_2^2 - \frac{99}{256}c^2 * u_1 + \frac{243}{4096}c^4u_2 + \frac{81}{512}c^2u_1^3\\
&\quad + \frac{81}{512}c^2u_2^3 - \frac{117}{256}c^2u_1^2 - \frac{27}{64}u_1^3u_2 - \frac{9}{32}u_1^2u_2 - \frac{9}{32}u_1u_2^2 + \frac{15}{16}u_1u_2 + \frac{243}{4096}c^4u_1 - \frac{81}{128}c^2u_1u_2),\\
f_5^{eq} &= \rho(\frac{81}{512}c^2u_1u_2 + \frac{45}{256}c^2u_1^2 + \frac{9}{512}c^2u_2^2 + \frac{45}{1024}c^2u_1 + \frac{27}{128}c^2u_2 - \frac{243}{4096}c^4u_2 + \frac{81}{2048}c^2u_1^3\\
&\quad - \frac{81}{512}c^2u_2^3 - \frac{3}{32}u_1u_2 + \frac{9}{32}u_1u_2^2 + \frac{9}{32}u_1^2 * u_2^2 + \frac{27}{128}u_1u_2^3 + \frac{243}{16384}c^4u_1 - \frac{9}{256}c^2 - \frac{9}{80}u_1 - \frac{9}{256}u_2^4\\
&\quad - \frac{9}{80}u_2 - \frac{9}{32}u_1^2 + \frac{9}{64}u_2^2 - \frac{9}{256}u_1^3 + \frac{81}{5120}u_1^5 + \frac{9}{32}u_2^3 + \frac{9}{128}u_1^4 + \frac{99}{4096}c^4 - \frac{81}{1280}u_2^5),\\
f_{13}^{eq} &= \rho(\frac{243}{8192}c^4u_1 + \frac{81}{1024}c^2u_1^3 + \frac{81}{2560}u_1^5 - \frac{3}{2048}c^4 + \frac{3}{64}c^2u_1^2 - \frac{15}{256}c^2u_2^2 + \frac{3}{64}u_1^4 - \frac{3}{32}u_1^2u_2^2\\
&\quad - \frac{3}{128}u_2^4 - \frac{27}{512}c^2u_1 - \frac{9}{128}u_1^3 + \frac{1}{128}c^2 - \frac{1}{16}u_1^2 + \frac{3}{32}u_2^2 + \frac{1}{40}u_1),
\end{aligned}$$

$$f_2^{eq}(u_1,u_2)=f_1^{eq}(-u_1,u_2),\quad f_3^{eq}(u_1,u_2)=f_1^{eq}(-u_1,-u_2),\quad f_4^{eq}(u_1,u_2)=f_1^{eq}(u_1,-u_2),$$
$$f_6^{eq}(u_1,u_2)=f_5^{eq}(u_2,u_1),\quad f_7^{eq}(u_1,u_2)=f_5^{eq}(u_2,-u_1),\quad f_8^{eq}(u_1,u_2)=f_5^{eq}(-u_1,u_2),$$
$$f_9^{eq}(u_1,u_2)=f_5^{eq}(-u_1,-u_2),\quad f_{10}^{eq}(u_1,u_2)=f_5^{eq}(-u_2,-u_1),\quad f_{11}^{eq}(u_1,u_2)=f_5^{eq}(-u_2,u_1),$$
$$f_{12}^{eq}(u_1,u_2)=f_5^{eq}(u_1,-u_2),\quad f_{14}^{eq}(u_1,u_2)=f_{13}^{eq}(u_2,u_1),\quad f_{15}^{eq}(u_1,u_2)=f_{13}^{eq}(-u_1,u_2),$$
$$f_{16}^{eq}(u_1,u_2)=f_{13}^{eq}(-u_2,u_1).$$

The total energy EDF h_q^{eq} can be computed by

$$h_q^{eq}=\left[(E+(\boldsymbol{\xi}_q-\mathbf{u})\cdot\mathbf{u}\right]f_q^{eq}+\varpi_q\frac{p}{\tilde{c}^2}RT,$$

where T_c denotes the characteristic temperature, $\tilde{c}=\sqrt{RT_c}$. $\varpi_0=0$, $\varpi_{1-4}=-17/56$, $\varpi_{5-12}=-1/8$, and $\varpi_{13-16}=-3/56$.

2.2. DUGKS for the velocity and total energy distribution functions

we first discrete the Boltzmann-BGK equation for the density distribution function by DUGKS. Integrating (2.1) over a control volume (CV) V_j located at the cell center $\mathbf{x}_j$ from time t_n to t_{n+1}, we get the following equation [37]

$$f_j^{n+1}-f_j^n+\frac{\Delta t}{|V_j|}F_{j,meso}^{n+1}=-\frac{\Delta t}{\tau_f}(f_j^{n+1}-f_j^{eq,n+1}),\tag{2.8}$$

where $F_{j,meso}^{n+1}$ represents the flux of the density distribution function across the cell interface defined as

$$F_{j,meso}^{n+1}=\int_{\partial V_j}(\boldsymbol{\xi}\cdot\mathbf{n})f(\mathbf{x}_b,\boldsymbol{\xi},t_{n+1})dS,$$

∂V_j and $|V_j|$ are the surface and volume of CV V_j, f_j the cell-averaged value of the density distribution function, $\mathbf{n}$ the outward unit normal of the cell interface. (2.8) is simplified as

$$f_j^{n+1}=\left(f_j^n-\frac{\Delta t}{|V_j|}F_{j,meso}^{n+1}+\frac{\Delta t}{\tau_f}f_j^{eq,n+1}\right)/\left(1+\frac{\Delta t}{\tau}\right).\tag{2.9}$$

Note that $F_{j,meso}^{n+1}$ is implicit. To compute $F_{j,meso}^{n+1}$, f at the cell interface $\mathbf{x}_b$ and time t^{n+1} are needed. We discrete and integrate the kinetic equations (2.1) along the particle velocity characteristic lines in one time step Δt, and employ the trapezoidal rule, we obtain

$$\bar{f}(\mathbf{x}_b,\boldsymbol{\xi},t_n+\Delta t)=\bar{f}^{+}(\mathbf{x}_b-\boldsymbol{\xi}\Delta t,\boldsymbol{\xi},t_n),\tag{2.10}$$

where $\bar{f}^{+}=f+\frac{\Delta t}{2}\Omega_f$, $\bar{f}=f-\frac{\Delta t}{2}\Omega_f$, $\Omega_f=-\frac{1}{\tau_f}(f-f^{eq})$. $\bar{f}^{+}(\mathbf{x}_b-\boldsymbol{\xi}\Delta t,\boldsymbol{\xi},t_n)$ is evaluated by

$$\bar{f}^{+}(\mathbf{x}_b-\boldsymbol{\xi}\Delta t,\boldsymbol{\xi},t_n)=\begin{cases}\bar{f}^{+}(\mathbf{x}_j,\boldsymbol{\xi},t_n)+(\mathbf{x}_b-\boldsymbol{\xi}\Delta t-\mathbf{x}_j)\cdot\mathcal{L}[\bar{f}^{+}(\mathbf{x},\boldsymbol{\xi},t),\mathbf{x}_j]\nabla\bar{f}^{+}(\mathbf{x}_j,\boldsymbol{\xi},t_n), & \text{if}\quad \boldsymbol{\xi}\cdot\mathbf{n}\geq 0,\\ \bar{f}^{+}(\mathbf{x}_i,\boldsymbol{\xi},t_n)+(\mathbf{x}_b-\boldsymbol{\xi}\Delta t-\mathbf{x}_i)\cdot\mathcal{L}[\bar{f}^{+}(\mathbf{x},\boldsymbol{\xi},t),\mathbf{x}_i]\nabla\bar{f}^{+}(\mathbf{x}_i,\boldsymbol{\xi},t_n), & \textit{otherwise},\end{cases}$$

where $\mathbf{x}_i$ is the cell center of $\mathbf{x}_j$'s neighbor cell. $\mathcal{L}$ the Venkatakrishnan limiter function [42].

DUGKS for (2.1) can also be applied similarly for (2.2). Integrating over CV V_j from t_n to t_{n+1}, one can get the following updating equation for h_j for the total energy distribution function

$$h_j^{n+1}=\left[h_j^n-\frac{\Delta t}{|V_j|}H_{j,meso}^{n+1}+\frac{\Delta t}{\tau_{hf}}(\boldsymbol{\xi}\cdot\mathbf{u})(f_j^{n+1}-f_j^{eq,n+1})\right]/(1+\frac{\Delta t}{\tau_h}),\tag{2.11}$$

where

$$H_{j,meso}^{n+1} = \int_{\partial V_j} (\boldsymbol{\xi} \cdot \mathbf{n}) h(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1}) dS$$

is also implicit. To obtain $h(\mathbf{x}_b, \boldsymbol{\xi}, t)$, we integrate (2.2) within one time step in a similar way and get

$$\bar{h}(\mathbf{x}_b, \boldsymbol{\xi}, t_n + \Delta t) = \bar{h}^+(\mathbf{x}_b - \boldsymbol{\xi}\Delta t, \boldsymbol{\xi}, t_n), \tag{2.12}$$

where $\bar{h} = f - \frac{\Delta t}{2}\Omega_h$, $\bar{h}^+ = h + \frac{\Delta t}{2}\Omega_h$, $\Omega_h = -\frac{1}{\tau_h}(h - h^{eq}) + \frac{1}{\tau_{hf}}(\boldsymbol{\xi} \cdot \mathbf{u})(f - f^{eq})$. $\bar{h}^+(\mathbf{x}_b - \boldsymbol{\xi}\Delta t, \boldsymbol{\xi}, t_n)$ is also calculated by

$$\bar{h}^+(\mathbf{x}_b - \boldsymbol{\xi}\Delta t, \boldsymbol{\xi}, t_n) = \begin{cases} \bar{h}^+(\mathbf{x}_j, \boldsymbol{\xi}, t_n) + (\mathbf{x}_b - \boldsymbol{\xi}\Delta t - \mathbf{x}_j) \cdot \mathcal{L}[\bar{h}^+(\mathbf{x}, \boldsymbol{\xi}, t), \mathbf{x}_j] \nabla \bar{h}^+(\mathbf{x}_j, \boldsymbol{\xi}, t_n), & \text{if} \quad \boldsymbol{\xi} \cdot \mathbf{n} \geq 0, \\ \bar{h}^+(\mathbf{x}_i, \boldsymbol{\xi}, t_n) + (\mathbf{x}_b - \boldsymbol{\xi}\Delta t - \mathbf{x}_i) \cdot \mathcal{L}[\bar{h}^+(\mathbf{x}, \boldsymbol{\xi}, t), \mathbf{x}_i] \nabla \bar{h}^+(\mathbf{x}_i, \boldsymbol{\xi}, t_n), & \textit{otherwise}. \end{cases}$$

Next, the macroscopic variables at $\mathbf{x}_b$ are calculated with $\bar{f}(\mathbf{x}_b, \boldsymbol{\xi}, t_n + \Delta t)$ and $\bar{h}(\mathbf{x}_b, \boldsymbol{\xi}, t_n + \Delta t)$ in the following form

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \end{pmatrix} = \begin{pmatrix} \int \bar{f} d\boldsymbol{\xi} \\ \int \boldsymbol{\xi} \bar{f} d\boldsymbol{\xi} \\ \int \bar{h} d\boldsymbol{\xi} \end{pmatrix}.$$

Thus, we can obtain the EDFs at the cell interface $\mathbf{x}_b$ based on the computed macroscopic variables. The calculation of $F_{j,meso}^{n+1}$ and $H_{j,meso}^{n+1}$ needs $f(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1})$ and $h(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1})$, which are got from $\bar{f}(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1})$ and $\bar{h}_b(\mathbf{x}, \boldsymbol{\xi}, t_{n+1})$ by

$$f(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1}) = \frac{2\tau_f}{2\tau_f + \Delta t} \bar{f}(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1}) + \frac{\Delta t}{2\tau_f + \Delta t} f^{eq}(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1}),$$

$$\begin{aligned} h(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1}) &= \frac{2\tau_h}{2\tau_h + \Delta t} \bar{h}(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1}) + \frac{\Delta t}{2\tau_h + \Delta t} h^{eq}(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1}) \\ &\quad + \frac{\tau_h \Delta t}{\tau_{hf}(2\tau_h + \Delta t)} (\boldsymbol{\xi} \cdot \mathbf{u}) \left[f(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1}) - f^{eq}(\mathbf{x}_b, \boldsymbol{\xi}, t_{n+1}) \right]. \end{aligned}$$

In (2.8) and (2.11), $f_j^{eq,n+1}$ and $h_j^{eq,n+1}$ are unknown. To solve this, the macroscopic variables at the cell center V_j has to be computed. The macroscopic variables can be deduced from the moments of the Boltzmann equation, which can be written as

$$\mathbf{W}_j^{n+1} = \mathbf{W}_j^n - \frac{\Delta t}{|V_j|} \mathbf{F}_{j,macro},$$

where

$$\mathbf{F}_{j,macro} = \begin{cases} \oint_{\partial V_j} (\boldsymbol{\xi} \cdot \mathbf{n}) f(\mathbf{x}_b, \boldsymbol{\xi}, t_n) d\boldsymbol{\xi} dS, \\ \oint_{\partial V_j} (\boldsymbol{\xi} \cdot \mathbf{n}) f(\mathbf{x}_b, \boldsymbol{\xi}, t_n) \boldsymbol{\xi} d\boldsymbol{\xi} dS, \\ \oint_{\partial V_j} (\boldsymbol{\xi} \cdot \mathbf{n}) h(\mathbf{x}_b, \boldsymbol{\xi}, t_n) d\boldsymbol{\xi} dS. \end{cases}$$

Now, the update of f and h can be executed in accordance with (2.8) and (2.11).

The CFL condition determines the time step size Δt in the following form

$$\Delta t = \eta \Delta x_{min}/C,$$

where C is the maximal discrete velocity, Δx_{min} the minimum grid spacing, and η the CFL number. The treatment of the boundary condition is implemented by the ghost cell method and the non-equilibrium extrapolation scheme [43]. More details can be found in [22].

2.3. Parallel implementation of DUGKS

To imrpove the computational time, parallel algorithm is required. To do the parallel computing, the first thing is to decompose the unstructured grid into irregular subdomains. In our work, we partition the unstructured grid into irregular subdomains by an open source package ParMETIS [39], which ensures the minimum of the parallel communication among subdomains and promises the load balance among processor cores. An example of an unstructured grid and its partition into 16 subdomains using ParMETIS is shown in Figure 2. Each subdomain is distributed to one processor core after decomposition, and the calculations on subdomains are done concurrently. In the DUGKS. most operations are local. Only the density distribution functions and total energy distribution functions of the outmost cells in a subdomain are required by neighboring processor cores, where the parallel communication occurs. We implement the parallel algorithm based on PETSc from Argonne National Laboratory [40]. Further reading about the parallel strategy for the DUGKS can be found [20].

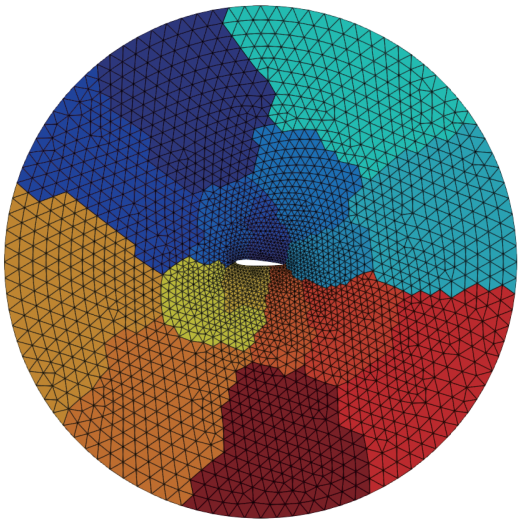


Figure 2. Partition of an unstructured grid. Different color represents different subdomains.

3. Numerical simulations

In this section, we validate our method by two numerical siulations, including flows over the NACA0012 airfoil and flows past a circular cylinder. All computations are executed on the Tianhe-2A supercomputer at the National Supercomputer Center in Guangzhou, China. Each node consists of two 12-core Intel Ivy Bridge Xeon CPUs and 64 GB memory. The specific heat ratio γ is 1.4, and the Prandtl number Pr 0.71 in the simulations.

3.1. Flows around the NACA0012 airfoil

The first benchmark test is the flow around the NACA0012 airfoil with two different configurations. For the first configuration, the Mach number at farfield $M_\infty = 0.5$, the angle of attack (AoA) $\alpha = 0°$. The Reynolds number is set to $Re = 5000$. The Mach number at farfield for the second configuration is$M_\infty = 2.0$, AoA $\alpha = 10°$, and $Re = 1000$. We adopt an unstructured grid with 2.87×10^5 quadrangular cells in these two numerial simulations. Figure 3 illustrates the computational domain and boundary configurations. A nonslip boundary condition is imposed on the airfoil surface, the non-equilibrium extrapolation method is employed at the inlet and outlet.

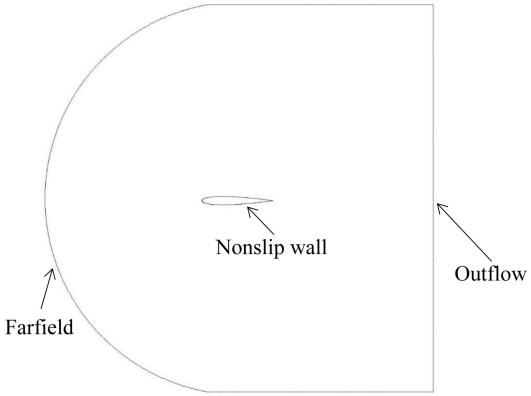
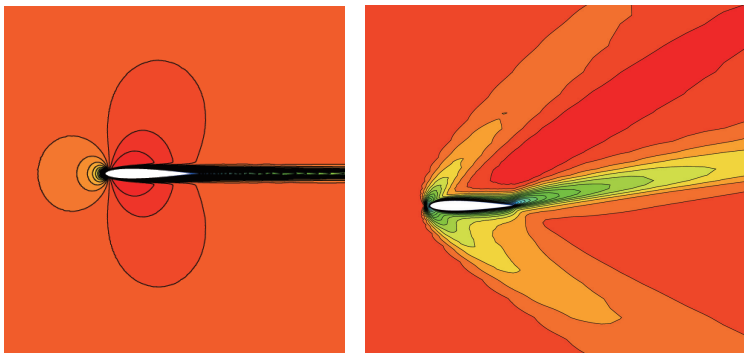


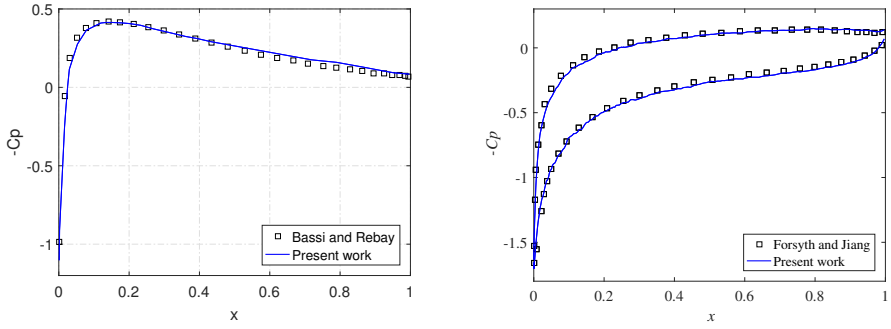
Figure 3. The computational domain and boundary configurations for flows around the NACA0012 airfoil.

The Mach number contours around the airfoil for the first simulation is shown in Figure 4(a), which is inaccrodance with the Mach number contours in [44]. Figure 5(a) gives the comparision of the calculated pressure coefficient with the one in [44], which shows that the results agree well with the reference data. The Mach number contours and pressure coefficients for the second simulation are given in Figures 4(b) and 5(b), respectively, which indicate that the simulated results are in good accordance with the results in [45].



(a) (b)

Figure 4. Mach number contours of the flow around the NACA0012 airfoil, (a) $\alpha = 0°$, $M_\infty = 0.5$, $Re = 5000$, (b) $\alpha = 10°$, $M_\infty = 2$, $Re = 1000$.



(a) (b)

Figure 5. Pressure coefficient around the NACA0012 airfoil, (a) $\alpha = 0°$, $M_\infty = 0.5$, $Re = 5000$, (b) $\alpha = 10°$, $M_\infty = 2$, $Re = 1000$.

3.2. Flows past a circular cylinder

In this subsection, two simulations are performed to compute supersonic flows past a circular cylinder [46] including (1) $M_\infty = 1.7$ and (2) $M_\infty = 2.5$. The Reynolds number is 2×10^5. The computational domain is a square $[0, 1] \times [0, 1]$ on the ξ-η plane. The boundary on the x-y plane is produced by (as shown in Figure 6)

$$\begin{cases} x = -\left(R_x - (R_x - 1)\,\dfrac{1}{a}\tanh(d\eta)\right)\cos\left(\dfrac{\pi}{2}\xi\right), \\ y = \left(R_y - \left(R_y - 1\right)\dfrac{1}{a}\tanh(d\eta)\right)\sin\left(\dfrac{\pi}{2}\xi\right). \end{cases}$$

Here $R_x = 4$, $R_y = 10$, $a = \tanh(d)$, and $d = 2.3$. We employ an unstructured grid with $32,188$ quadrangular cells in these two simulations. The inflow boundary is imposed at $\eta = 0$, where the distribution functions are fixed at their equilibrium states, the outflow boundary and the symmetric boundary are used at $\xi = 1$ and $\xi = 0$, repsectively, where the nonequilibrium extrapolation method is applied [43], the nonslip boundary is imposed at the cylinder surface $\eta = 1$.

The Mach number and pressure contours for $M_\infty = 1.7$ and $M_\infty = 2.5$ are illustrated in Figures 7 and 8, respectively. A bow shock located at $x = -3$ and $x = -1.9$ is formed at the upstream. The pressure coefficients Cp for the two cases along the surface of the cylinder compared with the ones in [38, 46, 47] are shown in Figure 9. In [47], an immersed boundary method is adopted on Cartesian-grid. The results in [46] are computed by a FD LBM. A GKS is used to compute the pressure coefficient in [38], which are different from our DUGKS. It can be seen that our results agree well with the ones in the literature.

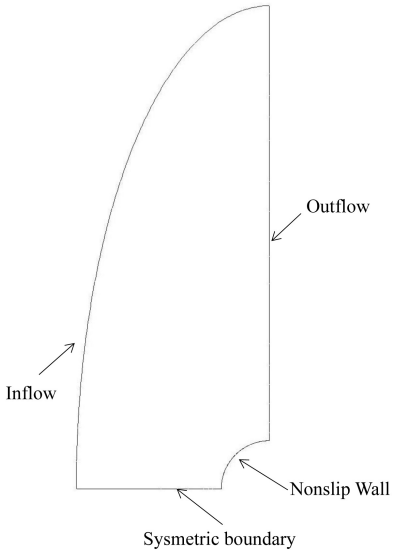


Figure 6. The setup of the flows past a circular cylinder on the x-y plane.

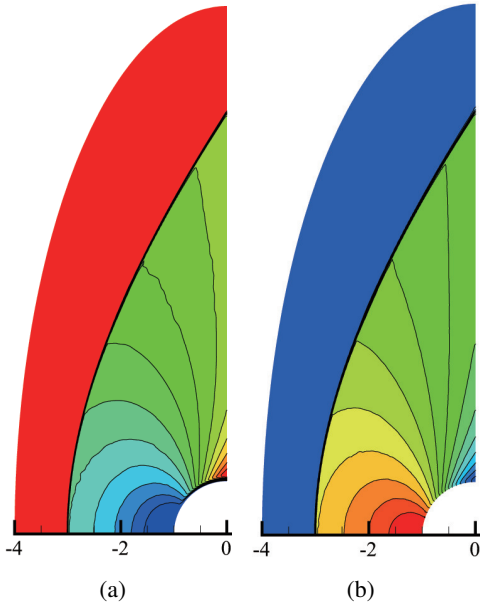
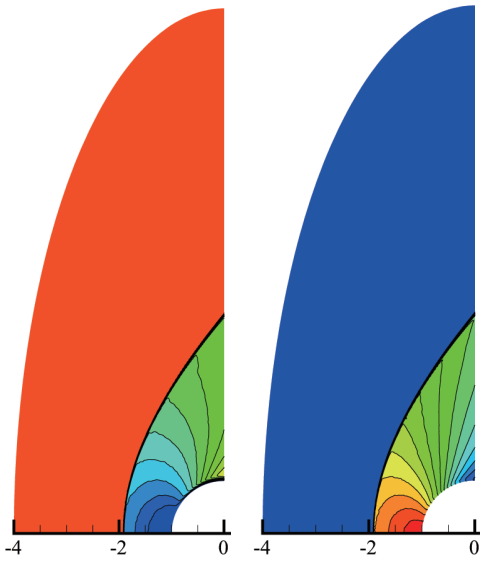
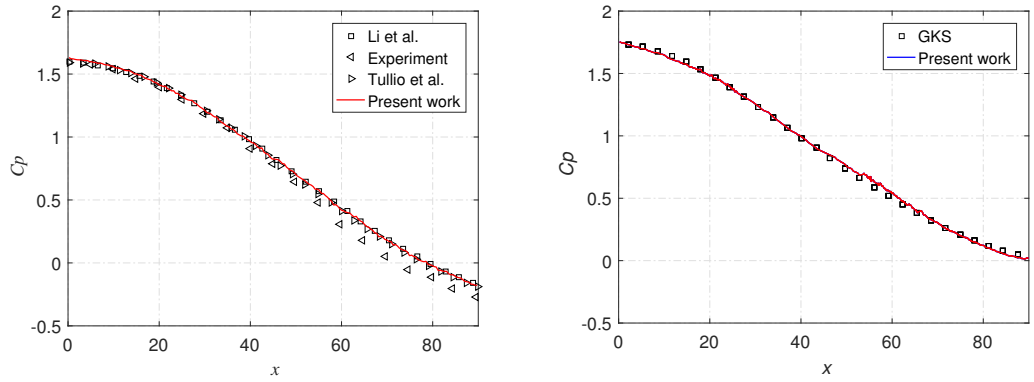


Figure 7. Mach number contours (a) and pressure contours (b) of the flow past a circular cylinder, $M_\infty = 1.7$.



(a) (b)

Figure 8. Mach number contours (a) and pressure contours (b) of the flow past a circular cylinder, $M_\infty = 2.5$.



(a) (b)

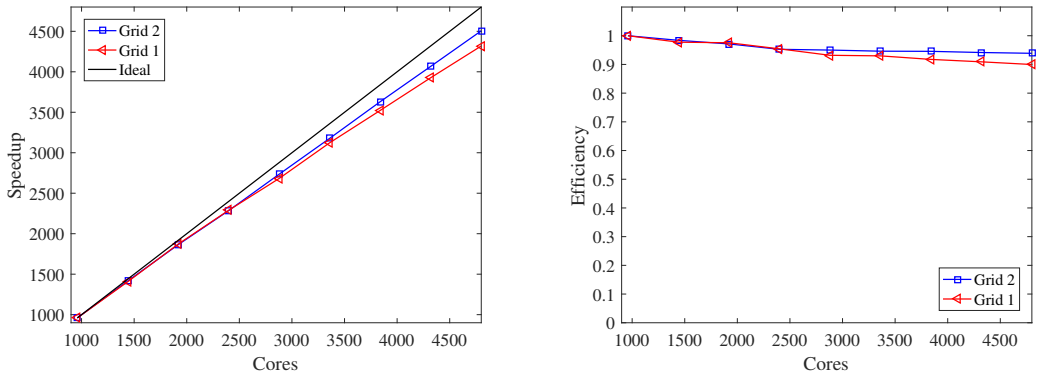
Figure 9. Pressure coefficient along the circular cylinder surface (a) $M_\infty = 1.7$, (b) $M_\infty = 2.5$.

3.3. Parallel scalability

The parallel performance of the proposed scheme is evaluated on the Tianhe-2A supercomputer [48]. In the test of parallel performance, a flow around the NACA0012 airfoil with two different grid sizes is used. The compute time and parallel efficiency of the scheme is shown in Table 1, in which the first 100 iterations are used to count the compute time. In the table, "Nodes" represent the number of compute nodes, "Cores" the number of processor cores, "Time" the compute time, and "Efficiency" the parallel efficiency, respectively. The speedup and parallel efficiency is illustrated in Figure 10. The larger the grid size is, the higher the speedup and parallel efficiency achieved.

Table 1. The strong scalability results for a flow around the NACA0012 airfoil with two different grid sizes. Grid 1 consists of 6.42×10^7 triangle cells, and grid 2 has 1.22×10^8 triangle cells.

Nodes	Cores	Grid 2		Grid 1	
		Time (s)	Efficiency	Time (s)	Efficiency
40	960	225.60	100.0%	122.46	100.0%
60	1440	152.85	98.40%	83.52	97.75%
80	1920	116.12	97.14%	62.77	97.55%
100	2400	94.72	95.27%	51.33	95.43%
120	2880	79.15	95.00%	43.81	93.18%
140	3360	68.12	94.62%	37.63	92.98%
160	3840	59.64	94.56%	33.37	91.75%
180	4320	53.25	94.15%	29.94	90.91%
200	4800	48.06	93.88%	27.22	89.97%



(a) (b)

Figure 10. The speedup (a) and efficiency (b) for a flow past the NACA0012 airfoil with 6.42×10^7 (Grid 1) and 1.22×10^8 (Grid 2) triangle cells. Here, "Ideal" refers to the ideal speedup.

4. Conclusions

In this paper, we presented a discrete unified gas kinetic scheme on unstructured grids and its parallel algorithm to simulate high Mach number viscous compressible flows, which is validated by two benchmark fluid problems. The validation indicates that the shock waves of high Mach number viscous compressible flows can be accurately captured. The parallel performance of the scheme investigated on the Tianhe-2A supercomputer shows that the parallel algorithm has good scalability on thousands of processor cores, which is suitable to perform high-resolution simulations of high Mach number viscid compressible flows. Based on our presented method, other discrete velocity models and equilibrium distribution functions for compressible flows are also be able to incorporate with our solver, and a

similar scalability can be expected. It is convenient to move to three-dimensional compressble flows, we just replace the discrete velocity model and its corresponding equilibrium distribution functions. We expect that our numerical methods can be used for solving some related inverse problems of practical importance [1, 2]; see also [49–53] for more related background discussion on inverse problems.

Acknowledgments

This work was supported by the National Key R&D Program of China (Grant No. 2018YFE0198400), the Shenzhen grant (Grant Nos. RCYX20200714114735074, JCYJ20220531100611025 and JCYJ20200109115422828), and the NSFC (Grant Nos. 62161160312, 12071461 and 12101588).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. Y. Deng, J. Li, H. Liu, On identifying magnetized anomalies using geomagnetic monitoring, *Arch. Rational Mech. Anal.*, **231** (2019), 153–187. https://doi.org/10.1007/s00205-018-1276-7
2. Y. Deng, J. Li, H. Liu, On identifying magnetized anomalies using geomagnetic monitoring within a magnetohydrodynamic model, *Arch. Rational Mech. Anal.*, **235** (2020), 691–721. https://doi.org/10.1007/s00205-019-01429-x
3. Y. Deng, H. Liu, W.-Y. Tsui, Identifying varying magnetic anomalies using geomagnetic monitoring, *Discrete Contin. Dyn. Syst.*, **40** (2020), 6411–6440. https://doi.org/10.3934/dcds.2020285
4. Y. Deng, Y. Gao, J. Li, H. Liu, R. Chen, Locating multiple magnetized anomalies by geomagnetic monitoring, unpublished work.
5. W. Li, S. Liu, S. Osher, Controlling conservation laws II: Compressible Navier-Stokes equations, *J. Comput. Phys.*, **463** (2022), 111264. https://doi.org/10.1016/j.jcp.2022.111264
6. M. Natarajan, R. Grout, W. Zhang, M. Day, A moving embedded boundary approach for the compressible Navier-Stokes equations in a block-structured adaptive refinement framework, *J. Comput. Phys.*, **465** (2022), 111315. https://doi.org/10.1016/j.jcp.2022.111315
7. G. Ju, C. Chen, R. Chen, J. Li, K. Li, S. Zhang, Numerical simulation for 3D flow in flow channel of aeroengine turbine fan based on dimension splitting method, *Electron. Res. Arch.*, **28** (2020), 837–851. https://doi.org/10.3934/era.2020043
8. M. E. Danis, J. Yan, A new direct discontinuous Galerkin method with interface correction for two-dimensional compressible Navier-Stokes equations, *J. Comput. Phys.*, **452** (2022), 110904. https://doi.org/10.1016/j.jcp.2021.110904
9. Z. Qiao, X. Yang, A multiple-relaxation-time lattice Boltzmann method with Beam-Warming scheme for a coupled chemotaxis-fluid model, *Electron. Res. Arch.*, **28** (2020), 1207–1225. https://doi.org/10.3934/era.2020066

10. K. Xu, A gas-kinetic BGK scheme for the Navier-Stokes equations and its connection with artificial dissipation and Godunov method, *J. Comput. Phys.*, **171** (2001), 289–335. https://doi.org/10.1006/jcph.2001.6790

11. C. Shu, Y. Wang, C. J. Teo, J. Wu, Development of lattice Boltzmann flux solver for simulation of incompressible flows, *Adv. Appl. Math. Mech.*, **6** (2014), 436–460. https://doi.org/10.4208/aamm.2014.4.s2

12. Z. H. Li, H. X. Zhang, Gas-kinetic numerical studies of three-dimensional complex flows on spacecraft re-entry, *J. Comput. Phys.*, **228** (2009), 1116–1138. https://doi.org/10.1016/j.jcp.2008.10.013

13. K. Xu, J. C. Huang, A unified gas-kinetic scheme for continuum and rarefied flows, *J. Comput. Phys.*, **229** (2010), 7747–7764. https://doi.org/10.1016/j.jcp.2010.06.032

14. Z. Guo, K. Xu, R. Wang, Discrete unified gas kinetic scheme for all Knudsen number flows: low-speed isothermal case, *Phys. Rev. E*, **88** (2013), 033305. https://doi.org/10.1103/PhysRevE.88.033305

15. A. U. Shirsat, S. G. Nayak, D. V. Patil, Simulation of high-Mach-number inviscid flows using a third-order Runge-Kutta and fifth-order WENO-based finite-difference lattice Boltzmann method, *Phys. Rev. E*, **106** (2022), 025314. https://doi.org/10.1103/PhysRevE.106.025314

16. J. Huang, X.-C. Cai, C. Yang, A fully implicit method for lattice Boltzmann equations, *SIAM J. Sci. Comput.*, **37** (2015), S291–S313. https://doi.org/10.1137/140975346

17. J. Huang, C. Yao, X.-C. Cai, A nonlinearly preconditioned inexact Newton algorithm for steady state lattice Boltzmann equations, *SIAM J. Sci. Comput.*, **38** (2015), A1701–A1724. https://doi.org/10.1137/15M1028078

18. R. Matin, M. K. Misztal, A. Hernandez-Garcia, J. Mathiesen, Finite element lattice Boltzmann simulations of contact line dynamics, *Phys. Rev. E*, **97** (2018), 013307. https://doi.org/10.1103/PhysRevE.97.013307

19. J. Wu, M. Shen, C. Liu, Study of flow over object problems by a nodal discontinuous Galerkin-lattice Boltzmann method, *Phys. Fluids*, **30** (2018), 040903. https://doi.org/10.1063/1.5010964

20. L. Xu, J. Li, R. Chen, A scalable parallel unstructured finite volume lattice Boltzmann method for three-dimensional incompressible flow simulations, *Int. J. Numer. Methods Fluids*, **93** (2021), 2744–2762. https://doi.org/10.1002/fld.4996

21. L. Xu, R. Chen, Scalable parallel finite volume lattice Boltzmann method for thermal incompressible flows on unstructured grids, *Int. J. Heat Mass Tran.*, **160** (2020), 120156. https://doi.org/10.1016/j.ijheatmasstransfer.2020.120156

22. L. Xu, R. Chen, X.-C. Cai, Parallel finite-volume discrete Boltzmann method for inviscid compressible flows on unstructured grids, *Phys. Rev. E*, **103** (2021), 023306. https://doi.org/10.1103/PhysRevE.103.023306

23. C. Sun, A. T. Hsu, Three-dimensional lattice Boltzmann model for compressible flows, *Phys. Rev. E*, **68** (2003), 016303. https://doi.org/10.1103/PhysRevE.68.016303

24. K. Qu, C. Shu, Y. T. Chew, Alternative method to construct equilibrium distribution functions in lattice-Boltzmann method simulation of inviscid compressible flows at high Mach number, *Phys. Rev. E*, **75** (2007), 036706. https://doi.org/10.1103/PhysRevE.75.036706
25. K. Li, C. Zhong, A lattice Boltzmann model for simulation of compressible flows, *Int. J. Numer. Methods Fluids*, **77** (2015), 334–357. https://doi.org/10.1002/fld.3984
26. K. Qu, Development of lattice Boltzmann method for compressible flows, Ph.D thesis, National University of Singapore, Singapore, 2008.
27. R.-F. Qiu, C.-X. Zhu, R.-Q. Chen, J.-F. Zhu, Y.-C. You, A double-distribution-function lattice Boltzmann model for high-speed compressible viscous flows, *Comput. Fluids*, **166** (2018), 24–31. https://doi.org/10.1016/j.compfluid.2018.01.039
28. Z. Liu, R. Chen, L. Xu, Parallel unstructured finite volume lattice Boltzmann method for high-speed viscid compressible flows, *International Journal of Modern Physics C*, **33** (2022), 2250066. https://doi.org/10.1142/S0129183122500668
29. L. Zhu, S. Chen, Z. Guo, dugksFoam: An open source OpenFOAM solver for the Boltzmann model equation, *Comput. Phys. Commun.*, **213** (2017), 155–164. https://doi.org/10.1016/j.cpc.2016.11.010
30. Z. Guo, K. Xu, Discrete unified gas kinetic scheme for multiscale heat transfer based on the phonon Boltzmann transport equation, *Int. J. Heat Mass Tran.*, **102** (2016), 944–958. https://doi.org/10.1016/j.ijheatmasstransfer.2016.06.088
31. P. Wang, S. Tao, Z. Guo, A coupled discrete unified gas-kinetic scheme for Boussinesq flows, *Comput. Fluids*, **120** (2015), 70–81. https://doi.org/10.1016/j.compfluid.2015.07.012
32. P. Wang, Y. Zhang, Z. Guo, Numerical study of three-dimensional natural convection in a cubical cavity at high Rayleigh numbers, *Int. J. Heat Mass Tran.*, **113** (2017), 217–228. https://doi.org/10.1016/j.ijheatmasstransfer.2017.05.057
33. H. Liu, M. Kong, Q. Chen, L. Zheng, Y. Cao, Coupled discrete unified gas kinetic scheme for the thermal compressible flows in all Knudsen number regimes, *Phys. Rev. E*, **98** (2018), 053310. https://doi.org/10.1103/PhysRevE.98.053310
34. Y. Wang, C. Zhong, S. Liu, Arbitrary Lagrangian-Eulerian-type discrete unified gas kinetic scheme for low-speed continuum and rarefied flow simulations with moving boundaries, *Phys. Rev. E*, **100** (2019), 063310. https://doi.org/10.1103/PhysRevE.100.063310
35. Y. Zhang, L. Zhu, P. Wang, Z. Guo, Discrete unified gas kinetic scheme for flows of binary gas mixture based on the McCormark model, *Phys. Fluids*, **31** (2019), 017101. https://doi.org/10.1063/1.5063846
36. J. Shang, Z. Chai, H. Wang, B. Shi, Discrete unified gas kinetic scheme for nonlinear convection-diffusion equations, *Phys. Rev. E*, **101** (2020), 023306. https://doi.org/10.1103/PhysRevE.101.023306
37. M. Zhong, S. Zou, D. Pan, C. Zhuo, C. Zhong, A simplified discrete unified gas kinetic scheme for incompressible flow, *Phys. Fluids*, **32** (2020), 093601. https://doi.org/10.1063/5.0021332
38. M. Zhong, S. Zou, D. Pan, C. Zhuo, C. Zhong, A simplified discrete unified gas–kinetic scheme for compressible flow, *Phys. Fluids*, **33** (2021), 036103. https://doi.org/10.1063/5.0033911

39. G. Karypis, K. Schloegel, PARMETIS: Parallel graph partitioning and sparse matrix ordering library version 4.0, Technical Report, 97-060.

40. S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, et al., PETSc/TAO users manual, Argonne National Laboratory, ANL-21/39 - Revision 3.17, 2022. Available from: `https://petsc.org/`

41. Z. Guo, C. Zheng, B. Shi, T. S. Zhao, Thermal lattice Boltzmann equation for low Mach number flows: decoupling model, *Phys. Rev. E*, **75** (2007), 036704. https://doi.org/10.1103/PhysRevE.75.036704

42. V. Venkatakrishnan, Convergence to steady-state solutions of the Euler equations on unstructured grids with limiters, *J. Comput. Phys.*, **118** (1995), 120–130. https://doi.org/10.1006/jcph.1995.1084

43. Z. Guo, C. Zheng, B. Shi, Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method, *Chinese Phys.*, **11** (2002), 366–375. https://doi.org/10.1088/1009-1963/11/4/310

44. F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, *J. Comput. Phys.*, **131** (1997), 267–279. https://doi.org/10.1006/jcph.1996.5572

45. P. A. Forsyth, H. Jiang, Nonlinear iteration methods for high speed laminar compressible Navier-Stokes equations, *Comput. Fluids*, **26** (1997), 249–279. https://doi.org/10.1016/S0045-7930(96)00041-2

46. Q. Li, Y. L. He, Y. J. Gao, Implementation of finite-difference lattice Boltzmann method on general body-fitted curvilinear coordinates, *International Journal of Modern Physics C*, **19** (2008), 1581–1595. https://doi.org/10.1142/S0129183108013126

47. M. D. De Tullio, P. De Palma, G. Iaccarino, G. Pascazio, M. Napolitano, An immersed boundary method for compressible flows using local grid refinement, *J. Comput. Phys.*, **225** (2007), 2098–2117. https://doi.org/10.1016/j.jcp.2007.03.008

48. X. Liao, L. Xiao, C. Yang, Y. Lu, MilkyWay-2 supercomputer: system and application, *Front. Comput. Sci.*, **8** (2014), 345–356. https://doi.org/10.1007/s11704-014-3501-3

49. Y. T. Chow, Y. Deng, Y. He, H. Liu, X. Wang, Surface-localized transmission eigenstates, super-resolution imaging, and pseudo surface plasmon modes, *SIAM J. Imaging Sci.*, **14** (2021), 946–975. https://doi.org/10.1137/20M1388498

50. Z. Bai, H. Diao, H. Liu, Q. Meng, Stable determination of an elastic medium scatterer by a single far-field measurement and beyond, *Calc. Var.*, **61** (2022), 170. https://doi.org/10.1007/s00526-022-02278-5

51. H. Liu, On local and global structures of transmission eigenfunctions and beyond, *J. Inverse Ill-Posed Probl.*, **30** (2022), 287–305. https://doi.org/10.1515/jiip-2020-0099

52. Y. Gao, H. Liu, X. Wang, K. Zhang, On an artificial neural network for inverse scattering problems, *J. Comput. Phys.*, **448** (2022), 110771. https://doi.org/10.1016/j.jcp.2021.110771

53. E. L. K. Blasten, H. Liu, Scattering by curvatures, radiationless sources, transmission eigenfunctions, and inverse scattering problems, *SIAM J. Math. Anal.*, **53** (2021), 3801–3837. https://doi.org/10.1137/20M1384002

AIMS Press