Physics of Fluids

An efficient discrete velocity method with inner iteration for steady flows in all flow regimes

Cite as: Phys. Fluids **34**, 027110 (2022); https://doi.org/10.1063/5.0084547
Submitted: 07 January 2022 • Accepted: 26 January 2022 • Published Online: 11 February 2022

L. M. Yang (杨鲤铭), C. Shu (舒昌), J. Wu (吴杰), et al.

Physics of Fluids ARTICLE scitation.org/journal/phf

An efficient discrete velocity method with inner iteration for steady flows in all flow regimes

Cite as: Phys. Fluids **34**, 027110 (2022); doi: 10.1063/5.0084547
Submitted: 7 January 2022 · Accepted: 26 January 2022 ·
Published Online: 11 February 2022

L. M. Yang (杨鲤铭),[1,a)] C. Shu (舒昌),[2] J. Wu (吴杰),[1] Y. Y. Liu (刘阳阳),[2] and X. Shen (沈翔)[3]

AFFILIATIONS

[1]Department of Aerodynamics, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Yudao Street, Nanjing 210016, Jiangsu, China
[2]Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 117576
[3]Department of Mechanical and Construction Engineering, Northumbria University, Newcastle upon Tyne NE1 8ST, United Kingdom

[a)]Author to whom correspondence should be addressed: lmyang@nuaa.edu.cn

ABSTRACT

An efficient improved discrete velocity method (IDVM) with inner iteration is presented to simulate the steady flows in all flow regimes in this work. It is an extension of our previous implicit IDVM to achieve a faster convergence rate. In the previous method, both the discrete velocity Boltzmann equation (DVBE) and the corresponding macroscopic governing equations are solved synchronously, where the computational discrete cost is dominated by the calculation of the DVBE since the number of distribution functions is far larger than that of macroscopic conservative variables. Furthermore, the convergence rate of the calculation of the DVBE is affected by the predicted equilibrium state obtained from the solution of macroscopic governing equations. To provide a more accurate predicted equilibrium state for the fully implicit discretization of the DVBE, an inner iteration is introduced into the solution of macroscopic governing equations, and the flux Jacobian of these equations is evaluated by the difference of numerical fluxes of Navier–Stokes equations rather than the Euler equation-based flux splitting method used in the previous implicit IDVM. This more accurate prediction procedure endows the developed method to accelerate the computation greatly, especially in the continuum flow regime. Numerical results indicate that, in the continuum flow regime, the present method is about one order of magnitude faster than the previous implicit IDVM and one to two orders of magnitude faster than the conventional semi-implicit DVM.

I. INTRODUCTION

The rarefication of a fluid flow problem is usually represented by the Knudsen number, which can be computed by the ratio of the mean free path of molecular to the characteristic length. Since the mean free path of molecular increases with the altitude, the reentry vehicle will encounter rarefied situations at a high altitude.[1–4] Besides, the microelectromechanical systems (MEMS) will also face rarefied flow problems due to the very small characteristic scale.[5–7] Moreover, the local Knudsen number in practical engineering problems may vary significantly over several orders of magnitude, which results in the flows covering all flow regimes. Taking the hypersonic flow over a flying vehicle at Reynolds number of 59 373 and Mach number of 4 as an example, the minimum and maximum local Knudsen numbers are 7.61×10^{-5} and 19.9, respectively.[8,9] Thus, an efficient and accurate numerical algorithm for all Knudsen number flows is always desired for such problems.

The Boltzmann–Bhatnagar–Gross–Krook (BGK) equation, which is free from the continuum assumption, is often utilized to resolve the above problems. This equation originates from the kinetic theory, and it describes the evolution of gas distribution functions in the velocity space, physical space, and time. Thus, to solve this equation with a deterministic method, the discretization should be carried out in both the physical space and the velocity space. Due to the discretization in the velocity space, this kind of approach is collectively referred to as the discrete velocity method (DVM),[10–17] which solves the discrete velocity Boltzmann equation (DVBE) by a certain computational fluid dynamics (CFD) approach, such as the finite volume method (FVM), the finite difference method (FDM), the finite element method (FEM), and the physics-informed neural network method.[18–21] In the framework of DVM, for simplicity and intuitive implementation, many studies aim to solve the DVBE individually.[22–25] Alternatively, thanks to the correspondence between the Boltzmann–BGK equation and the macroscopic

governing equations of mass, momentum, and energy, these two sets of equations are solved simultaneously in some of the variant discrete velocity methods, such as the unified gas kinetic scheme (UGKS),[26–29] the improved discrete velocity method (IDVM),[30,31] the general synthetic iteration scheme (GSIS),[32,33] and the modified discrete unified gas kinetic scheme (DUGKS).[34,35] Since the number of variables of the DVBE is far larger than that of macroscopic governing equations, the computational cost of the DVM is actually dominated by the calculation of the DVBE, which should be minimized as much as possible to improve the overall efficiency.

For the first kind of DVM,[22–25] i.e., the DVBE is solved individually, the numerical flux across the cell interface is the key to evolving the discrete distribution function at the cell center when the FVM is utilized to do the spatial discretization. Inspired by the reconstruction strategy used in the conventional CFD method, the upwind scheme is usually utilized to reconstruct the left and right states of the discrete distribution function at the cell interface, and the local solution to the Boltzmann equation without the collision term is directly adopted to calculate the numerical flux.[36,37] For illustration, we named this kind of approach the conventional DVM in this work. Since the equilibrium state is not required for the calculation of numerical flux, the distribution function of each discrete velocity in the conventional DVM can be updated individually, resulting in a convenient implementation. To achieve a faster convergence rate, the implicit scheme has been introduced into the conventional DVM. However, since the equilibrium state or the macroscopic flow variables at the new time level is still unknown, the collision term of the DVBE is usually discretized by the semi-implicit scheme, i.e., the equilibrium state is approximated by its value at the current time level.[38–40] This approximation will decelerate the convergence rate greatly in the continuum flow regime as pointed out by Mieussens.[41] To overcome this defect, Mieussens introduced a linearized implicit scheme to estimate the equilibrium distribution function at the new time level.[41] However, since a Jacobian matrix in the scale of the total number of discrete distribution functions needs to be calculated, this strategy complicates significantly the algorithm. Moreover, the conventional DVM usually requires the mesh size to be at least on the order of the mean free path of molecular due to the ignoring of the collisional effect in the evaluation of numerical flux across the cell interface, which makes it hard to provide accurate and efficient predictions in the continuum flow regime. By using the discrete characteristic solution of the DVBE to reconstruct the numerical flux across the cell interface, the DUGKS was proposed to overcome the above defects of the conventional DVM.[42,43] However, if only the DVBE is solved, the DUGKS will also encounter the difficulty in developing a fully implicit counterpart.

Another type of DVM resolves the DVBE and the macroscopic governing equations of mass, momentum, and energy simultaneously.[26–33] A representative is the UGKS,[26–29] which evaluates the numerical fluxes of both the DVBE and the macroscopic governing equations at the cell interface by locally reconstructing the integral solution of the DVBE. On the one hand, the UGKS can provide reasonable results in all flow regimes without the limitation of mesh size on the order of the mean free path of molecular due to the incorporation of the collisional effect into the calculation of fluxes. On the other hand, the UGKS can be easily extended to the fully implicit version by predicting the equilibrium distribution function at the new time level from the solution of macroscopic governing equations. Specifically, solving both the DVBE and the macroscopic governing equations by the lower–upper symmetric Gauss–Seidel (LU-SGS) method, two versions of the fully implicit UGKS were presented successively for steady flows[44] and unsteady flows.[45] Furthermore, to achieve a faster convergence rate, the multi-prediction implicit scheme was developed recently for steady flows in all flow regimes.[46] In this method, an inner iteration is introduced into the solutions of the DVBE as well as the macroscopic governing equations, which results in a more accurate predicted equilibrium state for the fully implicit discretization of the DVBE as compared with the implicit UGKS. This accurate prediction procedure endows the new method with high convergence speed in all flow regimes, especially in the continuum flow regime. In addition, the GSIS also introduces inner iterations into the solutions of both the DVBE and the macroscopic governing equations to improve the overall efficiency.[32,33] More recently, a two-step implicit UGKS was proposed to further enhance the computational efficiency of the original implicit UGKS.[47] In this method, one more prediction step for solving the macroscopic governing equations is introduced to incorporate the viscous term into the evaluation of macroscopic numerical flux.

The IDVM is another method to overcome the defects of the conventional DVM in the continuum flow regime.[30,31] This method also solves the DVBE and the macroscopic governing equations synchronously, while the collisional effect is only involved in the calculation of the numerical flux of macroscopic governing equations. As a result, the IDVM well keeps the simplicity of the conventional method and provides reasonable predictions in all flow regimes. To further improve the efficiency, two versions of the fully implicit IDVM were developed for the two-dimensional (2D) case[48] and the three-dimensional (3D) case,[49] respectively. In the implicit IDVM, the LU-SGS method is introduced into the solutions of both the macroscopic governing equations and the DVBE, and it is executed only once in each iteration for updating the predicted macroscopic flow variables and the discrete distribution functions. However, since the computational cost of the IDVM is actually dominated by the calculation of the DVBE, the strategy used in the previous implicit IDVM may not be optimal. In this work, an inner iteration will be introduced into the solution of macroscopic governing equations to provide a more accurate predicted equilibrium state for the fully implicit discretization of DVBE. Meanwhile, the flux Jacobian of macroscopic governing equations, which is estimated by the Euler equation-based flux splitting method in the previous implicit IDVM, will be evaluated by the difference of numerical fluxes of Navier–Stokes equations. These two strategies can accelerate greatly the computation of the DVBE so as to further improve the overall efficiency of the previous implicit IDVM, especially in the continuum flow regime. The influence of the inner iteration on the overall efficiency of the developed method will be investigated comprehensively to determine the preferable inner iteration numbers first, and then several 2D and 3D test examples from the continuum flow regime to the free molecular flow regime will be simulated to further assess the overall performance of the present method.

II. IMPROVED DISCRETE VELOCITY METHOD FOR ALL KNUDSEN NUMBER FLOWS

The original Boltzmann equation is an integral–differential equation, and the collision term is a fivefold integral, which is too complicated for engineering applications. To simplify the collision term, the BGK model is introduced by Bhatnagar, Gross, and Krook.[50]

However, this model will lead to unit Prandtl number. To be suitable for fluid flow problems with arbitrary Prandtl numbers, the Shakhov model is proposed.[51] The discrete form of the Boltzmann–BGK equation with the Shakhov model in the velocity space, namely, the DVBE, can be expressed as

$$\frac{\partial f_\alpha}{\partial t} + \boldsymbol{\xi}_\alpha \cdot \nabla f_\alpha = \frac{f_\alpha^S - f_\alpha}{\tau}, \quad \alpha = 1, \ldots, N_V, \tag{1}$$

where the subscript α is the index in the discrete velocity space and N_V denotes the total number of discrete velocity points. $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_D)$ represents the molecular velocity vector in D-dimensional space. f_α and f_α^S are the discrete distribution function and its equilibrium state. For the monatomic gas, f^S is given by

$$f^S = f^{eq}\left[1 + (1 - Pr)\frac{\mathbf{c}\cdot\mathbf{q}}{5pR_gT}\left(\frac{c^2}{R_gT} - 5\right)\right] \tag{2}$$

with

$$f^{eq} = \frac{\rho}{(2\pi R_gT)^{3/2}}\exp\left[-\frac{c^2}{2R_gT}\right], \tag{3}$$

where p is the pressure, ρ is the density, T is the temperature, Pr is the Prandtl number, $\mathbf{q}$ is the heat flux vector, $\mathbf{c} = \boldsymbol{\xi} - \mathbf{u}$ is the molecular thermal velocity vector, $c = |\mathbf{c}|$ is the magnitude of $\mathbf{c}$, and R_g is the gas constant. In addition, $\tau = \mu/p$ is the collision time and μ is the dynamic viscosity.

Taking the conservative moment integration to Eq. (1), we can obtain the corresponding macroscopic governing equations as follows:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{0}, \tag{4}$$

where the conservative flow variable vector $\mathbf{W}$ and the flux vector $\mathbf{F}$ are calculated by

$$\mathbf{W} = (\rho, \rho\mathbf{u}, \rho E)^T = \langle \boldsymbol{\psi} f \rangle_\alpha, \tag{5}$$

$$\mathbf{F} = (F_\rho, F_{\rho\mathbf{u}}, F_{\rho E})^T = \langle \boldsymbol{\xi}\boldsymbol{\psi} f \rangle_\alpha. \tag{6}$$

Here, $\boldsymbol{\psi} = (1, \boldsymbol{\xi}, \xi^2/2)^T$ denotes the moment vector. The notation $\langle f \rangle_\alpha = \sum_{\alpha=1}^{N_V} w_\alpha f_\alpha$ defines the quadrature of discrete distribution functions f_α in the velocity space, and w_α represents the quadrature weight corresponding to the discrete velocity $\boldsymbol{\xi}_\alpha$.

As introduced in Sec. I, the solution of rarefied fluid flow problems can be given by Eq. (1) purely.[22–25] However, to ensure the multiscale property and make the scheme free from the limitation of mesh size on the order of the mean free path, both the DVBE [Eq. (1)] and the macroscopic governing equations [Eq. (4)] are often resolved synchronously.[26–33] Another advantage of resolving the macroscopic governing equations is that the collision term of DVBE can be discretized in a fully implicit way, which can speed up the convergence rate of the computation.

A. Solving discrete velocity Boltzmann equation

Since only the steady flows are considered in this work, the temporal term in Eq. (1) can be dropped out. Thus, the fully implicit discretization to Eq. (1) yields

$$\boldsymbol{\xi}_\alpha \cdot \nabla f_\alpha^{n+1} = \frac{\bar{f}_\alpha^{S,n+1} - f_\alpha^{n+1}}{\tau^{n+1}}, \tag{7}$$

where n denotes the current time level and $n+1$ represents the new time level. $\bar{f}_\alpha^{S,n+1}$ is the predicted equilibrium state, which is defined at the new time level and given by the solution of Eq. (4), as shown in Sec. II B. If $\bar{f}_\alpha^{S,n+1}$ and τ^{n+1} are replaced by the equilibrium state and the collision time at the current time level, Eq. (7) will reduce to the semi-implicit discretization of the conventional DVM.[38–40]

To facilitate the application in fluid flow problems with curved boundaries, the finite volume discretization over a control volume V_i is adopted to Eq. (7), which yields

$$\frac{V_i}{\tau_i^{n+1}}\Delta f_{i,\alpha}^n + \sum_{j\in N(i)} \mathbf{n}_{ij}\cdot\boldsymbol{\xi}_\alpha \Delta f_{ij,a}^n S_{ij} = \mathrm{Res}_{i,\alpha}^n \tag{8}$$

with

$$\mathrm{Res}_{i,\alpha}^n = \frac{V_i}{\tau_i^{n+1}}\left(\bar{f}_{i,\alpha}^{S,n+1} - f_{i,\alpha}^n\right) - \sum_{j\in N(i)} \mathbf{n}_{ij}\cdot\boldsymbol{\xi}_\alpha f_{ij,a}^n S_{ij}, \tag{9}$$

where $\Delta f_{i,\alpha}^n = f_{i,\alpha}^{n+1} - f_{i,\alpha}^n$ represents the time increment of the discrete distribution function. $N(i)$ denotes the set of neighboring cells of cell i. $\mathbf{n}_{ij}$ and S_{ij} are the outward unit normal vector and the area of the interface shared by cell i and cell j, respectively.

To resolve Eq. (8), the discrete distribution function $f_{ij,a}^n$ and its time increment $\Delta f_{ij,\alpha}^n$ at the cell interface have to be determined in advance. Like the conventional DVM,[38–40] in the IDVM,[48,49] $\Delta f_{ij,\alpha}^n$ is computed by the first-order upwind scheme and $f_{ij,\alpha}^n$ is calculated by the local solution to the Boltzmann equation without the collision term at the cell interface as follows:

$$\Delta f_{ij,\alpha}^n = \mathrm{H}(\boldsymbol{\xi}_\alpha\cdot\mathbf{n}_{ij})\Delta f_{i,\alpha}^n + [1 - \mathrm{H}(\boldsymbol{\xi}_\alpha\cdot\mathbf{n}_{ij})]\Delta f_{j,\alpha}^n, \tag{10}$$

$$f_{ij,a}^n = \mathrm{H}(\boldsymbol{\xi}_\alpha\cdot\mathbf{n}_{ij}) f_{ij,\alpha}^{L,n} + [1 - \mathrm{H}(\boldsymbol{\xi}_\alpha\cdot\mathbf{n}_{ij})] f_{ij,\alpha}^{R,n}, \tag{11}$$

where $\mathrm{H}(\boldsymbol{\xi}_\alpha\cdot\mathbf{n}_{ij})$ is the heaviside function, and $f_{ij,\alpha}^{L,n}$ and $f_{ij,\alpha}^{R,n}$ are the left and right states of the discrete distribution function at the cell interface. In this work, $f_{ij,\alpha}^{L,n}$ and $f_{ij,\alpha}^{R,n}$ are interpolated from those at cell centers with a slope limiter function.

Once $\Delta f_{ij,\alpha}^n$ and $f_{ij,\alpha}^n$ are obtained, Eq. (8) can be solved by the classical LU-SGS method with the following forward and backward sweeps:[52,53]

$$D_{i,\alpha}\widetilde{\Delta f}_{i,a}^{\,n} + \sum_{j\in L(i)} D_{j,\alpha}\widetilde{\Delta f}_{j,a}^{\,n} = \mathrm{Res}_{i,\alpha}^n, \tag{12}$$

$$D_{i,\alpha}\Delta f_{i,a}^n + \sum_{j\in U(i)} D_{j,\alpha}\Delta f_{j,a}^n = D_{i,\alpha}\widetilde{\Delta f}_{i,a}^{\,n}, \tag{13}$$

where $D_{i,\alpha} = \left(\frac{V_i}{\tau_i^{n+1}} + \frac{1}{2}\sum_{j\in N(i)}|\mathbf{n}_{ij}\cdot\boldsymbol{\xi}_\alpha|S_{ij}\right)$ and $D_{j,\alpha} = \frac{1}{2}(\mathbf{n}_{ij}\cdot\boldsymbol{\xi}_\alpha - |\mathbf{n}_{ij}\cdot\boldsymbol{\xi}_\alpha|)S_{ij}$. $\widetilde{\Delta f}_{i,a}^{\,n}$ and $\widetilde{\Delta f}_{j,a}^{\,n}$ are the intermediate results of the forward sweep. The subset $L(i) \in N(i)$ represents the neighboring cells with indices less than i, and the subset $U(i) \in N(i)$ denotes the ones with indices larger than i.

B. Solving macroscopic governing equations

To ensure the multiscale property and achieve the fully implicit discretization of Eq. (1), the macroscopic governing equations (4) are

also resolved in the IDVM. The implicit discretization of Eq. (4) over a control volume V_i yields

$$\frac{V_i}{\Delta t_i^n}\Delta \mathbf{W}_i^n = -\mathbf{R}_i^{n+1}, \tag{14}$$

where $\Delta \mathbf{W}_i^n = \overline{\mathbf{W}}_i^{n+1} - \mathbf{W}_i^n$ represents the time increment of the conservative variable vector and Δt_i denotes the local time step for the implicit discretization, which can be calculated by the Courant–Friedrichs–Lewy (CFL) condition with Courant number σ. $\mathbf{R}_i$ is the residual of macroscopic governing equations, which are calculated by

$$\mathbf{R}_i = \sum_{j\in N(i)} \mathbf{n}_{ij}\cdot \mathbf{F}_{ij} S_{ij}. \tag{15}$$

To incorporate the collisional effect into the calculation of numerical flux, the local solution to the Boltzmann–BGK equation at the cell interface is utilized to evaluate $\mathbf{F}_{ij}$ in the IDVM. The detailed derivation can be found in Ref. 30, and the final expression of $\mathbf{F}_{ij}$ can be written as

$$\begin{aligned}\mathbf{F}_{ij} &= \beta\langle \boldsymbol{\xi}\boldsymbol{\psi} f_{DVM}(\mathbf{x}_{ij},\boldsymbol{\xi},\Delta t_p)\rangle_\alpha + (1-\beta)\langle \boldsymbol{\xi}\boldsymbol{\psi} f_{NS}(\mathbf{x}_{ij},\boldsymbol{\xi},\Delta t_p)\rangle_\alpha \\ &= \beta \mathbf{F}_{ij,DVM} + (1-\beta)\mathbf{F}_{ij,NS}\end{aligned} \tag{16}$$

with

$$\beta = e^{-\Delta t_p/\tau}, \tag{17}$$

where $f_{DVM}(\mathbf{x}_{ij},\boldsymbol{\xi}_\alpha,\Delta t_p) = f_{ij,a}^n$ represents the local solution to the Boltzmann equation without the collision term at the cell interface, which is given by Eq. (11). $f_{NS}(\mathbf{x}_{ij},\boldsymbol{\xi}_\alpha,\Delta t_p)$ defines the distribution function truncated to the Navier–Stokes level. Δt_p denotes the virtual time step, which is only used for the solution reconstruction and determined by the CFL condition with Courant number $\sigma_p = 0.95$ in the IDVM. In the implementation, $\mathbf{F}_{ij,DVM}$ is computed by substituting Eq. (11) into its definition and $\mathbf{F}_{ij,NS}$ is calculated by the Navier–Stokes solver directly.

To resolve Eq. (14), the macroscopic residual needs to be linearized to the following form:

$$\mathbf{R}_i^{n+1} = \mathbf{R}_i^n + \Delta \mathbf{R}_i^n = \mathbf{R}_i^n + \sum_{j\in C(i)} \frac{\partial \mathbf{R}_i^n}{\partial \mathbf{W}_j}\Delta \mathbf{W}_j^n, \tag{18}$$

where $\Delta \mathbf{R}_i^n$ is the time increment of the macroscopic residual. $C(i)$ represents a set, which contains cell i and its neighbor cells. In the previous implicit IDVM,[48,49] the last term of Eq. (18) is estimated by the following Euler equations-based flux splitting method:

$$\sum_{j\in C(i)} \frac{\partial \mathbf{R}_i^n}{\partial \mathbf{W}_j}\Delta \mathbf{W}_j^n = \frac{1}{2}\sum_{j\in N(i)} \left[\mathbf{n}_{ij}\cdot\left(\Delta \mathbf{F}_{c,i}^n + \Delta \mathbf{F}_{c,j}^n\right) + r_{ij}^n\left(\Delta \mathbf{W}_i^n - \Delta \mathbf{W}_j^n\right)\right] S_{ij}, \tag{19}$$

where

$$\Delta \mathbf{F}_{c,j}^n = \mathbf{F}_c\left(\mathbf{W}_j^n + \Delta \mathbf{W}_j^n\right) - \mathbf{F}_c\left(\mathbf{W}_j^n\right),$$

$$r_{ij}^n = \left(|\mathbf{n}_{ij}\cdot \mathbf{u}_{ij}^n| + c_{s,ij}^n\right) + \max\left(\frac{4}{3\rho_{ij}^n}, \frac{\gamma}{\rho_{ij}^n}\right)\frac{\mu_{ij}^n}{Pr|\mathbf{x}_j - \mathbf{x}_i|}.$$

Here, $\mathbf{F}_c = (\rho\mathbf{u}, \rho\mathbf{u}\mathbf{u} + p\mathbf{I}, (\rho E + p)\mathbf{u})^T$ denotes the macroscopic convective flux vector, c_s represents the sound speed, γ is the specific heat ratio, and $\mathbf{x}_i$ and $\mathbf{x}_j$ are the centroids of cell i and cell j, respectively. It should be indicated that the Euler equation-based flux splitting method may oversimplify for calculation of $\Delta \mathbf{R}_i^n$.

Substituting Eqs. (18) and (19) into Eq. (14), the resultant equation can be then solved by the LU-SGS method as follows:[52,53]

$$Q_i^n \widetilde{\Delta \mathbf{W}}_i^n + \frac{1}{2}\sum_{j\in L(i)}\left[\mathbf{n}_{ij}\cdot \Delta \mathbf{F}_{c,j}^n - r_{ij}^n \widetilde{\Delta \mathbf{W}}_j^n\right] S_{ij} = -\mathbf{R}_i^n, \tag{20}$$

$$Q_i^n \Delta \mathbf{W}_i^n + \frac{1}{2}\sum_{j\in U(i)}\left[\mathbf{n}_{ij}\cdot \Delta \mathbf{F}_{c,j}^n - r_{ij}^n \Delta \mathbf{W}_j^n\right] S_{ij} = Q_i^n \widetilde{\Delta \mathbf{W}}_i^n, \tag{21}$$

where $Q_i^n = \frac{V_i}{\Delta t_i^n} + \frac{1}{2}\sum_{j\in N(i)} r_{ij}^n S_{ij}$. By using the forward sweep (20) and the backward sweep (21), the time increment of the conservative variable vector $\Delta \mathbf{W}_i^n$ can be obtained and the predicted conservative variable vector can be then computed by $\overline{\mathbf{W}}_i^{n+1} = \mathbf{W}_i^n + \Delta \mathbf{W}_i^n$. Finally, $\overline{f}_\alpha^{S,n+1}$ can be calculated by Eq. (2) from $\overline{\mathbf{W}}_i^{n+1}$.

III. IMPROVED DISCRETE VELOCITY METHOD WITH INNER ITERATION

In the IDVM, both the discrete distribution functions and the conservative variables are evolved with one forward sweep and one backward sweep in each step by solving the DVBE and the macroscopic governing equations, respectively. However, since the number of discrete distribution functions in the velocity space is far larger than that of conservative variables, the computational cost of the IDVM is actually dominated by the calculation of the DVBE. Furthermore, the accuracy of the predicted equilibrium state given by the macroscopic governing equations will affect the convergence rate of the calculation of DVBE, especially in the continuum flow regime. To obtain a sought-after predicted equilibrium state, the inner iteration is introduced into the solution of macroscopic governing equations.

A. Inner iteration for solving the macroscopic governing equations

Since the temporal term will not affect the steady-state solution, we can multiply the left-hand side of Eq. (14) by the adaptive parameter β defined in Eq. (17), which yields

$$\beta^n \frac{V_i}{\Delta t_i^n}\left(\overline{\mathbf{W}}_i^{n+1} - \mathbf{W}_i^n\right) = -\mathbf{R}_i^{n+1} = -\left(\mathbf{R}_i^n + \Delta \mathbf{R}_i^n\right). \tag{22}$$

The effect of β in the above equation is to enlarge the local time step in the continuum flow regime to achieve a faster convergence rate while remaining its value unchanged in the rarefied flow regime to guarantee good numerical stability. According to Eq. (15), $\Delta \mathbf{R}_i^n$ can be written as

$$\Delta \mathbf{R}_i^n = \sum_{j\in N(i)} \mathbf{n}_{ij}\cdot\left(\mathbf{F}_{ij}^{n+1} - \mathbf{F}_{ij}^n\right) S_{ij} \approx \sum_{j\in N(i)} \mathbf{n}_{ij}\cdot\left(\overline{\mathbf{F}}_{ij,NS}^{n+1} - \mathbf{F}_{ij,NS}^n\right) S_{ij}, \tag{23}$$

where $\mathbf{F}_{ij,NS}^n$ and $\overline{\mathbf{F}}_{ij,NS}^{n+1}$ are the numerical fluxes of Navier–Stokes equations calculated by the macroscopic flow variables at the current time level and the predicted macroscopic flow variables at the new time level, respectively. In Eq. (23), the time increment of the macroscopic residual $\Delta \mathbf{R}_i^n$ is approximated by the difference of numerical fluxes of Navier–Stokes equations. This strategy is more accurate than that used

in the previous implicit IDVM, as shown in Eq. (19). Since $\overline{\mathbf{F}}_{ij,NS}^{n+1}$ is still unknown, Eq. (22) cannot be solved directly. By introducing a time derivative term with respect to the pseudo time η into the left-hand side of Eq. (22), we have

$$V_i\frac{\partial\overline{\mathbf{W}}_i^{n+1}}{\partial\eta}+\beta^n\frac{V_i}{\Delta t_i^n}\left(\overline{\mathbf{W}}_i^{n+1}-\mathbf{W}_i^n\right)=-\left(\mathbf{R}_i^n+\Delta\mathbf{R}_i^n\right). \tag{24}$$

Substituting Eq. (23) into Eq. (24) and discretizing the resultant equation by the backward Euler scheme, we can get

$$\frac{V_i}{\Delta\eta_i^{n+1,m}}\Delta\overline{\mathbf{W}}_i^{n+1,m}=-\left(\mathbb{R}_i^{n+1,m}+\Delta\mathbb{R}_i^{n+1,m}\right) \tag{25}$$

with

$$\mathbb{R}_i^{n+1,m}=\mathbf{R}_i^n+\sum_{j\in N(i)}\mathbf{n}_{ij}\cdot\left(\overline{\mathbf{F}}_{ij,NS}^{n+1,m}-\mathbf{F}_{ij,NS}^n\right)S_{ij}+\beta^n\frac{V_i}{\Delta t_i^n}\left(\overline{\mathbf{W}}_i^{n+1,m}-\mathbf{W}_i^n\right), \tag{26}$$

$$\Delta\mathbb{R}_i^{n+1,m}=\sum_{j\in N(i)}\mathbf{n}_{ij}\cdot\Delta\overline{\mathbf{F}}_{ij,NS}^{n+1,m}S_{ij}+\beta^n\frac{V_i}{\Delta t_i^n}\Delta\overline{\mathbf{W}}_i^{n+1,m}. \tag{27}$$

Here, $\Delta\overline{\mathbf{W}}_i^{n+1,m}=\overline{\mathbf{W}}_i^{n+1,m+1}-\overline{\mathbf{W}}_i^{n+1,m}$ and $\Delta\overline{\mathbf{F}}_{ij,NS}^{n+1,m}=\overline{\mathbf{F}}_{ij,NS}^{n+1,m+1}-\overline{\mathbf{F}}_{ij,NS}^{n+1,m}$ are the time increments of the predicted conservative variable vector and the flux vector of Navier–Stokes equations at the pseudo time level m of the inner iteration. At the end of the inner iteration, we can take $\overline{\mathbf{W}}_i^{n+1,m+1}$ as the approximation of $\overline{\mathbf{W}}_i^{n+1}$.

In the inner iteration, $\sum_{j\in N(i)}\mathbf{n}_{ij}\cdot\Delta\overline{\mathbf{F}}_{ij,NS}^{n+1,m}S_{ij}$ can be further approximated by the following Euler equations-based flux splitting method:

$$\sum_{j\in N(i)}\mathbf{n}_{ij}\cdot\Delta\overline{\mathbf{F}}_{ij,NS}^{n+1,m}S_{ij}=\frac{1}{2}\sum_{j\in N(i)}\left[\mathbf{n}_{ij}\cdot\Delta\overline{\mathbf{F}}_{c,j}^{n+1,m}+r_{ij}^{n+1,m}\left(\Delta\overline{\mathbf{W}}_i^{n+1,m}-\Delta\overline{\mathbf{W}}_j^{n+1,m}\right)\right]S_{ij}, \tag{28}$$

where $\Delta\overline{\mathbf{F}}_{c,j}^{n+1,m}=\mathbf{F}_c(\overline{\mathbf{W}}_j^{n+1,m}+\Delta\overline{\mathbf{W}}_j^{n+1,m})-\mathbf{F}_c(\overline{\mathbf{W}}_j^{n+1,m})$. In the above equation, $\sum_{j\in N(i)}\mathbf{n}_{ij}\cdot\Delta\overline{\mathbf{F}}_{c,i}^{n+1,m}S_{ij}=0$ for an arbitrary closed region has been used. Note that the above approximation will not destroy the accuracy of $\Delta\mathbf{R}_i^n$ estimated by Eq. (23) since the converged solution of Eq. (25) will always yield $\overline{\mathbf{F}}_{ij,NS}^{n+1}=\overline{\mathbf{F}}_{ij,NS}^{n+1,m+1}$. Substituting Eq. (28) into Eq. (25), we have

$$G_i^{n+1,m}\Delta\overline{\mathbf{W}}_i^{n+1,m}+\frac{1}{2}\sum_{j\in N(i)}\left[\mathbf{n}_{ij}\cdot\Delta\overline{\mathbf{F}}_{c,j}^{n+1,m}-r_{ij}^{n+1,m}\Delta\overline{\mathbf{W}}_j^{n+1,m}\right]S_{ij}=-\mathbb{R}_i^{n+1,m}, \tag{29}$$

where $G_i^{n+1,m}=\beta^n\frac{V_i}{\Delta t_i^n}+\frac{V_i}{\Delta\eta_i^{n+1,m}}+\frac{1}{2}\sum_{j\in N(i)}r_{ij}^{n+1,m}S_{ij}$. Equation (29) can be resolved by the following forward and backward sweeps:

$$G_i^{n+1,m}\widetilde{\Delta\overline{\mathbf{W}}}_i^{n+1,m}+\frac{1}{2}\sum_{j\in L(i)}\left[\mathbf{n}_{ij}\cdot\Delta\overline{\mathbf{F}}_{c,j}^{n+1,m}-r_{ij}^{n+1,m}\widetilde{\Delta\overline{\mathbf{W}}}_j^{n+1,m}\right]S_{ij}=-\mathbb{R}_i^{n+1,m}, \tag{30}$$

$$G_i^{n+1,m}\Delta\overline{\mathbf{W}}_i^{n+1,m}+\frac{1}{2}\sum_{j\in U(i)}\left[\mathbf{n}_{ij}\cdot\Delta\overline{\mathbf{F}}_{c,j}^{n+1,m}-r_{ij}^{n+1,m}\Delta\overline{\mathbf{W}}_j^{n+1,m}\right]S_{ij}=G_i^{n+1,m}\widetilde{\Delta\overline{\mathbf{W}}}_i^{n+1,m}. \tag{31}$$

Here, $\widetilde{\Delta\overline{\mathbf{W}}}_i^{n+1,m}$ denotes the intermediate result of the forward sweep. In the practical calculation, we can set $\overline{\mathbf{W}}_i^{n+1,1}=\mathbf{W}_i^n$ and $\overline{\mathbf{F}}_{ij,NS}^{n+1,1}=\mathbf{F}_{ij,NS}^n$ in the first step of the inner iteration ($m=1$) and choose $\Delta\eta_i^{n+1,m}=\Delta t_i^n$ for simplicity. The effect of the inner iteration on the overall efficiency and accuracy will be assessed in Sec. IV.

B. Velocity space discretization and computational sequence

In the simulation, both the Gauss–Hermite rule and the rectangle rule are adopted for numerical integration in the velocity space. For low-speed flow at a low Knudsen number, since the distribution function is almost symmetric about the original point and it is very close to the equilibrium state, the Gauss–Hermite rule can achieve high accuracy with relatively few discrete points in the velocity space. In this quadrature rule, both the abscissas and weights are determined by the Hermite polynomials, which cannot be chosen freely. For high-speed flow or high Knudsen number flow, since the distribution function may become bizarre, the rectangle rule is more preferable. When using the rectangle rule, the unstructured mesh is a natural way to discretize the velocity space. In this way, the abscissas and weights are, respectively, the centroids and the volumes of cells. Due to its flexibility, the unstructured mesh in the velocity space can be easily refined in the region where the value of the distribution function is large and coarsened in the region where its value is small. As a result, the desired accuracy can be achieved by applying relatively fewer discrete points in the velocity space as compared with the uniform mesh.

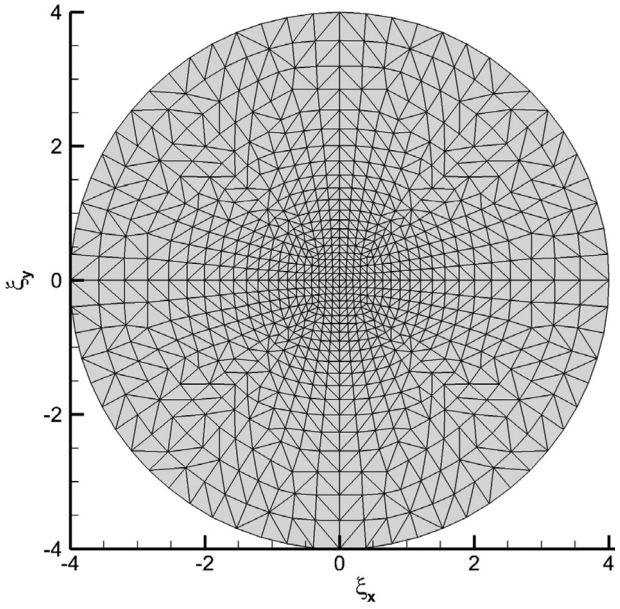

FIG. 1. Unstructured mesh in the velocity space for 2D lid-driven cavity flow.

In the DVM, since the conservative variable vector and the heat flux vector are calculated by the numerical quadrature of the discrete distribution functions and they are further applied to compute the equilibrium state, a difference between the numerical quadrature and the analytical integration may prevent the system from converging to its physical solution. An effective way to overcome this defect is the conservative correction, which enforces the following relationship:

$$\langle \boldsymbol{\psi}_1 (f^S - f) \rangle_\alpha = (0, \mathbf{0}, 0, -Pr\mathbf{q})^T \tag{32}$$

for all control volumes in the physical space.[54,55] Here, $\boldsymbol{\psi}_1 = (1, \boldsymbol{\xi}, \xi^2/2, \mathbf{c}c^2/2)^T$. Equation (32) is a nonlinear system for variables $(\rho, \mathbf{u}, T, \mathbf{q})^T$, which can be solved efficiently by the Newton iteration. Usually, one or two iterations are enough since the Newton algorithm converges rapidly. In our simulation, we set the convergence criterion of the Newton iteration as the infinite norm of the error vector of two adjacent iterations being less than 10^{-9}.

Since the inner iteration is only introduced into the solution of macroscopic governing equations, the main process of the present method is the same as that of the previous implicit IDVM. Suppose the indicator of the outer loop is n; the computational process of the developed method can be briefly summarized as follows:

(1) Calculate the left and right states of the discrete distribution function at the cell interface, $f_{ij,\alpha}^{L,n}$ and $f_{ij,\alpha}^{R,n}$, from those at cell centers by the second-order interpolation with a slope limiter function.
(2) Reconstruct the discrete distribution function at the cell interface $f_{ij,a}^n$ by Eq. (11) and compute the macroscopic flux attributed to $f_{ij,a}^n$ by the first term on the right-hand side of Eq. (16), $\mathbf{F}_{ij,DVM}^n$.
(3) Start the inner iteration for resolving the macroscopic governing equations. Suppose the indicator of the inner loop is m, the calculation process of the inner iteration can be summarized as step (a) to step (d).
    (a) Compute $\overline{\mathbf{F}}_{ij,NS}^{n+1,m}$ by the Navier–Stokes solver from the predicted conservative variable vector $\overline{\mathbf{W}}_i^{n+1,m}$. If $m = 1$,

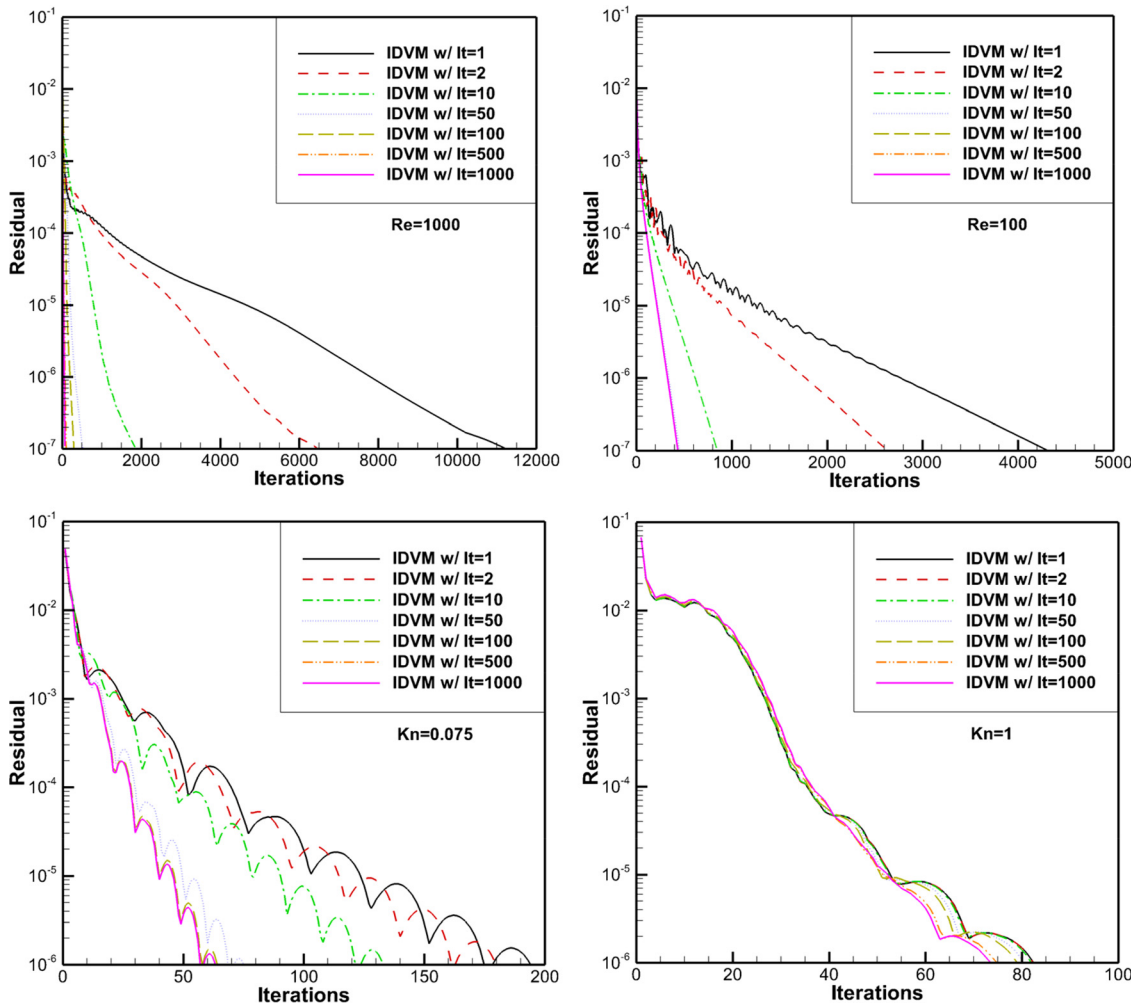

FIG. 2. Convergence histories of IDVM with different inner iterations for 2D lid-driven cavity flow at different Reynolds/Knudsen numbers.

TABLE I. Effect of the inner iteration on the total computational time (minutes) of the present method for 2D lid-driven cavity flow at different Reynolds/Knudsen numbers.

Inner iterations	$Re = 1000$	$Re = 100$	$Kn = 0.075$	$Kn = 1$	$Kn = 10$
1	192.99	78.60	4.76	4.14	4.20
2	118.98	49.87	4.35	4.14	4.21
10	36.44	17.01	3.27	4.09	4.21
50	13.63	12.11	1.92	4.11	4.31
100	10.67	15.71	1.71	4.15	4.39
500	10.06	44.88	2.27	4.53	5.13
1000	12.50	82.56	3.01	5.28	6.10

TABLE II. Comparison of computational times (minutes) of different methods for 2D lid-driven cavity flow at different Reynolds/Knudsen numbers. Note: Speed-up ratio 1 and speed-up ratio 2 are defined by the ratios of computational time of IDVM w/o It and Semi DVM over that of IDVM w/ It, respectively.

Schemes	$Re = 1000$	$Re = 100$	$Kn = 0.075$	$Kn = 1$	$Kn = 10$
IDVM w/ It	13.63	12.11	1.92	4.11	4.31
IDVM w/o It	192.99	78.60	4.76	4.14	4.20
Semi DVM	6975.18	322.54	5.07	4.13	4.20
Speed-up ratio 1	14.16	6.49	2.48	1.01	0.97
Speed-up ratio 2	511.75	26.63	2.64	1.00	0.97

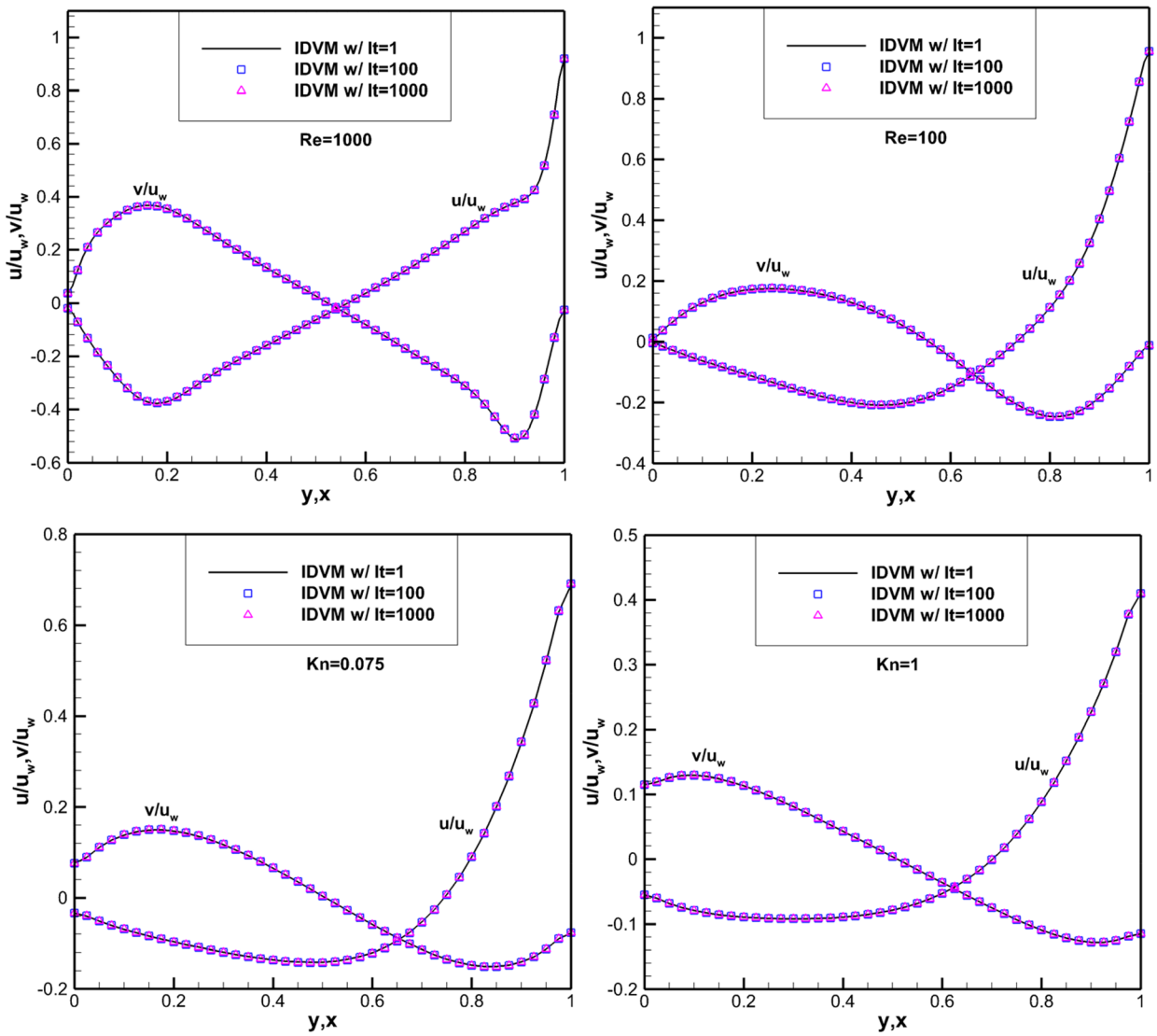

FIG. 3. Velocity profiles along the vertical and horizontal central lines obtained by the IDVM with different inner iterations for 2D lid-driven cavity flow at different Reynolds/Knudsen numbers.

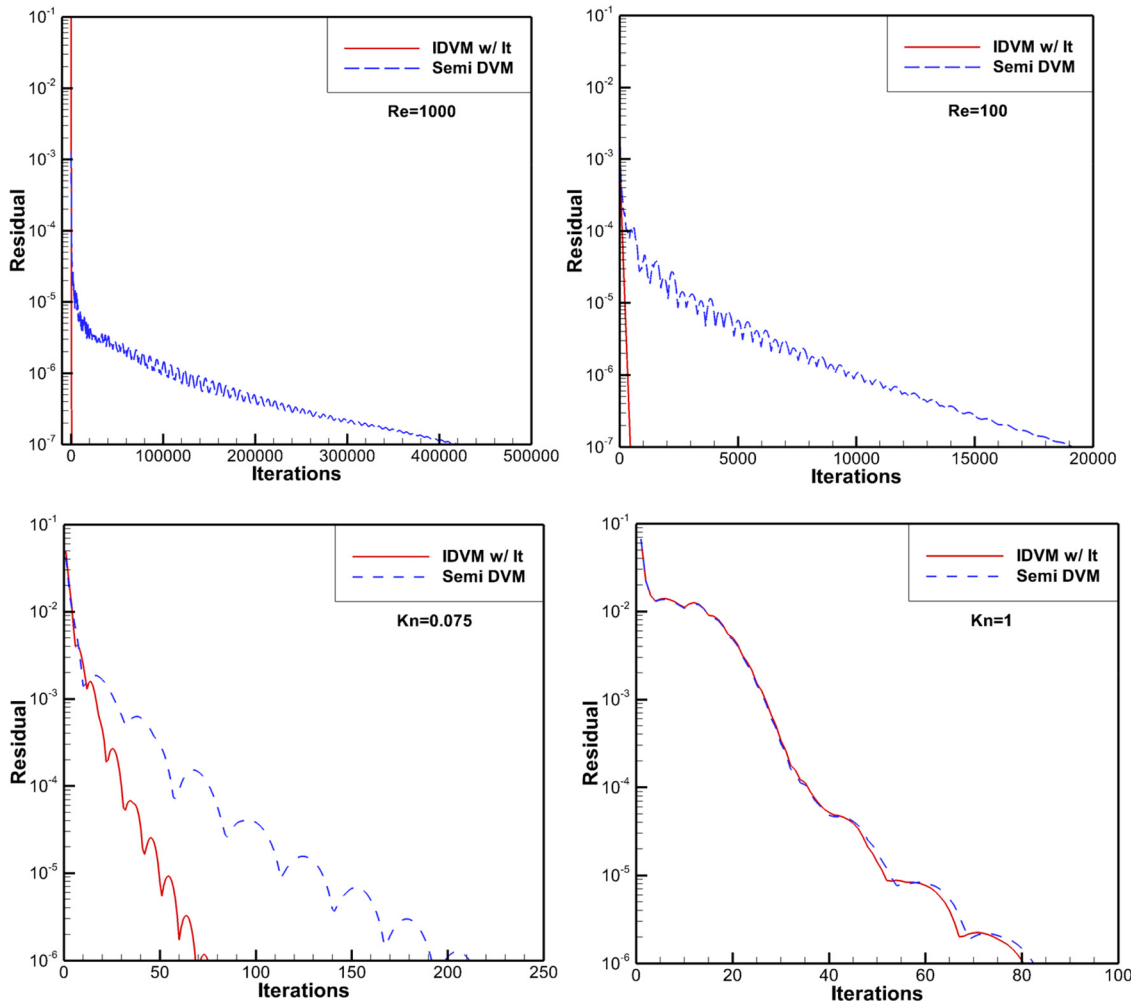

FIG. 4. Comparison of convergence histories of the IDVM with inner iteration and semi-implicit DVM for 2D lid-driven cavity flow at different Reynolds/Knudsen numbers.

set $\overline{\mathbf{W}}_i^{n+1,1} = \mathbf{W}_i^n$ and $\overline{\mathbf{F}}_{ij,NS}^{n+1,1} = \mathbf{F}_{ij,NS}^n$, and calculate the macroscopic residual by $\mathbf{R}_i^n = \sum_{j\in N(i)} \mathbf{n}_{ij} \cdot \mathbf{F}_{ij}^n S_{ij}$, where $\mathbf{F}_{ij}^n = \beta^n \mathbf{F}_{ij,DVM}^n + (1-\beta^n)\mathbf{F}_{ij,NS}^n$.

(b) Calculate $\mathbb{R}_i^{n+1,m}$ by Eq. (26) and compute the time increment of the predicted conservative variable vector $\Delta\overline{\mathbf{W}}_i^{n+1,m}$ by Eq. (29).

(c) Update the predicted conservative variable vector by $\overline{\mathbf{W}}_i^{n+1,m+1} = \overline{\mathbf{W}}_i^{n+1,m} + \Delta\overline{\mathbf{W}}_i^{n+1,m}$.

(d) Repeat steps (a)–(c) until the maximum iteration number of the inner iteration is arrived. At the end of the inner iteration, set $\overline{\mathbf{W}}_i^{n+1} = \overline{\mathbf{W}}_i^{n+1,m+1}$.

(4) Calculate the predicted equilibrium state $\overline{f}_{i,\alpha}^{S,n+1}$ by Eqs. (2) and (3) from the solution of macroscopic governing equations, $\overline{\mathbf{W}}_i^{n+1}$. In this step, the heat flux vector can be approximated by its value at the last time step $\mathbf{q}^n$.

(5) Compute the microscopic residual by Eq. (9) and solve Eq. (8) by the LU-SGS method to determine the time increment of the discrete distribution function $\Delta f_{i,a}^n$.

(6) Update the discrete distribution function by $f_{i,\alpha}^{n+1} = f_{i,\alpha}^n + \Delta f_{i,\alpha}^n$ and the conservative variable vector $\mathbf{W}_i^{n+1}$ as well as the heat flux vector $\mathbf{q}_i^{n+1}$ by taking moments of $f_{i,\alpha}^{n+1}$.

(7) Solve Eq. (32) by the Newton iteration to correct the conservative variable vector and the heat flux vector.

(8) Repeat steps (1)–(7) until the convergence criterion is satisfied.

IV. NUMERICAL EXAMPLES

Four test cases will be resolved to validate the overall performance of the present method in this section. First, we will investigate the influence of the inner iteration on the efficiency and accuracy of the developed method by the test case of 2D lid-driven cavity flow.

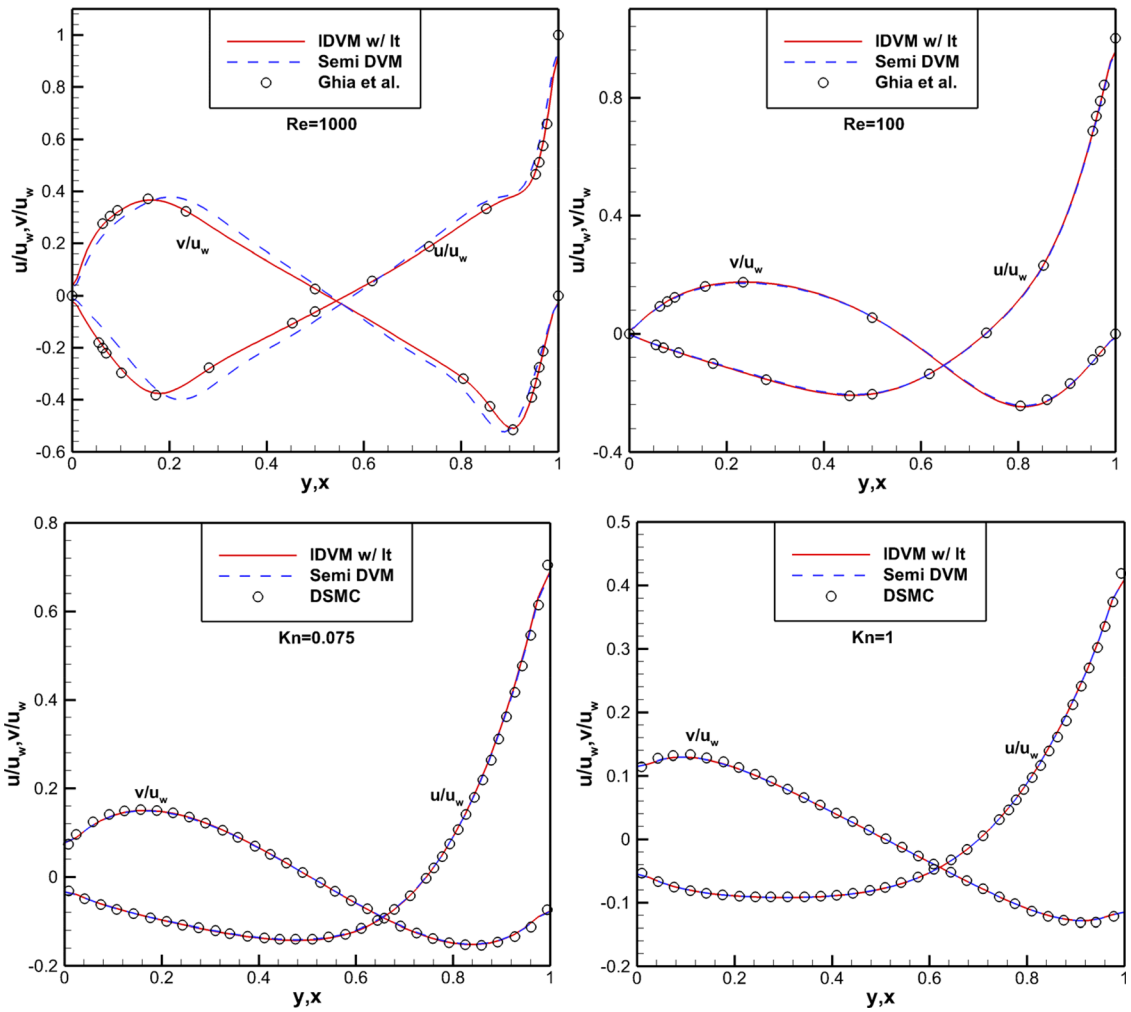

FIG. 5. Comparison of velocity profiles along the vertical and horizontal central lines obtained by the IDVM with inner iteration and the semi-implicit DVM for 2D lid-driven cavity flow at different Reynolds/Knudsen numbers.

Guidance for choosing the maximum iteration number of the inner iteration is then suggested for flows from the continuum flow regime to the free molecular flow regime. After that, we will focus on the comparison of the performance of the current method with the semi-implicit DVM by the rest test cases. Unless otherwise stated, the argon gas is assumed in the simulation, which results in the specific heat ratio of $5/3$, the Prandtl number of $2/3$, and the gas constant of 208.13. The Courant number σ for determination of Δt is adjusted individually for different cases to achieve better numerical stability. In addition, the first two cases are carried out on a personal computer (PC) with a processor of Intel(R) Xeon(R) Gold 6226R CPU at 2.9 GHz, and no parallel computation is utilized, while the open multi-processing (OpenMP) parallel technique with 16 and 24 cores is adopted to accelerate the calculation for the third and the fourth cases, respectively. For brevity, the results obtained by the semi-implicit DVM, the IDVM without inner iteration (i.e., the previous implicit IDVM) and the current method are denoted as "Semi DVM," "IDVM w/o It" and "IDVM w/ It" in the following texts, respectively.

Case 1: 2D lid-driven cavity flow

A classic test example for validating the performance of a newly developed method in all flow regimes is the lid-driven cavity flow. In this test case, the argon gas is filled in a square cavity with an edge length of L. All walls of the cavity are stationary except the top wall moving with a velocity of $u_W = 0.15\sqrt{2R_g T_0}$, where T_0 is the reference temperature. The temperature of all walls is fixed at T_0. In the simulation, two test cases with different Reynolds numbers of $Re = 100$ and 1000 and three test cases with different Knudsen numbers of $Kn = 0.075$, 1 and 10 are resolved. The test cases of $Re = 100$ and 1000 belong to the continuum flow regime, and test examples of $Kn = 0.075$, 1, and 10, respectively, correspond to the slip flow regime, the transition flow regime, and the free molecular flow regime. For the continuous case (the first two test cases), the dynamic viscosity is determined by

$$\mu = \frac{\rho_0 u_W L}{Re}, \tag{33}$$

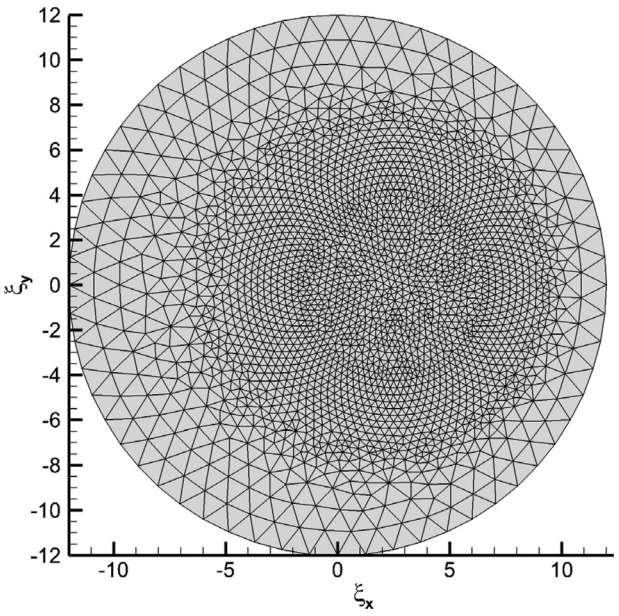

FIG. 6. Unstructured mesh in the velocity space for flow around a circular cylinder.

where ρ_0 is the reference density and Re is the Reynolds number. For the rarefied case (the last three test cases), the dynamic viscosity is computed by

$$\mu = \mu_0 \left(\frac{T}{T_0}\right)^w. \tag{34}$$

Here, w is a coefficient relevant to the intermolecular interaction model, which is taken as 0.81 in this test example. The reference viscosity μ_0 is calculated by

$$\frac{\mu_0}{L} = \frac{5\rho_0(2\pi R_g T_0)^{1/2}}{16} Kn, \tag{35}$$

where Kn is the Knudsen number. For discretization of the physical space, the unstructured uniform mesh with 10 000 and 1600 quadrilateral cells is used for the continuous case and the rarefied case, respectively. For discretization of the velocity space, the Gauss–Hermite quadrature rule with 8×8 and 28×28 mesh points is, respectively, used in the continuous case and the case of $Kn = 0.075$. For the cases of $Kn = 1$ and 10, an unstructured mesh with 1648 triangular cells is utilized to discretize the velocity space and the rectangle rule is adopted for numerical quadrature, as shown in Fig. 1. The Courant number for evaluation of Δt is taken as $\sigma = 100$.

First, the influence of the inner iteration on the efficiency and accuracy of the developed method is investigated in all flow regimes. Seven different maximum iteration numbers of the inner iteration are considered. They are set as MaxIt = 1, 2, 10, 50, 100, 500, and 1000. Note that MaxIt = 1 corresponds to the IDVM without inner iteration. The convergence histories of the developed method with different inner iterations are shown in Fig. 2. It can be seen that the larger value of MaxIt has a faster convergence rate, while the acceleration becomes insignificant when MaxIt $\geq$ 50. In addition, the acceleration effect of the inner iteration is dramatically evident for the continuous case and becomes less noticeable as the Knudsen number increases. Indeed, the convergence histories of the test case of $Kn = 10$ with different inner iterations are the same, which are not shown in Fig. 2. The effect of the inner iteration on the total computational time of the present method for 2D lid-driven cavity flow at different Reynolds/Knudsen numbers is tabulated in Table I. Theoretically, the total computational time consists of the calculation of outer and inner iterations. However, since the inner iteration is only introduced into the calculation of macroscopic governing equations, the total computational time is still dominated by the outer iteration, especially for the test case with a large number of discrete distribution functions in the velocity space. As reported in this table, the total computational time decreases first and then increases with the increase of inner iteration numbers for all test cases, while the trend in the continuous case is more evident than that in the rarefied case. When MaxIt = 50, the speedup ratio of the present method to the IDVM without inner iteration is about 14 for the test case of $Re = 1000$, as reported in Table II. Since MaxIt = 50 has a good comprehensive benefit in all flow regimes, it will be adopted to the rest test cases in this work. Except for the efficiency, the effect of the inner iteration on the accuracy of the present method is depicted in Fig. 3. Clearly, the inner iteration will not affect the accuracy of the present method.

Second, the efficiency and accuracy of the developed method are compared with the semi-implicit DVM. The convergence histories of these two methods at different Reynolds/Knudsen numbers are shown in Fig. 4, and the velocity profiles along the vertical and horizontal central lines are displayed in Fig. 5. Also depicted in Fig. 5 are the reference data calculated by Ghia *et al.*[56] for the test cases of $Re = 100$ and 1000 and by using the Direct Simulation Monte Carlo (DSMC) method[57] for the rest test cases. Evidently, a significant acceleration is achieved for the present method as compared with the semi-implicit DVM, especially for the test cases of $Re = 100$ and 1000. In addition, since the collisional effect is included in the calculation of numerical flux, the present method can predict the velocity distribution correctly in all flow regimes, while an obvious deviation is observed for the semi-implicit DVM in the test case of $Re = 1000$. In comparison, the semi-implicit DVM usually requires using a very small cell size, which is on the order of the mean free path of molecular, to get reasonable solutions as commented by Xu.[58] The comparison of the computational times of the developed method and the semi-implicit DVM is tabulated in Table II. A speed-up ratio of one to two orders of magnitude is achieved for the present method in the continuum flow regime, while for high Knudsen number flows, the computational time of the present method is roughly the same as that of the semi-implicit DVM. This test example validates the high efficiency of the present method in all flow regimes.

Case 2: Flow around a circular cylinder

To assess the performance of the present method for high-speed flows, the hypersonic rarefied flow around a circular cylinder is simulated in this subsection. In this test case, the free-stream Mach number and Knudsen number are taken as $Ma = 5$ and $Kn = 0.1$, respectively. The reference viscosity is determined by

$$\frac{\mu_0}{R_0} = \frac{15\rho_0(2\pi R_g T_0)^{1/2}}{2(5-2w)(7-2w)} Kn, \tag{36}$$

where $R_0 = 0.01$ m is the radius of the cylinder, $\rho_0 = 8.582 \times 10^{-5}$ kg/m^3 is the free stream density, and $T_0 = 273$ K is the free stream temperature. In the simulation, the dynamic viscosity is

computed by Eqs. (34) and (36) using the variable hard-sphere (VHS) model with $w = 0.81$. The wall temperature is fixed at T_0. The computational domain, which has a far-field boundary at $15R_0$ away from the geometrical center, is discretized by a non-uniform mesh with 61 points in the radial direction and 60 points in the circumferential direction. The velocity space is discretized by an unstructured mesh with 4648 triangular cells, and the rectangle rule is adopted for numerical quadrature, as shown in Fig. 6. Besides, the Courant number for evaluation of Δt is taken as $\sigma = 1$ for better numerical stability.

Figure 7 compares the density, temperature, x, and y components of the heat flux vector obtained by the IDVM with inner iteration and the semi-implicit DVM. Since the Knudsen number of this test case is relatively large, the results of these two methods are consistent with each other. The comparison of density, u-velocity, and temperature distributions along the stagnation line and convergence histories calculated by the two methods are depicted in Fig. 8, and the comparison of pressure, shear stress, and normal heat flux distributions along the cylindrical surface from the stagnation point to the trailing edge are shown in Fig. 9. Also displayed in Figs. 8 and 9 are the results obtained by the DSMC method.[43] It can be seen that the results of these two methods are basically the same, and they agree well with the reference data. As for the convergence history, the IDVM with inner iteration is slightly faster than the semi-implicit DVM. The total computational times spent by the IDVM with inner iteration and the semi-implicit DVM are 1.00 and 1.07 h, respectively. This test example indicates that the accuracy, efficiency, and stability of these two methods are almost the same for solving hypersonic rarefied flows.

Case 3: 3D lid-driven cavity flow

This test example is the extension of the 2D case in the y-direction, as shown in Fig. 10. At the top wall, the lid is moving with a velocity of $u_W = 0.15\sqrt{2R_g T_0}$ in the x-direction. Other settings, including the gas properties, the wall temperature, and the calculation of the dynamic viscosity, are consistent with the 2D scenario. Four cases with different Knudsen numbers or Reynolds numbers are simulated in this test. They are the cases of $Kn = 1$ in the transition flow regime, $Kn = 0.1$ in the slip flow regime, and $Re = 100$ and 1000 in the

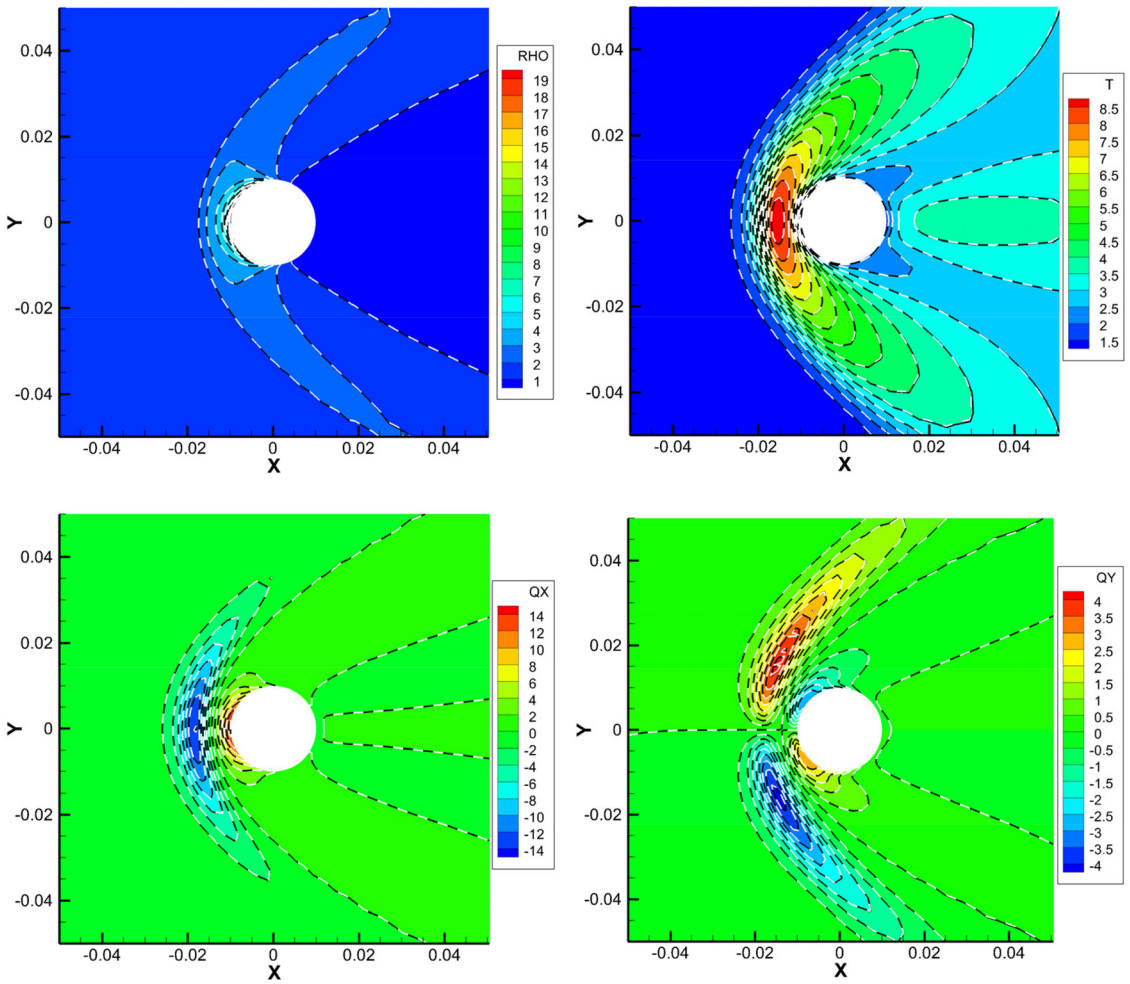

FIG. 7. Comparison of density, temperature, x and y components of the heat flux vector for flow around a circular cylinder (IDVM w/ It: colored background with black solid line; semi DVM: white dash line).

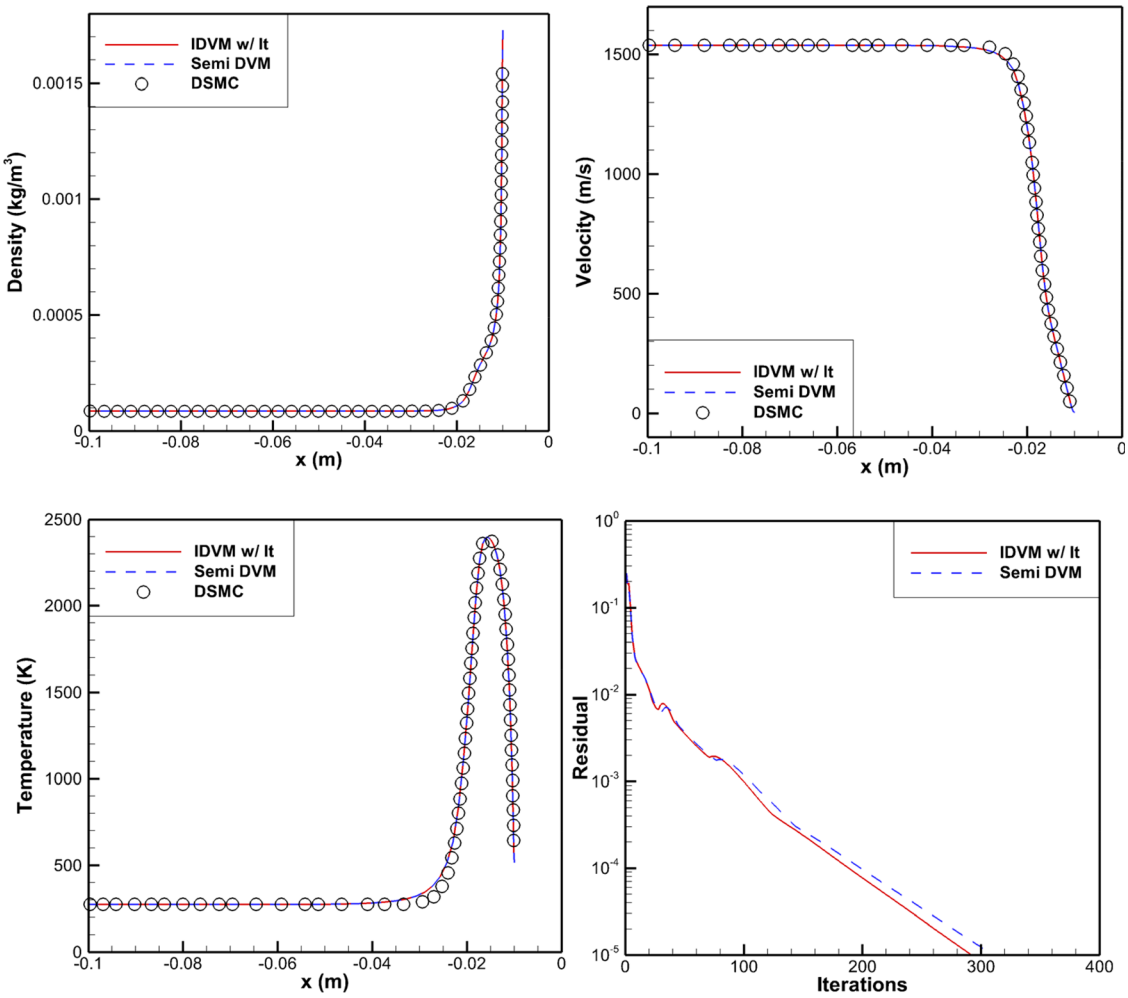

FIG. 8. Density, *u*-velocity, and temperature profiles along the stagnation line and convergence history for flow around a circular cylinder.

continuum flow regime. For $Kn = 1$, an unstructured mesh with 4000 hexahedral cells is used to discretize the physical space, and the velocity space is discretized by an unstructured mesh with 22 909 tetrahedral cells, as shown in Fig. 10. For $Kn = 0.1$, the physical space is discretized by an unstructured mesh with 13 500 hexahedral cells, and the numerical integration in the velocity space is performed by the Gauss–Hermite rule with $18 \times 18 \times 18$ mesh points. For the test cases in the continuum flow regime, a finer unstructured mesh with 10 800 hexahedral cells is adopted to discretize the physical space, and the Gauss–Hermite quadrature rule with a coarser mesh of $8 \times 8 \times 8$ points is used to do the numerical integration. The Courant number for evaluation of Δt is taken as $\sigma = 100$ in all simulations.

The comparison of convergence histories of the developed method and the semi-implicit DVM at different Reynolds/Knudsen numbers is depicted in Fig. 11. It can be seen that the present method has a faster convergence rate than the semi-implicit DVM, especially for the test cases in the continuum flow regime. Accordingly, a significant acceleration is achieved for the present method. As indicated in Table III, the speed-up ratio is about one to two orders of magnitude in the continuum flow regime. Figure 12 compares the velocity distributions along the vertical and horizontal central lines at $y = 0.5$ plane calculated by the developed method and the semi-implicit DVM. Also displayed in the plots of $Re = 100$ and 1000 are the benchmark data obtained by the differential quadrature (DQ) method.[59] For the test cases of $Re = 100$, $Kn = 0.1$ and 1, the results of the present method and the semi-implicit DVM are consistent with each other, while for the test case of $Re = 1000$, the results obtained by the present method are closer to the reference data than the semi-implicit DVM. The reason is that the collisional effect is involved in the calculation of numerical flux in the present method, while it is neglected in the semi-implicit DVM, which will result in a large numerical dissipation as the flow goes into the continuum flow regime at large Reynolds numbers. Figure 13 shows the temperature contours computed by the present method for test cases of $Kn = 0.1$ and 1, which are in line with the results of IDVM without inner iteration.[48]

Case 4: Flow around a 3D blunt body

First, supersonic rarefied flow with a free-stream Mach number of 3.834 and a Knudsen number of 0.03 around a sphere is simulated.

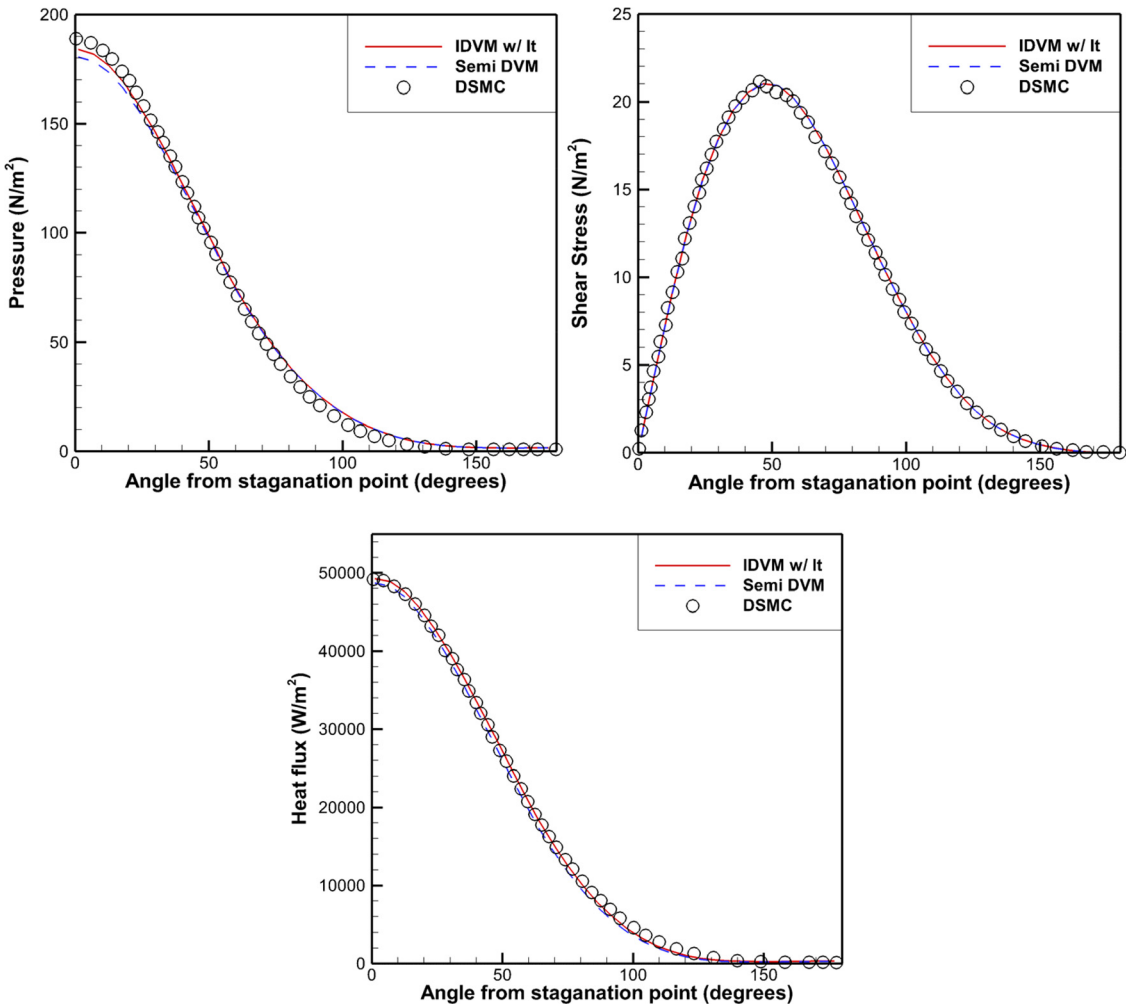

FIG. 9. Pressure, shear stress, and heat flux distributions along the cylindrical surface from the stagnation point to the trailing edge for flow around a circular cylinder.

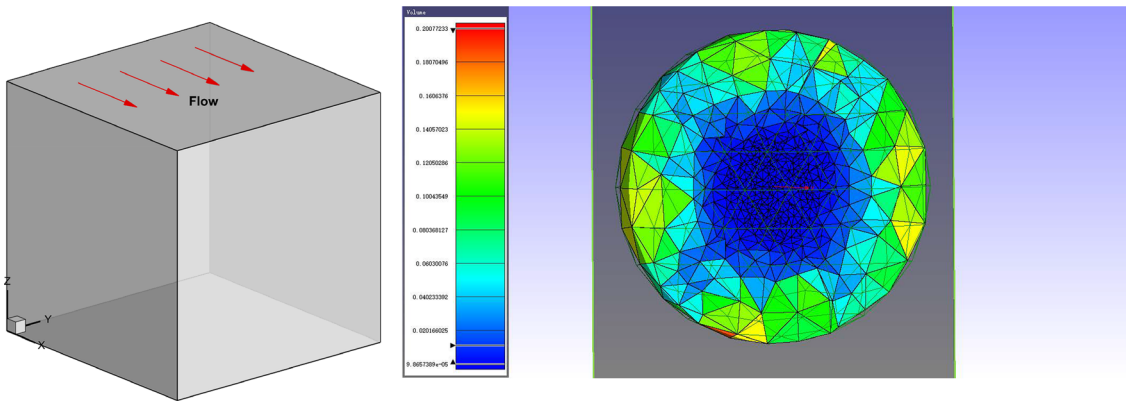

FIG. 10. Geometry (left) and unstructured mesh in the velocity space (right) for 3D lid-driven cavity flow.

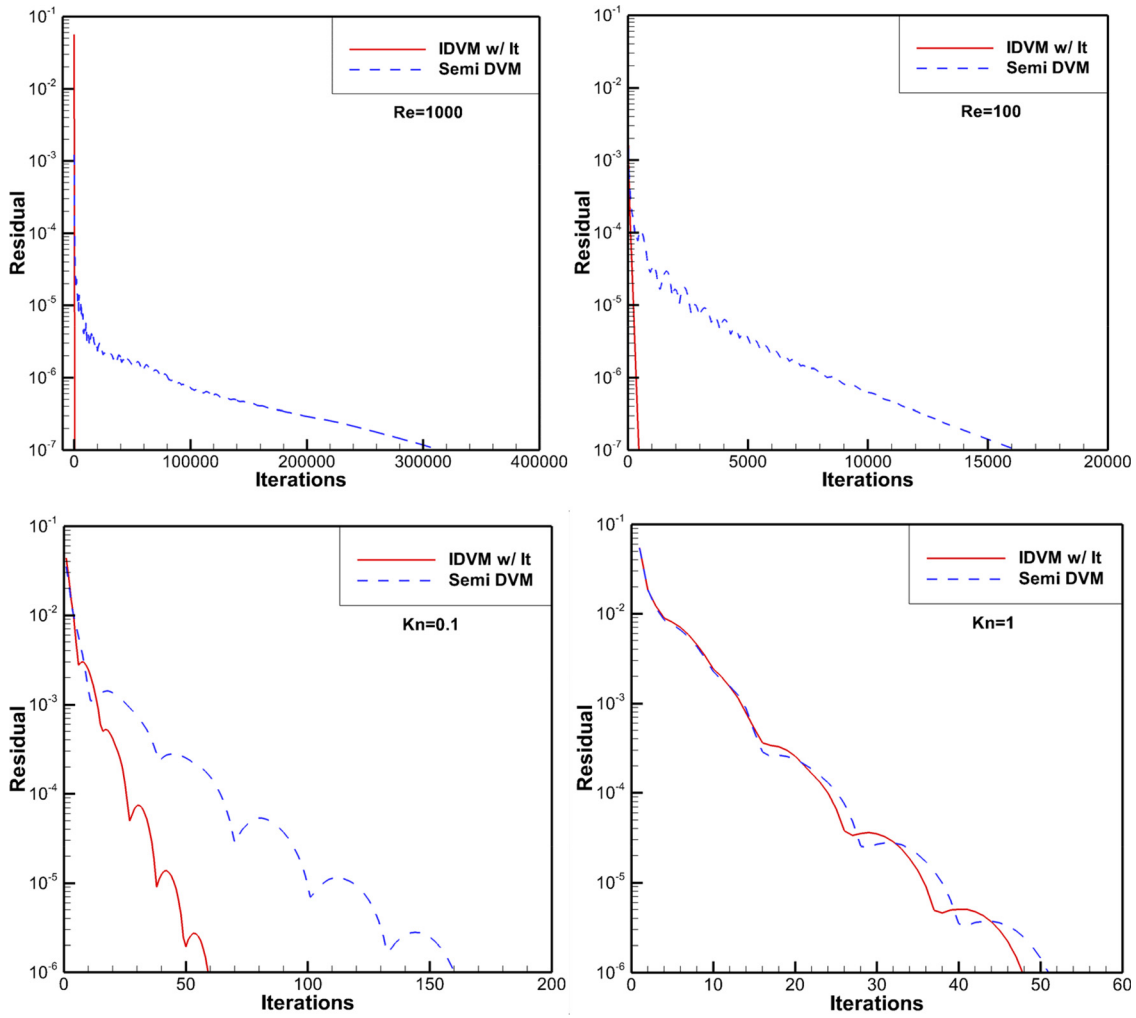

FIG. 11. Comparison of convergence histories of the IDVM with inner iteration and semi-implicit DVM for 3D lid-driven cavity flow at different Reynolds/Knudsen numbers.

In the simulation, the dynamic viscosity is calculated by Eqs. (34) and (35) with $w = 0.75$, in which the characteristic length is taken as the sphere diameter. The wall temperature is fixed at $T_W = [1 + (\gamma - 1)Ma^2/2]T_0$, where T_0 is the reference temperature. The spherical surface is discretized by 768 quadrilateral elements. The physical space, which has a far-field boundary at four times of sphere diameter away from the geometrical center, is discretized by an unstructured mesh with 20 736 hexahedral cells. The velocity space is discretized by an unstructured mesh with 36 053 tetrahedral cells, and the rectangle rule is adopted for numerical quadrature, as shown in Fig. 14. Figure 15 depicts the density and temperature contours around the sphere. The temperature and u-velocity distributions along the stagnation line are shown in Fig. 16 and compared with those of Li and Zhang computed by the gas kinetic unified algorithm (GKUA)[60] and Bird *et al.* by the DSMC method.[61] It can be found that the present results basically match well with the reference data. In addition, the drag coefficients calculated by the developed method, the GKUA, and the DSMC method are 1.4295, 1.3749, and 1.4122, respectively. The relative error of the present result over the result of the DSMC method is 1.2%. The total outer iteration numbers and the computational time of this test example by using the OpenMP parallel technique with 24 cores are 222 and 13.59 h, respectively.

TABLE III. Comparison of computational times (hours) of different methods for 3D lid-driven cavity flow at different Reynolds/Knudsen numbers.

Schemes	$Re = 1000$	$Re = 100$	$Kn = 0.1$	$Kn = 1$
IDVM w/ It	4.98	4.08	0.46	0.18
Semi DVM	1798.38	98.15	1.21	0.18
Speed-up ratio	361.12	24.06	2.63	1.00

Second, hypersonic rarefied flow with a free-stream Mach number of 5, a Knudsen number of 0. 601, and an angle of attack of 26° around an Orion Crew Module is tested to further assess the flexibility

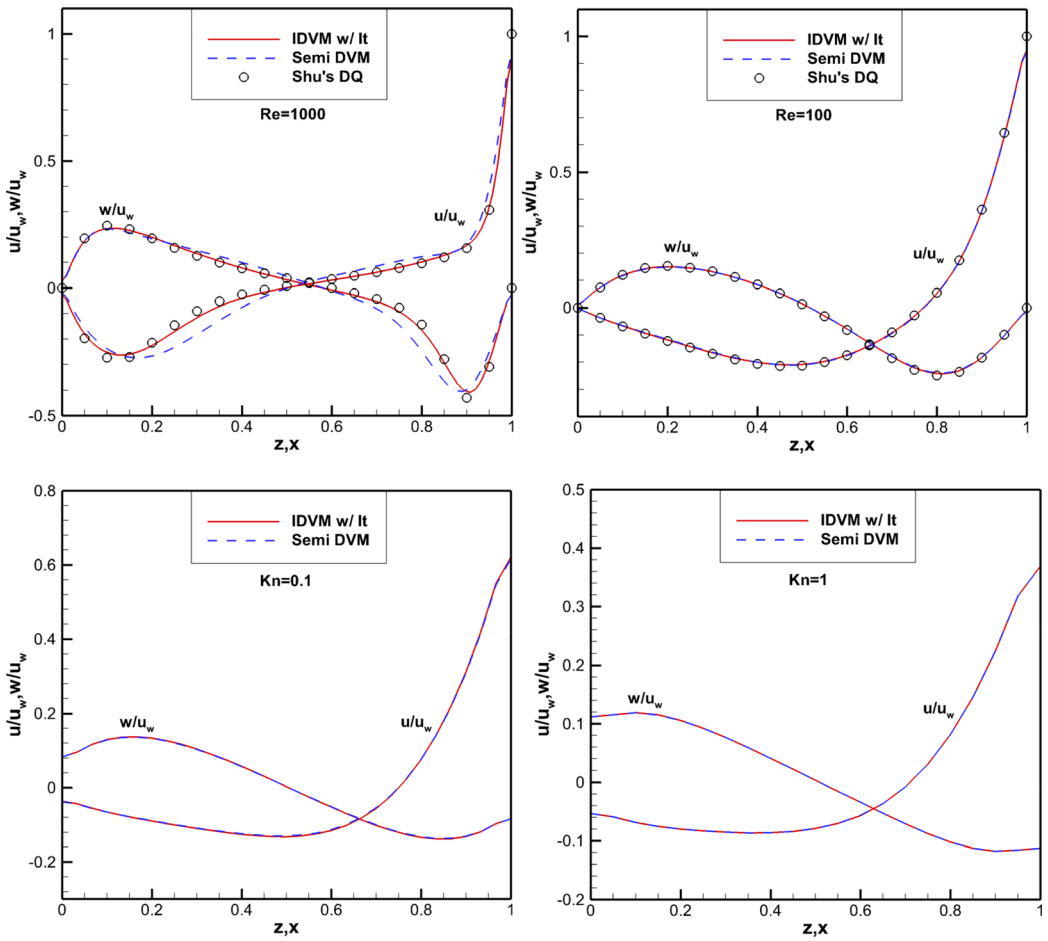

FIG. 12. Comparison of velocity profiles along the vertical and horizontal central lines at $y = 0.5$ plane obtained by the IDVM with inner iteration and the semi-implicit DVM for 3 D lid-driven cavity flow at different Reynolds/Knudsen numbers.

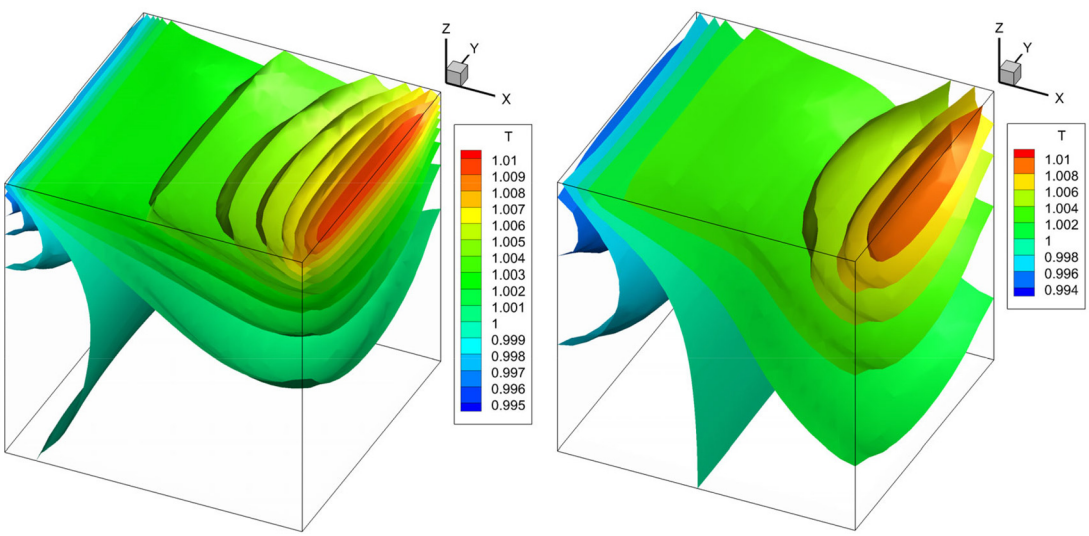

FIG. 13. Temperature contours for 3D lid-driven cavity flow at $Kn = 0.1$ (left) and $Kn = 1$ (right).

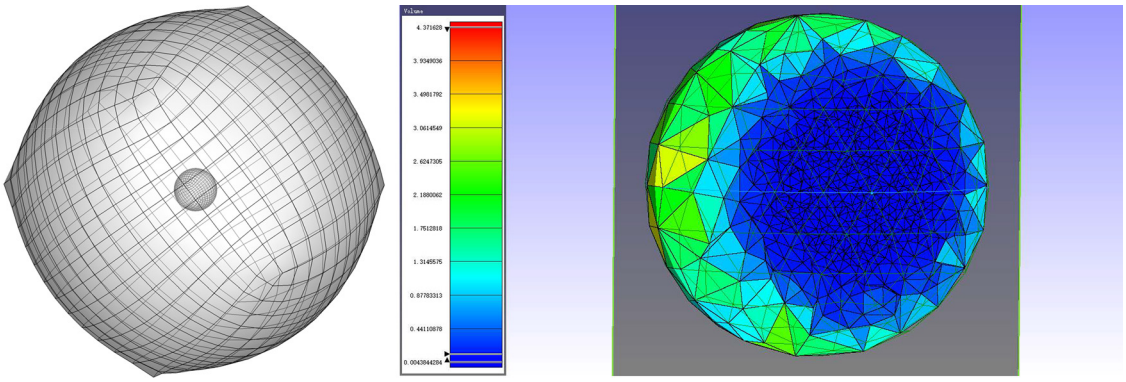

FIG. 14. Meshes in the physical space (left) and the velocity space (right) for supersonic flow around a sphere.

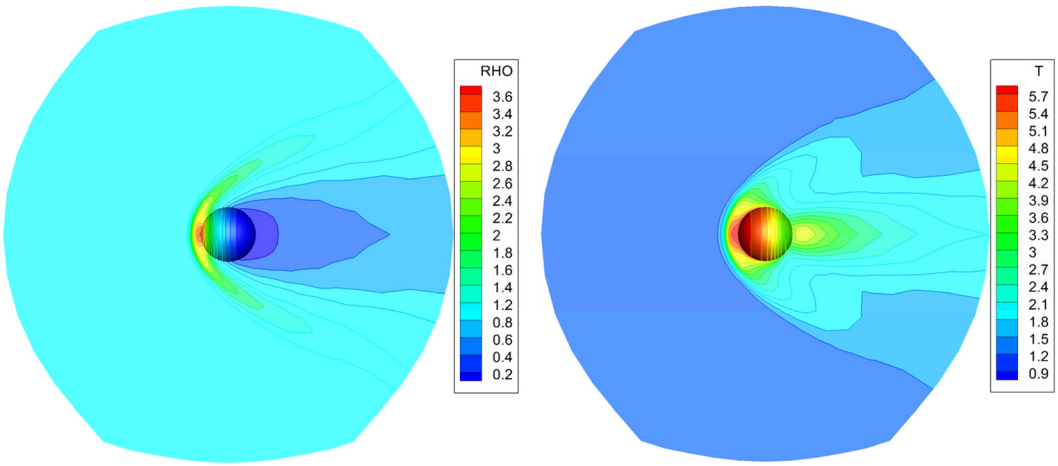

FIG. 15. Density (left) and temperature (right) contours for supersonic flow around a sphere.

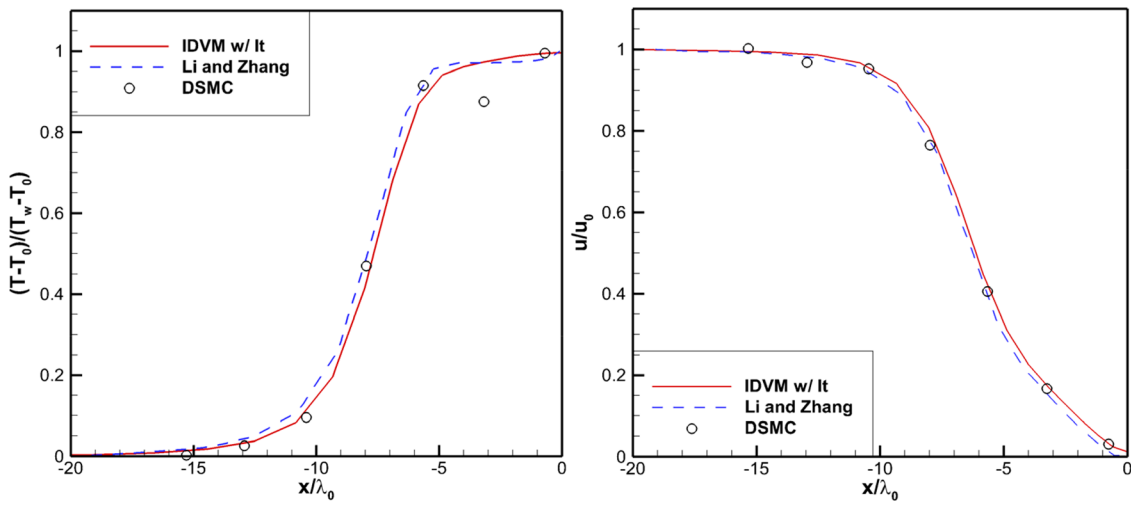

FIG. 16. Temperature (left) and *u*-velocity (right) profiles along the stagnation line for supersonic flow around a sphere.

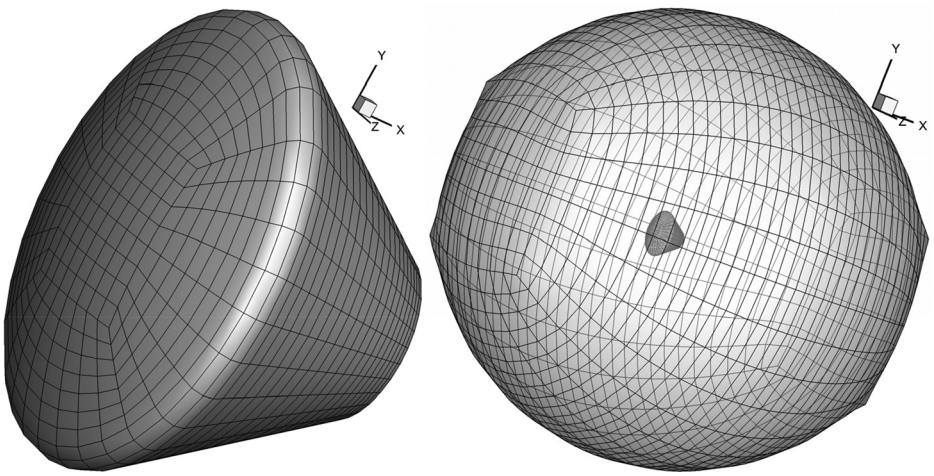

FIG. 17. Surface mesh (left) and mesh in the physical space (right) for hypersonic flow around an orion crew module.

and stability of the present method for solving more complex problems. The details of the geometry can be found in Ref. 62, which is discretized by 966 quadrilateral elements. As shown in Fig. 17, the physical space is discretized by an unstructured mesh with 37 440 hexahedral cells. The velocity space is discretized by an unstructured mesh with 36 053 tetrahedral cells, and the rectangle rule is adopted for numerical quadrature. In the simulation, the dynamic viscosity is computed by Eqs. (34) and (35) with $w = 0.5$, where the characteristic length is taken as $L = 5.0292$ m. Other parameters are corresponding to the condition at the altitude of 120 km, i.e., the average molecular weight is 26.159 g/mol, the number density is $n_0 = 5.2127 \times 10^{17}\ \mathrm{m}^{-3}$, the free-stream density is $\rho_0 = 2.2642 \times 10^{-8}\ \mathrm{kg/m}^3$, the free-stream temperature is $T_0 = 368$ K, the wall temperature is $T_W = 567$ K, and the reference area is 19.865 m^2. Figure 18 shows the density and the pressure contours at $z = 0$ plane as well as the surface of the orion crew module. Due to the relatively large Knudsen number, a diffused bow shock wave is observed in front of the object. The distributions of the surface pressure coefficient and the magnitude of the heat flux vector over the orion crew module are shown in Fig. 19, where the heat flux is normalized by $\rho_0(2R_gT_0)^{3/2}$. Since the wall temperature is far less than the stagnation temperature in this test case, the largest pressure and heating are formed at the stagnation point. The computed normal and axial force coefficients are 0.497 and 1.648, respectively. The total outer iteration numbers and the computational time of this test example by using the OpenMP parallel technique with 24 cores are 77 and 11.45 h, respectively.

V. CONCLUSIONS

In the previous implicit IDVM, the total computational cost is dominated by the calculation of the DVBE since the number of discrete distribution functions is far larger than that of the macroscopic conservative variables. Thus, the key to improving the overall efficiency of the previous IDVM is to minimize as much as possible the computational cost of the DVBE. To achieve this goal, an inner iteration is introduced into the solution of macroscopic governing equations to provide a more accurate predicted equilibrium state for the fully implicit discretization of the DVBE so as to speed up the convergence rate of its calculation. Since the predicted results are actually the solution of Navier–Stokes equations in the continuum flow regime, the convergence rate of the developed method can be accelerated greatly in this regime.

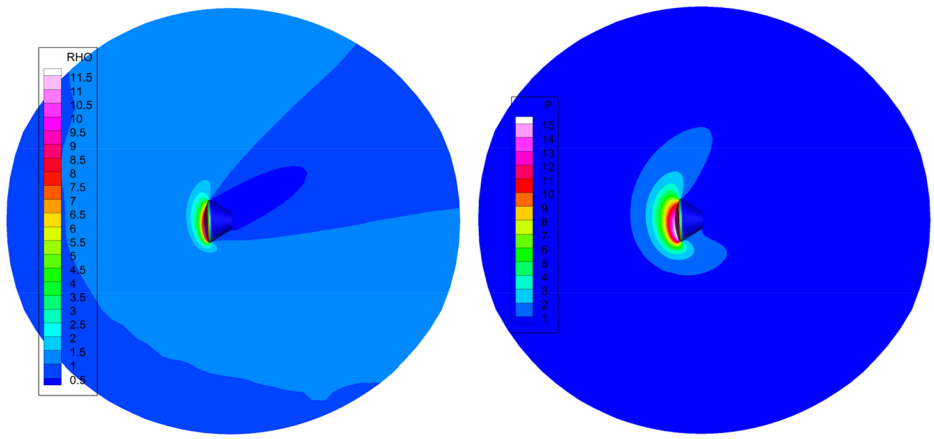

FIG. 18. Density (left) and pressure (right) contours for hypersonic flow around an orion crew module.

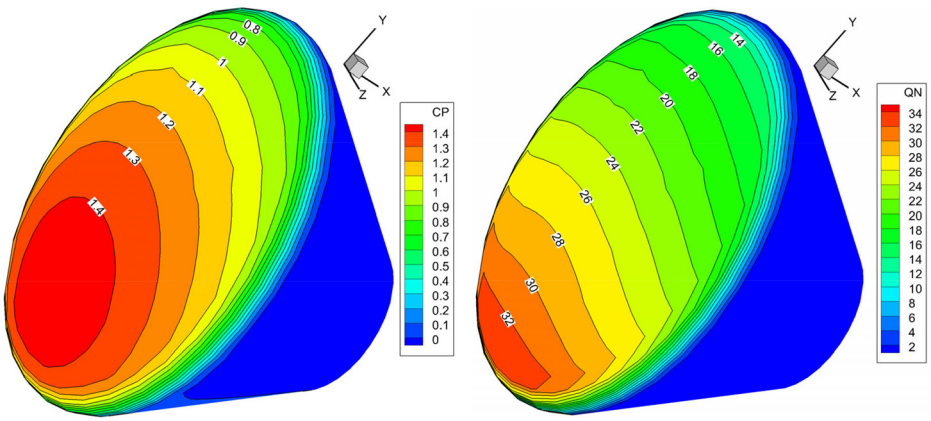

FIG. 19. Pressure coefficient (left) and magnitude of the heat flux vector (right) distributions on the surface of the orion crew module.

Four test examples, including the 2D lid-driven cavity flow, the hypersonic flow around a circular cylinder, the 3D lid-driven cavity flow, and the supersonic/hypersonic flow around a 3D blunt body, are resolved for comprehensive evaluation of the developed method in all flow regimes. Numerical results show that the inner iteration will not affect the accuracy and stability of the IDVM but improve the efficiency significantly in the continuum flow regime. As shown in the test case of lid-driven cavity flow at $Re = 1000$, the speed-up ratios of one order of magnitude and two orders of magnitude are achieved for the present method as compared with the previous implicit IDVM and the semi-implicit DVM, respectively.

ACKNOWLEDGMENTS

The research was supported by the Natural Science Foundation of Jiangsu Province (Grant No. BK20210273) and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

AUTHOR DECLARATIONS

Conflict of Interest

The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

DATA AVAILABILITY

The data that support the findings of this study are available within the article.

REFERENCES

[1]I. D. Boyd, "Modeling of associative ionization reactions in hypersonic rarefied flows," Phys. Fluids **19**(9), 096102 (2007).
[2]S. Mungiguerra, G. Zuppardi, and R. Savino, "Rarefied aerodynamics of a deployable re-entry capsule," Aerosp. Sci. Technol. **69**, 395–403 (2017).
[3]S. J. Plimpton, S. G. Moore, A. Borner, A. K. Stagg, T. P. Koehler, J. R. Torczynski, and M. A. Gallis, "Direct simulation Monte Carlo on petaflop supercomputers and beyond," Phys. Fluids **31**(8), 086101 (2019).
[4]Y. Chen, Y. Zhu, and K. Xu, "A three-dimensional unified gas-kinetic wave-particle solver for flow computation in all regimes," Phys. Fluids **32**(9), 096108 (2020).
[5]J. Fan and C. Shen, "Statistical simulation of low-speed rarefied gas flows," J. Comput. Phys. **167**(2), 393–412 (2001).
[6]S. Jaiswal, A. Pikus, A. Strongrich, I. B. Sebastião, J. Hu, and A. A. Alexeenko, "Quantification of thermally-driven flows in microsystems using Boltzmann equation in deterministic and stochastic contexts," Phys. Fluids **31**(8), 082002 (2019).
[7]X. Wang, W. Zhang, T. Su, Z. Zhang, and S. Zhang, "Gas-surface interaction effects on rarefied gas flows around microbeams induced by temperature fields," Int. J. Heat Mass Transfer **172**, 121186 (2021).
[8]D. Jiang, "Study of the gas-kinetic scheme based on the analytic solution of model equations," Ph.D. thesis (China Aerodynamics Research and Development Center, 2016).
[9]K. Xu and C. Liu, "A paradigm for modeling and computation of gas dynamics," Phys. Fluids **29**(2), 026101 (2017).
[10]J. E. Broadwell, "Study of rarefied shear flow by the discrete velocity method," J. Fluid Mech. **19**(3), 401–414 (1964).
[11]C. Buet, "A discrete-velocity scheme for the Boltzmann operator of rarefied gas dynamics," Transp. Theory Stat. Phys. **25**(1), 33–60 (1996).
[12]V. I. Kolobov, R. R. Arslanbekov, V. V. Aristov, A. A. Frolova, and S. A. Zabelok, "Unified solver for rarefied and continuum flows with adaptive mesh and algorithm refinement," J. Comput. Phys. **223**(2), 589–608 (2007).
[13]D. Valougeorgis and S. Naris, "Acceleration schemes of the discrete velocity method: Gaseous flows in rectangular microchannels," SIAM J. Sci. Comput. **25**(2), 534–552 (2003).
[14]C. Liu, K. Xu, Q. Sun, and Q. Cai, "A unified gas-kinetic scheme for continuum and rarefied flows. IV. Full Boltzmann and model equations," J. Comput. Phys. **314**, 305–340 (2016).
[15]W. Q. Hu and Z. H. Li, "Investigation on different discrete velocity quadrature rules in gas-kinetic unified algorithm solving Boltzmann model equation," Comput. Math. Appl. **75**(11), 4179–4200 (2018).
[16]Z. H. Li, W. Q. Hu, A. P. Peng, J. L. Wu, and C. H. Lee, "Gas-kinetic unified algorithm for plane external force-driven flows covering all flow regimes by modeling of Boltzmann equation," Int. J. Numer. Methods Fluids **92**(8), 922–949 (2020).
[17]Z. H. Li, W. Q. Hu, J. L. Wu, and A. P. Peng, "Improved gas-kinetic unified algorithm for high rarefied to continuum flows by computable modeling of the Boltzmann equation," Phys. Fluids **33**(12), 126114 (2021).
[18]L. B. Barichello and C. E. Siewert, "A new version of the discrete-ordinates method," in *Proceedings of the 2nd International Conference on Computational Heat and Mass Transfer (COPPE/EE/UFRJ)* (Federal University of Rio de Janeiro, 2001), pp. 22–26.
[19]L. B. Barichello, M. Camargo, P. Rodrigues, and C. E. Siewert, "Unified solutions to classical flow problems based on the BGK model," Z. Angew. Math. Phys. **52**(3), 517–534 (2001).
[20]E. Schiassi, R. Furfaro, C. Leake, M. De Florio, H. Johnston, and D. Mortari, "Extreme theory of functional connections: A fast physics-informed neural network method for solving ordinary and partial differential equations," Neurocomputing **457**, 334–356 (2021).
[21]M. De Florio, E. Schiassi, B. D. Ganapol, and R. Furfaro, "Physics-informed neural networks for rarefied-gas dynamics: Thermal creep flow in the Bhatnagar–Gross–Krook approximation," Phys. Fluids **33**(4), 047110 (2021).
[22]A. N. Kudryavtsev and A. A. Shershnev, "A numerical method for simulation of microflows by solving directly kinetic equations with WENO schemes," J. Sci. Comput. **57**(1), 42–73 (2013).

[23]C. Baranger, J. Claudel, N. Hérouard, and L. Mieussens, "Locally refined discrete velocity grids for stationary rarefied flow simulations," J. Comput. Phys. **257**, 572–593 (2014).
[24]Z. H. Li, L. Bi, H. X. Zhang, and L. Li, "Gas-kinetic numerical study of complex flow problems covering various flow regimes," Comput. Math. Appl. **61**(12), 3653–3667 (2011).
[25]J. L. Wu, Z. H. Li, A. P. Peng, X. C. Pi, and Z. H. Li, "Numerical study on rarefied unsteady jet flow expanding into vacuum using the gas-kinetic unified algorithm," Comput. Fluids **155**, 50–61 (2017).
[26]K. Xu and J. C. Huang, "A unified gas-kinetic scheme for continuum and rarefied flows," J. Comput. Phys. **229**(20), 7747–7764 (2010).
[27]S. Chen, K. Xu, C. Lee, and Q. Cai, "A unified gas kinetic scheme with moving mesh and velocity space adaptation," J. Comput. Phys. **231**(20), 6643–6664 (2012).
[28]S. Liu, P. Yu, K. Xu, and C. Zhong, "Unified gas-kinetic scheme for diatomic molecular simulations in all flow regimes," J. Comput. Phys. **259**, 96–113 (2014).
[29]L. Mieussens, "On the asymptotic preserving property of the unified gas kinetic scheme for the diffusion limit of linear kinetic models," J. Comput. Phys. **253**, 138–156 (2013).
[30]L. M. Yang, C. Shu, W. M. Yang, Z. Chen, and H. Dong, "An improved discrete velocity method (DVM) for efficient simulation of flows in all flow regimes," Phys. Fluids **30**(6), 062005 (2018).
[31]L. M. Yang, X. Zhao, C. Shu, and Y. J. Du, "Parametric reduced order modeling-based discrete velocity method for simulation of steady rarefied flows," J. Comput. Phys. **430**, 110037 (2021).
[32]W. Su, M. T. Ho, Y. Zhang, and L. Wu, "GSIS: An efficient and accurate numerical method to obtain the apparent gas permeability of porous media," Comput. Fluids **206**, 104576 (2020).
[33]L. Zhu, X. Pi, W. Su, Z. H. Li, Y. Zhang, and L. Wu, "General synthetic iterative scheme for nonlinear gas kinetic simulation of multi-scale rarefied gas flows," J. Comput. Phys. **430**, 110091 (2021).
[34]H. Liu, Y. Cao, Q. Chen, M. Kong, and L. Zheng, "A conserved discrete unified gas kinetic scheme for microchannel gas flows in all flow regimes," Comput. Fluids **167**, 313–323 (2018).
[35]R. Yuan and C. Zhong, "A conservative implicit scheme for steady state solutions of diatomic gas flow in all flow regimes," Comput. Phys. Commun. **247**, 106972 (2020).
[36]L. Mieussens, "Discrete-velocity models and numerical schemes for the Boltzmann-BGK equation in plane and axisymmetric geometries," J. Comput. Phys. **162**(2), 429–466 (2000).
[37]K. Aoki, P. Degond, and L. Mieussens, "Numerical simulations of rarefied gases in curved channels: Thermal creep, circulating flow, and pumping effect," Commun. Comput. Phys. **6**(5), 919–954 (2009).
[38]J. Y. Yang and J. C. Huang, "Rarefied flow computations using nonlinear model Boltzmann equations," J. Comput. Phys. **120**(2), 323–339 (1995).
[39]V. A. Titarev, "Numerical method for computing two-dimensional unsteady rarefied gas flows in arbitrarily shaped domains," Comput. Math. Math. Phys. **49**(7), 1197–1211 (2009).
[40]V. A. Titarev, "Implicit numerical method for computing three-dimensional rarefied gas flows on unstructured meshes," Comput. Math. Math. Phys. **50**(10), 1719–1733 (2010).
[41]L. Mieussens, "Discrete velocity model and implicit scheme for the BGK equation of rarefied gas dynamics," Math. Models Methods Appl. Sci. **10**(08), 1121–1149 (2000).
[42]Z. Guo, K. Xu, and R. Wang, "Discrete unified gas kinetic scheme for all Knudsen number flows: Low-speed isothermal case," Phys. Rev. E **88**(3), 033305 (2013).
[43]L. Zhu, Z. Guo, and K. Xu, "Discrete unified gas kinetic scheme on unstructured meshes," Comput. Fluids **127**, 211–225 (2016).
[44]Y. Zhu, C. Zhong, and K. Xu, "Implicit unified gas-kinetic scheme for steady state solutions in all flow regimes," J. Comput. Phys. **315**, 16–38 (2016).
[45]Y. Zhu, C. Zhong, and K. Xu, "An implicit unified gas-kinetic scheme for unsteady flow in all Knudsen regimes," J. Comput. Phys. **386**, 190–217 (2019).
[46]R. Yuan, S. Liu, and C. Zhong, "A multi-prediction implicit scheme for steady state solutions of gas flow in all flow regimes," Commun. Nonlinear Sci. Numer. Simul. **92**, 105470 (2021).
[47]X. Xu, Y. Zhu, C. Liu, and K. Xu, "General implicit iterative method for unified gas-kinetic scheme," arXiv:2105.00392 (2021).
[48]L. M. Yang, Z. Chen, C. Shu, W. M. Yang, J. Wu, and L. Q. Zhang, "Improved fully implicit discrete-velocity method for efficient simulation of flows in all flow regimes," Phys. Rev. E **98**(6), 063313 (2018).
[49]L. M. Yang, C. Shu, W. M. Yang, and J. Wu, "An improved three-dimensional implicit discrete velocity method on unstructured meshes for all Knudsen number flows," J. Comput. Phys. **396**, 738–760 (2019).
[50]P. L. Bhatnagar, E. P. Gross, and M. Krook, "A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems," Phys. Rev. **94**(3), 511–525 (1954).
[51]E. M. Shakhov, "Generalization of the Krook kinetic relaxation equation," Fluid Dyn. **3**(5), 95–96 (1972).
[52]A. Jameson and S. Yoon, "Lower-upper implicit schemes with multiple grids for the Euler equations," AIAA J. **25**(7), 929–935 (1987).
[53]V. A. Titarev, "Implicit high-order method for calculating rarefied gas flow in a planar microchannel," J. Comput. Phys. **231**(1), 109–134 (2012).
[54]V. A. Titarev, M. Dumbser, and S. Utyuzhnikov, "Construction and comparison of parallel implicit kinetic solvers in three spatial dimensions," J. Comput. Phys. **256**, 17–33 (2014).
[55]L. M. Yang, C. Shu, W. M. Yang, J. Wu, and M. Q. Zhang, "Numerical investigation on performance of three solution reconstructions at cell interface in DVM simulation of flows in all Knudsen number regimes," Int. J. Numer. Methods Fluids **90**(11), 545–563 (2019).
[56]U. Ghia, K. N. Ghia, and C. T. Shin, "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method," J. Comput. Phys. **48**(3), 387–411 (1982).
[57]J. C. Huang, K. Xu, and P. Yu, "A unified gas-kinetic scheme for continuum and rarefied flows. II. Multi-dimensional cases," Commun. Comput. Phys. **12**(3), 662–690 (2012).
[58]K. Xu, "To overcome memory barrier of kinetic solvers for non-equilibrium flow study," Sci. Bull. **62**, 99–101 (2017).
[59]C. Shu, L. Wang, and Y. T. Chew, "Numerical computation of three-dimensional incompressible Navier-Stokes equations in primitive variable form by DQ method," Int. J. Numer. Methods Fluids **43**(4), 345–368 (2003).
[60]Z. H. Li and H. X. Zhang, "Gas-kinetic numerical studies of three-dimensional complex flows on spacecraft re-entry," J. Comput. Phys. **228**(4), 1116–1138 (2009).
[61]F. W. Vogenitz, G. A. Bird, J. E. Broadwell, and H. Rungaldier, "Theoretical and experimental study of rarefied supersonic flows about several simple shapes," AIAA J. **6**(12), 2388–2394 (1968).
[62]J. Moss, K. Boyles, and F. Greene, "Orion aerodynamics for hypersonic free molecular to continuum conditions," AIAA Paper No. 2006-8081 (2006).