Journal of Computational Physics 376 (2019) 973–991

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A high-order hybridizable discontinuous Galerkin method with fast convergence to steady-state solutions of the gas kinetic equation

Wei Su, Peng Wang, Yonghao Zhang, Lei Wu *

James Weir Fluids Laboratory, Department of Mechanical and Aerospace Engineering, University of Strathclyde, G1 1XJ Glasgow, United Kingdom

ARTICLE INFO

Article history:
Received 21 March 2018
Received in revised form 10 August 2018
Accepted 29 August 2018
Available online 1 September 2018

Keywords:
Hybridizable discontinuous Galerkin
Gas kinetic equation
Synthetic iterative scheme
Fast convergence

ABSTRACT

The mass flow rate of Poiseuille flow of rarefied gas through long ducts of two-dimensional cross-sections with arbitrary shape is critical in the pore-network modeling of gas transport in porous media. Here, for the first time, the high-order hybridizable discontinuous Galerkin (HDG) method is used to find the steady-state solution of the linearized Bhatnagar–Gross–Krook equation on two-dimensional triangular meshes. The velocity distribution function and its traces are approximated in piecewise polynomial spaces (of degree up to 4) on the triangular meshes and mesh skeletons, respectively. By employing a numerical flux that is derived from the first-order upwind scheme and imposing its continuity weakly on the mesh skeletons, global systems for unknown traces are obtained with fewer coupled degrees of freedom when compared to the original discontinuous Galerkin formulation. To achieve fast convergence to the steady-state solution, a diffusion-like equation for flow velocity, which is asymptotic-preserving into the fluid dynamic limit, is solved by the HDG simultaneously on the same meshes. The proposed HDG-synthetic iterative scheme is proved to be accurate and efficient. Specifically, for flows in the near-continuum regime, numerical simulations have shown that, to achieve the same level of accuracy, our scheme could be faster than the conventional iterative scheme by two orders of magnitude, also it is faster than the synthetic iterative scheme based on the finite difference discretization in the spatial space by one order of magnitude. In addition, the implicit HDG method is more efficient than an explicit discontinuous Galerkin gas kinetic solver, as well as the implicit discontinuous Galerkin scheme when the degree of approximating polynomial is larger than 2. The HDG-synthetic iterative scheme is ready to be extended to simulate rarefied gas mixtures and the Boltzmann collision operator.

1. Introduction

Accurate physical models and efficient numerical methods are needed to describe the gas flow spanning a wide range of rarefactions. The conventional Navier–Stokes (NS) equations, however, are valid in the continuum flow regime only, where the Knudsen number Kn, i.e. the ratio of the mean free path of gas molecules λ to the flow characteristic dimension H, is

* Corresponding author.
E-mail address: lei.wu.100@strath.ac.uk (L. Wu).

https://doi.org/10.1016/j.jcp.2018.08.050

less than 0.001. Beyond this regime, gas flows deviate from equilibrium and the Boltzmann equation from the gas kinetic theory should be used. According to the Chapman–Enskog expansion, NS equations are the approximated solution of the Boltzmann equation to the first-order of the Knudsen number [1]. As Kn increases, higher-order terms beyond the linear constitutive relations begin to dominate, and NS equations gradually lose their validity. The non-equilibrium effects not only cause velocity slip and temperature jump at the solid surface in the slip flow regime ($0.001 \leq Kn < 0.1$), but also modify the constitutive relations, such as the Newton's law for stress and strain as well as the Fourier's law for heat flux and temperature gradient in the transition ($0.1 \leq Kn < 10$) and free-molecular ($10 \leq Kn$) flow regimes. In these non-equilibrium flow regimes, the profile of the local velocity distribution function (VDF) of gas molecules is not known a priori. Instead, the VDF should be determined by solving the Boltzmann equation numerically. Two categories of numerical approaches have been developed for this task. One is the direct simulation Monte Carlo method [2] that uses a collection of particles to mimic the molecular behavior stochastically, and the other is the deterministic method, which relies on the discretization of governing equations over computational grids [3]. Generally speaking, the particle-based methods are efficient and robust for high-speed flows, since they are unconditionally stable and easy to implement high-fidelity physico-chemical models, while the deterministic methods which yield noise-free solutions are promising for low-speed flow simulations [4].

In the past decades, due to the rapid development of micro-electro-mechanical systems and the shale gas revolution in North America, extensive works have been devoted to constructing efficient deterministic schemes. These methods often adopt a numerical quadrature to approximate the integration with respect to the molecular velocity on a discrete set of velocities [5]. Then, the VDF, which is discrete in the velocity space but continuous in the spatial space and time, is resolved by the finite difference method (FDM), finite volume method (FVM), and finite element method (FEM) [6–9]. Compared to the NS equations, numerical simulation of the Boltzmann equation is expensive in terms of computation time and memory. First of all, additional dimensions of the molecular velocity space are discretized, resulting in a system of governing equations for each discrete velocity. Generally speaking, flows with large values of Kn require a large number of discrete velocities to resolve the significant variations and discontinuities in the VDF [9,10]. Second, most of the deterministic schemes treat the streaming and collision separately. Therefore, in order to suppress the numerical diffusion errors, the size of spatial cell and time interval should be smaller than the mean free path and the mean collision time, respectively [11]. For this reason, the deterministic technique becomes costly for near-continuum flows. Finally, the iteration scheme to find steady-state solution converges extremely slowly for flows at low Knudsen numbers, since the exchange of information (e.g. perturbation in the flow field) through streaming becomes very inefficient when binary collisions dominate [12].

Great efforts have been devoted to overcoming the above limitations in various aspects. In addition to the commonly used techniques such as high-order discretization scheme or automatically adaptive refinement in the spatial and velocity spaces [13–15], two alternative approaches are worth mentioning here. One is proposed to handle the streaming and collision simultaneously so that the restrictions on cell size and time step are significantly relaxed. This strategy has been realized in the unified gas-kinetic scheme (UGKS) [16–19] by calculating the time-evolution of the flux at the cell interface due to convection and collision. Its advantage of asymptotic-preserving into the NS limit enables UGKS to capture the essential flow physics on coarse grids [20]. Nevertheless, since information is exchanged through the evolution of VDF, UGKS still needs a large number of time steps to obtain steady-state solutions in near-continuum flows [21,22]. The other strategy, known as the synthetic iterative scheme (SIS), achieves high efficiency and accuracy in particular with fast convergence property by solving the kinetic equation and a diffusion-like equation for macroscopic quantities simultaneously [23,24]. Since the VDF is guided by the macroscopic flow quantities from the diffusion equation at each iterative step, information propagates accurately and fast even on the coarse grid when Kn is small. Moreover, the macroscopic equation contains high-order moments of VDF to take into account non-equilibrium effects, thus the SIS also preserves accuracy in the simulation of high Kn flows. Based on the FDM in the spatial space, the SIS has been successfully applied to Poiseuille flow using the Bhatnagar–Gross–Krook (BGK) kinetic model for single-species gases [25], and flows of binary and ternary gas mixtures driven by local pressure, temperature and concentration gradients using the McCormack model [26–29]. Recently, an SIS is proposed to solve the linearized Boltzmann equation, where the role of realistic intermolecular potentials in Poiseuille and thermal transpiration flows has been analyzed [12] and compared to experiment [30].

In the present paper, to further achieve high-order discretization and enable the capability of dealing with complex geometries, the high-order hybridizable discontinuous Galerkin (HDG) discretization and SIS are coupled to solve the linearized BGK equation for Poiseuille flow through two-dimensional cross-section of arbitrary shape. The developed HDG-SIS has important application in the simulation of rarefied gas flow through complex porous media via the pore-network modeling, where three-dimensional pores with various shapes of two-dimensional cross-sections are extracted [31], e.g. from the ultra-tight shale strata. Accurate and efficient numerical method for solving the gas kinetic equation is urgently needed to find the mass flow rate or apparent permeability of these pores, such that the permeability of the porous media can be obtained by the "Kirchhoff's circuit law" based on the pore-network modeling.

The remainder of the paper is organized as follows. In Sec. 2, the BGK equation and its synthetic macroscopic equation for the fast convergence of flow velocity in Poiseuille flow are introduced. In Sec. 3 the numerical scheme is described with details in the HDG formulation, flux construction, and implementation of boundary conditions. Four different problems are simulated in Sec. 4 to assess the accuracy and efficiency of the proposed HDG-SIS scheme. Conclusions and outlooks are presented in Sec. 5.

2. The gas kinetic equation

The Boltzmann equation describes the evolution of the VDF depending on spatial position $\boldsymbol{x}' = (x_1', x_2', x_3')$, molecular velocity $\boldsymbol{v}' = (v_1', v_2', v_3')$, and time t'. In Cartesian coordinates it has the form of:

$$\frac{\partial f'}{\partial t'} + \boldsymbol{v}' \cdot \frac{\partial f'}{\partial \boldsymbol{x}'} + \boldsymbol{a}' \cdot \frac{\partial f'}{\partial \boldsymbol{v}'} = \mathcal{C}\left(f'\right). \tag{1}$$

Here, f' is the VDF that is defined so that the number density of gas molecules at time t', with velocity within the limits $\boldsymbol{v}'$ and $\boldsymbol{v}' + \mathrm{d}\boldsymbol{v}'$, and spatial location within $\boldsymbol{x}'$ and $\boldsymbol{x}' + \mathrm{d}\boldsymbol{x}'$, is equal to $f'\mathrm{d}\boldsymbol{v}'\mathrm{d}\boldsymbol{x}'$. $\boldsymbol{a}' = (a_1', a_2', a_3')$ is the external acceleration, while $\mathcal{C}(f')$ is the collision operator, which describes the change in VDF after binary collisions [1].

Due to complexity of the collision operator, the full Boltzmann equation is amenable to analytical solutions only for few special cases. In practice, deterministic solution is commonly sought for gas kinetic models that reduce $\mathcal{C}(f')$ to simpler collision operators; frequently used are the BGK [32], ellipsoidal statistical BGK [33], and Shakhov [34] models. Here we develop the numerical scheme based on the following BGK equation, which is written in the non-dimensional form as:

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f}{\partial \boldsymbol{x}} + \boldsymbol{a} \cdot \frac{\partial f}{\partial \boldsymbol{v}} = \delta n T^{1-\omega}\left(F_{\mathrm{eq}} - f\right), \tag{2}$$

where $\boldsymbol{v}$ is $\boldsymbol{v}'$ normalized by the most probable speed $v_{\mathrm{m}} = \sqrt{2RT_0}$ at the reference temperature T_0 with R being the specific gas constant, $\boldsymbol{x}$ is $\boldsymbol{x}'$ normalized by the characteristic flow length H, $\boldsymbol{a}$ is $\boldsymbol{a}'$ normalized by v_{m}^2/H, t is t' normalized by H/v_{m}, n is the number density of gas molecules normalized by the average number density n_0 at T_0, T is the gas temperature normalized by T_0, and f is f' normalized by n_0/v_{m}^3. The coefficient ω is the viscosity index, i.e. the shear viscosity μ of the gas is proportional to T^{ω}. The normalized equilibrium VDF is defined as:

$$F_{\mathrm{eq}} = \frac{n}{(\pi T)^{3/2}} \exp\left(-\frac{|\boldsymbol{v} - \boldsymbol{u}|^2}{T}\right), \tag{3}$$

where $\boldsymbol{u} = (u_1, u_2, u_3)$ is the macroscopic flow velocity normalized by v_{m}. Finally, the equivalent rarefaction parameter δ is defined as the inversed Knudsen number:

$$\delta = \frac{\sqrt{\pi}}{2Kn} = \frac{p_0 H}{\mu_0 \sqrt{2RT_0}}, \tag{4}$$

with p_0 and μ_0 being the pressure and shear viscosity of the gas at reference temperature T_0, respectively.

When the flow velocity is sufficiently small compared to v_{m}, and the external acceleration is also small, we can linearize the VDF about the global equilibrium state f_{eq} as:

$$f = f_{\mathrm{eq}}(1 + h), \quad f_{\mathrm{eq}} = \frac{\exp\left(-|\boldsymbol{v}|^2\right)}{\pi^{3/2}}, \tag{5}$$

and the VDF $h(\boldsymbol{x}, \boldsymbol{v})$ for the perturbation is governed by the following linearized BGK equation [35]:

$$\begin{aligned} &\boldsymbol{v} \cdot \frac{\partial h}{\partial \boldsymbol{x}} - 2\boldsymbol{a} \cdot \boldsymbol{v} = \mathcal{L}\left(\varrho, \boldsymbol{u}, \tau, \boldsymbol{v}\right) - \delta h, \\ &\mathcal{L}\left(\varrho, \boldsymbol{u}, \tau, \boldsymbol{v}\right) = \delta\left[\varrho + 2\boldsymbol{u} \cdot \boldsymbol{v} + \tau\left(|\boldsymbol{v}|^2 - \frac{3}{2}\right)\right], \end{aligned} \tag{6}$$

in which we have omitted the derivative with respect to the time since we are only interested in the steady-state solution. The macroscopic gas variables, including the perturbed number density ϱ, the flow velocity $\boldsymbol{u}$, and the perturbed temperature τ, are calculated from the velocity moments of the perturbed VDF:

$$\varrho = \int h f_{\mathrm{eq}} \mathrm{d}\boldsymbol{v}, \quad \boldsymbol{u} = \int \boldsymbol{v} h f_{\mathrm{eq}} \mathrm{d}\boldsymbol{v}, \quad \tau = \frac{2}{3}\int |\boldsymbol{v}|^2 h f_{\mathrm{eq}} \mathrm{d}\boldsymbol{v} - \varrho. \tag{7}$$

2.1. The discrete velocity model

The deterministic approach relies on the discrete velocity method (DVM), in which a set of M_{v} discrete velocities $\boldsymbol{v}^j = (v_1^{j_1}, v_2^{j_2}, v_3^{j_3})$ are chosen to represent the VDF [5]. If we denote $h^j = h(\boldsymbol{x}, \boldsymbol{v}^j)$, $\mathcal{L}^j\left(\varrho, \boldsymbol{u}, \tau\right) = \mathcal{L}\left(\varrho, \boldsymbol{u}, \tau, \boldsymbol{v}^j\right)$, and $f_{\mathrm{eq}}^j = f_{\mathrm{eq}}\left(\boldsymbol{v}^j\right)$, the linearized BGK equation is replaced by a system of differential equations for h^j that are discrete in the velocity space but still continuous in the spatial space:

$$\boldsymbol{v}^j \cdot \frac{\partial h^j}{\partial \boldsymbol{x}^j} - 2\boldsymbol{a} \cdot \boldsymbol{v}^j = \mathcal{L}^j\left(\varrho, \boldsymbol{u}, \tau\right) - \delta h^j, \quad j = 1, \ldots, M_{\mathrm{v}}, \tag{8}$$

which are usually solved by the following conventional iterative scheme (CIS):

$$\delta h^{j,(t+1)} + \boldsymbol{v}^j \cdot \frac{\partial h^{j,(t+1)}}{\partial \boldsymbol{x}} = \mathcal{L}^j\left(\varrho^{(t)}, \boldsymbol{u}^{(t)}, \tau^{(t)}\right) + 2\boldsymbol{a}\cdot\boldsymbol{v}^j, \tag{9}$$

where the superscripts (t) and $(t+1)$ represent two consecutive iteration steps. The iteration is terminated when the convergence to the steady solution is achieved. For conciseness, we will omit the index of iteration step in the remainder of the paper unless necessary.

Note that $\mathcal{L}^j$ contains macroscopic variables, which can be evaluated using some numerical quadratures:

$$\varrho = \sum_{j=1}^{M_\mathrm{v}} h^j f_{\mathrm{eq}}^j \varpi^j, \quad \boldsymbol{u} = \sum_{j=1}^{M_\mathrm{v}} \boldsymbol{v}^j h^j f_{\mathrm{eq}}^j \varpi^j, \quad \tau = \frac{2}{3}\sum_{j=1}^{M_\mathrm{v}} |\boldsymbol{v}^j|^2 h^j f_{\mathrm{eq}}^j \varpi^j - \varrho, \tag{10}$$

where ϖ^j is the weight of the quadrature rule. Various quadratures are used, including the Gauss-Hermite quadrature [36] and the composite Newton–Cote rule with uniform and non-uniform velocity discretizations [6,9].

2.2. The synthetic iterative scheme for fast convergence

It is well known that the iterative scheme (9) is very efficient in the free-molecular flow regime where binary collisions are negligible. However, for near-continuum flows the iteration scheme converges slowly and the results are very likely to be biased by accumulated rounding errors. The SIS, which has the asymptotic-preserving property in the NS limit and enables rapid convergence to the steady-state, has been developed for the linearized kinetic equations [23,24,12] to achieve high efficiency and accuracy.

In this paper, we consider the steady gas flow along a channel of arbitrary cross-section in the x_1-x_2 plane, subject to a small pressure gradient in the x_3 direction. It is assumed that the channel length is significantly larger than the other dimensions of its cross-section plane as well as the mean free path of gas molecules, thus the end effects can be neglected and the flow field only varies in x_1 and x_2 directions. Suppose the pressure gradient X_P, which is normalized by p_0/H, is very small, then the term $2\boldsymbol{a}\cdot\boldsymbol{v}^j$ in the linearized BGK equation (9) can be replaced by $-X_\mathrm{P} v_3^j$, and the synthetic diffusion equation for the flow velocity u_3 in the x_3 direction is given by [12]:

$$\frac{\partial^2 u_3}{\partial x_1^2} + \frac{\partial^2 u_3}{\partial x_2^2} = X_\mathrm{P}\delta - \frac{1}{4}\left(\frac{\partial^2 F_{2,0,1}}{\partial x_1^2} + 2\frac{\partial^2 F_{1,1,1}}{\partial x_1 \partial x_2} + \frac{\partial^2 F_{0,2,1}}{\partial x_2^2}\right), \tag{11}$$

where $F_{m,n,l}(x_1,x_2) = \sum_{j=1}^{M_\mathrm{v}} f_{\mathrm{eq}}^j h^j H_m(v_1) H_n(v_2) H_l(v_3) \varpi^j$ are high-order moments, with $H_n(v)$ being the n-th order physicists' Hermite polynomial.

It should be noted that Eq. (11) is derived from the linearized BGK equation with no approximation, which only works for Poiseuille flow when the flow velocity is perpendicular to the computational domain. In the near-continuum flow regime where δ is large, the diffusion equation is reduced to the NS equation $\partial^2 u_3/\partial x_1^2 + \partial^2 u_3/\partial x_2^2 = X_\mathrm{P}\delta$. That is to say, it is asymptotic-preserving to the hydrodynamic limit. Since the diffusion equation exchanges the information very efficiently, fast convergence and high accuracy in the near-continuum flow regime can be easily achieved by solving the gas kinetic equation (9) in parallel with the diffusion equation (11). On the other hand, when δ is very small, i.e. the flow is highly rarefied, high-order moments will play significant roles. We assume $X_\mathrm{P} = -1$ in the following calculations.

3. The HDG method

The discontinuous Galerkin (DG) finite element method was initially introduced for the neutron transport equation [37]. In the last few decades, after its success in solving nonlinear hyperbolic conservation laws and many convection-dominated problems [38,39], it is recognized as one of the most promising methods for next generation computational fluid dynamics. Similar to the FVM, DG methods assume discontinuous solution space, where the resulting equations are closed by approximation of the numerical flux on the cell interfaces. Instead of reconstructing the solution on large stencils, high-order spatial accuracy of the DG solution is sought by means of element-by-element polynomial functions. The compactness and their discontinuous nature make the methods ideal for parallelization and the implementation of *hp*-adaptive schemes.

In recent years, DG methods have been applied to the gas kinetic model equations [40], and the linearized/full Boltzmann equations [41–43] for the simulation of non-equilibrium gas flows. For kinetic model equations, it has been shown that the second-order DG discretization combined with the explicit Runge–Kutta time marching is more efficient than the second-order FVM scheme [40]. Despite these advantages, classical DG methods are computationally more expensive than their continuous Galerkin counterparts for steady problem or implicit scheme. This is mainly due to the large number of degrees of freedom in approximating field variables resulting from the discontinuous nature. This shortcoming is enlarged when solving the diffusion equation, where additional auxiliary variables are introduced to approximate the derivatives of the solution [44].

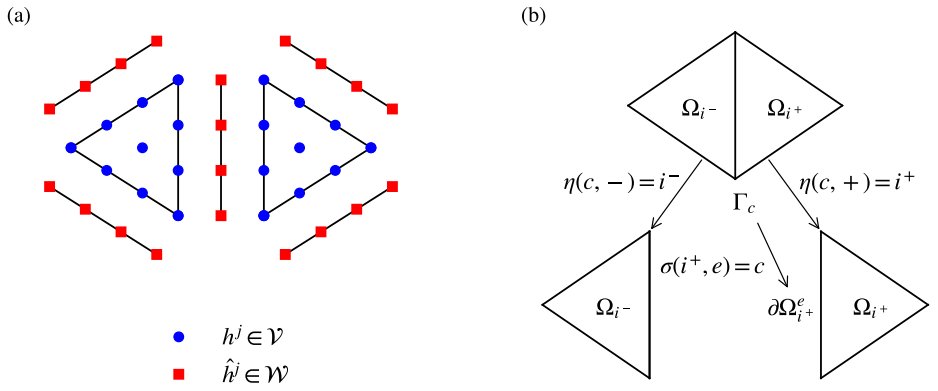

Fig. 1. (a) Nodal points and solution spaces for $k=3$ to approximate the distribution function h^j and its trace $\hat{h}^j$. (b) Schematic demonstration of the index mapping functions that relates the local edge of a triangle $\partial\Omega_i^e$ to a global face Γ_c.

The HDG method was proposed to overcome this disadvantage [45]. By producing a final system in terms of the degrees of freedom interpolating traces of the field variables, HDG could significantly reduce the number of global unknowns, since the traces are defined on cell interfaces and single-valued. Therefore, the HDG method is more appropriate for steady and implicit solvers. This advantage is prominent for the gas kinetic simulation, where a cumbersome system of governing equations needs to be solved. The majority of HDG applications in fluid dynamics to date includes convection–diffusion flow [45], stokes flow [46], wave propagation problem [47] and incompressible/compressible NS flows [48–50]. In this paper, for the first time, the HDG method is designed for gas kinetic equations, which will be detailed below.

3.1. Hybridizable discontinuous Galerkin formulation

We apply the discontinuous Galerkin method to discretize the system in the spatial space. Let $\Omega \in \mathbb{R}^2$ be a two-dimensional domain with boundary $\partial\Omega$ in the x_1-x_2 plane. Then, Ω is partitioned into M_{el} disjoint regular triangles Ω_i:

$$\Omega = \cup_i^{M_{\text{el}}} \Omega_i. \tag{12}$$

The boundaries $\partial\Omega_i$ of the triangles define a group of M_{fc} faces Γ_c:

$$\Gamma = \cup_i^{M_{\text{el}}} \{\partial\Omega_i\} = \cup_c^{M_{\text{fc}}} \{\Gamma_c\}. \tag{13}$$

The HDG method provides an approximate solution to h^j on Ω_i as well as an approximation to its trace $\hat{h}^j$ on Γ_c in some piecewise finite element spaces $\mathcal{V} \times \mathcal{W}$ in the following forms:

$$\begin{aligned} \mathcal{V} &= \{\varphi :\ \varphi|_{\Omega_i} \in \mathcal{P}^k(\Omega_i),\ \forall\, \Omega_i \subset \Omega\}, \\ \mathcal{W} &= \{\psi :\ \psi|_{\Gamma_c} \in \mathcal{P}^k(\Gamma_c),\ \forall\, \Gamma_c \subset \Gamma\}, \end{aligned} \tag{14}$$

where $\mathcal{P}^k(D)$ denotes the space of k-th order polynomials on the domain D, as shown in Fig. 1(a). Before describing the HDG formulation, we first define a collection of index mapping functions, namely σ and η that relates the local edge of a triangle $\partial\Omega_i^e$ to a global face Γ_c [51]. Since the e-th edge of the triangle $\partial\Omega_i$ is the c-th face Γ_c, we set $\sigma(i,e)=c$ so that $\partial\Omega_i^e = \Gamma_{\sigma(i,e)}$. Similarly, since the interior face $\Gamma_c \in \Gamma\backslash\partial\Omega$ is the intersection of the two triangles, namely left triangle Ω_{i^-} and right triangle Ω_{i^+}, we set $\eta(c,+)=i^+$ and $\eta(c,-)=i^-$, then we denote $\Gamma_c = \partial\Omega_{\eta(c,+)} \cap \partial\Omega_{\eta(c,-)}$. At the boundary face $\Gamma_c \in \partial\Omega$, only the right triangle is involved. The mapping functions are illustrated in Fig. 1(b).

3.1.1. Formulation of HDG method

The HDG method solves the problem in two steps [45]. First, a global problem is set up to determine the trace $\hat{h}^j$ on the face Γ. Then, a local problem with $\hat{h}^j$ as the boundary condition on $\partial\Omega_i$ is solved element-by-element to obtain the solutions for h^j. Generally speaking, when moving from the interior of the triangle element Ω_i to its boundary $\partial\Omega_i$, $\hat{h}^j$ defines what the value of h^j on the boundary should be. In the HDG method, it is assumed that $\hat{h}^j$ is singled-valued on each face.

Introducing $(\cdot)$ and $\langle\cdot\rangle$ as $(a,b)_D = \int_{D\subset\mathbb{R}^2}(a\cdot b)\mathrm{d}x_1\mathrm{d}x_2$ and $\langle a,b\rangle_D = \int_{D\subset\mathbb{R}^1}(a\cdot b)\mathrm{d}\Gamma$, respectively, the weak formulation of Eq. (9) for the VDF h^j in each element Ω_i is:

$$-\left(\nabla\varphi, \boldsymbol{v}^j h^j\right)_{\Omega_i} + \sum_{e=1}^{3} \langle\varphi, \hat{\boldsymbol{F}}\cdot\boldsymbol{n}\rangle_{\partial\Omega_i^e} + (\varphi, \delta h^j)_{\Omega_i} = (\varphi, s^j)_{\Omega_i}, \quad \text{for all } \varphi \in \mathcal{V}, \tag{15}$$

where $\boldsymbol{n}$ is the outward unit normal vector, $s^j = \mathcal{L}^j - X_P v_3^{j_3}$, and $\hat{\boldsymbol{F}}$ is the numerical trace of the flux defined as [52]:

$$\hat{\boldsymbol{F}}^j \cdot \boldsymbol{n} = \boldsymbol{v}^j \cdot \boldsymbol{n} \hat{h}^j + \alpha \left(h^j - \hat{h}^j \right), \tag{16}$$

where α is a stabilization parameter on each edge $\partial\Omega_i^e$ calculated as [48]:

$$\alpha = |\boldsymbol{v}^j \cdot \boldsymbol{n}|. \tag{17}$$

On inserting Eq. (16) into Eq. (15), we can express the solution of h^j on each triangle as a function of $\hat{h}^j$. In matrix form, it is written as

$$\mathbf{H}^{i,j} = \left[\mathbf{A}^{i,j}\right]^{-1} \mathbf{S}^{i,j} + \left[\mathbf{A}^{i,j}\right]^{-1} \hat{\mathbf{A}}^{i,j} \hat{\mathbf{H}}^{i,j}, \tag{18}$$

where $\mathbf{H}^{i,j}$ and $\hat{\mathbf{H}}^{i,j}$ are the vectors of degrees of freedom of h^j and $\hat{h}^j$ on Ω_i and $\partial\Omega_i$, respectively. The coefficient matrices $\mathbf{A}^{i,j}$, $\mathbf{S}^{i,j}$ and $\hat{\mathbf{A}}^{i,j}$ are given in the Appendix A.

The global problem, used to determine $\hat{h}^j$, is obtained by imposing the continuity of the normal flux at cell interfaces. For all $\psi \in \mathcal{W}$, the weak formulation is:

$$\begin{aligned} &\langle \psi, \hat{\boldsymbol{F}} \cdot \boldsymbol{n}_{\eta(c,+)} \rangle_{\Gamma_c} + \langle \psi, \hat{\boldsymbol{F}} \cdot \boldsymbol{n}_{\eta(c,-)} \rangle_{\Gamma_c} = 0, \quad \text{on } \Gamma \backslash \partial\Omega, \\ &\langle \psi, \hat{\boldsymbol{F}} \cdot \boldsymbol{n}_{\eta(c,+)} \rangle_{\Gamma_c} + \langle \psi, \hat{\boldsymbol{G}} \cdot \boldsymbol{n} \rangle_{\Gamma_c} = 0, \quad \text{on } \Gamma \cap \partial\Omega, \end{aligned} \tag{19}$$

where $\hat{\boldsymbol{F}} \cdot \boldsymbol{n}_{\eta(c,+)}$ and $\hat{\boldsymbol{F}} \cdot \boldsymbol{n}_{\eta(c,-)}$ denote the numerical fluxes calculated from the left and right triangles, respectively, and $\hat{\boldsymbol{G}} \cdot \boldsymbol{n}$ is the flux defined over the boundary $\partial\Omega$ flowing into the computational domain. By inserting the definition of the numerical flux (16), we obtain the following matrix system for the global problem:

$$\begin{aligned} &\hat{\mathbf{B}}^{c,j} \hat{\mathbf{H}}^{c,j} = \mathbf{B}^{\eta(c,+),j} \mathbf{H}^{\eta(c,+),j} + \mathbf{B}^{\eta(c,-),j} \mathbf{H}^{\eta(c,-),j}, \quad \text{on } \Gamma \backslash \partial\Omega, \\ &\hat{\mathbf{B}}^{c,j} \hat{\mathbf{H}}^{c,j} = \mathbf{B}^{\eta(c,+),j} \mathbf{H}^{\eta(c,+),j} + \hat{\mathbf{S}}^{c,j}, \quad \text{on } \Gamma \cap \partial\Omega, \end{aligned} \tag{20}$$

where $\hat{\mathbf{H}}^{c,j}$ is the vector of degrees of freedom of $\hat{h}^j$ on Γ_c. Other coefficient matrices are given in the Appendix A.

After eliminating the unknowns $\mathbf{H}^{i,j}$ with Eq. (18) and assembling Eq. (20) over all the faces, the global problem becomes:

$$\mathbb{K}^j \hat{\mathbf{H}}^j = \mathbb{R}^j, \tag{21}$$

where $\hat{\mathbf{H}}^j$ is the vector of degrees of freedom of $\hat{h}^j$ on all the faces Γ, $\mathbb{K}^j$ is the global matrix of the linear system of equations. Once the values of $\hat{h}^j$ are obtained, an element-by-element reconstruction of the approximation of h^j is implemented according to Eq. (18).

3.1.2. Strategy to solve the large sparse linear system

It is noted that the linear system (21) is highly sparse, in which only face unknowns that involve two adjacent triangles are coupled at each row. Compared to a standard linear DG system, the trace system is much smaller and sparser when $k > 1$ [53]. However, for large-scale applications, the trace system is still the major bottleneck for efficient computation, so linear iterative solvers might be suitable [54,55].

However, when solving the gas kinetic equation, we notice that the global matrices $\mathbb{K}^j$ remain unchanged during all iterations. Thus, in this paper we still employ the direct solver for the linear systems (21). First, we assemble the trace systems for all discrete velocities into a large one as:

$$\begin{bmatrix} \mathbb{K}^1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbb{K}^j & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \mathbb{K}^{M_v} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{H}}^1 \\ \vdots \\ \hat{\mathbf{H}}^j \\ \vdots \\ \hat{\mathbf{H}}^{M_v} \end{bmatrix} = \begin{bmatrix} \mathbb{R}^1 \\ \vdots \\ \mathbb{R}^j \\ \vdots \\ \mathbb{R}^{M_v} \end{bmatrix}. \tag{22}$$

Second, the *LU*-decomposition is completed before iteration. Third, at each iterative step, the traces are obtained by direct substitution. Both the *LU*-decomposition and substitution phases are executed by calling the large-sparse linear solver PARDISO [56]. Note that the computational cost in terms of the CPU time listed in Sec. 4 only counts the computational time used in iterations, that is, the time to set up and factorize the global system is not included, which is negligible compared to the overall iteration time.

3.1.3. Implementation of boundary condition

Before describing the implementation of the boundary condition, we take an insight into the form of the numerical flux. If inserting the expression of flux (16) into the continuity equation (19) at interior faces, we immediately have:

$$\langle \psi, \hat{h}^j \rangle = \frac{1}{2} \langle \psi, h^j_{\eta(c,+)} + h^j_{\eta(c,-)} \rangle. \tag{23}$$

That is, the trace $\hat{h}^j$ at the interior face is equal, in a weak sense, to the average of $h^j_{\eta(c,+)}$ and $h^j_{\eta(c,-)}$, which are evaluated at the interface from the left and right triangles, respectively. Then we obtain an equivalent expression for $\hat{\boldsymbol{F}} \cdot \boldsymbol{n}$:

$$\hat{\boldsymbol{F}} \cdot \boldsymbol{n}_{\eta(c,\pm)} = \begin{cases} \boldsymbol{v}^j \cdot \boldsymbol{n}_{\eta(c,\pm)} h^j_{\eta(c,\pm)}, & \boldsymbol{v}^j \cdot \boldsymbol{n}_{\eta(c,\pm)} \geq 0 \\ \boldsymbol{v}^j \cdot \boldsymbol{n}_{\eta(c,\pm)} h^j_{\eta(c,\mp)}, & \boldsymbol{v}^j \cdot \boldsymbol{n}_{\eta(c,\pm)} < 0 \end{cases}, \tag{24}$$

which is exactly the upwind scheme.

The flux $\hat{\boldsymbol{G}} \cdot \boldsymbol{n}$ needs to be specified at the boundary $\partial\Omega$ to complete the formulation. To be consistent with the evaluation of fluxes at interior faces, we calculate the boundary flux as:

$$\hat{\boldsymbol{G}} \cdot \boldsymbol{n} = \boldsymbol{v}^j \cdot \boldsymbol{n} \hat{h}^j + \alpha \left(g^j - \hat{h}^j \right), \tag{25}$$

where g^j is the boundary value of h^j and $\boldsymbol{n}$ is the outward unit normal vector at the boundary pointing into the flow field. In this paper, the fully diffuse boundary condition is used to determine the perturbed VDF g^j at the solid surface. Suppose the solid wall is static and has the temperature T_0, the perturbed VDF for the reflected molecules at the wall (i.e., when $\boldsymbol{v}^j \cdot \boldsymbol{n} > 0$) is given by $g^j = -2\sqrt{\pi} \sum_{\boldsymbol{v}^j \cdot \boldsymbol{n} < 0} (\boldsymbol{v}^j \cdot \boldsymbol{n}) f^j_{\mathrm{eq}} h^j \varpi^j$, which is always zero in this specific problem due to the symmetry condition $h(v_3) = -h(-v_3)$ [10].

Other types of boundary conditions, such as the diffuse-specular boundary condition with tangential momentum accommodation coefficient less than one, symmetry/periodic boundaries, as well as pressure inlet/outlet boundary could be incorporated straightforwardly [40,30].

3.2. HDG for the synthetic equation

The HDG method for solving the diffusion equation (11) has been well developed [45,52], in which two auxiliary variables are introduced to approximate the derivatives of u_3, thus the HDG approximation is synchronously taken for the flow velocity u_3, its derivatives ∇u_3, and its trace $\hat{u}_3$. Here, we skip the details of the scheme, and discuss several modifications tailored for the current problem.

First, since the second-order partial derivatives of the high-order moments appear in the equation (11), we rewrite the equation into a first-order system in the form as:

$$\begin{aligned} &\nabla \cdot \boldsymbol{q} = X_{\mathrm{P}} \delta, \\ &\boldsymbol{q} + \nabla u_3 + \boldsymbol{r} = \boldsymbol{0}, \end{aligned} \tag{26}$$

where the vector $\boldsymbol{r}$ is

$$\boldsymbol{r} = \frac{1}{4} \left[\frac{\partial F_{2,0,1}}{\partial x_1} + \frac{\partial F_{1,1,1}}{\partial x_2}, \frac{\partial F_{1,1,1}}{\partial x_1} + \frac{\partial F_{0,2,1}}{\partial x_2} \right]^T. \tag{27}$$

That is, the auxiliary variable $\boldsymbol{q}$ is introduced to approximate the combination of the derivatives of u_3 and high-order moments, which guarantees the stability and local solvability. Then, the flow velocity u_3 and its trace $\hat{u}_3$, as well as the vector $\boldsymbol{q}$ are discretized in the piecewise finite element spaces $\mathcal{V} \times \mathcal{W} \times [\mathcal{V}]^2$.

Second, to specify the boundary condition of $\hat{u}_3$, we evaluate it from the perturbed VDF as:

$$\langle \psi, \hat{u}_3 \rangle_{\Gamma_c} = \langle \psi, \sum_j^N v_3^{j_3} f^j_{\mathrm{eq}} h^j \varpi^j \rangle_{\Gamma_c}, \quad \text{on } \Gamma \cap \partial\Omega. \tag{28}$$

This could guarantee the proper value of the flow velocity at the boundary, especially when the slip velocity at the solid surface is large for highly rarefied gas flows.

The procedure of the SIS for the linearized BGK equation is described as follows:

- When $h^{j,(t)}$ and $u_3^{(t)}$ are known at the t-th iteration step, calculate the VDF $h^{j,(t+1)}$ at $(t+1)$-th step by solving the conventional iteration scheme (9), with the right-hand side given by $2\delta u_3 v_3^{j_3} + X_{\mathrm{P}} v_3^{j_3}$. Note that the variations in density and temperature are zero in this specific flow;
- From $h^{j,(t+1)}$, calculate the vector $\boldsymbol{r}$, i.e. the high-order moments $F_{2,0,1}$, $F_{1,1,1}$ and $F_{0,2,1}$ in Eq. (27);

- From $h^{j,(t+1)}$, calculate the flow velocity trace $\hat{u}_3^{(t+1)}$ at boundary, see Eq. (28);
- Calculate $u_3^{(t+1)}$ by solving the synthetic diffusion equation (26), with the boundary condition obtained from the previous step.

The above iterative procedure is continued until the steady-state is reached. For the following calculation, the stabilization parameter appears in the expression of the numerical flux for $\hat{\boldsymbol{q}} \cdot \boldsymbol{n}$ (Eq. (8) in Ref. [52]) is set to be 1.

4. Results and discussions

The HDG method of k up to 4 is applied to solve the linearized BGK kinetic model equation (9) in parallel with the synthetic diffusion equation (11). The convergence criterion for the iterative procedure described in Sec. 3.2 is that the global relative residual in flow velocity between two successive iteration steps is less than 10^{-5}. The residual is defined as

$$\mathcal{R} = \frac{|\int u_3^{(t+1)} - u_3^{(t)} \mathrm{d}\Omega|}{|\int u_3^{(t)} \mathrm{d}\Omega|}. \tag{29}$$

In addition to the profiles of flow velocity, we are interested in the property of dimensionless mass flow rate (MFR):

$$M = \frac{1}{H^2} \int u_3 \mathrm{d}\Omega. \tag{30}$$

To assess the accuracy and efficiency of the proposed scheme, our numerical results are compared with the discrete UGKS (DUGKS) solutions, which have been verified from the continuum to free-molecular flow regimes [22], or available data from literature. In the four test cases below, the convergence tests in terms of the discrete velocities are performed first to determine the number of points in the molecular velocity space: the convergence is said to be reached if further refinement would only improve the solutions by a magnitude no more than 0.5%. The entire tests are done in double precision on a workstation with Intel Xeon-E5-2680 processors and 132 GB RAM. During iteration, we call the relative routines in Intel Math Kernel Library (MKL) to invert the matrix. Moreover, to solve the HDG global equation, we call the direct sparse solver, PARDISO. The first two tests are done on single processor, and the internal parallelism for MKL functions is not activated. The last two simulations are run on multiple processors using OpenMP.

4.1. Fast convergence of the SIS: Poiseuille flow between two parallel plates

Poiseuille flow between two parallel plates with a distance of H is used to assess the accuracy and fast convergence of the proposed HDG solver. The one-dimensional flow is resolved on a two-dimensional domain of $\Omega = [0, 0.5] \times [0, 1.0]$ with 4 uniform isosceles right triangles being set along the direction perpendicular to the plates, say, the x_2 direction. Therefore, the height of each triangle is equal to 0.354, which is larger than the mean free path when $Kn < 0.354$ or equivalently $\delta > 2.50$.

The MFR at different rarefaction parameter δ, obtained from the SIS with $k = 3$, is illustrated in Fig. 2(a) and compared with those from the DUGKS and the CIS. In the CIS, only the linearized BGK model equation (9) is solved. 24 non-uniform points are employed with a truncation of $[-4, 4]$ in each direction to discretize the molecular velocity space. Simulation parameters including the numbers of grid points employed in the DUGKS could be found in Ref. [22]. It is shown that when δ increases, the MFR first drops to the minimum value at $\delta \sim 1$ and then rapidly increases. The Knudsen minimum in the mass flow rate is due to the competition of two effects: when Kn increases, the slip velocity at the plates becomes larger, while the velocity profile becomes flatter [10]. The SIS could obtain MFRs with high accuracy on such a coarse grid over a wide range of flow regimes. The relative L_2 errors of the SIS results to the ones of the DUGKS are within 1.1%. However, the CIS results possess obvious errors when $\delta \gtrsim 150$. For example, the MFR from the CIS is about 61.7% smaller than that of the DUGKS at $\delta = 886.2$. This is due to the fact that the spatial resolution is too low such that the numerical viscosity is not negligible in comparison with the physical viscosity of the gas in the CIS, while in the SIS the macroscopic diffusion equation (11) is solved with the physical viscosity.

Another superiority of the SIS over the CIS is immediately seen from Fig. 2(b), which shows the iteration steps to reach the steady-state solution for both CIS and SIS. When the CIS is used, the number of iteration steps increases rapidly with the rarefaction parameter in the near-continuum flow regime ($\delta \geq 10$), whereas those of the SIS only increases slightly. In the late transition flow regime ($\delta < 1$), however, the numbers of iterative steps are almost the same for both schemes. This is further confirmed in Table 1, where the relative L_2 error of MFRs (the DUGKS results are used as the reference solutions), the number of iteration steps, and the total CPU time are listed for various rarefaction parameters δ and degrees of approximation polynomials in the HDG method. It is interesting to note that with the same number of triangles, the number of iterative steps of the CIS reaches a constant value as the degree of polynomials in the HDG discretization increases. While at large δ, the number of iterative steps of the SIS drops when higher-order approximation polynomials are employed. Compared to the kinetic equation, the time to solving the fluid-dynamic equation (11) is negligible, so the CPU time saving is proportional to the reduction of iteration steps. Therefore, the SIS needs significantly less time to reach converged solutions than the CIS. At $\delta = 8.862$, the SIS with $k = 4$ is 10 times faster than the CIS, while at $\delta = 88.62$ it is nearly 140 times faster.

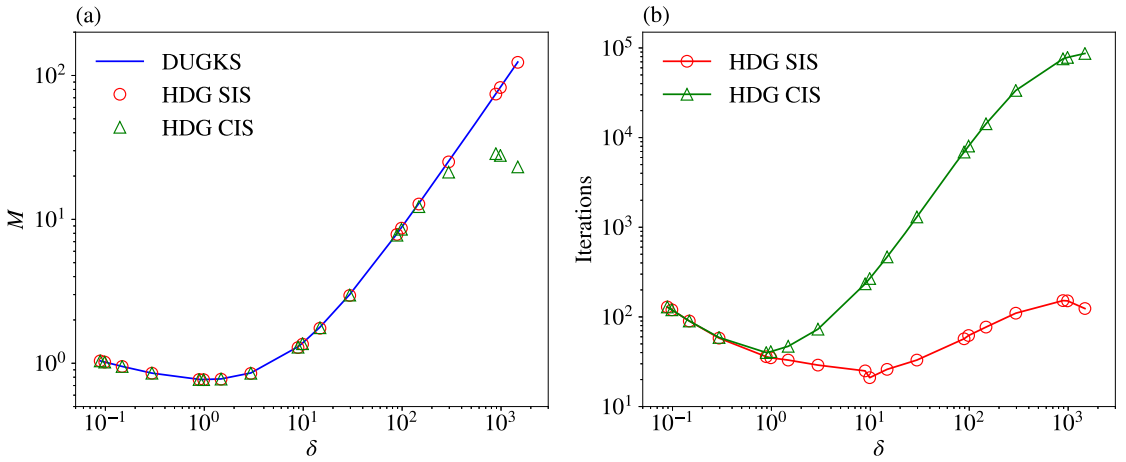

Fig. 2. Poiseuille flow between two parallel plates over a wide range of gas rarefaction. (a) Mass flow rates; (b) Number of iteration steps.

Table 1
Comparisons between the CIS (only the linearized BGK equation is solved) and SIS (the linearized BGK equation is solved in parallel with the diffusion-like equation for u_3) in terms of the accuracy, the number of iterations (Itr denotes the number of iteration steps to reach the convergence criterion $\mathcal{R} < 10^{-5}$), and the CPU time t_c. Poiseuille flow between two parallel plates is considered.

δ	k	CIS			SIS		
		L_2 error	Itr	t_c, [s]	L_2 error	Itr	t_c, [s]
88.62	1	2.17×10^{-1}	6115	253.8	3.96×10^{0}	213	9.2
	2	2.43×10^{-2}	6876	459.3	9.70×10^{-3}	87	5.9
	3	2.42×10^{-2}	6886	622.6	9.73×10^{-3}	58	5.6
	4	2.42×10^{-2}	6886	956.0	9.35×10^{-3}	45	6.9
8.862	1	6.78×10^{-2}	224	10.1	3.96×10^{-1}	45	1.9
	2	7.80×10^{-3}	234	16.1	2.10×10^{-2}	30	2.0
	3	7.21×10^{-3}	234	22.3	1.35×10^{-2}	25	2.4
	4	7.01×10^{-3}	234	34.7	1.01×10^{-2}	23	3.5
0.8862	1	7.65×10^{-3}	40	1.8	1.90×10^{-3}	36	1.5
	2	2.04×10^{-3}	40	2.8	4.21×10^{-3}	36	2.4
	3	2.00×10^{-3}	40	3.8	2.51×10^{-3}	36	3.3
	4	1.99×10^{-3}	40	5.9	2.17×10^{-3}	36	5.3
0.08862	1	1.92×10^{-3}	129	5.8	2.12×10^{-3}	129	5.2
	2	8.94×10^{-4}	130	8.7	9.37×10^{-4}	129	8.3
	3	9.14×10^{-4}	129	11.8	9.20×10^{-4}	129	11.8
	4	9.13×10^{-4}	129	18.5	9.14×10^{-4}	129	18.9

To illustrate how the SIS works in the near-continuum flow regime, convergence histories of the SIS and CIS are plotted in Fig. 3 when $\delta = 88.62$. Staring from the zero disturbance, the flow velocity gradually increases from zero due to the gas-gas and gas-surface collisions. From Fig. 3(a) we see that, near the wall the flow velocity quickly approaches the converged value, while the velocity in the bulk adjusts rather slowly. As a result, a large number of iterations is required in the CIS to promote the flow velocity reaching to the maximum value. However, this situation is completely changed in the SIS, where the macroscopic diffusion equation (11) quickly generates the parabolic velocity profile (the second-order derivative $\partial^2 u_3/\partial x_2^2$ is very close to $-\delta$) in the bulk, which boots the convergence significantly. From Fig. 3(b) it is found that the velocity profile of the SIS is already very close to the final converged solution, even after two iterations.

4.2. Comparison of HDG, FDM and DG: flow along a channel of square cross-section

The performance of the HDG-SIS is now assessed in the Poiseuille flow along a channel with a square cross-section of height H, by comparing with solutions obtained from the same SIS but with the second-order FDM for the approximation of spatial derivatives [12]. The flow is resolved on a domain of $\Omega = [0, 1] \times [0, 1]$. As shown in Fig. 4(a), the computational domain is partitioned with uniform triangles. For the discretization of the velocity space, $24 \times 24 \times 24$ non-uniform points are used with a truncation of $[-4, 4]$ in each direction. The typical flow velocity contours obtained by the HDG-SIS at $\delta = 100$, 10 and 1 are shown in Fig. 4(b)–(d), respectively. It is observed that the maximum velocity emerges in the center of the flow field. As the rarefaction parameter δ decreases from 100 to 1, the maximum velocity reduces while the slip velocity in the vicinity of solid surfaces increases.

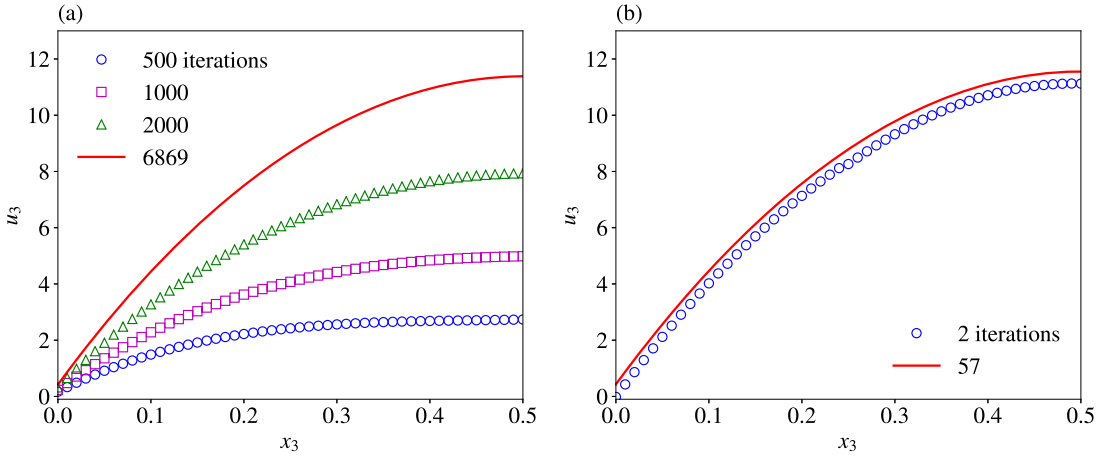

Fig. 3. Comparison of the CIS (a) and SIS (b) in terms of the convergence history for the velocity in Poiseuille flow between two parallel plates. The rarefaction parameter is $\delta = 88.62$, and the order of HDG is $k = 3$. Red lines are the converged result. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

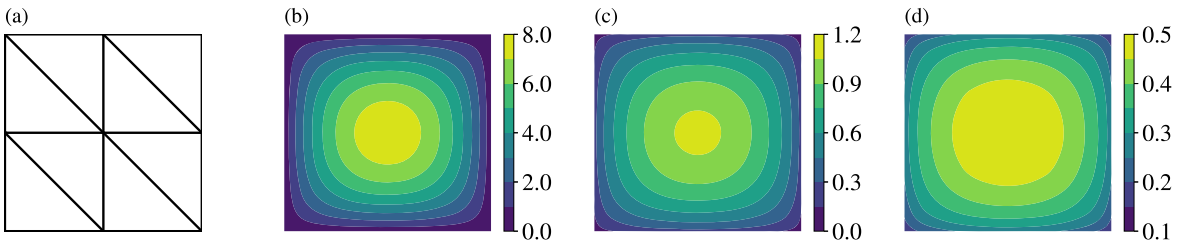

Fig. 4. Poiseuille flows along a channel of square cross-section: (a) geometry and mesh; (b) u_3 contour at $\delta = 100$ with $M_{el} = 50$, $k = 4$; (c) u_3 contour at $\delta = 10$ with $M_{el} = 50$, $k = 4$; (d) u_3 contour at $\delta = 1$ with $M_{el} = 18$, $k = 4$.

Table 2
Poiseuille flow along a channel with square cross-section solved by the HDG-SIS. Itr denotes the number of iteration steps to satisfy the convergence criterion $\mathcal{R} < 10^{-5}$, and t_c is the CPU time.

k	$\delta = 100$				$\delta = 10$				$\delta = 1$			
	M_{el}	L_2 error	Itr	t_c, [s]	M_{el}	L_2 error	Itr	t_c, [s]	M_{el}	L_2 error	Itr	t_c, [s]
1	8	4.32×10^{0}	164	13.0	2	1.90×10^{0}	51	1.2	2	1.26×10^{-3}	14	0.3
	18	1.83×10^{0}	128	23.5	8	4.94×10^{-1}	37	3.0	8	8.01×10^{-3}	14	1.2
	32	9.66×10^{-1}	109	35.4	18	2.03×10^{-1}	32	5.7	18	3.40×10^{-3}	13	2.4
	50	5.73×10^{-1}	94	48.8	32	9.85×10^{-2}	29	8.8	32	9.07×10^{-4}	13	4.1
2	8	1.41×10^{-1}	81	10.6	2	1.17×10^{-1}	30	1.1	2	6.90×10^{-3}	13	0.5
	18	4.44×10^{-2}	69	21.2	8	2.90×10^{-3}	26	3.5	8	1.46×10^{-3}	13	1.7
	32	1.35×10^{-2}	58	32.0	18	8.52×10^{-3}	24	6.9	18	1.71×10^{-3}	13	3.8
	50	2.98×10^{-3}	50	43.8	32	9.29×10^{-3}	22	11.7	32	1.66×10^{-3}	13	7.0
3	8	1.78×10^{-2}	64	12.2	2	2.71×10^{-2}	28	1.5	2	3.33×10^{-3}	13	0.7
	18	8.56×10^{-3}	49	21.4	8	1.23×10^{-2}	23	4.4	8	1.66×10^{-3}	13	2.5
	32	6.49×10^{-3}	40	32.8	18	7.84×10^{-3}	21	9.1	18	7.84×10^{-4}	13	5.7
	50	5.59×10^{-3}	35	43.6	32	5.74×10^{-3}	21	16.3	32	4.69×10^{-4}	13	10.4
4	8	8.49×10^{-3}	48	14.9	2	1.30×10^{-2}	23	2.0	2	1.33×10^{-3}	13	1.2
	18	6.01×10^{-3}	37	26.6	8	7.16×10^{-3}	21	6.5	8	6.98×10^{-4}	13	4.1
	32	5.12×10^{-3}	31	40.7	18	4.92×10^{-3}	21	15.1	18	5.72×10^{-4}	13	9.3
	50	4.56×10^{-3}	27	54.4	32	3.98×10^{-3}	21	26.2	32	4.92×10^{-4}	13	16.2

For the HDG-SIS, the L_2 errors of the MFR, the numbers of iterative steps, and the CPU time to obtain the converged solutions are listed in Table 2, for various numbers of triangles and degrees of approximation polynomials. The results obtained by the FDM-SIS are also listed in Table 3, where M_p denotes the number of equally-distributed discrete points in the spatial space. The L_2 errors are calculated using the DUGKS results as reference. For the DUKGS simulations, the same discrete velocity grid as that in the SIS is employed, while 48×48 and 72×72 points are located in the spatial space for cases with $\delta < 10$ and $\delta \geq 10$, respectively.

Table 3
Poiseuille flow along a channel of square cross-section solved by the FDM-SIS. M_p is the number of discrete points in the spatial space, Itr is the number of iteration steps to satisfy the convergence criterion $\mathcal{R} < 10^{-5}$, and t_c is the CPU time.

M_p	$\delta = 100$			$\delta = 10$			$\delta = 1$		
	L_2 error	Itr	t_c, [s]	L_2 error	Itr	t_c, [s]	L_2 error	Itr	t_c, [s]
9^2	2.22×10^{-1}	310	4.4	1.62×10^{-1}	51	1.0	7.76×10^{-2}	13	0.3
15^2	8.66×10^{-2}	188	7.5	6.69×10^{-2}	38	1.9	3.11×10^{-2}	13	0.8
25^2	3.49×10^{-2}	119	14.8	2.83×10^{-2}	30	4.7	1.26×10^{-2}	13	1.9
35^2	1.99×10^{-2}	91	28.0	1.63×10^{-2}	29	11.5	7.03×10^{-3}	13	4.9
45^2	1.36×10^{-2}	75	65.5	1.10×10^{-2}	32	32.6	4.59×10^{-3}	13	11.5
55^2	9.36×10^{-3}	63	138.0	7.75×10^{-3}	26	79.5	2.75×10^{-3}	13	27.1

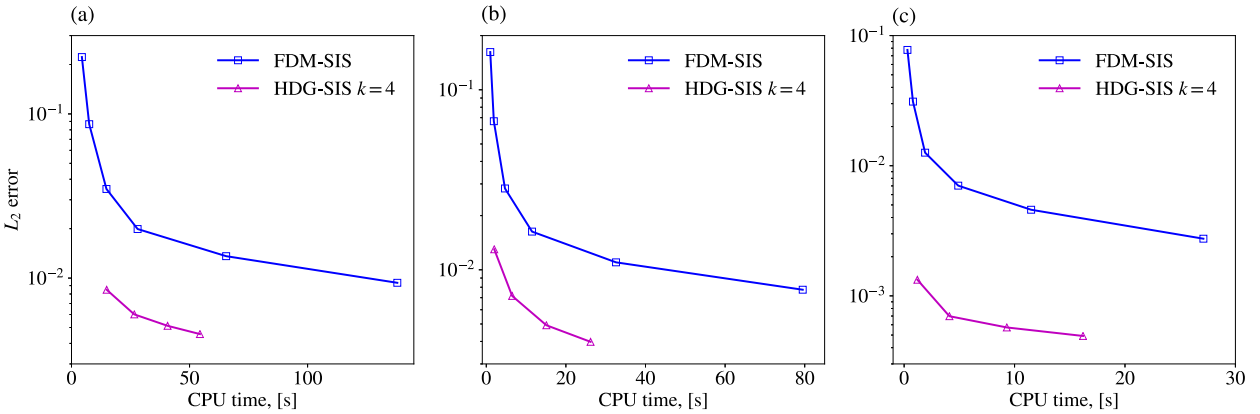

Fig. 5. Convergence of the MFR in the Poiseuille flow along a channel of square cross-section: the comparison between FDM-SIS and the HDG-SIS with the order $k = 4$ when (a) $\delta = 100$, (b) $\delta = 10$, and (c) $\delta = 1$.

It is found from Table 2 that for the spatial grids with the same number of triangles, the HDG-SIS solutions with higher-order accuracy are obtained with higher-order approximation polynomials. Therefore, to achieve the same order of accuracy, solvers with higher-order polynomials require spatial grids with fewer triangles. For example, when $\delta = 100$, the solver with 3^{rd}-order polynomials has an error of about 0.8% in the MFR using 18 triangles, while the one with 4^{th}-order polynomials reaches this accuracy with only 8 triangles. Moreover, as the rarefaction parameter decreases, fewer triangles are needed to obtain high-accuracy results. As far as the convergence speed is concerned, for all the rarefaction levels, solvers with different degrees of polynomials require almost the same number of iterations to obtain solutions with the same order of accuracy. For example, when $\delta = 100$, about 50 iterations are required to obtain MFR with L_2 error less than 1%. Since fewer triangles are needed, the higher order the solver, the less the CPU time. At $\delta = 100$, the CPU time to obtain solution with an error of $\sim 0.8\%$ with $k = 4$ is about 70% of that with $k = 3$.

For the comparison of the HDG-SIS and the second-order FDM-SIS, we find that the HDG discretization is much more efficient. At $\delta = 100$, the FDM-SIS predicts the MFR with an error less than 1% on the spatial grid with 55×55 discrete points, while the HDG-SIS obtains the solution with the same order of accuracy on 50, 18 and 8 triangles for $k = 2$, 3 and 4 solvers, respectively. Meanwhile, at $\delta = 1$, the FDM-SIS obtains the MFR with an error less than 1% on 35×35 points, while the HDG-SIS obtains the solution only on 2 triangles for all solvers. The HDG solver of $k = 4$ could be 9.3 times (more than 12 times) faster than the FDM solver to obtain converged results at $\delta = 100$ ($\delta = 10$). Fig. 5 shows the convergence history of the MFR in terms of the CPU time for the FDM-SIS and HDG-SIS with $k = 4$. It is clear that the HDG method is significantly more efficient especially for highly rarefied flows. Although higher-order FDM could achieve better efficiency, it needs much more computational effort since stencils involving a large number of points are required in the FDM scheme, which is extremely difficult to be implemented for complex geometries.

It is also interesting to compare the performance of the implicit HDG solver and the original DG method. The comparison is based on the CIS scheme for the flow at $\delta = 10$. Here, we consider two different DG methods, one is based on the implicit iterative scheme like the one used in the current HDG solver, the other is the explicit Runge–Kutta DG (RKDG) gas kinetic solver [40]. The detailed formulation of the implicit DG (IDG) scheme is presented in the Appendix B. The number of degree of freedom N_{dof}, the L_2 errors of the MFR, the numbers of iterative steps, iterative time interval Δt in the RKDG solver, and the CPU time are listed in Table 4 for the solvers with different approximating polynomials. Our numerical results show that the implicit HDG and DG with the same order on the same mesh yield the same solution (MFRs have at least 7 same significant digits) using the same number of iterative steps, since the numerical flux in the HDG is equivalent to the upwind scheme used in the IDG. Note that N_{dof} refers to the number of degrees of freedom that appear independently in each scheme. For the HDG solver, it is calculated based on the trace unknowns as $N_{dof} = M_v M_{fc}(k+1)$, while for the IDG or RKDG, it is based on the field unknowns as $N_{dof} = M_v M_{el}(k+1)(k+2)/2$. For the structured triangular mesh used, the

Table 4
Comparison between the HDG-CIS, IDG-CIS and RKDG-CIS for Poiseuille flow along a channel with square cross-section at $\delta = 10$, where Itr is the number of iteration steps to satisfy the convergence criterion $\mathcal{R} < 10^{-5}$, N_{dof} denotes the number of degree of freedom involving in each method, Δt is the iterative time interval and t_c is the CPU time. Note that Itr and L_2 error in the HDG-CIS and IDG-CIS are exactly the same, while N_{dof} for IDG-CIS and RKDG-CIS is the same.

k	M_{el}	HDG-CIS				IDG-CIS		RKDG-CIS			
		$N_{\text{dof}}/M_{\text{v}}$	L_2 error	Itr	t_c, [s]	$N_{\text{dof}}/M_{\text{v}}$	t_c, [s]	L_2 error	Δt, [s]	Itr	t_c, [s]
1	8	32	9.92×10^{-2}	131	10.8	24	4.9	9.97×10^{-2}	5.30×10^{-5}	529	17.2
	32	112	2.32×10^{-2}	139	45.0	96	21.4	2.46×10^{-2}	2.65×10^{-5}	1015	133.5
	72	240	1.02×10^{-2}	141	160.0	216	54.5	1.24×10^{-2}	1.77×10^{-5}	1438	446.3
	128	416	6.22×10^{-3}	141	180.2	384	116.6	9.22×10^{-3}	1.33×10^{-5}	1830	1004.6
2	8	48	6.62×10^{-3}	141	18.0	48	11.6	7.65×10^{-3}	3.54×10^{-5}	804	122.0
	32	168	2.93×10^{-3}	141	71.5	192	52.5	5.16×10^{-3}	1.77×10^{-5}	1444	931.2
	72	360	2.51×10^{-3}	141	169.7	432	140.3	5.93×10^{-3}	1.18×10^{-5}	2020	2950.3
	128	624	2.39×10^{-3}	141	316.1	768	276.6	6.99×10^{-3}	8.84×10^{-6}	2555	6430.1
3	8	64	2.84×10^{-3}	141	24.2	80	23.2				
	32	224	2.39×10^{-3}	141	97.8	320	110.2		–		
	72	480	2.30×10^{-3}	141	240.9	720	284.5				
	128	832	2.28×10^{-3}	141	465.8	1280	523.5				
4	8	80	2.44×10^{-3}	141	37.7	120	44.0				
	32	280	2.29×10^{-3}	141	152.4	480	216.1		–		
	72	600	2.27×10^{-3}	141	381.2	1080	545.0				
	128	1040	2.26×10^{-3}	141	735.1	1920	893.8				

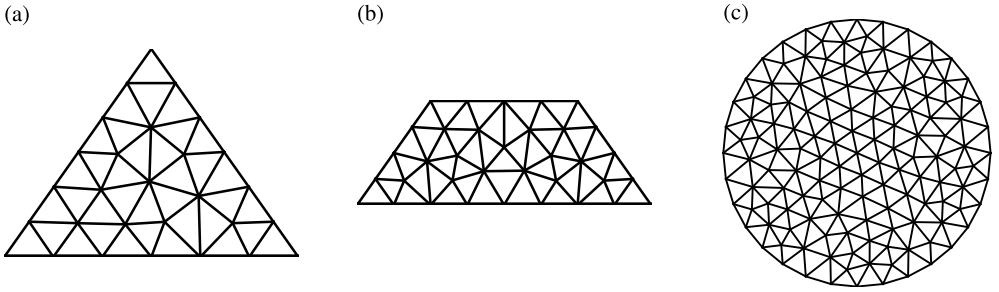

Fig. 6. Schematics of the geometry and spatial meshes for Poiseuille flows along channels of triangular, trapezoidal, and circular cross-sections.

number of faces M_{fc} is around 1.7 times the number of triangles M_{el}. Therefore, it is found that N_{dof} in HDG is smaller than the one in the original DG scheme when $k \geq 2$. The higher order and more triangles, the more significant this difference will be. Actually, N_{dof} of the HDG becomes closer to that of the continuous higher-order FEM [53]. As a consequence, the HDG method costs less CPU time than the IDG method when $k > 2$. Although N_{dof} of the HDG with $k = 2$ is smaller than that of the IDG, it is not more efficient. This is due to the fact that extra time is required to recover field solution from trace solution in the HDG. Another advantage of the HDG due to reduction in number of coupled degrees of freedom is to promote further parallelism.

Compared to the HDG or IDG method, the RKDG solver needs more CPU time to obtain converged solutions. When the order of polynomials and the number of triangular elements increase, the number of iterative steps in the HDG or IDG solver approaches a fixed value (around 141 for the current case). For the RKDG solver, the iteration number increases substantially, as its time interval is restricted by the Counrant–Firedirchs–Lewy condition, i.e. $|\boldsymbol{v}^j|_{\max} \Delta t \leq C H_{\min}$, where $H_{\min}$ is the minimum altitude of triangles. We have set the constant $C = 0.3$ for $k = 1$ and 0.2 for $k = 2$, respectively. Therefore, the implicit solvers are more efficient. To obtain the solution with L_2 error less than 1% on 128 triangles, the HDG solver with $k = 1$ could be 10 times faster than the RKDG solver.

4.3. Accuracy of the SIS: flows along the channels of various cross-sections

We further assess the accuracy of HDG-SIS by solving the Poiseuille flows along the channels of triangular, trapezoidal, and circular cross-sections, and comparing the MFRs to existing data. The geometries and meshes are illustrated in Fig. 6. The isosceles triangular and trapezoidal cross-sections have an acute angle of $\theta = 54.74°$, and the ratio of the small and large base in the trapezoid is equal to 0.5. Totally 36, 118 and 240 triangles are used for the HDG solver with $k = 3$. The molecular velocity space is discretized in the range of $[-4, 4]$ by 32 non-uniform points in each direction. The characteristic length H for the flow in the triangular and trapezoidal channels is set as its hydraulic diameter, i.e. 4 times the ratio of area and perimeter. For the circular channel, the radius is chosen as the characteristic length.

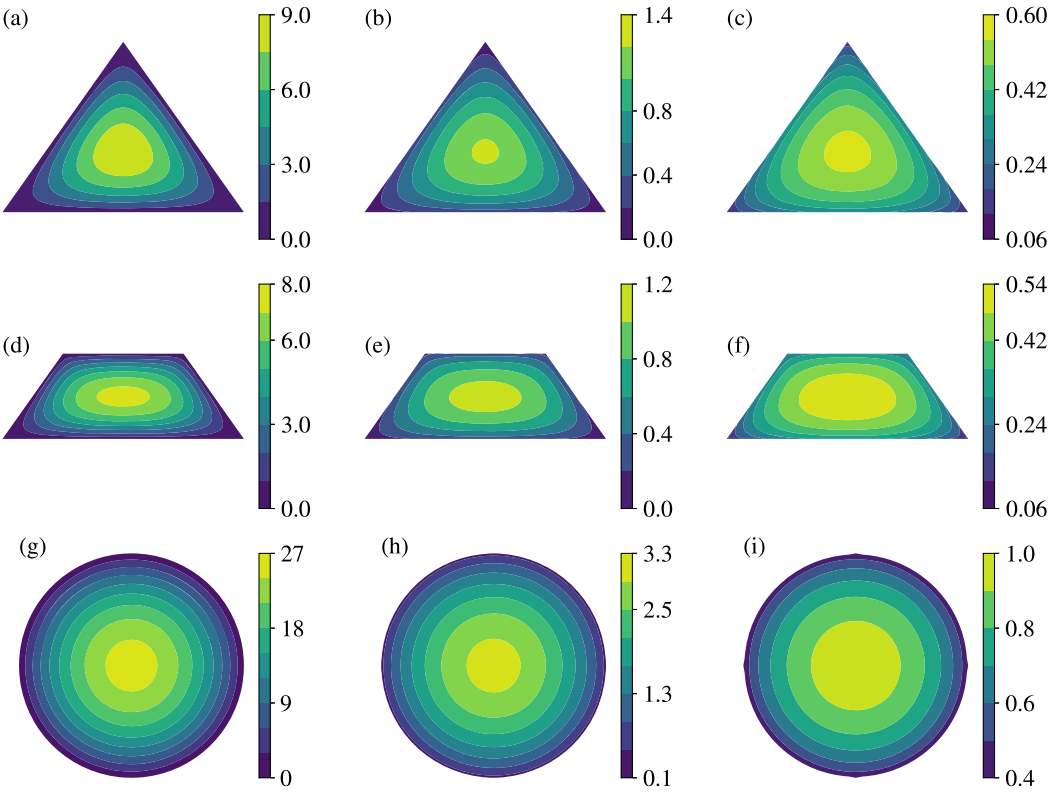

Fig. 7. Velocity contours in the Poiseuille flows along the channels of triangle, trapezoid, and circle cross-sections. The rarefaction parameters in the first, second, and third columns are $\delta = 100$, 10, and 1, respectively.

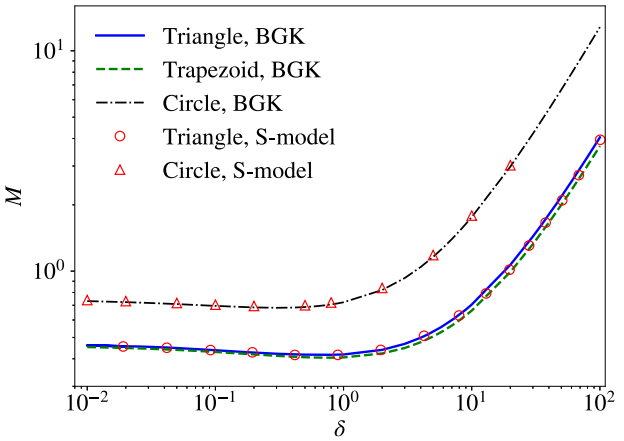

Fig. 8. MFRs of the Poiseuille flow along the channels of triangle, trapezoid, and circle cross-sections. The lines are solutions from the HDG-SIS solver based on the linearized BGK kinetic model, the symbols are solutions from the FDM solver based on the linearized Shakhov kinetic model (S-model) [57,58].

The velocity contours at $\delta = 100$, 10, and 1 are shown in Fig. 7. Similar to flows in the square channel, the maximum velocities appear in the center of the flow field, which decrease as the rarefaction parameter decreases. MFRs over a wide range of gas rarefaction are plotted in Fig. 8. It is found that MFRs for the triangular and trapezoidal channels are close to each other due to the fact that the hydraulic diameter is chosen as the characteristic length to non-dimensionalize the problem. If using the radius as the characteristic length, the MFR in the circular channel is larger than those in the other two channels. The Knudsen minimum again arises at $\delta \sim 1$. In all cases, the HDG-SIS results agree well with those in the literature [57,58], which demonstrates the accuracy of the proposed HDG-SIS scheme. It is worth to mention that the results in the literature were calculated from the linearized Shakhov kinetic model equation, where the additional correction of the heat flux in the collision operator has no effect on the MFR in this problem.

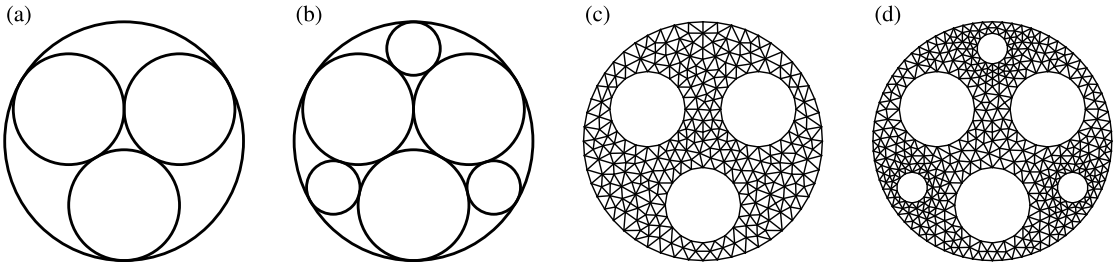

Fig. 9. (a)–(b): Schematics of the geometry for the original Level-1 and Level-2 Apollonian gaskets. (c)–(d): Schematics of the geometry and mesh for the Poiseuille flows along the Level-1 and Level-2 Apollonian gasket channels.

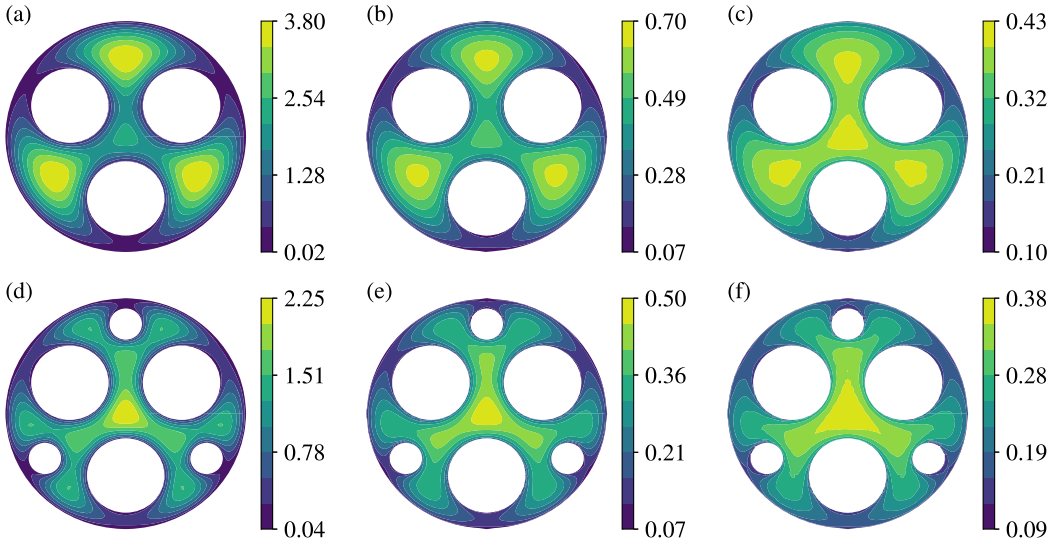

Fig. 10. Velocity contours in the Poiseuille flows along the Apollonian gasket channels. The rarefaction parameters in the first, second, and third columns are $\delta = 100$, 10, and 1, respectively.

4.4. Capability to handle complex geometry: flows along Apollonian gasket channels

Finally, the HDG-SIS is applied to calculate the Poiseuille flows along the channels with cross-section described by the Apollonian fractal gasket, to demonstrate its capability of handling complex geometries. The cross-section of the original Apollonian fractal gasket is a fractal generated starting from a circle, which is filled by three circles with the same radius, each is tangent to the others (including the internal tangent with the outer circle, see Fig. 9(a)). Then for the next level, the structure is filled with three more circles, each is tangent to another three, see Fig. 9(b). Here, for the geometry we calculated, the inner circles are not tangent to anyone of the others, while their centers coincide to those in the original Apollonian gaskets and their radii are determined such that the porosity (the ratio of void area) is 0.7 for the first level and on this basis, the porosity of the second level is 0.65. The resulting geometries and meshes of the Level-1 and Level-2 structures are presented in Fig. 9(c)–(d). In the current simulation, we treat the inner small circles as solids and the fluid flowing through the gaps between the outer circle and the inner ones. To determine the rarefaction parameter, the radius of the outer circle is set as the characteristic length. Total 494 and 1082 triangles are employed in the spatial discretization and the velocity grid is the same as the previous test. Here the regular triangle elements without distortion are used. Therefore, the curved boundaries are approximated by line segments. Note that elements allowing curved boundaries could be used to achieve lower geometrical discretization error on mesh with fewer triangles.

Fig. 10 displays the velocity contours in the different geometries with varying rarefaction parameters, where the velocity distributions possess an axial symmetry. When there is no solid inside the outer circle (the last row in Fig. 7), the maximum velocity is at the center of the domain. However, for the Level-1 geometry, the large flow velocities move along the radial direction to the outer boundary. While for the Level-2 geometry, the large flow velocities emerge in the center of the domain again. Fig. 11 shows the MFRs in the Poiseuille flow through the Apollonian gasket channels together with the one through the circular channel. As the recursion level increases, the porosity of the Apollonian gasket channel decreases, so as the MFR. The Knudsen minimum in the MFR can be seen, however, the location of the minimum MFR shifts towards larger values of δ in the Apollonian gasket channels compared to the one in the circle channel. This is because, in the calculation

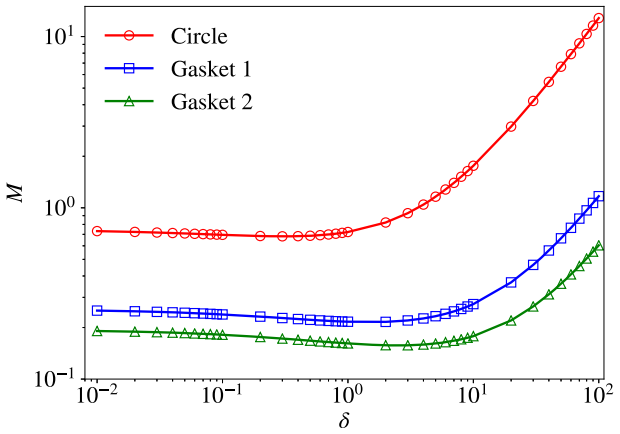

Fig. 11. MFRs of the Poiseuille flow through the Apollonian gasket channels solved by the HDG-SIS.

of δ the characteristic flow length H is selected to be the radius of the outer circle, which is larger than the radius of the solid near which the flow velocity is maximum.

5. Conclusions

In summary, based on the high-order HDG discretization, we have developed an accurate and efficient numerical method to find the steady-state solution of the linearized BGK model equation, for rarefied Poiseuille gas flow through channels with cross-sections of arbitrary shapes. First, an HDG solver with approximation polynomial of degree up to 4 has been developed. The velocity distribution functions and their traces are approximated on arbitrary triangular mesh and the mesh skeleton, respectively. Based on the first-order upwind scheme, a numerical flux has been designed to evaluate the convection between adjacent cells. By imposing the continuity of the normal flux, a final global system for traces of the velocity distribution function is obtained. Since the traces are defined on the cell interfaces and have single-values, the global coupled degrees of freedom of the unknowns are significantly reduced compared to the classical discontinuous Galerkin method. The boundary condition has been implemented in a unified framework, the same as the calculation of flux on interfaces.

In parallel to the HDG solver for the gas kinetic equation, a macroscopic diffusion equation for flow velocity is synchronously solved on the same mesh. At each iterative step, the collision operator $\mathcal{L}$ in the bulk region is corrected by the flow velocity from the diffusion equation. Since the macroscopic equation boosts the exchange of information, fast convergence with asymptotic-preserving into the hydrodynamic limit is realized for the steady-state solution in the near-continuum flow regime. On the other hand, high-order moments of the velocity distribution function in the diffusion equation preserve the accuracy of the scheme in highly rarefied gas flows.

Four different problems of Poiseuille flow along channels with various cross-sections have been presented to show accuracy and capability of the proposed scheme. Several conclusions are summarized through the performance analysis:

- Compared to the conventional iterative scheme, the synthetic iterative scheme can significantly reduce the number of iterative steps to reach the steady-state solution in the near-continuum flow regime: the synthetic iterative scheme could be more than 100 times faster.
- To obtain the results with the same order of accuracy, the HDG solver with higher degree of approximation polynomial requires fewer triangles in spatial discretization. As a result, the computational time and memory consumption can be further reduced.
- Compared to the synthetic iterative scheme solved by the finite difference method, the HDG discretization is much more efficient. To obtain the results with the same order of accuracy, the HDG scheme can be faster than the finite difference method by one order of magnitude.
- Compared to the implicit discontinuous Galerkin scheme, the HDG solver is more efficient when the degree of approximating polynomial is larger than 2, since the number of coupled degrees of freedom is reduced. Compared to the explicit Runge–Kutta discontinuous Galerkin gas kinetic solver, the HDG solver requires significantly fewer iterative steps thus less CPU time to obtain converged solutions.

It is worth mentioning that the basic HDG formulation developed in this paper is not limited to the linearized BGK equation. It is straightforward to be extended for other gas kinetic model equations, or even the full Boltzmann equation by adopting a proper method (e.g. fast spectral method [59] and conservative projection method [60]) to calculate the Boltzmann collision operator. Since the computational cost of the Boltzmann collision operator is much higher than that of the gas kinetic models, and the HDG with higher degree of approximation polynomial can reduce the spatial triangular meshes

(and hence the nodal points where the Boltzmann collision operator is evaluated), the advantage of using HDG method will become more pronounced. Also, the HDG-SIS is ready to be extended for the simulation of rarefied gas mixtures.

Acknowledgements

This work is jointly founded by the Royal Society of Edinburgh and National Natural Science Foundation of China under Grant No. 51711530130, the Carnegie Research Incentive Grant for the Universities in Scotland, and the Engineering and Physical Sciences Research Council (EPSRC) in the UK under grant EP/M021475/1. Wei Su is grateful to Dr. Xi Zou at the University of Nottingham for his valuable suggestion on solving large sparse linear systems.

Appendix A

Details for implementing the HDG method for the linearized BGK model equation are presented. The weak form of the HDG local and global problems are:

$$-\left(\nabla\varphi, \boldsymbol{v}^j h^j\right)_{\Omega_i} + \sum_{e=1}^{3}\langle\varphi, \left(\boldsymbol{v}^j\cdot\boldsymbol{n}-\alpha\right)\hat{h}^j\rangle_{\partial\Omega_i^e} + \sum_{e=1}^{3}\langle\varphi, \alpha h^j\rangle_{\partial\Omega_i^e} + (\varphi, \delta h^j)_{\Omega_i} = (\varphi, s^j)_{\Omega_i}, \tag{A.1}$$

and

$$\begin{aligned}\langle\psi, \hat{h}^j\rangle_{\Gamma_c} &= \frac{1}{2}\langle\psi, h^j_{\eta(c,+)} + h^j_{\eta(c,-)}\rangle_{\Gamma_c}, \quad \text{on } \Gamma\backslash\partial\Omega,\\ \langle\psi, \hat{h}^j\rangle_{\Gamma_c} &= \frac{1}{2}\langle\psi, h^j_{\eta(c,+)} + g^j\rangle_{\Gamma_c}, \quad \text{on } \Gamma\cap\partial\Omega,\end{aligned} \tag{A.2}$$

where $i = 1, \ldots, M_{\text{el}},\ j = 1, \ldots, M_{\text{v}}$ with M_{el} being the number of triangular elements and M_{v} being the number of discrete velocities. The local problem (A.1) represents a system of equations for each triangle Ω_i and discrete velocity $\boldsymbol{v}^j$, which allow unknown h^j as a function of the trace unknown $\hat{h}^j$. Then, replaced in Eq. (A.2), a global system is built in terms of the unknown trace only.

In this paper, unknowns are approximated by nodal shape functions N_l in each triangle Ω_i or by $\hat{N}_l$ on each face Γ_c, which have the form given below:

$$\begin{aligned}h_i^j &= \sum_{l=1}^{K_{\text{el}}} N_i^l H_{i,l}^j, \quad \text{in } \Omega_i\\ \hat{h}_c^j &= \sum_{l=1}^{K_{\text{fc}}} \hat{N}_c^l \hat{H}_{c,l}^j, \quad \text{on } \Gamma_c\end{aligned} \tag{A.3}$$

where $K_{\text{el}} = (k+1)(k+2)/2$ and $K_{\text{fc}} = k+1$ are the numbers of degree of freedom, when the approximations are sought in the finite element space of polynomials of degree up to k. If we denote $\mathbf{H}^{i,j}$ as the vector of nodal value of h^j on each triangle Ω_i, $\hat{\mathbf{H}}^{i,j}$ as the vector collecting all the nodal value of $\hat{h}^j$ on the three faces of triangle Ω_i, and $\hat{\mathbf{H}}^{c,j}$ as the vector of nodal value of $\hat{h}^j$ on each face Γ_c, both the local and global problems can be rewritten in the following matrix form as:

$$\mathbf{H}^{i,j} = \left[\mathbf{A}^{i,j}\right]^{-1}\mathbf{S}^{i,j} + \left[\mathbf{A}^{i,j}\right]^{-1}\hat{\mathbf{A}}^{i,j}\hat{\mathbf{H}}^{i,j}, \tag{A.4}$$

and

$$\begin{aligned}\hat{\mathbf{B}}^{c,j}\hat{\mathbf{H}}^{c,j} &= \mathbf{B}^{\eta(c,+),j}\mathbf{H}^{\eta(c,+),j} + \mathbf{B}^{\eta(c,-),j}\mathbf{H}^{\eta(c,-),j}, \quad \text{on } \Gamma\backslash\partial\Omega,\\ \hat{\mathbf{B}}^{c,j}\hat{\mathbf{H}}^{c,j} &= \mathbf{B}^{\eta(c,+),j}\mathbf{H}^{\eta(c,+),j} + \hat{\mathbf{S}}^{c,j}, \quad \text{on } \Gamma\cap\partial\Omega,\end{aligned} \tag{A.5}$$

where

$$
\begin{aligned}
\mathbf{A}_{ml}^{i,j} &= \delta\left(N_i^m, N_i^l\right)_{\Omega_i} + \sum_{e=1}^{3}\alpha\langle N_i^m, N_i^l\rangle_{\partial\Omega_i^e} - \left(\boldsymbol{v}^j\cdot\nabla N_i^m, N_i^l\right)_{\Omega_i},\\
\hat{\mathbf{A}}_{ml}^{i,j,e} &= \left(\alpha - \boldsymbol{v}^j\cdot\boldsymbol{n}\right)\langle N_i^m, \hat{N}_{\sigma(i,e)}^l\rangle_{\partial\Omega_i^e},\\
\mathbf{S}_m^{i,j} &= \left(N_i^m, s^j\right)_{\Omega_i},\\
\hat{\mathbf{B}}_{ml}^{c,j} &= \langle\hat{N}_c^m, \hat{N}_c^l\rangle_{\Gamma_c},\\
\mathbf{B}_{ml}^{\eta(c,\pm),j} &= \frac{1}{2}\langle\hat{N}_c^m, N_{\eta(c,\pm)}^l\rangle_{\Gamma_c},\\
\hat{\mathbf{S}}_m^{c,j} &= \frac{1}{2}\langle\hat{N}_c^m, g^j\rangle_{\Gamma_c}.
\end{aligned}
\tag{A.6}
$$

By eliminating the unknowns $\mathbf{H}^{i,j}$ with Eq. (A.4) and assembling the equations of global problem over all faces, the global problem becomes

$$
\mathbb{K}^j\hat{\mathbf{H}}^j = \mathbb{R}^j,
\tag{A.7}
$$

where $\hat{\mathbf{H}}^j$ is the vector of nodal value of $\hat{h}^j$ that is the sum of all the faces in the computational domain, and

$$
\begin{aligned}
\mathbb{K}^j &= \mathop{\mathsf{A}}_{c=1}^{M_{\mathrm{fc}}}\left\{\hat{\mathbf{B}}^{c,j} - \mathbf{B}^{\eta(c,\pm),j}\left[\mathbf{A}^{\eta(c,\pm),j}\right]^{-1}\hat{\mathbf{A}}^{\eta(c,\pm),j}\right\},\\
\mathbb{R}^j &= \mathop{\mathsf{A}}_{c=1}^{M_{\mathrm{fc}}}\left\{\mathbf{B}^{\eta(c,\pm),j}\left[\mathbf{A}^{\eta(c,\pm),j}\right]^{-1}\mathbf{S}^{\eta(c,\pm),j} + \hat{\mathbf{S}}^{c,j}\right\},
\end{aligned}
\tag{A.8}
$$

where A is the conventional assembly operator. To obtain the global matrix $\mathbb{K}^j$ and vector $\mathbb{R}^j$, the dense matrices $\mathbf{A}^{i,j}$ with dimension $K_{\mathrm{el}}\times K_{\mathrm{el}}$ for each $i=1,\ldots,M_{\mathrm{el}},\ j=1,\ldots,M_{\mathrm{v}}$ need to be inverted. Then, the sparse unsymmetrical linear system (A.7) is directly solved to determine $\hat{\mathbf{H}}^j$. Finally, $\mathbf{H}^{i,j}$ is updated in an element-by-element fashion according to Eq. (A.4).

Appendix B

We present the formulation of the implicit DG method for Eq. (8). Unlike the HDG method, the original DG method only provides the approximate solution to h^j on Ω_i in the piecewise finite element space $\mathcal{V}$. At each iterative step, the following weak formula is solved to determine the field unknowns:

$$
-\left(\nabla\varphi, \boldsymbol{v}^j h^j\right)_{\Omega_i} + \sum_{e=1}^{3}\langle\varphi, \hat{\boldsymbol{F}}\cdot\boldsymbol{n}\rangle_{\partial\Omega_i^e} + \left(\varphi, \delta h^j\right)_{\Omega_i} = \left(\varphi, s^j\right)_{\Omega_i}, \quad \text{for all } \varphi\in\mathcal{V},
\tag{B.1}
$$

where the numerical flux is defined based on the upwind scheme:

$$
\hat{\boldsymbol{F}}\cdot\boldsymbol{n} = \frac{1}{2}\left[\boldsymbol{v}^j\cdot\boldsymbol{n}h^j + \boldsymbol{v}^j\cdot\boldsymbol{n}h_{\mathrm{ext}(i,e)}^j + |\boldsymbol{v}^j\cdot\boldsymbol{n}|\left(h^j - h_{\mathrm{ext}(i,e)}^j\right)\right],
\tag{B.2}
$$

with $h_{\mathrm{ext}(i,e)}^j$ being the VDF in the neighboring triangle $\Omega_{\mathrm{ext}(i,e)}$ that shares the face e with Ω_i. If the face is at the boundary $\partial\Omega$, $h_{\mathrm{ext}(i,e)}^j = g^j$.

Assembling the weak formulation over all triangles, we obtain the global system in the following matrix form:

$$
\mathbb{K}^j\mathbf{H}^j = \mathbb{R}^j,
\tag{B.3}
$$

where $\mathbf{H}^j$ is the vector of the nodal value of h^j that is the sum of all the triangles in the computational domain, and

$$
\begin{aligned}
\mathbb{K}^j &= \mathop{\mathsf{A}}_{i=1}^{M_{\mathrm{el}}}\left\{\mathbf{A}^{i,j} + \sum_{e=1}^{3}\mathbf{A}^{\mathrm{ext}(i,e),j}\right\},\\
\mathbb{R}^j &= \mathop{\mathsf{A}}_{i=1}^{M_{\mathrm{el}}}\mathbf{S}^{i,j},
\end{aligned}
\tag{B.4}
$$

$$\begin{aligned}
\mathbf{A}_{ml}^{i,j} &= \delta\left(N_i^m, N_i^l\right)_{\Omega_i} + \sum_{e=1}^{3}\frac{1}{2}\left(\boldsymbol{v}^j\cdot\boldsymbol{n} + |\boldsymbol{v}^j\cdot\boldsymbol{n}|\right)\langle N_i^m, N_i^l\rangle_{\partial\Omega_i} - \left(\boldsymbol{v}^j\cdot\nabla N_i^m, N_i^l\right)_{\Omega_i},\\
\mathbf{A}_{ml}^{\mathrm{ext}(i,e),j} &= \frac{1}{2}\left(\boldsymbol{v}^j\cdot\boldsymbol{n} - |\boldsymbol{v}^j\cdot\boldsymbol{n}|\right)\langle N_i^m, N_{\mathrm{ext}(i,e)}^l\rangle_{\partial\Omega_i^e},\\
\mathbf{S}_m^{i,j} &= \left(N_i^m, s^j\right)_{\Omega_i},
\end{aligned} \tag{B.5}$$

with $N_{\mathrm{ext}(i,e)}^l$ being the nodal function in the neighboring triangle. Here, $\mathbf{A}$ is the conventional assembly operator. To solve the linear system, direct solver is applied. The global matrices $\mathbb{K}^j$ remain unchanged during all iterations. Before iteration, the coefficient matrices $\mathbb{K}^j$ are assembled for all discrete velocities and the *LU*-decomposition is conducted. At each iterative step, h^j are obtained by direct substitution. Both the *LU*-decomposition and substitution phases are executed by calling PARDISO.

References

[1] S. Chapman, T. Cowling, The Mathematical Theory of Non-uniform Gases, 3rd edition, Cambridge University press, New York, 1970.
[2] G.A. Bird, Molecular Gas Dynamics and the Direct Simulation, Clarendon, Oxford, 1994.
[3] V.V. Aristov, Direct Methods for Solving the Boltzmann Equation and Study of Nonequilibrium Flows, Springer, Netherlands, Dordrecht, 2001.
[4] N.G. Hadjiconstantinou, A.L. Garcia, M.Z. Bazant, G. He, Statistical error in particle simulations of hydrodynamic phenomena, J. Comput. Phys. 187 (1) (2003) 274–297.
[5] A.B. Huang, D.P. Giddens, The discrete ordinate method for the linearized boundary value problems in kinetic theory of gases, in: C.L. Brundin (Ed.), Rarefied Gas Dynamics, vol. 1, 1967, p. 481.
[6] J. Yang, J. Huang, Rarefied flow computations using nonlinear model Boltzmann equations, J. Comput. Phys. 120 (2) (1995) 323–339.
[7] V. Kolobov, R. Arslanbekov, V. Aristov, A. Frolova, S. Zabelok, Unified solver for rarefied and continuum flows with adaptive mesh and algorithm refinement, J. Comput. Phys. 223 (2) (2007) 589–608.
[8] L. Mieussens, Discrete-velocity models and numerical schemes for the Boltzmann-BGK equation in plane and axisymmetric geometries, J. Comput. Phys. 162 (2) (2000) 429–466.
[9] L. Wu, J.M. Reese, Y. Zhang, Solving the Boltzmann equation deterministically by the fast spectral method: application to gas microflows, J. Fluid Mech. 746 (2014) 53–84.
[10] W. Su, S. Lindsay, H. Liu, L. Wu, Comparative study of the discrete velocity and lattice Boltzmann methods for rarefied gas flows through irregular channels, Phys. Rev. E 96 (2017) 023309.
[11] J.M. Burt, I.D. Boyd, A low diffusion particle method for simulating compressible inviscid flows, J. Comput. Phys. 227 (9) (2008) 4653–4670.
[12] L. Wu, J. Zhang, H. Liu, Y. Zhang, J.M. Reese, A fast iterative scheme for the linearized Boltzmann equation, J. Comput. Phys. 338 (2017) 431–451.
[13] R.R. Arslanbekov, V.I. Kolobov, A.A. Frolova, Kinetic solvers with adaptive mesh in phase space, AIP Conf. Proc. 1501 (1) (2012) 294–301.
[14] C. Baranger, J. Claudel, N. Hérouard, L. Mieussens, Locally refined discrete velocity grids for stationary rarefied flow simulations, J. Comput. Phys. 257 (2014) 572–593.
[15] A. Alekseenko, E. Josyula, Deterministic solution of the spatially homogeneous Boltzmann equation using discontinuous Galerkin discretizations in the velocity space, J. Comput. Phys. 272 (2014) 170–188.
[16] K. Xu, J.-C. Huang, A unified gas-kinetic scheme for continuum and rarefied flows, J. Comput. Phys. 229 (20) (2010) 7747–7764.
[17] C. Liu, K. Xu, Q. Sun, Q. Cai, A unified gas-kinetic scheme for continuum and rarefied flows IV: full Boltzmann and model equations, J. Comput. Phys. 314 (2016) 305–340.
[18] 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.
[19] Z. Guo, R. Wang, K. Xu, Discrete unified gas kinetic scheme for all Knudsen number flows. II. Thermal compressible case, Phys. Rev. E 91 (3) (2015) 033313.
[20] S. Chen, K. Xu, A comparative study of an asymptotic preserving scheme and unified gas-kinetic scheme in continuum flow limit, J. Comput. Phys. 288 (2015) 52–65.
[21] Y. Zhu, C. Zhong, K. Xu, Implicit unified gas-kinetic scheme for steady state solutions in all flow regimes, J. Comput. Phys. 315 (2016) 16–38.
[22] P. Wang, M.T. Ho, L. Wu, Z. Guo, Y. Zhang, A comparative study of discrete velocity methods for low-speed rarefied gas flows, Comput. Fluids 161 (2018) 33–46.
[23] D. Valougeorgis, S. Naris, Acceleration schemes of the discrete velocity method: Gaseous flows in rectangular microchannels, SIAM J. Sci. Comput. 25 (2) (2003) 534–552.
[24] J. Lihnaropoulos, S. Naris, D. Valougeorgis, Formulation and stability analysis of rapidly convergent iteration schemes for the 2-D linearized BGK equation, Transp. Theory Stat. Phys. 36 (4–6) (2007) 513–528.
[25] L. Szalmás, D. Valougeorgis, A fast iterative model for discrete velocity calculations on triangular grids, J. Comput. Phys. 229 (11) (2010) 4315–4326.
[26] S. Naris, D. Valougeorgis, F. Sharipov, D. Kalempa, Discrete velocity modelling of gaseous mixture flows in MEMS, Superlattices Microstruct. 35 (3) (2004) 629–643.
[27] S. Naris, D. Valougeorgis, D. Kalempa, F. Sharipov, Flow of gaseous mixtures through rectangular microchannels driven by pressure, temperature, and concentration gradients, Phys. Fluids 17 (10) (2005) 100607.
[28] L. Szalmás, An accelerated discrete velocity method for flows of rarefied ternary gas mixtures in long rectangular channels, Comput. Fluids 128 (2016) 91–97.
[29] L. Szalmás, A fast iterative discrete velocity method for ternary gas mixtures flowing through long tubes, Comput. Phys. Commun. 200 (2016) 44–49.
[30] L. Wu, H. Struchtrup, Assessment and development of the gas kinetic boundary condition for the Boltzmann equation, J. Fluid Mech. 823 (2017) 511–537.
[31] Z. Jiang, K. Wu, G. Couples, M.I.J. van Dijke, K.S. Sorbie, J. Ma, Efficient extraction of networks from three-dimensional porous media, Water Resour. Res. 43 (12) (2007) 1–7.
[32] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (1954) 511–525.
[33] L.H. Holway, New statistical models for kinetic theory: methods of construction, Phys. Fluids 9 (9) (1966) 1658–1673.
[34] E. Shakhov, Generalization of the Krook kinetic relaxation equation, Fluid Dyn. 3 (5) (1968) 95–96.
[35] C. Cercignani, The Boltzmann Equation and Its Applications, Springer-Verlag, New York, 1988.

[36] B. Shizgal, A Gaussian quadrature procedure for use in the solution of the Boltzmann equation and related problems, J. Comput. Phys. 41 (2) (1981) 309–328.
[37] W.H. Reed, T.R. Hill, Triangular Mesh Methods for the Neutron Transport Equation, Tech. Rep. 836, 1973.
[38] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (2) (1998) 199–224.
[39] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (3) (2001) 173–261.
[40] W. Su, A.A. Alexeenko, G. Cai, A parallel Runge–Kutta discontinuous Galerkin solver for rarefied gas flows based on 2D Boltzmann kinetic equations, Comput. Fluids 109 (2015) 123–136.
[41] M. Gobbert, S. Webster, T. Cale, A Galerkin method for the simulation of the transient 2-D/2-D and 3-D/3-D linear Boltzmann equation, J. Sci. Comput. 30 (2) (2007) 237–273.
[42] L.L. Baker, N.G. Hadjiconstantinou, Variance-reduced Monte Carlo solutions of the Boltzmann equation for low-speed gas flows: a discontinuous Galerkin formulation, Int. J. Numer. Methods Fluids 58 (4) (2008) 381–402.
[43] G. Kitzler, J. Schöberl, A high order space-momentum discontinuous Galerkin method for the Boltzmann equation, Comput. Math. Appl. 70 (7) (2015) 1539–1554.
[44] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, SIAM J. Numer. Anal. 35 (6) (1998) 2440–2463.
[45] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 47 (2) (2009) 1319–1365.
[46] N. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for stokes flow, Comput. Methods Appl. Mech. Eng. 199 (9) (2010) 582–597.
[47] G. Giorgiani, S. Fernández–Méndez, A. Huerta, Hybridizable discontinuous Galerkin p-adaptivity for wave propagation problems, Int. J. Numer. Methods Fluids 72 (12) (2013) 1244–1262.
[48] J. Peraire, N. Nguyen, B. Cockburn, A hybridizable discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations, in: 48th AIAA Aerosp. Sci. Meet. Incl. New Horizons Forum Aerosp. Expo., 2010, pp. 1–11.
[49] N. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations, J. Comput. Phys. 230 (4) (2011) 1147–1170.
[50] M.J. Moghtader, High-Order Hybridizable Discontinuous Galerkin Method For Viscous Compressible Flows, Ph.D. thesis, Universitat Politècnica de Catalunya, 2016.
[51] R.M. Kirby, S.J. Sherwin, B. Cockburn, To CG or to HDG: a comparative study, J. Sci. Comput. 51 (1) (2012) 183–212.
[52] R. Sevilla, A. Huerta, Tutorial on Hybridizable Discontinuous Galerkin (HDG) for Second-Order Elliptic Problems, Springer International Publishing, Cham, 2016, pp. 105–129.
[53] H. Antonio, A. Aleksandar, R. Xevi, P. Jaime, Efficiency of high-order elements for continuous and discontinuous Galerkin methods, Int. J. Numer. Methods Eng. 96 (9) (2013) 529–560.
[54] A. Crivellini, F. Bassi, An implicit matrix-free discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations, Comput. Fluids 50 (1) (2011) 81–93.
[55] S. Muralikrishnan, M.-B. Tran, T. Bui-Thanh, An improved iterative HDG approach for partial differential equations, J. Comput. Phys. 367 (2018) 295–321.
[56] O. Schenk, K. Gärtner, Solving unsymmetric sparse systems of linear equations with PARDISO, Future Gener. Comput. Syst. 20 (3) (2004) 475–487.
[57] I. Graur, F. Sharipov, Non-isothermal flow of rarefied gas through a long pipe with elliptic cross section, Microfluid. Nanofluid. 6 (2) (2009) 267–275.
[58] K. Ritos, Y. Lihnaropoulos, S. Naris, D. Valougeorgis, Pressure-and temperature-driven flow through triangular and trapezoidal microchannels, Heat Transf. Eng. 32 (13–14) (2011) 1101–1107.
[59] L. Wu, C. White, T.J. Scanlon, J.M. Reese, Y. Zhang, Deterministic numerical solutions of the Boltzmann equation using the fast spectral method, J. Comput. Phys. 250 (2013) 27–52.
[60] F. Tcheremissine, Conservative evaluation of Boltzmann collision integral in discrete ordinates approximation, Comput. Math. Appl. 35 (1) (1998) 215–221.