A comparative study of the DSBGK and DVM methods for low-speed rarefied gas flows

A comparative study of the DSBGK and DVM methods for low-speed rarefied gas flows

Minh Tuan Ho[a], Jun Li[b,*], Lei Wu[a], Jason M. Reese[c], Yonghao Zhang[a]

[a] *James Weir Fluids Laboratory, Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow G1 1XJ, UK*
[b] *Center for Integrative Petroleum Research, College of Petroleum Engineering and Geosciences, King Fahd University of Petroleum & Minerals, Saudi Arabia*
[c] *School of Engineering, University of Edinburgh, Edinburgh EH9 3FB, UK*

Abstract

Low-speed rarefied gas flow in a lid-driven cavity is chosen as a test case in order to assess the accuracy and efficiency of both the Direct Simulation Bhatnagar-Gross-Krook (DSBGK) method and the Discrete Velocity Method (DVM) for solving the BGK kinetic equation. Various lid-speeds and a broad range of rarefaction levels, from slip to near free-molecular flows, are investigated. The DSBGK and DVM results are in satisfactory agreement for all the examined cases in 2D and 3D. As a statistical method, the stochastic noise of the DSBGK method is much smaller than that of the conventional Direct Simulation Monte Carlo (DSMC) method, and is independent of the Mach number. To achieve the required accuracy, the DSBGK simulations need more CPU time than the DVM simulations, i.e. for the 2D cases, a factor of 2 to 15 times more for convergence, and about 50 to 80 times more overall, including the time-averaging process. However, for 3D cases, the third direction in the DVM velocity grid is needed, so the computational cost of DSBGK is now only 0.16 to 0.51 times that of the DVM for the convergence process, and 1.6 to 5.8 times that of the DVM overall. The efficiency of the DSBGK method can also be expected to be enhanced in large-scale 3D simulations, where the computational cost for time-averaging becomes negligible in comparison with the convergence process. The DSBGK simulations require much less memory, even at low Mach numbers, than the DVM simulations; in the test cases with the required accuracy, about 10 simulated molecules per cell in the DSBGK simulations are sufficient for an arbitrary

*Corresponding author

Kn, while the DVM requires at least 4×24 and $4 \times 24 \times 12$ velocity grids for the 2D and 3D cases, respectively, even at Kn = 0.1. Finally, we discuss the ray effects of the DVM, which exist in flow problems with a discontinuous boundary and are caused by incompatibility of the velocity grid, the spatial grid, and the order of accuracy of the numerical scheme.

Keywords: Rarefied gas dynamics, Kinetic equation, Direct simulation BGK method, Discrete velocity method, Low speed flows, Ray effects

1. Introduction

When the mean free path of gas molecules becomes appreciable compared to the characteristic flow length, the conventional Navier-Stokes equations fail, and gas kinetic theory should be applied to study the rarefied gas dynamics. Low-speed rarefied gas flows are characteristic of both MEMS and tight porous media [1, 2]. While the Direct Simulation Monte Carlo (DSMC) method [3] is the most popular molecular-based technique for modelling rarefied flows [4, 5, 6], it is computationally expensive and essentially impractical for low-speed flows due to stochastic noise. For example, to find the gas permeability of porous media, a large pressure ratio is usually applied between the inlet and the outlet to increase the signal-to-noise ratio and hence reduce the sampling required [7, 8, 9]. In the near-continuum regime, however, the flow velocity is large, and the obtained permeability is very likely not independent of the pressure gradient because of the nonlinear Forchheimer effect. It is therefore important to develop efficient and accurate numerical methods in order to simulate low-speed rarefied gas flows.

The information-preservation (IP) DSMC is probably the first attempt to simulate low-speed flows efficiently [10]. In addition to the macroscopic quantities that can be obtained by conventional sampling, IP-DSMC introduces information quantities (such as the information velocity and information temperature) to reduce the statistical noise significantly. However, the evolution of these information quantities is ad-hoc; for example, the shear viscosity needs to be adjusted.

The Low-variance Deviational Simulation Monte Carlo (LVDSMC) solver has been proposed in Refs. [11, 12, 13]. In this method, computational efficiency is

significantly improved by simulating only the deviation from an equilibrium state. Since the computational cost does not depend on the Mach number (Ma), rarefied gas flows with Ma as low as 10^{-5} have been simulated. This is in sharp contrast to DSMC, in which the required statistical sampling leads to a computational cost that is proportional to Ma^{-2} [14]. LVDSMC has also been extended to solve the linearized Bhatnagar-Gross-Krook (BGK) kinetic model equation [15, 16] and the McCormack kinetic equation for binary gas mixtures [17]. Excellent agreement with other deterministic solvers has been reported. There have been other reports of stochastic techniques coupled with deterministic methods to reduce the variance of particle methods [18, 19, 20, 21].

The Direct Simulation BGK (DSBGK) method is also a particle-based approach and has been recently proposed for improving the efficiency of rarefied gas flow simulations at very low speeds [22]. It has been validated against DSMC simulations for several benchmark problems over a wide range of Knudsen numbers (Kn, defined as the ratio of the mean free path to the characteristic flow length) [23, 24]. Compared with the standard DSMC technique, the DSBGK method achieves high efficiency by avoiding generating a large number of random fractions in the intermolecular collision process, and by using increments (instead of transient values) of molecular variables to update macro quantities in each cell based on the conservation laws of mass, momentum and energy. This updating algorithm significantly reduces the stochastic noise due to discontinuous events of simulated molecules randomly moving into and out of cells. A comparative analysis of the algorithms of the DSMC and DSBGK methods, with comparisons of simulation results produced by each, is presented in Ref. [24].

The Discrete Velocity Method (DVM), on the other hand, deterministically solves the Boltzmann equation, or simplified models [25, 26]. DVM has been widely used to produce reliable data for rarefied gas flows from low to high speed [27, 28, 29]. Although the DVM offers accurate fluctuation-free solutions, it generally requires high dimensionality in the distribution function, which may lead to a high demand in computational memory and cost (although a memory reduction technique has recently been proposed [30]). In addition to the dimensions in spatial space, the

DVM also usually needs three-dimensional discretization in molecular velocity space, whereas particle-based methods (such as DSMC and DSBGK) only need a number of simulated molecules per cell to dynamically discretize the molecular velocity space. For 1D and 2D flow problems, the dimensions of DVM in molecular velocity space can be reduced [31]. Moreover, if only the steady state solution is of interest, DVM can accelerate its rate of convergence by using implicit time-marching schemes or other iterative schemes [26, 32]. To the best of the authors' knowledge, particle-based methods usually have no such acceleration opportunities without losing accuracy, due to their time-evolutionary nature. Although the timestep Δt in the DSMC method is not restricted by the Courant-Friedrichs-Lewy (CFL) stability condition, the error in the transport coefficients has been found to be proportional to Δt^2 [33, 34].

In this paper we assess the accuracy and computational efficiency of two different methods for solving the BGK kinetic equation — the DSBKG method and the DVM. Our chosen benchmark problems are the lid-driven cavity flows in 2D and 3D, which are characterized by shear-driven and flow compression phenomena that have been studied previously [35, 36, 37, 38, 39].

2. The BGK equation and its numerical solution

The Bhatnagar-Gross-Krook (BGK) kinetic model equation simplifies the Boltzmann equation by using a relaxation-time approximation [15]. It can produce good results when thermal effects are negligible. Therefore, the relaxation time is chosen to recover only the shear viscosity, according to the Chapman-Enskog expansion in the continuum flow limit. Without an external body force, the BGK equation takes the following form:

$$\frac{\partial f}{\partial t} + \boldsymbol{c} \cdot \nabla f = -\frac{1}{\tau} \left(f - f_{\mathrm{eq}} \right), \tag{1}$$

where $f = f(\boldsymbol{x}, \boldsymbol{c}, t)$ is the velocity distribution function of gas molecules with molecular velocity $\boldsymbol{c} = (c_x, c_y, c_z)$ at position $\boldsymbol{x} = (x, y, z)$ and time t, and f_{eq} is the equilibrium distribution function defined by the Maxwellian:

$$f_{\mathrm{eq}} = \frac{n}{(2\pi k_{\mathrm{B}} T/m)^{3/2}} \exp\left(-\frac{m\xi^2}{2k_{\mathrm{B}} T} \right), \tag{2}$$

where n, m, T and k_B are the gas number density, molecular mass, temperature and the Boltzmann constant, respectively; $\boldsymbol{\xi} = \boldsymbol{c} - \boldsymbol{u}$ is the peculiar velocity, with $\boldsymbol{u}$ the macroscopic flow velocity. Conservative flow variables $\boldsymbol{W} \equiv (n, n\boldsymbol{u}, ne)^T$ are calculated as velocity moments of the distribution function, i.e.

$$\boldsymbol{W} = \int \boldsymbol{\psi} f \mathrm{d}\boldsymbol{c}, \tag{3}$$

where $\boldsymbol{\psi} = (1, \boldsymbol{c}, c^2/2)^T$, and $e = (u^2 + 3k_\mathrm{B}T/m)/2$ is the specific total energy.

The relaxation time τ in Eq. (1) is related to the dynamic viscosity μ and the local pressure p by $\tau = \mu/p = \mu/(nk_\mathrm{B}T)$. For gas molecules interacting through the inverse power-law potential, the dynamic viscosity μ depends on the temperature T as

$$\mu = \mu_0 \left(\frac{T}{T_0}\right)^\omega, \tag{4}$$

where ω is the viscosity index and μ_0 is the reference viscosity at the reference temperature T_0. For a lid-driven cavity flow, the reference temperature is chosen as the bounding wall temperature $T_0 = T_\mathrm{w} = 273$ K. Without loss of generality, argon gas with $m = 6.63 \times 10^{-26}$ kg, $\mu_0 = 2.117 \times 10^{-5}$ Ns/m^2 and $\omega = 0.81$ is used here.

The mean free path λ_0 of gas molecules and the Knudsen number Kn are defined as

$$\lambda_0 = \frac{\mu_0}{p_0}\sqrt{\frac{\pi k_\mathrm{B} T_0}{2m}}, \quad \mathrm{Kn} = \frac{\lambda_0}{L_\mathrm{char}}, \tag{5}$$

respectively, where L_char is the characteristic length. The Mach number Ma is defined as

$$\mathrm{Ma} = \frac{u_\mathrm{w}}{\sqrt{\gamma k_\mathrm{B} T_0/m}}, \tag{6}$$

where γ is the specific heat ratio and u_w is the speed of the moving lid.

To fully determine the rarefied gas flow, the gas-surface interaction should be specified. In this paper, we consider the Maxwell diffuse boundary condition at the solid wall, i.e. the velocity distribution function for gas molecules entering the computational domain is given by

$$f_\mathrm{B,diff}(\boldsymbol{c}_\mathrm{r}) = n_\mathrm{eff} \left(\frac{m}{2\pi k_\mathrm{B} T_\mathrm{w}}\right)^{3/2} \exp\left(-\frac{m|\boldsymbol{c}_\mathrm{r}|^2}{2k_\mathrm{B}T_\mathrm{w}}\right), \tag{7}$$

where $\boldsymbol{c}_\mathrm{r}$ is the reflected velocity of gas molecules relative to the wall, and the effective number density n_eff is determined from the impermeable condition, that is,

the number of gas molecules moving to the wall is equal to that reflected from the same wall.

2.1. The DSBGK method

As the DSBGK method is a relatively new technique, some significant description is appropriate here. The DSBGK method for solving the BGK equation is proposed in Ref. [22], and further detailed in Ref. [24] where the extension to problems with an external body force is discussed. The simulation timestep Δt and computational domain cell size Δx are selected as in the DSMC method when simulating problems of high Kn. Each simulated molecule l carries four molecular variables: position $\boldsymbol{x}_l$, molecular velocity $\boldsymbol{c}_l$, number N_l of real molecules represented by the simulated molecule, and F_l that is equal to $f(\boldsymbol{x}_l, \boldsymbol{c}_l, t)$. The variables $n_{\mathrm{tr},j}, \boldsymbol{u}_{\mathrm{tr},j}, T_{\mathrm{tr},j}$ of each cell j are updated using $\boldsymbol{x}_l, \boldsymbol{c}_l$ and the *increment* of N_l based on the mass, momentum and energy conservation principles of the intermolecular collision process. They are then used in turn to update the molecular variables according to the BGK equation and an extrapolation of the acceptance-rejection scheme. The cell variables $n_{\mathrm{tr},j}, \boldsymbol{u}_{\mathrm{tr},j}, T_{\mathrm{tr},j}$ are transitional variables and converge to $n_j, \boldsymbol{u}_j, T_j$ that are defined by the moments of f, as discussed in Ref. [24] (after Eq. (13) in that paper).

At the initial state, $\boldsymbol{x}_l$ and $\boldsymbol{c}_l$ are selected according to the uniform initial distribution $f_0 = f_{\mathrm{eq},0}$, and the initial N_l is the same for all the simulated molecules (as in DSMC simulations). The initial F_l can then be determined accordingly, i.e., $F_l = f_0(\boldsymbol{x}_l, \boldsymbol{c}_l, 0)$. In the simulation process, each simulated molecule l moves on a uniform trajectory until encountering boundaries. During each Δt, the trajectory of each simulated molecule may be divided into several segments by the cell interfaces. The time interval used by the simulated molecule l for the segment located inside the cell j is denoted by $\Delta_j t_l$. F_l can be updated by the integration of the BGK equation along each trajectory segment in sequence, i.e.

$$F_{l,\mathrm{new}} = f_{\mathrm{eq},j} + (F_l - f_{\mathrm{eq},j}) \exp(-\Delta_j t_l / \tau), \tag{8}$$

where $f_{\mathrm{eq},j}$ is the local equilibrium distribution defined by using the transient $n_{\mathrm{tr},j}$, $\boldsymbol{u}_{\mathrm{tr},j}$ and $T_{\mathrm{tr},j}$ of the cell j.

According to an extrapolation of the acceptance-rejection scheme [22], $[\boldsymbol{x}_l, \boldsymbol{c}_l, N_l(f_2/f_1)_l]_{\text{all}}$ is a representative sample of f_2 if $[\boldsymbol{x}_l, \boldsymbol{c}_l, N_l]_{\text{all}}$ is a representative sample of f_1, where $(f_2/f_1)_l$ is the ratio of f_2 and f_1 at the point $(\boldsymbol{x}_l, \boldsymbol{c}_l)$. Thus, N_l can be updated according to F_l for each trajectory segment:

$$N_{l,\text{new}} = N_l F_{l,\text{new}}/F_l, \tag{9}$$

from which we obtain $\Delta_j N_l = N_{l,\text{new}} - N_l$ for the trajectory segment of the simulated molecule l located inside the cell j. This is the number increment of real molecules of class $\boldsymbol{c}_l$ due to the intermolecular collisions inside the cell j during the current timestep. We compute the summation $\sum_{\in j} \Delta_j N_l$ over those trajectory segments located inside cell j during the current timestep. Mass conservation in the intermolecular collision process inside cell j requires $\sum_{\in j} \Delta_j N_l = 0$. Thus, we reduce $n_{\text{tr},j}$ if $\sum_{\in j} \Delta_j N_l > 0$ and then $\sum_{\in j} \Delta_j N_l$ will be reduced at the next timestep according to Eqs. (2), (8) and (9), and vice versa. This auto-regulation scheme ensures that $\sum_{\in j} \Delta_j N_l$ approaches zero. Similarly, $\sum_{\in j}(\Delta_j N_l m \boldsymbol{c}_l)$ and $\sum_{\in j}(\Delta_j N_l m \boldsymbol{c}_l^2/2)$ can be used to regulate $\boldsymbol{u}_{\text{tr},j}, T_{\text{tr},j}$ according to the momentum and energy conservations. The auto-regulation schemes used in the ordinary DSBGK simulations to update the cell variables after each timestep are therefore

$$\begin{aligned}
n_{\text{tr},j}^{\text{new}} &= \frac{n_{\text{tr},j} V_j - \sum_{\in j} \Delta_j N_l}{V_j}, \\
\boldsymbol{u}_{\text{tr},j}^{\text{new}} &= \frac{n_{\text{tr},j} V_j \boldsymbol{u}_{\text{tr},j} - \sum_{\in j}(\Delta_j N_l \boldsymbol{c}_l)}{n_{\text{tr},j}^{\text{new}} V_j}, \\
T_{\text{tr},j}^{\text{new}} &= \frac{n_{\text{tr},j} V_j (3k_{\text{B}} T_{\text{tr},j}/2 + m\boldsymbol{u}_{\text{tr},j}^2/2) - \sum_{\in j}(\Delta_j N_l m \boldsymbol{c}_l^2/2) - n_{\text{tr},j}^{\text{new}} V_j m (\boldsymbol{u}_{\text{tr},j}^{\text{new}})^2/2}{n_{\text{tr},j}^{\text{new}} V_j 3k_{\text{B}}/2},
\end{aligned} \tag{10}$$

where V_j is the volume of cell j.

Now, we discuss how the DSBGK method reduces the stochastic noise in cell variables. When using particle-based methods to solve a kinetic equation, the velocity of each particle is usually updated independently according to the kinetic equation, which consequently satisfies the conservation laws on average but violates conservation during each timestep. The incurred stochastic noise due to this violation can be reduced when solving the Boltzmann equation by using a special particle simulation method [40]. In this regard, the DSBGK method also adopts the special

scheme in Eq. (10) to impose conservations in each cell for each timestep to reduce noise.

In addition to the noise caused by violation of conservation laws, there is another type of noise in particle simulation due to frequent and random events of simulated molecules moving into and out of each cell. The cell variables calculated by the transient molecular variables inside the concerned cell therefore suffer from significant noise since the number of simulated molecules inside each cell on average is small. This is the source of significant stochastic noise in DSMC and molecular dynamic (MD) simulations. Instead of using transient values of molecular variables, their increments along molecular trajectories are used in the DSBGK method to update/regulate the cell variables. Although the molecular variables entering into each cell are still random and noisy, by integrating the BGK equation along the molecular trajectories, their variations are smooth. Consequently, noise in the cell variables is significantly reduced by using Eq. (10) to update the cell variables, compared to the DSMC and other particle simulation methods that define the cell variables by using transient molecular variables.

At a wall boundary (with $\boldsymbol{e}_{\rm n}$ as the outward normal direction), $\boldsymbol{c}_l$ and then F_l are updated after molecular reflection at $\boldsymbol{x}_l$ on the wall; N_l remains unchanged to conserve mass. The reflected velocity $\boldsymbol{c}_{\rm r}$ is randomly selected, as in the DSMC method, and then $\boldsymbol{c}_l$ is updated to $\boldsymbol{c}_{\rm r} + \boldsymbol{u}_{\rm w}$, where $\boldsymbol{u}_{\rm w}$ is the wall velocity. (The subscript l has been omitted for clarity in the notation of the incoming and reflected velocities.)

We introduce $f_{\rm B}(\boldsymbol{c})$ as the distribution function f at the reflection point $\boldsymbol{x}_l$ at time t in a local Cartesian reference frame moving at $\boldsymbol{u}_{\rm w}$, so that $f_{\rm B}(\boldsymbol{c}) = f(\boldsymbol{x}_l, \boldsymbol{c} + \boldsymbol{u}_{\rm w}, t)$. With an appropriate expression for $f_{\rm B}(\boldsymbol{c})$, we then have $F_{l,\rm new} = f_{\rm B}(\boldsymbol{c}_{\rm r})$. The distribution $f_{\rm B}(\boldsymbol{c}_{\rm i})|_{\boldsymbol{c}_{\rm i}\cdot\boldsymbol{e}_{\rm n}<0}$ of the incoming molecules is known from the molecular information in the adjacent cell, and $f_{\rm B}(\boldsymbol{c}_{\rm r})|_{\boldsymbol{c}_{\rm r}\cdot\boldsymbol{e}_{\rm n}>0}$ is the distribution of reflected molecules.

Theoretically, $f_{\rm B}(\boldsymbol{c}_{\rm i})$ depends on the incoming molecules. When the wall velocity is small, and to further reduce stochastic noise, a simple boundary condition is proposed: we use cell variables rather than the molecular variables to determine

$f_{\mathrm{B}}(\boldsymbol{c}_{\mathrm{i}})$ as a local equilibrium distribution, i.e.

$$f_{\mathrm{B,simple}}(\boldsymbol{c}_{\mathrm{i}}) = n_{\mathrm{tr},j}\left(\frac{m}{2\pi k_{\mathrm{B}}T_{\mathrm{tr},j}}\right)^{3/2}\exp\left[\frac{-m(\boldsymbol{c}_{\mathrm{i}}-(\boldsymbol{u}_{\mathrm{tr},j}-\boldsymbol{u}_{\mathrm{w}}))^2}{2k_{\mathrm{B}}T_{\mathrm{tr},j}}\right], \tag{11}$$

where $n_{\mathrm{tr},j}, \boldsymbol{u}_{\mathrm{tr},j}, T_{\mathrm{tr},j}$ are the quantities of cell j close to the reflection point $\boldsymbol{x}_l$. Then, the number flux of incoming molecules per unit wall area per unit time is:

$$\begin{aligned} N_{\mathrm{in,simple}} &= -\int_{\boldsymbol{c}_{\mathrm{i}}\cdot\boldsymbol{e}_{\mathrm{n}}<0} f_{\mathrm{B,simple}}(\boldsymbol{c}_{\mathrm{i}})(\boldsymbol{c}_{\mathrm{i}}\cdot\boldsymbol{e}_{\mathrm{n}})\mathrm{d}\boldsymbol{c}_{\mathrm{i}} \\ &= n_{\mathrm{tr},j}\sqrt{\frac{k_{\mathrm{B}}T_{\mathrm{tr},j}}{2\pi m}}\left[\exp(-\hat{u}_{\mathrm{in}}^2)+\sqrt{\pi}\hat{u}_{\mathrm{in}}(1+\mathrm{erf}(\hat{u}_{\mathrm{in}}))\right], \end{aligned} \tag{12}$$

where $\hat{u}_{\mathrm{in}} = -(\boldsymbol{u}_{\mathrm{tr},j}-\boldsymbol{u}_{\mathrm{w}})\cdot\boldsymbol{e}_{\mathrm{n}}/\sqrt{2k_{\mathrm{B}}T_{\mathrm{tr},j}/m}$. Similarly, the number flux N_{out} of reflected molecules is:

$$\begin{aligned} N_{\mathrm{out}} &= \int_{\boldsymbol{c}_{\mathrm{r}}\cdot\boldsymbol{e}_{\mathrm{n}}>0} f_{\mathrm{B,diffuse}}(\boldsymbol{c}_{\mathrm{r}})(\boldsymbol{c}_{\mathrm{r}}\cdot\boldsymbol{e}_{\mathrm{n}})\mathrm{d}\boldsymbol{c}_{\mathrm{r}} \\ &= n_{\mathrm{eff}}\sqrt{\frac{k_{\mathrm{B}}T_{\mathrm{w}}}{2\pi m}}. \end{aligned} \tag{13}$$

Now $N_{\mathrm{out}} = N_{\mathrm{in,simple}}$ according to mass conservation, so we obtain an estimate for n_{eff}:

$$n_{\mathrm{eff,simple}} = n_{\mathrm{tr},j}\sqrt{\frac{T_{\mathrm{tr},j}}{T_{\mathrm{w}}}}[\exp(-\hat{u}_{\mathrm{in}}^2)+\sqrt{\pi}\hat{u}_{\mathrm{in}}(1+\mathrm{erf}(\hat{u}_{\mathrm{in}}))]. \tag{14}$$

Then we update F_l by $F_{l,\mathrm{new}} = f_{\mathrm{B,diffuse}}(\boldsymbol{c}_{\mathrm{r}})$, where $n_{\mathrm{eff}} = n_{\mathrm{eff,simple}}$. Compared to the statistically accurate boundary condition that we develop below, this simple boundary condition helps to reduce stochastic noise in the DSBGK simulations, especially at low Ma. The numerical error incurred by the simple boundary will be analyzed in Sections 3.1-3.3.

For closed flow problems, density drift has been observed in previous DSBGK simulations after a large number of timesteps when using this simple implementation of the boundary condition. To reduce the magnitude of the density drift, 2000 simulated molecules per cell are employed in the following study of 2D cavity flows. However, the DSBGK accuracy is almost unchanged when using only 10 simulated molecules per cell and more time-averaging samples. Correspondingly, the sampling process of density could stop after only about 100 timesteps to avoid deviation due to the slow density drift, because the transient density distribution has low stochastic noise. In the following simulations, we will use only 100 samples for

number density n and larger number of samples for other macroscopic parameters, unless stated otherwise. Note that this unphysical density drift disappears when simulating open flow problems, because fixed number densities are applied at the open boundaries (e.g., the channel flow problem [24]). By contrast, flow velocity and temperature are not subject to unphysical drift, even in closed problems, thanks to the specified constraints at the boundary. Additionally, the density drift in closed problems becomes unnoticeable if the perturbation is very small (e.g. for the 2D cavity flow problem with $u_{\rm w} = 10^{-6}$ m/s) [41].

If the flow velocity is not small (e.g. for the cavity flow problem with $u_{\rm w} = 50$ m/s), the density drift can be eliminated by using a statistically accurate boundary condition [41] in which the incoming number flux is directly calculated by using the information of incoming simulated molecules, although the correspondingly computed flux is noisy. As in the DSMC method, it is convenient for the DSBGK method to calculate the net flux $\Gamma(Q)$ of any molecular quantity $Q(\boldsymbol{c})$ in unit time and across unit area of the boundary surface, viz.

$$\Gamma(Q) = \frac{1}{\Delta t \Delta S} \sum_l N_l [Q(\boldsymbol{c}_{\rm i}) - Q(\boldsymbol{c}_{\rm r})]_l, \tag{15}$$

where the summation is over the simulated molecules reflected on the sub-area ΔS during the current timestep Δt; $Q(\boldsymbol{c}_{\rm i})$ and $Q(\boldsymbol{c}_{\rm r})$ are the incoming and reflected quantities, respectively. If $Q = m\boldsymbol{c}$ or $m\boldsymbol{c}^2/2$, then $\Gamma(Q)$ represents the stress or heat flux, respectively. Similarly, the incoming number flux is computed as:

$$N_{\rm in} = \frac{1}{\Delta t \Delta S} \sum_l N_l. \tag{16}$$

As $N_{\rm out} = N_{\rm in}$ again, we obtain a statistically accurate formula for $n_{\rm eff}$, i.e.

$$n_{\rm eff} = \sqrt{\frac{2\pi m}{k_{\rm B} T_{\rm w}}} \frac{1}{\Delta t \Delta S} \sum_l N_l, \tag{17}$$

where $\sum_l N_l$ usually contains large stochastic noise. $F_{l,\rm new} = f_{\rm B,diffuse}(\boldsymbol{c}_{\rm r})$ is implemented to update F_l during the simulation process, and $n_{\rm eff}$ is updated by using Eq. (17) after each Δt.

The workflow of a DSBGK simulation can then be summarized as follows:

1. Initialization. Generate the domain cells and simulated molecules and assign them initial values for $n_{\mathrm{tr},j}, \boldsymbol{u}_{\mathrm{tr},j}, T_{\mathrm{tr},j}$ and $\boldsymbol{x}_l, \boldsymbol{c}_l, F_l, N_l$, respectively.
2. Each simulated molecule l moves on a uniform trajectory until encountering boundaries. During each Δt, the trajectory of each simulated molecule may be divided into several segments by the cell interfaces. Then, $\boldsymbol{x}_l, F_l, N_l$ are *deterministically* updated along each segment in sequence. When encountering wall boundaries, $\boldsymbol{c}_l$ is updated to $\boldsymbol{c}_{\mathrm{r}} + \boldsymbol{u}_{\mathrm{w}}$ according to the reflection model, and F_l is correspondingly updated to $f_{\mathrm{B}}(\boldsymbol{c}_{\mathrm{r}})$. In open flow systems, simulated molecules are removed from the computational domain when they move across open boundaries, and new simulated molecules are generated at the open boundaries. The variables $n_{\mathrm{tr},j}, \boldsymbol{u}_{\mathrm{tr},j}, T_{\mathrm{tr},j}$ of each cell j are updated at the end of each Δt.
3. After convergence, $n_{\mathrm{tr},j}, \boldsymbol{u}_{\mathrm{tr},j}, T_{\mathrm{tr},j}$ provide the discrete solutions of the BGK equation at steady state.

2.2. The Discrete Velocity Method (DVM)

For a 2D case, two reduced velocity distribution functions are introduced to cast the 3D molecular velocity space into 2D [26], i.e.

$$g = \int f(\boldsymbol{x}, \boldsymbol{c}, t)\mathrm{d}c_z, \quad h = \int c_z^2 f(\boldsymbol{x}, \boldsymbol{c}, t)\mathrm{d}c_z. \tag{18}$$

For convenience, in what follows we denote $\boldsymbol{c} = (c_x, c_y)$, $\boldsymbol{\xi} = (\xi_x, \xi_y)$ and $\boldsymbol{x} = (x, y)$. Using g and h, macroscopic variables can be computed as $n = \int g\mathrm{d}\boldsymbol{c}$, $n\boldsymbol{u} = \int \boldsymbol{c}g\mathrm{d}\boldsymbol{c}$, and $ne = \frac{1}{2}\int (c^2 g + h)\mathrm{d}\boldsymbol{c}$. The governing equations for the two reduced distribution functions can be deduced from Eq. (1) in the form of the generic function $\phi = (h, g)$ as

$$\frac{\partial \phi}{\partial t} + \boldsymbol{c} \cdot \nabla \phi = -\frac{\phi - \phi_{\mathrm{eq}}}{\tau}, \tag{19}$$

where the reduced equilibrium distribution functions $\phi_{\mathrm{eq}} = (h_{\mathrm{eq}}, g_{\mathrm{eq}})$ are

$$g_{\mathrm{eq}}(\boldsymbol{x}, \boldsymbol{c}, t) = \int f_{\mathrm{eq}}(\boldsymbol{x}, \boldsymbol{c}, c_z, t)\mathrm{d}c_z = \frac{nm}{2\pi k_{\mathrm{B}} T} \exp\left[-\frac{m\xi^2}{2k_{\mathrm{B}}T}\right], \tag{20a}$$

$$h_{\mathrm{eq}}(\boldsymbol{x}, \boldsymbol{c}, t) = \int c_z^2 f_{\mathrm{eq}}(\boldsymbol{x}, \boldsymbol{c}, c_z, t)\mathrm{d}c_z = k_{\mathrm{B}} T g_{\mathrm{eq}}/m. \tag{20b}$$

DVM is one of the most common deterministic approaches for solving the Boltzmann equation and its simplified models [25, 26]. It projects the continuous molecular velocity space $\boldsymbol{c}$ into a set of fixed N_c discrete velocities $\boldsymbol{c}^{(ii)}$ $(ii = 1, 2, .., N_c)$. As a result, for the BGK model, the governing equation (19) is replaced by a system of N_c independent equations. Here, we discretize this system in time by a fully time-implicit Godunov-type scheme [26, 32]:

$$\begin{aligned} \left[\frac{1}{\Delta t^{(ts)}} + \boldsymbol{c}^{(ii)} \cdot \nabla + \frac{1}{\tau^{(ts)}}\right] \Delta\phi^{(ts)} &= \mathrm{RHS}^{(ts)}, \\ \mathrm{RHS}^{(ts)} = \frac{1}{\tau^{(ts)}} \left[\phi_{\mathrm{eq}}^{(ts)} - \phi^{(ts)}\right] &- \boldsymbol{c}^{(ii)} \cdot \nabla\phi^{(ts)}, \end{aligned} \tag{21}$$

where $\Delta\phi^{(ts)} = \phi^{(ts+1)} - \phi^{(ts)}$ needs to be determined at the timestep ts. $\mathrm{RHS}^{(ts)}$ is the explicit part, and the spatial derivative is approximated by a third-order upwind scheme. For instance, the derivative with respect to the x-direction at point x_{jj} is evaluated by:

$$\left.\frac{\partial\phi^{(ts)}}{\partial x}\right|_{jj} = \begin{cases} (2\phi_{jj+1}^{(ts)} + 3\phi_{jj}^{(ts)} - 6\phi_{jj-1}^{(ts)} + \phi_{jj-2}^{(ts)})/(6\Delta x), & c_x^{(ii)} > 0\,, \\ (-2\phi_{jj-1}^{(ts)} - 3\phi_{jj}^{(ts)} + 6\phi_{jj+1}^{(ts)} - \phi_{jj+2}^{(ts)})/(6\Delta x), & c_x^{(ii)} < 0\,. \end{cases} \tag{22}$$

The left-hand side of Eq. (21) is the implicit part, and the spatial derivative is approximated by a first-order upwind scheme. By marching in the appropriate direction, e.g. increasing x in the case of $c_x^{(ii)} > 0$, the unknown $\Delta\phi^{(ts)}$ can be obtained directly without solving a system of equations.

Note that Δt in Eq. (21) is a pseudo-timestep that is defined by the CFL condition, i.e. $\Delta t = \eta\Delta x^{\min}/c_x^{\max}$, where η is the CFL number, $\Delta x^{\min}$ is the minimum spatial grid size, and $c_x^{\max}$ is the maximum discrete speed. While η here can be smaller than 1 to capture transient behaviour, it can also be set as large as 10^4 (as it is in the studies below) to obtain a steady-state solution.

In order to evaluate the flow variables in Eq. (3), the product Gaussian quadratures in the velocity polar coordinates $\boldsymbol{c} = (c_p\cos\varphi, c_p\sin\varphi)$ and the velocity spherical coordinates $\boldsymbol{c} = (c_p\sin\theta\cos\varphi, c_p\sin\theta\sin\varphi, c_p\cos\theta)$ are used for 2D and 3D simulations, respectively [36, 42]. For example, the flow variables in the 3D case are

approximated as:

$$\boldsymbol{W} = \int_0^{\infty}\int_0^{2\pi}\int_0^{\pi} \boldsymbol{\psi} f c_p^2 \sin\theta \mathrm{d}\theta \mathrm{d}\varphi \mathrm{d}c_p \approx \sum_{k=1}^{N_{c_p}}\sum_{j=1}^{N_\varphi}\sum_{i=1}^{N_\theta} w_{c_p}^{(k)} w_\varphi^{(j)} w_\theta^{(i)} \boldsymbol{F}\left(c_p^{(k)}, \varphi^{(j)}, \theta^{(i)}\right), \tag{23}$$

where $\boldsymbol{F} = \boldsymbol{\psi} f c_p^2 \exp\left(c_p^2\right)$; N_{c_p}, N_φ and N_θ are the numbers of discretized points in the radius c_p, azimuth φ and inclination θ, respectively, of the velocity space; $c_p^{(k)}$ and $w_{c_p}^{(k)}$ are the half-range Gauss-Hermite abscissae and weights, while $\cos\theta^{(i)}$ and $w_\theta^{(i)}$ are the Gauss-Legendre abscissae and weights in $[-1, 1]$ [43, 44]. The nodes $\varphi^{(j)}$ are uniformly spaced on $[0, 2\pi]$ and $w_\varphi^{(j)} = 2\pi/N_\varphi$. We denote the total number of velocity grid points by $N_c = N_{c_p} \times N_\varphi$ and $N_c = N_{c_p} \times N_\varphi \times N_\theta$ for 2D and 3D velocity grids, respectively.

Our simulations start from the global equilibrium state. The convergence criterion for the steady-state, which is based on the velocity flow field, is checked every timestep as follows:

$$E(t) = \frac{\sum |\boldsymbol{u}(t) - \boldsymbol{u}(t - \Delta t)|}{\sum |\boldsymbol{u}(t)|} < 10^{-6}. \tag{24}$$

3. 2D cavity flows

As our first test case for comparison of the DSBGK method and the DVM, we consider the rarefied gas flow inside a square cavity of size $L = 1\ \mu m$ and this size is used as the characteristic length $L_{\mathrm{char}} = L$. Cartesian coordinates are used, with the origin located at the bottom left corner of the cavity, as shown in Fig. 1 (a), in which $X = x/L$ and $Y = y/L$. From the origin, the positive X- and Y-directions point towards the bottom right corner and the top left corner, respectively. The top lid moves in the positive X-direction with a constant speed u_{w}. The cavity walls are maintained at a constant temperature $T_{\mathrm{w}} = T_0$.

Lid-driven cavity flows in 2D are simulated by both the DSBGK method and the DVM over a wide range of Knudsen and Mach numbers. We choose the lid speeds to be $u_{\mathrm{w}} = 0.001$, 1, 10, and 50 m/s, which correspond to $\mathrm{Ma} = 3.2 \times 10^{-6}$, 3.2×10^{-3}, 3.2×10^{-2}, and 0.16, respectively. We also choose $\mathrm{Kn} = 0.1$, 1 and 8, to cover the slip, transition, and free-molecular flow regimes, respectively. Initially, the gas is in global equilibrium described by Eq. (2), with zero flow velocity and uniform

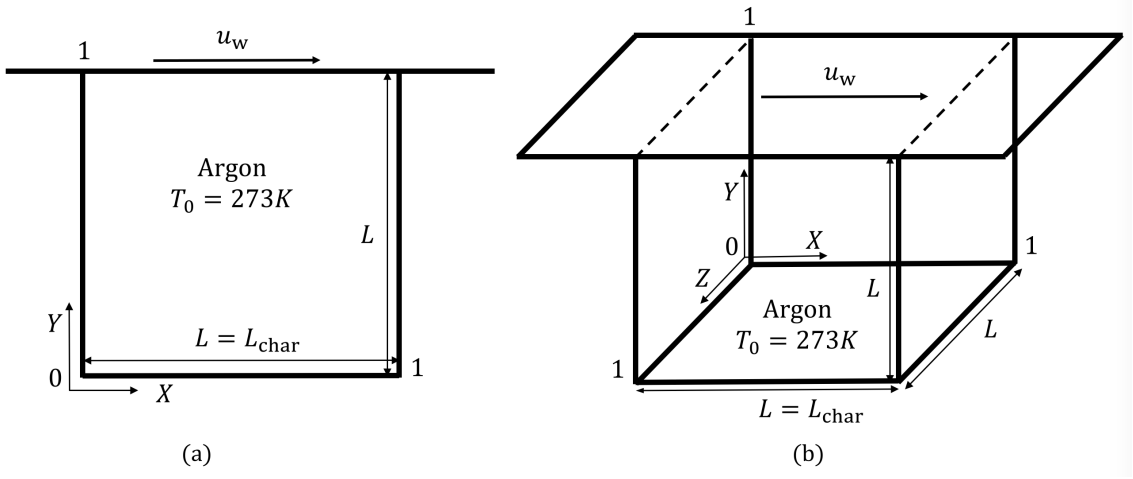


Figure 1: Schematic of the flow test cases: (a) 2D lid-driven cavity, (b) 3D lid-driven cavity.

temperature T_0. The uniform number density is adjusted to obtain the desired value of Kn.

Unless stated otherwise, the number of uniform spatial cells/grids is 60^2 for both the DSBGK and the DVM simulations. For simplicity of comparison between the two methods, uniform spatial grids are implemented in this study. However, DVM can be easily extended to non-uniform grids [45], and DSBGK is unchanged when using non-uniform grids [24]. We use an 8×80 velocity grid in the DVM, and employ 2000 simulated molecules per cell in the DSBGK method with the statistically accurate boundary condition Eq. (17) (or the simple boundary condition Eq. (14)) for Ma $= 0.16$ (or for Ma < 0.16). The DSBGK simulations need 500 samples to smooth the results for arbitrary Ma and Kn. Coarser spatial and velocity grids (or smaller numbers of simulated molecules per cell) will be tested in Section 3.4 below. Note that the DSBGK simulations use dimensional quantities, while the DVM simulations employ only dimensionless quantities scaled by a relevant reference value, e.g. $L_{\text{char}}, n_0, u_{\text{w}}, T_0$. Perturbed macroscopic quantities obtained by the two methods are reported in our results below, i.e.

$$\tilde{n} = \frac{n - n_0}{n_0}, \ \tilde{u} = \frac{u}{u_{\text{w}}}, \ \tilde{v} = \frac{v}{u_{\text{w}}}, \ \tilde{T} = \frac{T - T_0}{T_0}, \tag{25}$$

where u and v are the components in the X and Y directions, respectively, of the macroscopic velocity vector $\boldsymbol{u} = (u, v)$. (Hereafter, the tildes on these perturbed macroscopic quantities are omitted for simplicity.)

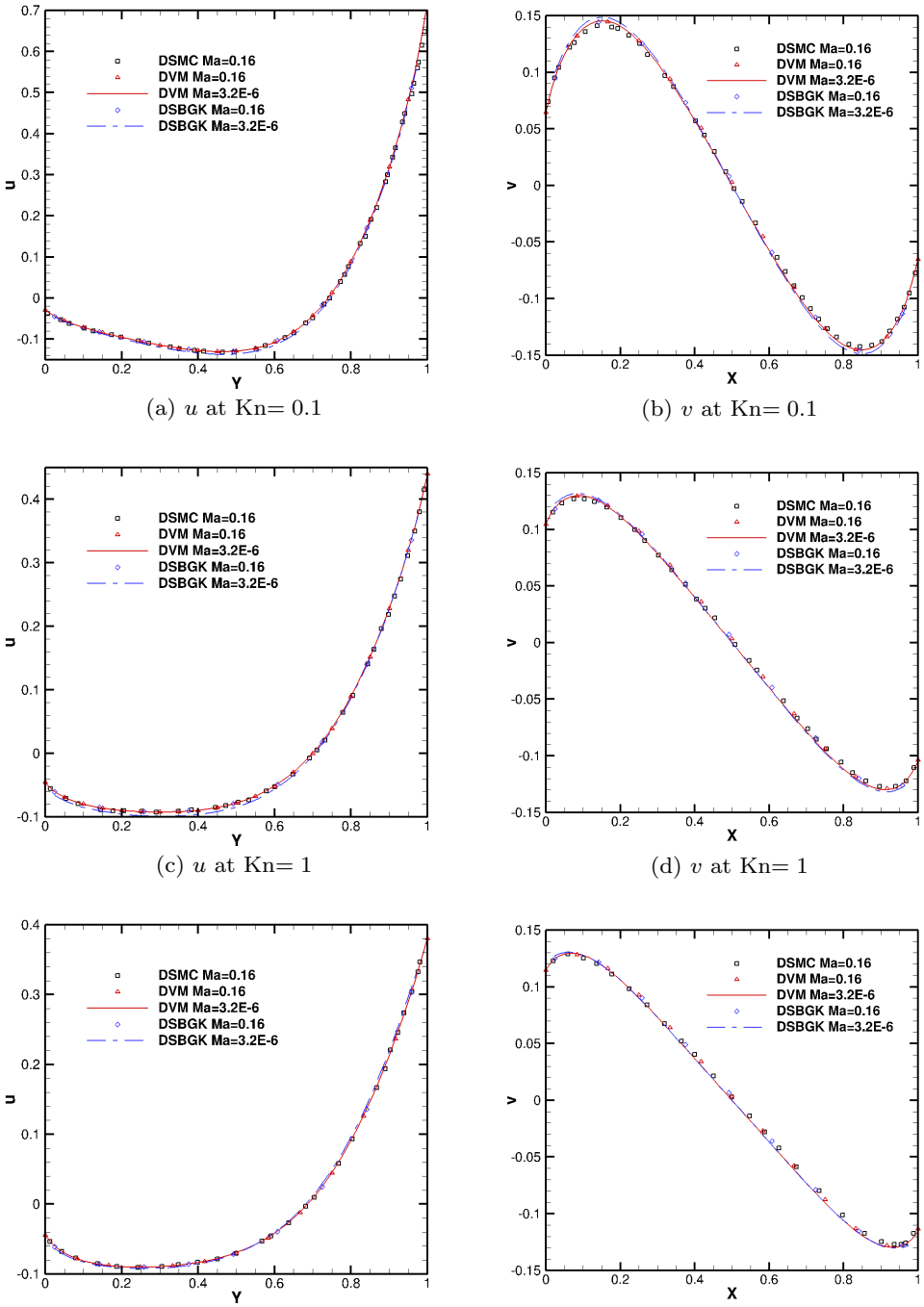
3.1. Validation

Figure 2 shows DVM, DSBGK and published DSMC [37] profiles of the perturbed u and v velocity components along the vertical centreline ($X = 0.5$) and the horizontal centreline ($Y = 0.5$), respectively, for various Ma and Kn. Figures 2 (a),(c),(e) show that the velocity slip at the moving lid increases considerably with Kn, while the increase at the bottom wall is negligible. For a moderate Mach number of 0.16, where heat flux is insignificant, the DVM and DSBGK (with the statistically accurate boundary condition) results agree very well with the published DSMC data [37] for Kn $= 0.1, 1$ and 8, in which the variable hard-sphere (VHS) molecular model with $\omega = 0.81$ was used.

For Ma < 0.16, the DSBGK results obtained using the statistically accurate boundary condition are dominated by stochastic noise (not shown here), so the simple but smooth boundary condition is used instead. Consequently, the DSBGK results have a small discrepancy from the DSMC data. Compared with the DVM method in solving the same BGK equation, the maximum discrepancy in the u profiles between the DVM and DSBGK methods is about 7.5% at Ma $= 3.2 \times 10^{-6}$, Kn $= 1$, and occurs only in the flow region with relatively small perturbation.

Similarly, Figs. 2 (b),(d),(f) show that the v profiles obtained by the two methods are in good agreement with the DSMC data at Ma $= 0.16$, Kn $= 0.1, 1$ and 8. The v profiles resemble a cosine function, with one maximum and one minimum near the left and right walls, respectively. These extrema are almost anti-symmetrical with respect to the cavity centre; this has also previously been shown using the linearized kinetic equation [46]. As the Mach number decreases, the v profile remains nearly unchanged in the DVM results, while a small discrepancy is observed in the DSBGK results due to the use of the simple boundary condition. The maximum discrepancy in the v profiles between the DVM and DSBGK methods is approximately 2.5% at Ma $= 3.2 \times 10^{-6}$ and Kn $= 0.1$.

Note that it has been reported independently in Ref. [37] (see Fig. 4 of that paper) that the u profile is independent of Ma, and the v profile remains essentially unchanged with Mach number for Ma ≤ 0.32. This is confirmed by the present DVM and DSBGK results. Although not shown here, the u and v profiles produced



(a) u at Kn= 0.1 (b) v at Kn= 0.1

(c) u at Kn= 1 (d) v at Kn= 1

(e) u at Kn= 8 (f) v at Kn= 8

Figure 2: Profiles of the perturbed horizontal velocity u (left column) and perturbed vertical velocity v (right column) along the vertical centreline ($X = 0.5$) and horizontal centreline ($Y = 0.5$), respectively, of the 2D lid-driven cavity. The DVM and DSBGK results are compared with published DSMC data [37]. Note that the flow velocity has been normalized by the lid velocity u_{w}.

by the DVM and DSBGK simulations for Ma $= 3.2 \times 10^{-2}$ and 3.2×10^{-3} are indistinguishable from those for Ma $= 3.2 \times 10^{-6}$.

3.2. Effect of Mach number

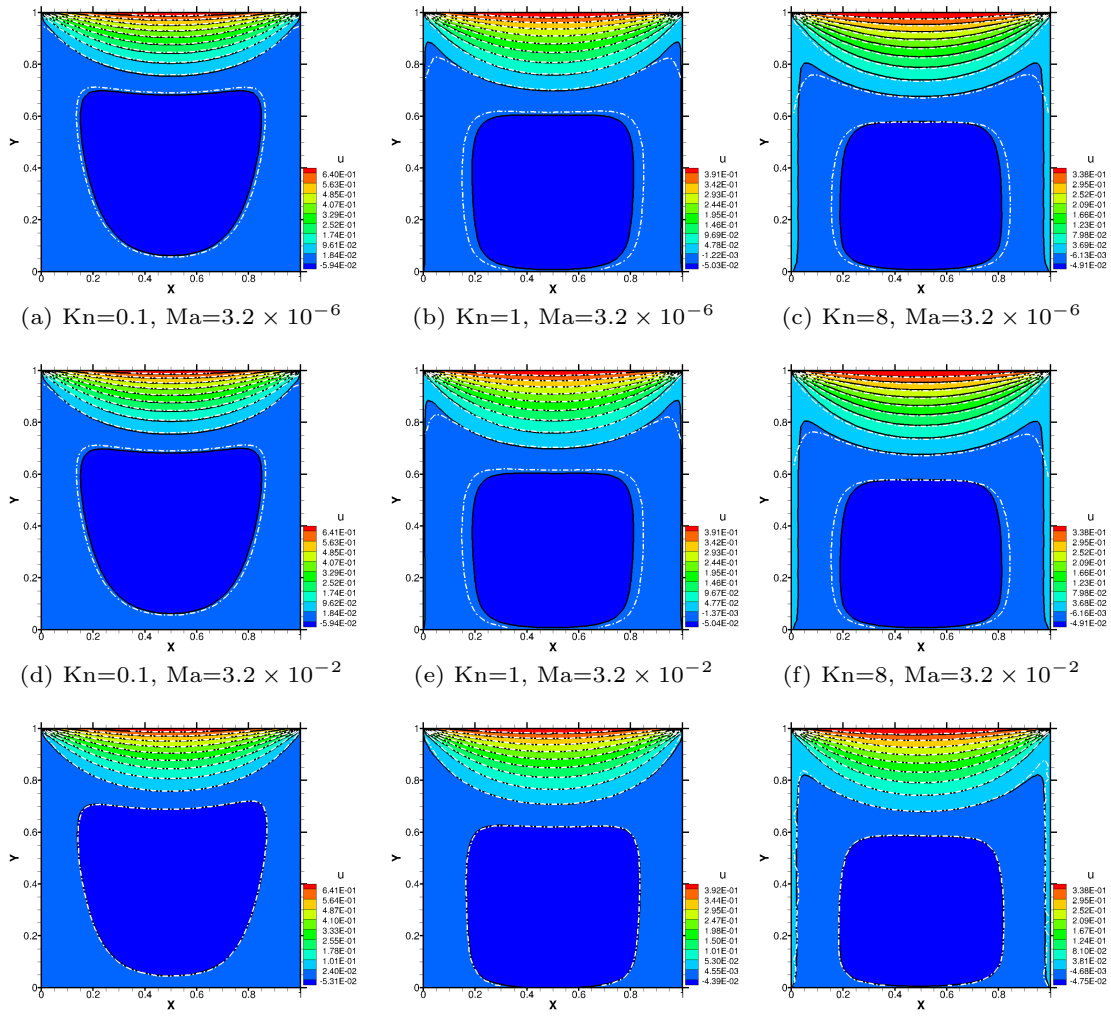
The impact of Mach number is investigated by fixing Kn $= 1$ and producing simulations for Ma $= 3.2 \times 10^{-6}$, 3.2×10^{-3}, 3.2×10^{-2}, and 0.16. DSMC results are, to the knowledge of the authors, not available in the literature for low-Ma flows, except results for Ma $= 0.16$ that have been included in Fig. 2. We therefore use the accurate DVM results as reference data, which are obtained using an 8×80 velocity grid.

In Figs. 3 and 4, we find that the values of u and v remain almost constant for all examined Ma, except their minima slightly increase for Ma $= 0.16$. The maximum u is located at the centre of the top wall, and u is negative in the major part of the cavity near the bottom, left and right walls. The positive and negative regions of v are found toward the left and right walls, respectively, and the extrema are located just under the top corners.

The u and v contour lines obtained by both the DVM and DSBGK simulations are identical at Ma $= 0.16$, in which the DSBGK uses the statistically accurate boundary condition, but are noticeably different at lower Mach numbers, in which the DSBGK uses the simple boundary condition. The maximum discrepancies are about 7.5% for u along $X = 0.5$, and 1.0% for v along $Y = 0.5$, at Ma $= 3.2 \times 10^{-6}$. The maximum discrepancies for the n and T profiles along the horizontal centerline, which are observed near the walls at Ma $= 3.2 \times 10^{-6}$, are 4.7% and 2%, respectively.

The contours of n are shown in Fig. 5. The gas is compressed and expanded near the top right and top left corners, respectively, causing a respective rise and fall in n at the top corners. The magnitude of the maximum and minimum in the n contours increases with Ma as the gas compression and expansion become stronger at higher lid speeds. The DVM and DSBGK results only differ slightly in their predicted n for all Ma.

The T contour plots in Fig. 6 show that the hot and cold regions in the flow field are toward the top right (gas compression) and top left (gas expansion) corners, respectively. Similar to the dependence of n on Ma, the maximum and minimum

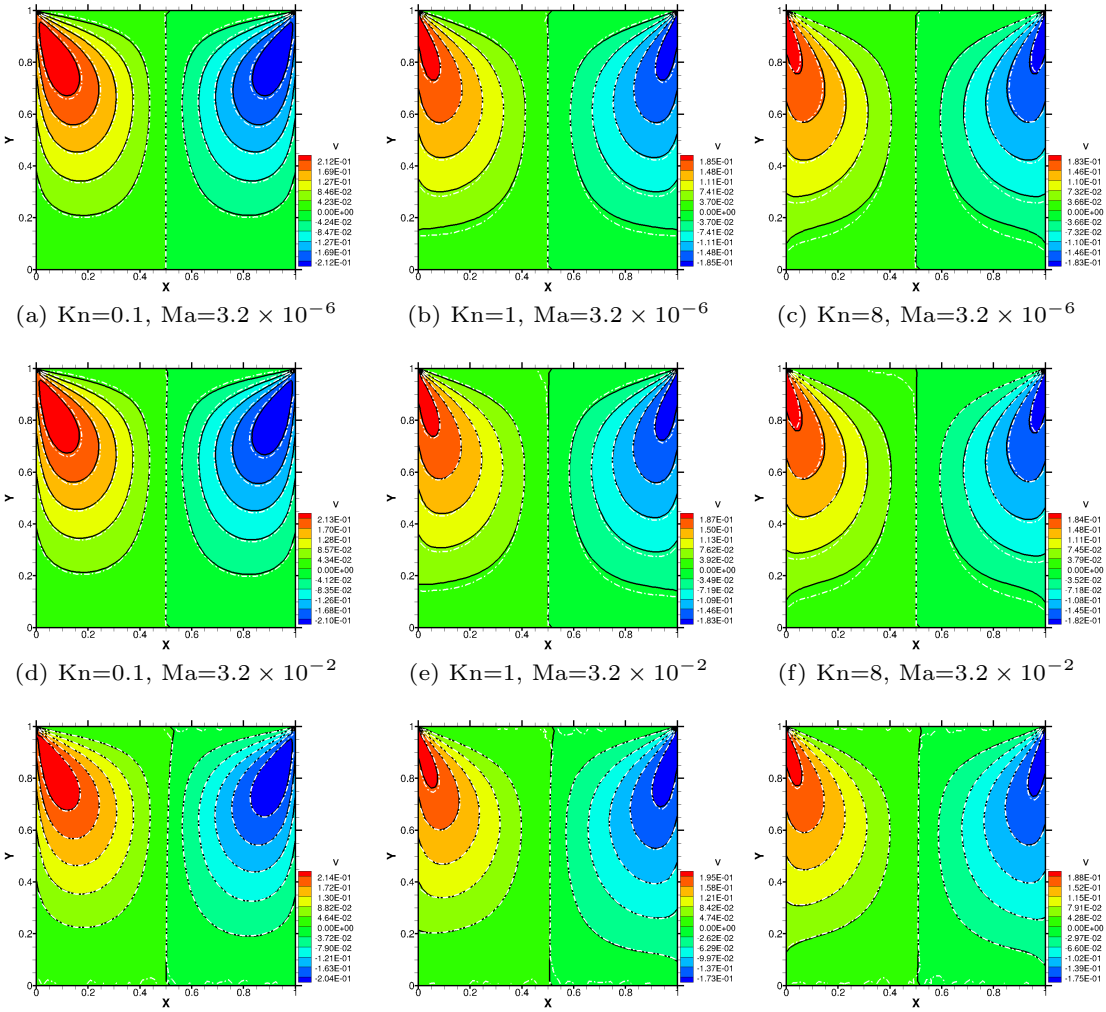


(a) Kn=0.1, Ma=3.2 × 10^{-6} (b) Kn=1, Ma=3.2 × 10^{-6} (c) Kn=8, Ma=3.2 × 10^{-6}

(d) Kn=0.1, Ma=3.2 × 10^{-2} (e) Kn=1, Ma=3.2 × 10^{-2} (f) Kn=8, Ma=3.2 × 10^{-2}

(g) Kn=0.1, Ma=0.16 (h) Kn=1, Ma=0.16 (i) Kn=8, Ma=0.16

Figure 3: Contours of the perturbed horizontal velocity u in the 2D lid-driven cavity flow obtained using the DVM (the black solid line and coloured background) and the DSBGK method (the white dash-dot line).

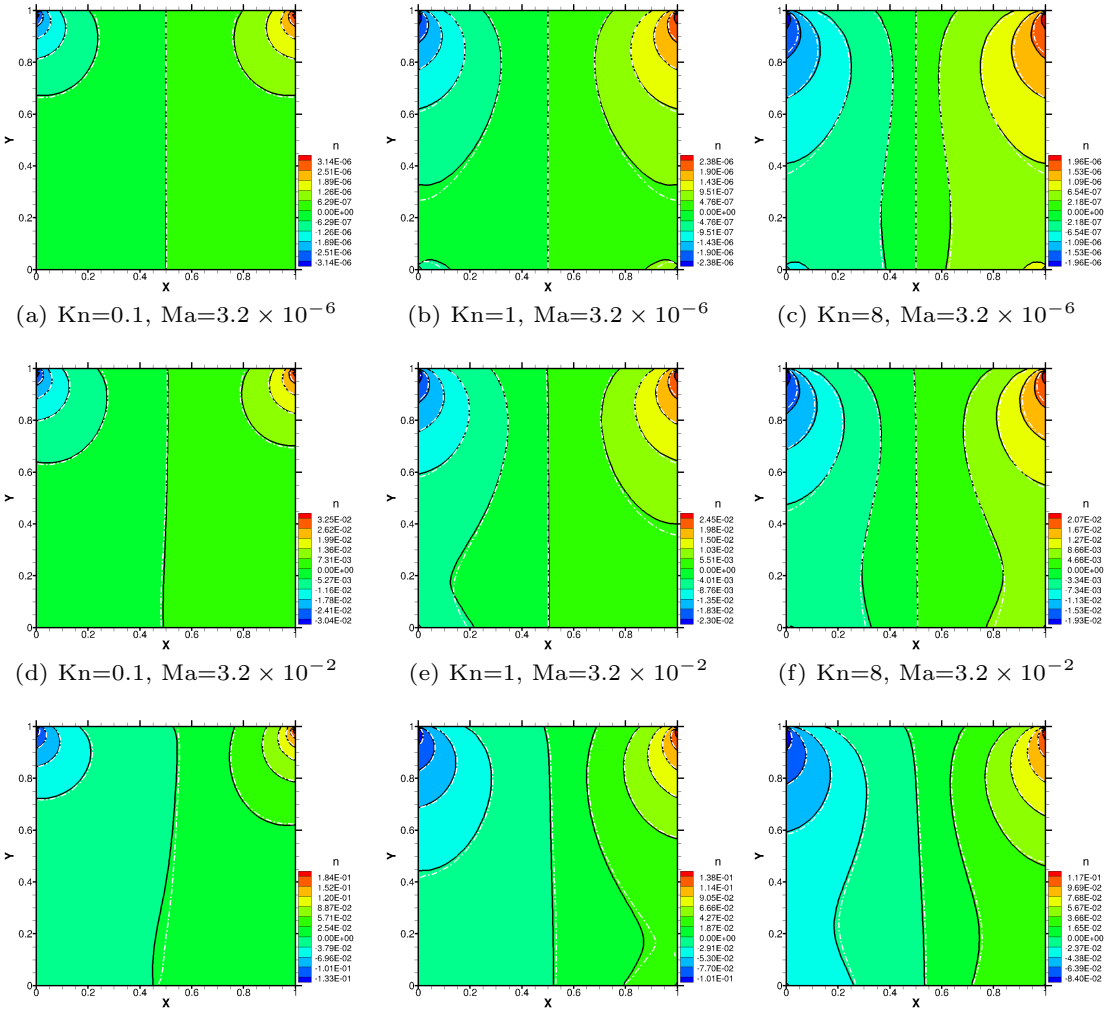


(a) Kn=0.1, Ma=3.2 × 10^{-6} (b) Kn=1, Ma=3.2 × 10^{-6} (c) Kn=8, Ma=3.2 × 10^{-6}

(d) Kn=0.1, Ma=3.2 × 10^{-2} (e) Kn=1, Ma=3.2 × 10^{-2} (f) Kn=8, Ma=3.2 × 10^{-2}

(g) Kn=0.1, Ma=0.16 (h) Kn=1, Ma=0.16 (i) Kn=8, Ma=0.16

Figure 4: Contours of the perturbed vertical velocity v in the 2D lid-driven cavity flow obtained using the DVM (the black solid line and coloured background) and the DSBGK method (the white dash-dot line).



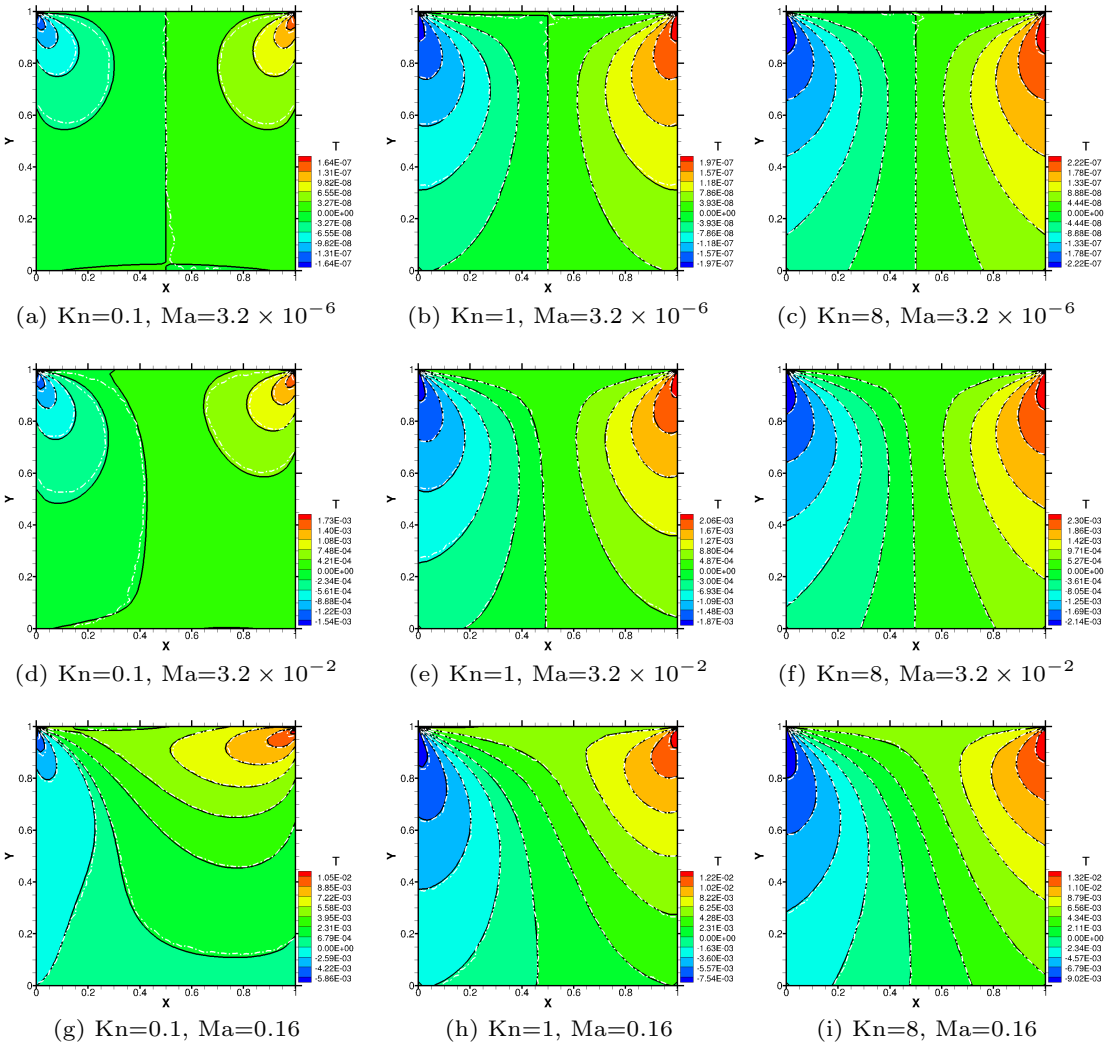
(a) Kn=0.1, Ma=3.2 × 10^{-6} (b) Kn=1, Ma=3.2 × 10^{-6} (c) Kn=8, Ma=3.2 × 10^{-6}

(d) Kn=0.1, Ma=3.2 × 10^{-2} (e) Kn=1, Ma=3.2 × 10^{-2} (f) Kn=8, Ma=3.2 × 10^{-2}

(g) Kn=0.1, Ma=0.16 (h) Kn=1, Ma=0.16 (i) Kn=8, Ma=0.16

Figure 5: Contours of the perturbed number density n in the 2D lid-driven cavity flow obtained using the DVM (the black solid line and coloured background) and the DSBGK method (the white dash-dot line).

temperatures near the top corners also vary with Ma. However, unlike with n, these significant variations of T are not confined to the top corners as the Mach number increases. Instead, the hot region expands leftward, while the cold region draws in towards the left and extends downwards. The T distributions produced by the DVM and DSBGK methods are generally in good agreement with each other for all Ma.



(a) Kn=0.1, Ma=3.2×10^{-6} (b) Kn=1, Ma=3.2×10^{-6} (c) Kn=8, Ma=3.2×10^{-6}

(d) Kn=0.1, Ma=3.2×10^{-2} (e) Kn=1, Ma=3.2×10^{-2} (f) Kn=8, Ma=3.2×10^{-2}

(g) Kn=0.1, Ma=0.16 (h) Kn=1, Ma=0.16 (i) Kn=8, Ma=0.16

Figure 6: Contours of the perturbed temperature T in the 2D lid-driven cavity flow obtained using the DVM (the black solid line and coloured background) and the DSBGK method (the white dash-dot line).

3.3. Effect of Knudsen number

To assess the effects of rarefaction, we compare our simulation results at Ma = 3.2×10^{-2}, and Knudsen numbers Kn = 0.1, 1 and 8. The macroscopic flow fields

are presented in subfigures d), e) and f) of Figs. 3 to 6. Although the u and v flow patterns remain almost unchanged, the magnitudes of u and v decline with increasing Kn. For Knudsen numbers increasing from 0.1 to 8, the magnitude of the maximum (minimum) u reduces around 47% (17%); for v, it is about 14% (13%).

Likewise, the magnitudes of the maxima and minima in the n field decrease with increasing Kn. From Kn $= 0.1 \rightarrow 8$, the magnitudes of the maximum and minimum of n reduce by about 37%. The regions of low and high n also expand downward with increasing Kn. The temperature in the hottest (coldest) region increases (decreases) by 33% (39%) as the Knudsen number rises from 0.1 to 8.

Using the simple boundary condition in DSBGK produces very good agreement with DVM for n and T contours at all Kn. The u and v contours of DSBGK deviate slightly from those of DVM at Kn $= 1, 8$, and this deviation diminishes at Kn $= 0.1$.

3.4. Effects of velocity and spatial grids, and computational costs

So far, we have focused on the accuracy of the DVM and DSBGK methods at different Kn and Ma using either a high resolution velocity grid or a large number of simulated molecules per cell. For practical applications, it is important to strike a balance between computational accuracy and efficiency. So we test different reduced velocity grids in the DVM simulations and reduced numbers of simulated molecules per cell in the DSBGK simulations, for Ma $= 3.2 \times 10^{-3}$ and $Kn = 0.1, 1$ and 8. The simple boundary is again used to stabilize the DSBGK simulations. The timestep in the DSBGK simulations is fixed at $\Delta t = 2.0\Delta x\sqrt{m/(2k_{\mathrm{B}}T_0)}$ for different Kn to clarify the relation between the timesteps and the corresponding CPU simulation time, which is the wall clock time for both DVM and DSBGK simulations, unless stated otherwise.

Allowing tolerances of a maximum local relative deviation (from the accurate DVM results obtained with an 8×80 velocity grid) of 10% in u, v, n, T profiles along the horizontal centreline, the velocity grids of the DVM method can be reduced to 4×24, 4×40, 4×48 at Kn $= 0.1, 1, 8$, respectively, while the number of simulated molecules per cell in the DSBGK method can be decreased to 10 for all Kn. Figures 7 to 9 show the contours of macroscopic quantities obtained by the DVM and the

DSBGK using these numerical parameters, alongside the reference contours obtained by the DVM with an 8×80 velocity grid.

The accuracy of the DSBGK method does not deteriorate when using only 10 simulated molecules per cell, except for a small drift in the density distribution. However, the stochastic noise is significantly increased, so 5000 samples are needed to smooth the results. Although the number of simulated molecules per cell used by the DSBGK method is small, the molecular velocities inside each cell are dynamically updated via the frequent and random events of simulated molecules moving into and out of each cell (from the perspective of the Eulerian description). The molecular velocities along all representative trajectories are also dynamically updated via the frequent and random molecular reflections at the boundary (from the perspective of the Lagrangian description). Thus, the *dynamic* discretization of using few simulated molecules per cell in the DSBGK method can sample from the whole velocity space and therefore allow as fine discretization of the unbounded molecular velocity space as desired with the increase of simulation time. This is the same as in the DSMC method.

The DVM contours with coarse velocity grids can be seen to oscillate around the reference solutions in the regions far from the two top corners. This can be explained as "ray effects", which are known as major shortcomings of the DVM when applied to neutron transport and radiative transfer problems involving discontinuous boundary conditions [47, 48, 49]. These effects are due to the finite discretization of velocity space tending to capture discontinuities, whereas the finite discretization of spatial space tending to smooth the flow field. The velocity grid therefore should be fine enough so that the error due to ray effects can be compensated by the error due to numerical diffusion, which is related to the spatial grid and spatial discretization scheme [50].

In Fig. 10, the ray effects (wavy contours) can be clearly observed in the DVM solution with a 60^2 spatial grid and a 4×40 velocity grid, for the case of $\mathrm{Ma} = 3.2 \times 10^{-3}, \mathrm{Kn} = 8$. The effects are diminished by increasing the velocity grid to 4×80, by reducing the spatial grid to 30^2, or by using a lower-order scheme, i.e. a first order upwind scheme, rather than Eq. (22). The contours obtained by the first

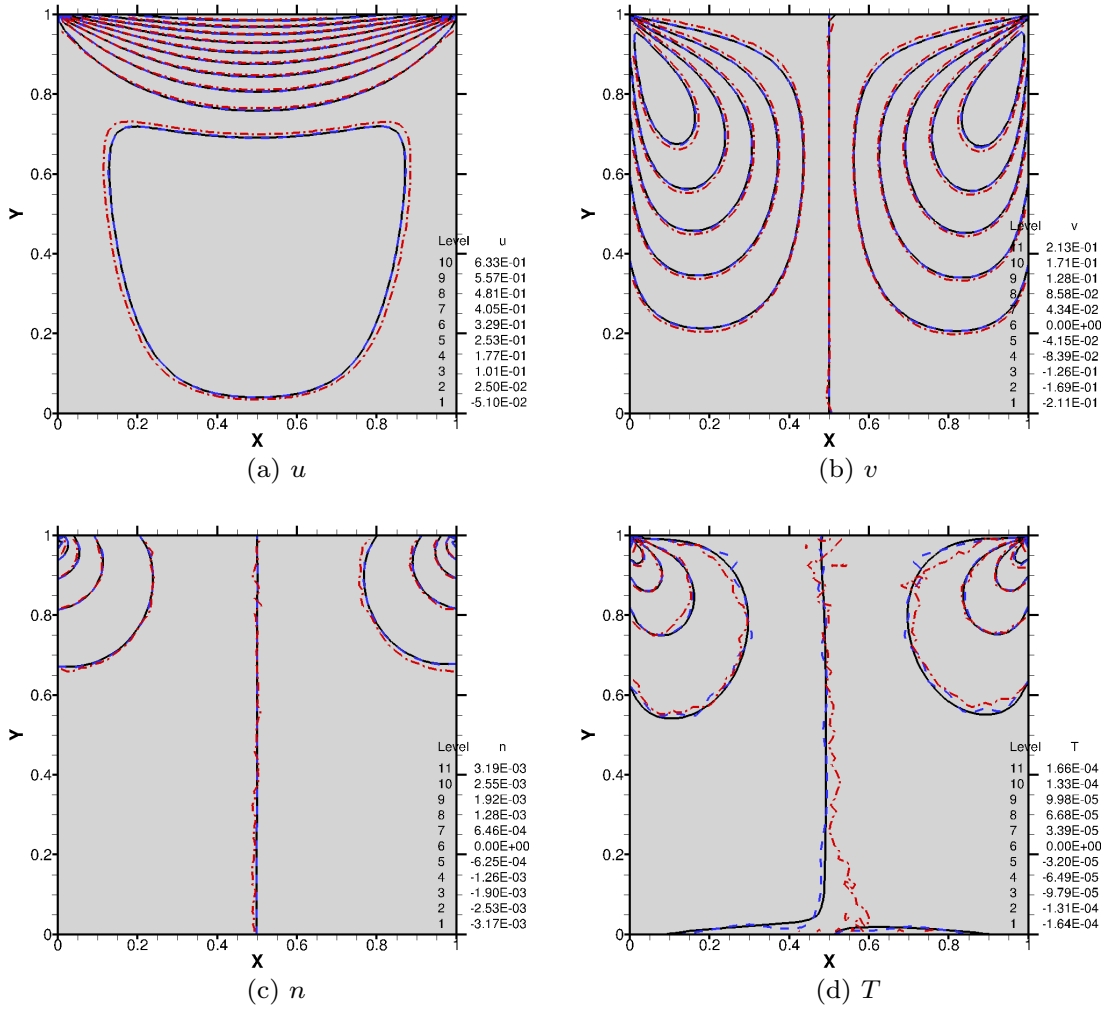
two options are almost identical to each other (and to the reference contours shown in Fig. 9) and slightly differ from the results using the third option. Among these three options for this case of high Kn, the second option, i.e. compatible coarse velocity and spatial grids with a high-order upwind scheme, provides accurate data at relatively cheap computational cost.

In addition to velocity grid size, the layout of velocity grid is also important to mitigate ray effects. For instance, a 32^2 velocity grid in Cartesian coordinates yields significant ray effects, whereas a 4×80 velocity grid in polar coordinates (about one-third size of the Cartesian grid) does not. These ray effects are sensitive to N_φ but not to N_{c_p}, e.g. a 4×80 velocity grid can provide an identical solution to an 8×80 velocity grid (see Figs. 9 and 10).

In addition, Figs. 7 to 9 show that ray effects increase with Knudsen number, since the collision process is dominated by the streaming process. To mitigate ray effects when a linearized kinetic equation is used, the perturbed distribution function can be split into two parts: the part induced by the wall velocity can be solved analytically along characteristic directions, whereas the other part is solved numerically [36].

It is also important to compare the computational costs of the DVM and DSBGK methods in achieving the required solution tolerances. A comparison of computational costs is given in Table 1. All the serial calculations are performed using a single CPU core on an Alfahd compute node (High-Performance Computing facility at King Fahd University of Petroleum and Minerals) with an Intel Xeon CPU E5-2680 v4 and 128 GB of memory. The codes are compiled with Intel Fortran Compiler version 18.0 using `-O3 -xCORE-AVX2` flags. The total number of timesteps required by the DVM simulations is minimal when $\mathrm{Kn} = 1$, while that required by the DSBGK method increases with Kn. The DSBGK method requires less than one minute of CPU time to obtain converged solutions, which is about 2 to 15 times more expensive than the DVM. However, the overall CPU time for a DSBGK simulation is much longer, about 50 to 80 times higher than that of DVM, due to the time-averaging process.

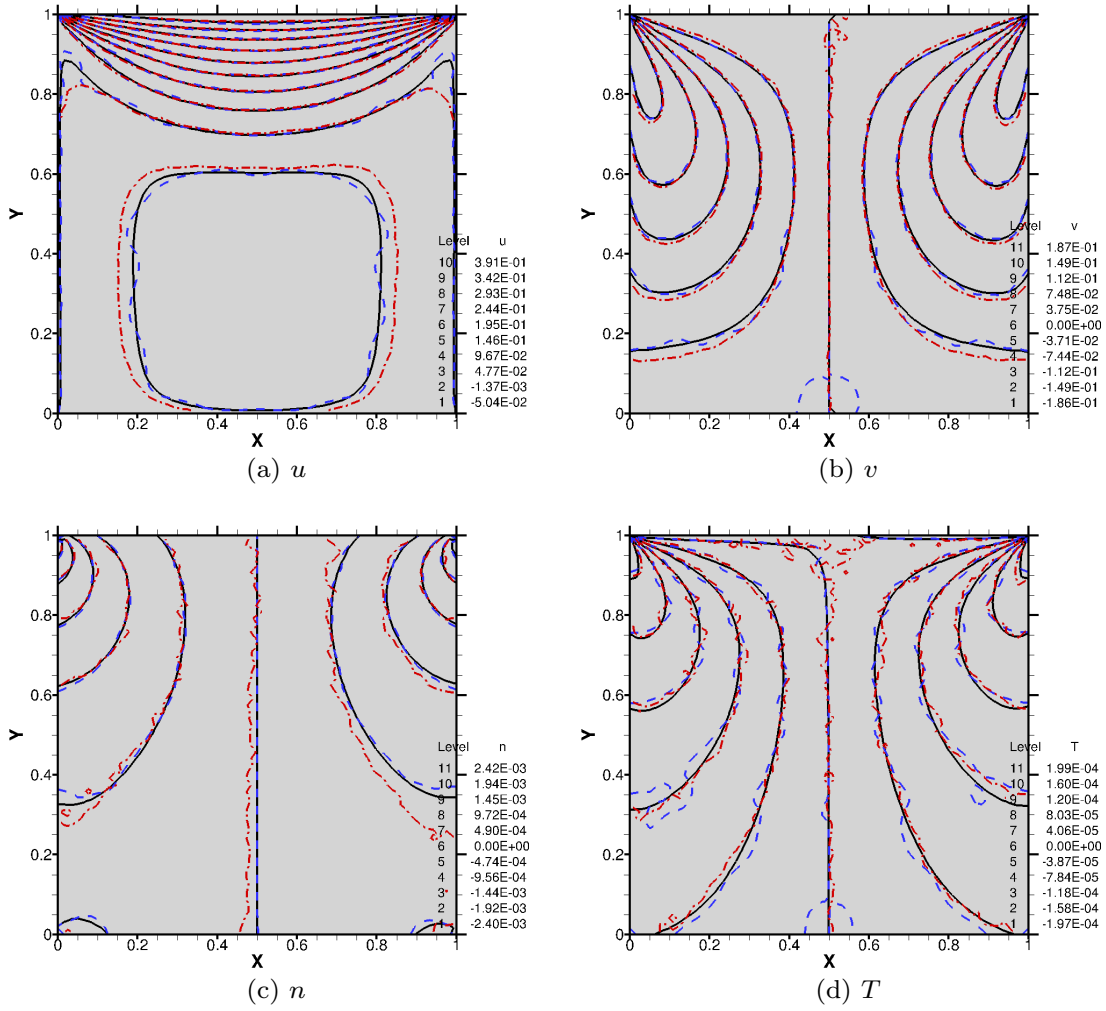
For large-scale simulations, the efficiency of the DSBGK method should be en-



(a) u (b) v

(c) n (d) T

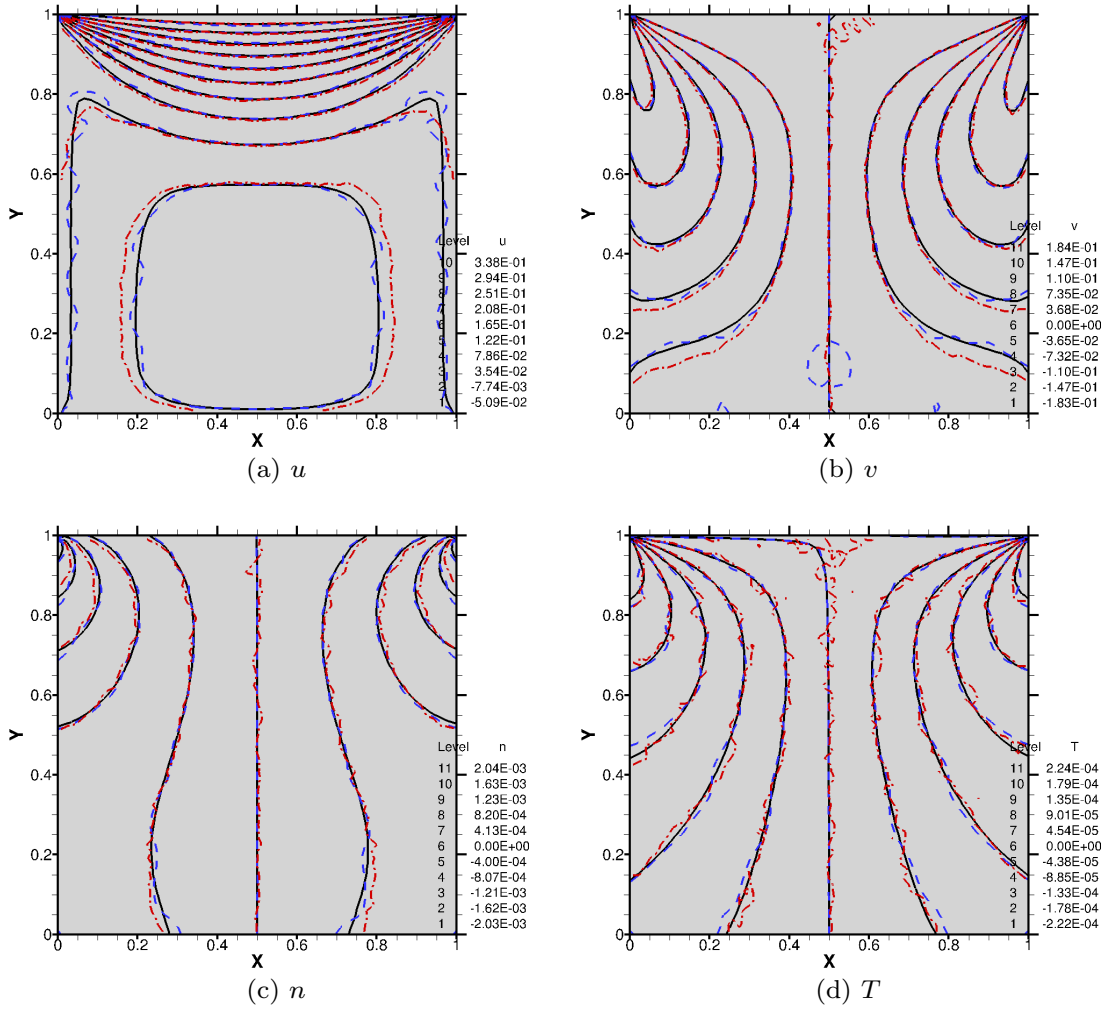
Figure 7: Contours of the perturbed u, v, n and T in the 2D lid-driven cavity flow obtained by the DVM using an 8×80 velocity grid (black solid lines), the DVM using a 4×24 velocity grid (blue dashed lines), and the DSBGK method using 10 simulated molecules per cell with the simple boundary condition and 5000 samples (red dash-dot lines); $\mathrm{Ma} = 3.2 \times 10^{-3}, \mathrm{Kn} = 0.1$.



(a) u (b) v

(c) n (d) T

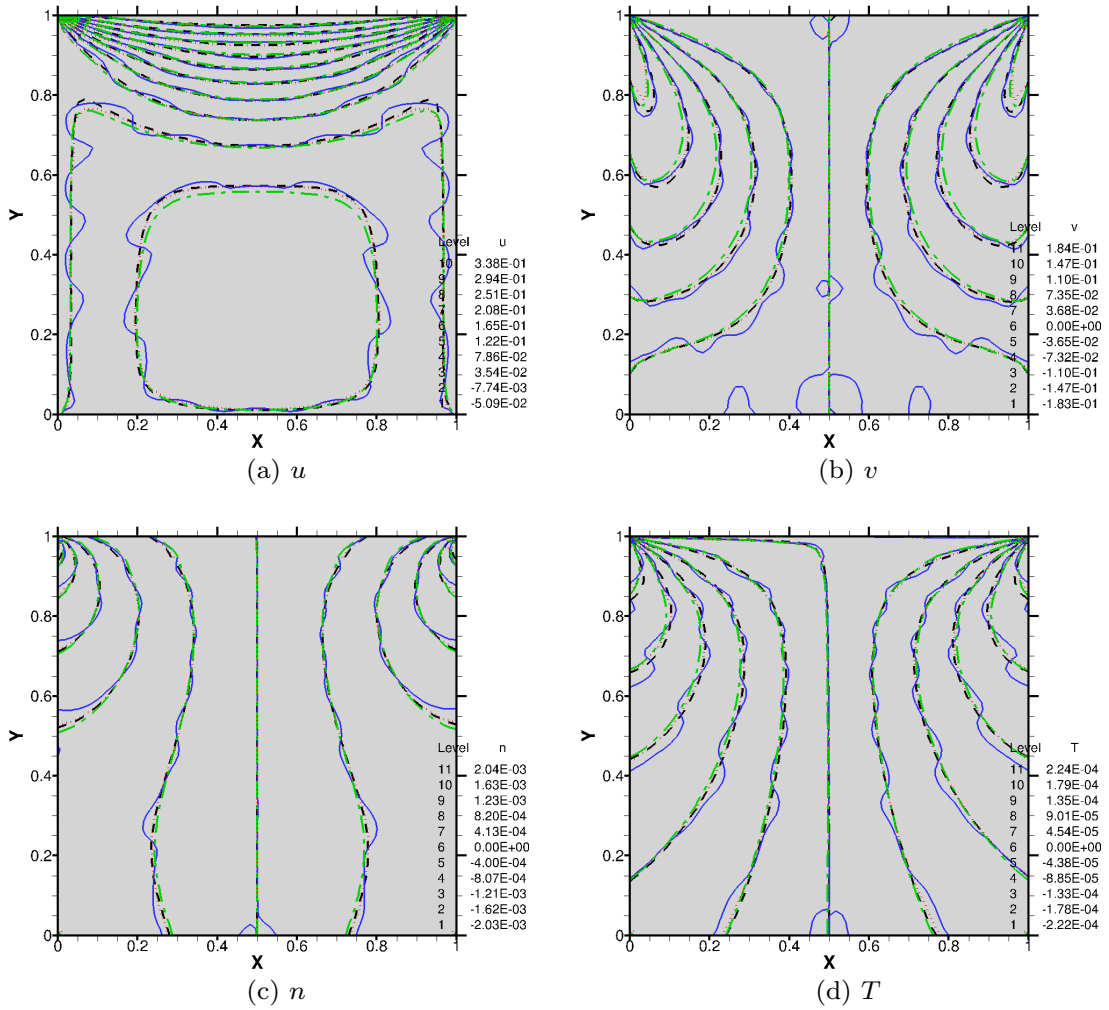
Figure 8: Contours of the perturbed u, v, n and T in the 2D lid-driven cavity flow obtained by the DVM using an 8×80 velocity grid (black solid lines), the DVM using a 4×40 velocity grid (blue dashed lines), and the DSBGK method using 10 simulated molecules per cell with the simple boundary condition and 5000 samples (red dash-dot lines); $\mathrm{Ma} = 3.2 \times 10^{-3}, \mathrm{Kn} = 1$.



(a) u (b) v

(c) n (d) T

Figure 9: Contours of the perturbed u, v, n and T in the 2D lid-driven cavity flow obtained by the DVM using an 8×80 velocity grid (black solid lines), the DVM using a 4×48 velocity grid (blue dashed lines), and the DSBGK method using 10 simulated molecules per cell with the simple boundary condition and 5000 samples (red dash-dot lines); $\mathrm{Ma} = 3.2 \times 10^{-3}, \mathrm{Kn} = 8$.



(a) u (b) v

(c) n (d) T

Figure 10: For the 2D flow case with $\mathrm{Ma} = 3.2 \times 10^{-3}, \mathrm{Kn} = 8$, ray effects (wavy contours) in the DVM solution when using a 4×40 velocity grid and a 60^2 spatial grid (blue solid lines) are diminished by increasing the velocity grid to 4×80 (black dash lines) or by decreasing the spatial grid to 30^2 (red dot lines) or by reducing the order of accuracy to a first order upwind scheme (green dash dot line).

Table 1: The computational costs of using the DVM and DSBGK methods in the 2D lid-driven cavity flow problem for Ma $= 3.2 \times 10^{-3}$ and with 60^2 spatial grids/cells.

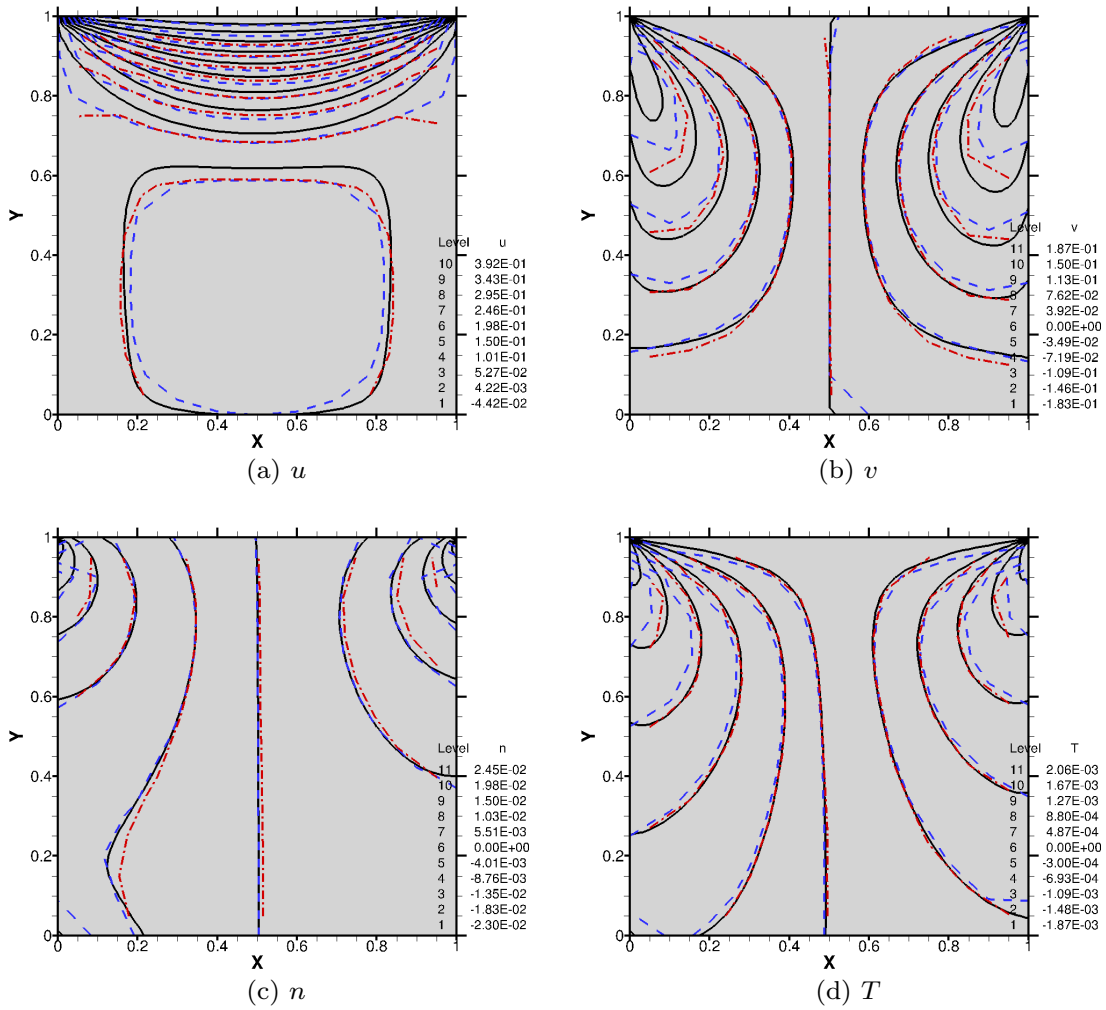
	DVM			DSBGK		
Kn	0.1	1	8	0.1	1	8
velocity grids $N_{c_p} \times N_\varphi$	4×24	4×40	4×48	-	-	-
simulated molecules per cell	-	-	-	10	10	10
timesteps for convergence	124	44	51	200	400	1600
timesteps in total	124	44	51	5200	5400	6600
CPU time (s) for convergence	0.97	0.65	1.05	2	4	16
overall CPU time (s)	0.97	0.65	1.05	51	53	65

hanced, since the fixed CPU time used for the time-averaging process (i.e. 5000 Δt, as we used here) will become negligible compared to the CPU time needed for convergence (e.g. increasing from 200 $\sim$ 1600 Δt used here to millions of Δt). Thus, the comparison of CPU time required for the convergence process alone is also an important indicator for large-scale problems. Note also that the comparison of efficiency made here is for this 2D case, where the discretization of velocity space in the third direction is not required in the DVM simulations. Further efficiency comparisons will be made below for the 3D case, where the full 3D velocity grid should be used in DVM.

The computational cost can be further reduced by using coarse spatial grids, as long as the cell size is smaller than the mean free path (as in DSMC simulations). We therefore report on how the simulation accuracy is affected when the number of spatial grids/cells reduces from 60^2 to 10^2, for the case with Kn $= 1$, Ma $= 3.2 \times 10^{-2}$. Since we focus on the discretization in physical space, high resolution is still maintained in the velocity space here, i.e. the DVM simulation uses an 8×80 velocity grid, while the DSBGK simulation employs 2000 simulated molecules per cell and the simple boundary condition.

Figure 11 shows the contours of the macroscopic quantities obtained by the DSBGK and DVM methods with the coarse spatial grid. The DVM results using a 60^2 spatial grid are also plotted as a reference solution. It can be seen that the n

and T contours given by both methods on the coarse spatial grid are in satisfactory agreement with the reference ones. However, the u and v contours of the coarse solutions of both methods have some deviations from the reference solutions. This numerical error is expected to occur also with other simulation methods when using coarse spatial grids to save computational cost. Balancing computational accuracy with efficiency becomes a key issue when simulating large-scale problems, e.g. gas flows in porous media.



(a) u (b) v (c) n (d) T

Figure 11: Contours of the macroscopic flow quantities in the 2D lid-driven cavity with Kn = 1 and 3.2×10^{-2} obtained using a 60^2 spatial grid (DVM: black solid lines) and a 10^2 spatial grid (DVM: blue dashed lines, DSBGK: red dash-dot lines).

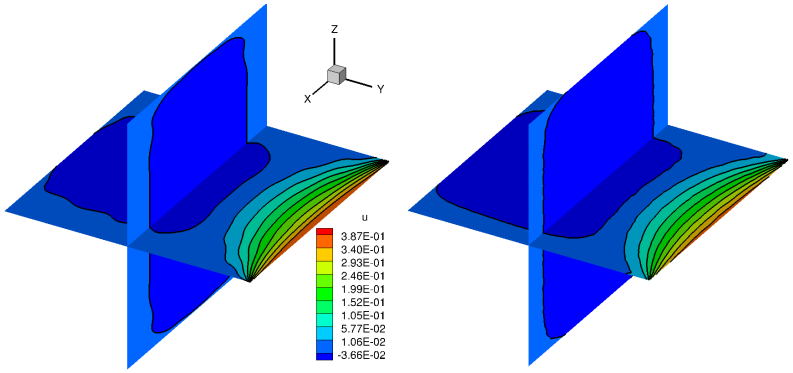
4. 3D cavity flows

We now consider the rarefied gas flow inside a cubic cavity of size $L = 1\ \mu m$, which is an extension in the Z-direction of the square cavity examined in Section 3. The top lid (in the $X - Z$ plane at $Y = 1$) moves in the positive X-direction with a constant speed $u_{\rm w}$, see Fig. 1 (b). We fix $\mathrm{Ma} = 3.2 \times 10^{-3}$ and choose $\mathrm{Kn} = 0.1$, 1 and 8, to cover the slip, transition, and free-molecular flow regimes. The number of uniform spatial cells/grids is 60^3 for both the DSBGK and DVM simulations. We use the accurate DVM results as reference data, which are obtained using a $4 \times 80 \times 40$ velocity grid.

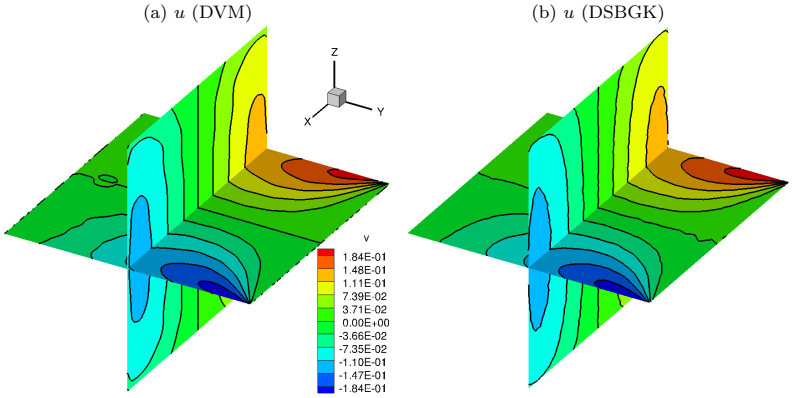
By comparing the u, v, n, T contours on the plane $Z = 0.5$ obtained by the 3D reference solutions with those obtained by the 2D reference solutions (shown in Figs. 7-9), the side wall ($Z = 0, 1$) effects on the middle plane are seen to be negligible when $\mathrm{Kn} = 0.1$. Although it is not illustrated here, the side wall effects increase with Knudsen number and change the T profiles most significantly.

We examine effects of using 10 simulated molecules per cell in the DSBGK method, and coarse velocity grids in the DVM. The coarse DVM grids used in the 2D case are now extended in the inclination θ with $N_\theta = 12$ for this 3D case, i.e. $4 \times 24 \times 12$, $4 \times 40 \times 12$, $4 \times 48 \times 12$ for $\mathrm{Kn} = 0.1$, 1, 8, respectively. The deviations of both methods from the reference solution on the plane $Z = 0.5$ are similar to the 2D case. For example, Fig. 12 shows the u, v, n, T contours on the planes $Z = 0.5$ and $Y = 0.5$ for the case with $\mathrm{Kn} = 8$. The DVM results (left column of Fig. 12) show that although the number of distinct discrete c_z ($N_{c_p} \times N_\theta$) is much less than that of distinct discrete c_x or c_y ($N_{c_p} \times N_\theta \times N_\varphi$), the ray effects (wavy contours) observed in the plane $Y = 0.5$ are less obvious than those in the plane $Z = 0.5$. Moreover, the ray effects in the plane $Y = 0.5$ are more pronounced near the two lateral walls at $Z = 0, 1$. These can be explained by the fact that the discontinuities in wall velocity with respect to the $X-$ and $Y-$directions exist in all the $Z-$planes, while those with respect to the $Z-$direction exist only in the planes $Z = 0, 1$.

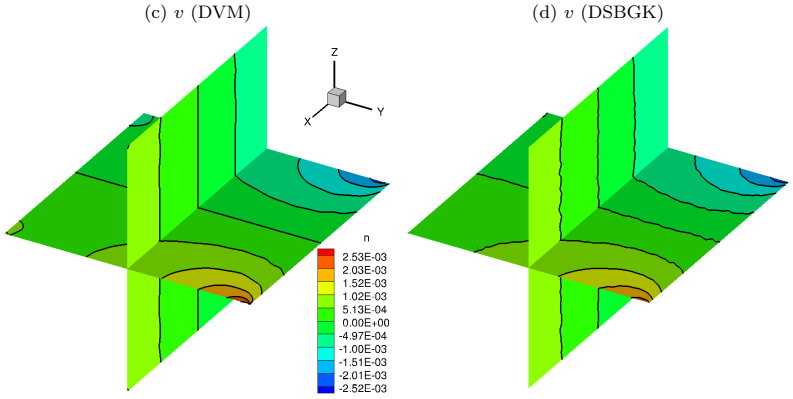
The serial computational costs of the two numerical methods for the 3D cavity problem, using the same machine as in the 2D cavity problem, are compared in Table 2. Dependence of the number of timesteps on Knudsen number recalls that



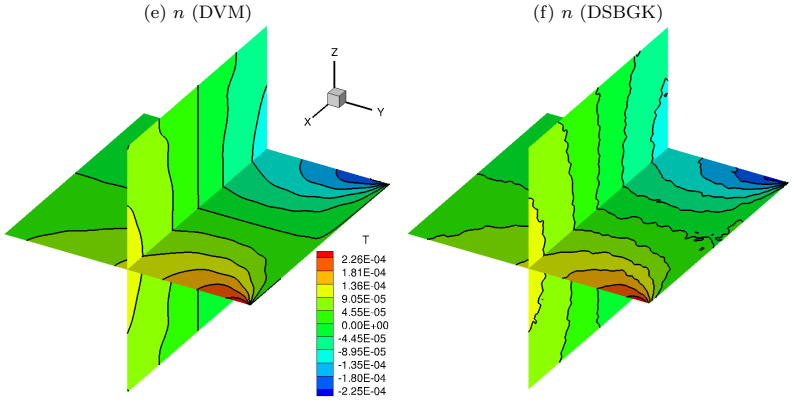
(a) u (DVM) (b) u (DSBGK)



(c) v (DVM) (d) v (DSBGK)



(e) n (DVM) (f) n (DSBGK)



(g) T (DVM) (h) T (DSBGK)

Figure 12: The 3D lid-driven case with $\mathrm{Ma} = 3.2 \times 10^{-3}$, $\mathrm{Kn} = 8$: contours of the perturbed u, v, n and T on the planes $Z = 0.5$ and $Y = 0.5$, obtained by the DVM using a $4 \times 48 \times 12$ velocity grid (left column), and the DSBGK method using 10 simulated molecules per cell with the simple boundary condition and 5000 samples (right column).

Table 2: The computational costs of using the DVM and DSBGK methods in the 3D lid-driven cavity flow problem for Ma $= 3.2 \times 10^{-3}$ and with 60^3 spatial grids/cells.

	DVM			DSBGK		
Kn	0.1	1	8	0.1	1	8
velocity grids $N_{c_p} \times N_\varphi \times N_\theta$	$4 \times 24 \times 12$	$4 \times 40 \times 12$	$4 \times 48 \times 12$	-	-	-
simulated molecules per cell	-	-	-	10	10	10
timesteps for convergence	146	67	278	200	400	2400
timesteps in total	146	67	278	5200	5400	7400
CPU time (s) for convergence	1162	882	4473	185	376	2279
overall CPU time (s)	1162	882	4473	4857	5115	7033

of the 2D case for both DVM and DSBGK. To reach steady state, the CPU time spent by DVM is about 2 to 6 times more than that needed by the DSBGK method. Including the time-averaging process, however, makes the overall computational cost of the DSBGK method from 1.6 to 5.8 times that of the DVM.

5. Conclusions

We have compared the DVM and DSBGK methods for solving the BGK gas kinetic model applied to low-speed lid-driven cavity flows over a range of Knudsen and Mach numbers. For the 2D case with Ma $= 0.16$, the velocity profiles along the centrelines predicted by the two methods are in good agreement with those reported using the standard DSMC method [37]. When the Mach number decreases, the velocity profiles predicted by the DVM and DSMC methods are essentially independent of Ma, while those of the DSBGK method vary by up to 7.5% (in the u profile) and 2.5% (in the v profile) due to the use of the simple boundary condition. This maximum discrepancy occurs only in the region with relatively small perturbations. Overall, the DVM and DSBGK techniques produce results in good agreement.

For 2D case, the magnitudes of the perturbed number density and perturbed temperature in the flow field are observed to increase with the Mach number, while the velocity field remains nearly unchanged. Among the investigated macroscopic quantities, only the temperature distribution is sensitive to the Mach number: the

cooler region constrict leftward and expands downward with increasing Mach number while the hotter region expands leftward. For both 2D and 3D cases, the variations of all the macroscopic quantities, except the temperature, reduce significantly with increasing Knudsen number. The regions of low/high number density and temperature in the flow field expand with increasing Knudsen number, while the velocity distributions remain unchanged.

To reduce the "ray effects" in the DVM simulations that are induced by the discontinuous boundary condition, the velocity grid should be compatible with the spatial grid and the accuracy order of the numerical scheme. With a third-order upwind scheme and a 60^2 spatial grid, the molecular velocity grids of the DVM in 2D can be reduced to $4 \times 24, 4 \times 40, 4 \times 48$ points for $\mathrm{Kn} = 0.1, 1, 8$, respectively, if we accept a maximum local relative error of 10% in the u, v, n, T profiles along the horizontal centreline of the 2D cavity. For the 3D cavity with a 60^3 spatial grid, the corresponding velocity grids of the DVM are $4 \times 24 \times 12, 4 \times 40 \times 12, 4 \times 48 \times 12$.

Similarly, the number of simulated molecules per cell in the DSBGK method can also be reduced to 10 for all Kn to achieve the same tolerance. The time-averaging process in the DSBGK method needs to be prolonged with decreasing number of simulated molecules per cell to reduce stochastic noise in the solution. This modest number of simulated molecules per cell does, however, result in a much smaller memory requirement for the DSBGK method than for the DVM.

The total number of DSBGK timesteps increases with Kn, while the required DVM timesteps are at a minimum at $\mathrm{Kn} = 1$. Compared to a deterministic method like the DVM, the DSBGK method needs additional computational effort for the time-averaging process. For 2D flow problems, the DVM velocity space can also be projected from 3D to 2D to make computational savings. Consequently, the overall CPU time consumed by the DSBGK simulations is much larger than that of the DVM in the 2D cavity case, although for the 3D case the computational costs of the two methods become comparable. The efficiency of the DSBGK method can be expected to be enhanced in large-scale 3D simulations, where the required CPU time for time-averaging becomes a negligible part of the overall simulation.

Acknowledgements

This publication is based on work supported by the College of Petroleum Engineering and Geosciences, King Fahd University of Petroleum and Minerals, Saudi Arabia. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper. All data within this publication can be freely accessed at http://dx.doi.org/

References

References

[1] C.-M. Ho, Y.-C. Tai, Micro-Electro-Mechanical- Systems (MEMS) and fluid flows, Annu Rev Fluid Mech 30 (1998) 579–612.

[2] H. Darabi, A. Ettehad, F. Javadpour, K. Sepehrnoori, Gas flow in ultra-tight shale strata, J Fluid Mech 710 (2012) 641–658.

[3] G. A. Bird, Molecular gas dynamics and the direct simulation of gas flows, Oxford Science Publications, 1994.

[4] C. Cercignani, Rarefied Gas Dynamics: From basic concepts to actual calculations, Cambridge University Press, 2000.

[5] S. Rjasanow, T. Schreiber, W. Wagner, Reduction of the number of particles in the Stochastic Weigthed Particle Method for the Boltzmann equation, J Comput Phys 145 (1998) 382–405.

[6] A. Crestetto, N. Crouseilles, M. Lemou, Kinetic/fluid micro-macro numerical schemes for Vlasov-Poisson-BGK equation using particles, Kinet Relat Model 5 (4) (2012) 787–816.

[7] C. White, T. J. Scanlon, R. E. Brown, Permeability of ablative materials under rarefied gas conditions, J Spacecr Rockets 53 (1) (2016) 134–142.

[8] R. Jambunathan, D. A. Levin, Advanced parallelization strategies using hybrid MPI-CUDA octree DSMC method for modeling flow through porous media, Comput Fluids 149 (2017) 70–87.

[9] G. Yang, B. Weigand, Investigation of the Klinkenberg effect in a micro/nanoporous medium by direct simulation Monte Carlo method, Phys Rev Fluids 3 (4) (2018) 1–17.

[10] J. Fan, C. Shen, Statistical simulation of low-speed rarefied gas flows, J Comput Phys 167 (2) (2001) 393–412.

[11] L. L. Baker, N. G. Hadjiconstantinou, Variance reduction for Monte Carlo solutions of the Boltzmann equation, Phys Fluids 17 (5) (2005) 1–4.

[12] T. M. M. Homolle, N. G. Hadjiconstantinou, A low-variance deviational simulation Monte Carlo for the Boltzmann equation, J Comput Phys 226 (2) (2007) 2341–2358.

[13] T. M. M. Homolle, N. G. Hadjiconstantinou, Low-variance deviational simulation Monte Carlo, Phys Fluids 19 (4) (2007) 041701.

[14] 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.

[15] P. Bhatnagar, E. Gross, M. Krook, A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems, Phys Rev 94 (3) (1954) 511–525.

[16] G. A. Radtke, N. G. Hadjiconstantinou, Variance-reduced particle simulation of the Boltzmann transport equation in the relaxation-time approximation, Phys Rev E 79 (5) (2009) 1–9.

[17] L. Szalmas, Variance-reduced DSMC for binary gas flows as defined by the McCormack kinetic model, J Comput Phys 231 (9) (2012) 3723–3738.

[18] G. Dimarco, L. Pareschi, Hybrid multiscale methods II. Kinetic equations, Multiscale Model Simul 6 (4) (2008) 1169–1197.

[19] G. Dimarco, L. Pareschi, Fluid solver independent hybrid methods for multiscale kinetic equations, SIAM J Sci Comput 32 (2) (2010) 603–634.

[20] P. Degond, G. Dimarco, L. Pareschi, The moment-guided Monte Carlo method, Int J Numer Methods Fluids 67 (2) (2011) 189–213.

[21] N. Crouseilles, G. Dimarco, M. Lemou, Asymptotically preserving and time diminishing schemes for rarefied gas dynamic, Kinet Relat Model 10 (3) (2017) 643–668.

[22] J. Li, Direct simulation method based on BGK equation, in: Proceedings of the 27th International Symposium on Rarefied Gas Dynamics, Vol. 1333, 2011, pp. 283–288.

[23] J. Li, Efficiency and stability of the DSBGK method, AIP Conference Proceedings 1501 (1) (2012) 849–856.

[24] J. Li, Comparison between the DSMC and DSBGK methods, arXiv.org physics.co (2012) 1207.1040.

[25] J. E. Broadwell, Study of rarefied shear flow by the discrete velocity method, J Fluid Mech 19 (3) (1964) 401–414.

[26] J. Yang, J. Huang, Rarefied flow computations using nonlinear model Boltzmann equations, J Comput Phys 120 (2) (1995) 323–339.

[27] K. Aoki, K. Kanba, S. Takata, Numerical analysis of a supersonic rarefied gas flow past a flat plate, Phys Fluids 9 (4) (1997) 1144–1161.

[28] F. Sharipov, V. Seleznev, Data on internal rarefied gas flows, J Phys Chem Ref Data 27 (3) (1998) 657–706.

[29] 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.

[30] L. M. Yang, C. Shu, W. M. Yang, J. Wu, An implicit scheme with memory reduction technique for steady state solutions of DVBE in all flow regimes, Phys Fluids 30 (4) (2018) 040901.

[31] C. K. Chu, Kinetic-theoretic description of the formation of a shock wave, Phys Fluids 8 (1) (1965) 12–22.

[32] V. A. Titarev, Conservative numerical methods for model kinetic equations, Comput Fluids 36 (9) (2007) 1446–1459.

[33] A. L. Garcia, W. Wagner, Time step truncation error in direct simulation Monte Carlo, Phys Fluids 12 (10) (2000) 2621–2633.

[34] N. G. Hadjiconstantinou, Analysis of discretization in the Direct Simulation Monte Carlo, Phys Fluids 12 (10) (2000) 2634–2638.

[35] U. Ghia, K. N. Ghia, C. T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J Comput Phys 48 (3) (1982) 387–411.

[36] S. Naris, D. Valougeorgis, The driven cavity flow over the whole range of the Knudsen number, Phys Fluids 17 (9) (2005) 1–12.

[37] B. John, X. J. Gu, D. R. Emerson, Investigation of heat and mass transfer in a lid-driven cavity under nonequilibrium flow conditions, Numer Heat Transf Part B 58 (5) (2010) 287–303.

[38] J. C. Huang, K. Xu, P. Yu, A unified gas-kinetic scheme for continuum and rarefied flows II: Multi-dimensional cases, Commun Comput Phys 12 (3) (2012) 662–690.

[39] 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.

[40] H. Babovsky, On a simulation scheme for the Boltzmann equation, Math Methods Appl Sci 8 (1986) 223–233.

[41] J. Li, Improved diffuse boundary condition for the DSBGK method to eliminate the unphysical density drift, arXiv.org physics.co (2014) 1403.3923.

[42] M. T. Ho, I. Graur, Heat transfer through rarefied gas confined between two concentric spheres, Int J Heat Mass Transf 90 (2015) 58–71.

[43] 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.

[44] K. Atkinson, Numerical integration on the sphere, J Aust Math Soc 23 (1982) 332–347.

[45] M. T. Ho, I. Graur, Numerical study of unsteady rarefied gas flow through an orifice, Vacuum 109 (2014) 253–265.

[46] L. Wu, J. M. Reese, Y. Zhang, Oscillatory rarefied gas flow inside rectangular cavities, J Fluid Mech 748 (2014) 350–367.

[47] K. D. Lathrop, Ray effects in discrete ordinates equations, Nucl Sci Eng 32 (3) (1968) 357–369.

[48] W. A. Fiveland, Discrete-ordinates solutions of the radiative transport equation for rectangular enclosures, J Heat Transfer 106 (4) (1984) 699.

[49] J. C. Chai, H. O. S. Lee, S. V. Patankar, Ray effect and false scattering in the discrete ordinates method, Numer Heat Transf Part B 24 (4) (1993) 373–389.

[50] P. J. Coelho, The role of ray effects and false scattering on the accuracy of the standard and modified discrete ordinates methods, J Quant Spectrosc Radiat Transf 73 (2-5) (2002) 231–238.