Journal of Computational Physics 228 (2009) 4532–4548

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Convergence behavior of a new DSMC algorithm

M.A. Gallis [a,*], J.R. Torczynski [a], D.J. Rader [a], G.A. Bird [b]

[a] Engineering Sciences Center, Sandia National Laboratories, P.O. Box 5800-0346, Albuquerque, NM 87185-0346, USA
[b] GAB Consulting Pty. Ltd., Sydney, NSW 2000, Australia

ARTICLE INFO

Article history:
Received 8 October 2008
Received in revised form 2 February 2009
Accepted 16 March 2009
Available online 25 March 2009

Keywords:
DSMC
Sophisticated DSMC
Algorithm
Convergence
Rarefied gas dynamics

ABSTRACT

The convergence rate of a new direct simulation Monte Carlo (DSMC) method, termed "sophisticated DSMC", is investigated for one-dimensional Fourier flow. An argon-like hard-sphere gas at 273.15 K and 266.644 Pa is confined between two parallel, fully accommodating walls 1 mm apart that have unequal temperatures. The simulations are performed using a one-dimensional implementation of the sophisticated DSMC algorithm. In harmony with previous work, the primary convergence metric studied is the ratio of the DSMC-calculated thermal conductivity to its corresponding infinite-approximation Chapman–Enskog theoretical value. As discretization errors are reduced, the sophisticated DSMC algorithm is shown to approach the theoretical values to high precision. The convergence behavior of sophisticated DSMC is compared to that of original DSMC. The convergence of the new algorithm in a three-dimensional implementation is also characterized. Implementations using transient adaptive sub-cells and virtual sub-cells are compared. The new algorithm is shown to significantly reduce the computational resources required for a DSMC simulation to achieve a particular level of accuracy, thus improving the efficiency of the method by a factor of 2.

1. Introduction

The direct simulation Monte Carlo (DSMC) method of Bird [1] is the most general and widely used method for simulating non-continuum gas dynamics. DSMC simulations have been shown to yield solutions to the Boltzmann equation in the limit of vanishing discretization error [2–4], and departures of DSMC simulations from such solutions have been shown to obey Green-Kubo theory for small but finite discretization errors [5–7]. As a result, DSMC is often used as the standard by which other methods for simulating non-continuum gas dynamics are assessed.

The generality and the accuracy of the DSMC method have allowed its application to areas outside the regime of hypersonic aerodynamics, such as material processing and micro- and nano-technology. These new areas have placed new demands on the method since the signal-to-noise ratio for subsonic flows is less favorable than for hypersonic flows. Consequently, a clear understanding of how to achieve a specified level of numerical accuracy with a minimum of computational effort is needed.

In response to this need, convergence studies of the DSMC method have been conducted [5–8]. These studies cover a wide range of flows, with the focus on one-dimensional Fourier and Couette flows, both steady and unsteady. The key convergence metric is the ratio of the DSMC-calculated bulk transport properties (thermal conductivity and viscosity) to their Chapman–Enskog (CE) theoretical values [3] although other functionals, such as heat flux and temperature have also been discussed [8].

* Corresponding author. Tel.: +1 505 844 7639; fax: +1 505 844 6620.
E-mail address: magalli@sandia.gov (M.A. Gallis).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.03.021

Four parameters are known to limit the numerical accuracy of the DSMC method: the number of independent samples per cell S_c, which is related to the statistical error of the method, and the number of simulators (computational molecules) per cell N_c, the time step Δt, and the cell size Δx, which are related to the discretization error of the method [6–8]. Bird observes that statistical fluctuations decrease with the inverse square root of the sample size and can be reduced (in principle) to any desired level by continuing the simulation or by repeating it with different initial random seeds [1]. Several authors [9,10] offer closed-form expressions that relate statistical error in DSMC simulations to the square root of the sample size.

Bird recently proposed a new variant of DSMC, termed "sophisticated DSMC" [11–13]. This new DSMC algorithm aims at improving the computational efficiency of DSMC without losing the accuracy of the original algorithm by reducing the discretization error of the algorithm for a particular selection of simulation parameters. To achieve this, significant modifications to the ways that simulators move and collide are introduced. More efficient grids and adaptive time steps that vary across the domain are used to gain computational efficiency during the move phase. In the collision phase, the new algorithm abandons the random selection of collision partners within a cell in favor of a nearest-neighbor selection scheme. All these modifications optimize critical simulation parameters at a relatively low cost, leading to a more efficient DSMC algorithm.

The new method retains many of the features of the original method and all of the physical models. However, the dependence on discretization parameters and therefore the convergence characteristics of the new and original algorithms differ, so a reevaluation of the convergence behavior is therefore necessary.

In this paper, the ability of the new DSMC algorithm to deliver improved computational efficiency is examined. In harmony with previous work [8], the benchmark case used for this purpose is one-dimensional heat transfer in a gas between two parallel walls at unequal temperatures. The main convergence metric used is the ratio of the DSMC-calculated thermal conductivity to its corresponding infinite-approximation CE theoretical value [3]. To ensure that the CE limit is achieved, DSMC simulations are performed at small system and local Knudsen numbers ($\sim$0.02). Under these conditions, the normal solution in the central region of the domain can be clearly differentiated from the Knudsen layers near the walls; it is within this central (near-continuum) region that the convergence behaviors of the functionals are investigated.

More than 700 simulations covering the regime from near-equilibrium to non-equilibrium conditions are performed. The results of the new DSMC method are compared with those of the original method for the same problem. From these results, the difference in the performance of the two algorithms is estimated. The present calculations employ sufficient samples to reduce statistical errors to levels that are negligible compared to the errors associated with the other three parameters. The remaining non-statistical error (hereafter referred to as the discretization error) is systematically investigated for the Fourier problem over wide ranges of the discretization parameters Δx, Δt, and N_c. Herein, DSMC07 and DSMC94 (i.e., as published in Bird's 1994 monograph [1]) are used to distinguish the new and original DSMC algorithms.

2. DSMC07: A new DSMC algorithm

The sophisticated DSMC07 algorithm retains the basic elements of the original DSMC94 algorithm described in Bird's monograph [1]. The key computational assumptions of DSMC, the uncoupling of molecular motion and collisions over a computational time step (usually a fraction of the mean collision time (MCT) or the mean transit time (MTT)) and the partitioning of the physical domain into cells (usually a fraction of the local mean free path), are maintained. The major modifications to the algorithm involve changing how simulators are selected for collisions and how collisions are distributed over the duration of a time step. Besides the global time that a DSMC algorithm keeps track of, simulator-based and cell-based times are calculated and kept track of as well. To achieve this, the DSMC07 global time is advanced in small global time steps that are typically a small fraction of the time step used by the DSMC94 algorithm. Unlike the DSMC94 algorithm, only a small fraction of the simulators move and collide at any global time step.

2.1. Collision partner selection procedures

In DSMC, the computational grid serves two purposes. The first one is the spatial discretization of sampled properties, such as the collision frequency and moments of the molecular velocity distribution (e.g., density, momentum, energy). The second one is to facilitate selection of collision partners that satisfy the basic requirement of geometrical proximity. In principle, computational cells are not necessary for the selection of collision partners. Simulators occupy positions in physical three-dimensional space, and, when collisions between them occur, each simulator could assess its neighboring simulators to identify possible collision partners. Computationally, such a scheme would be feasible only for one-dimensional flows where simulators could be easily sorted using their x-distance. In two- and three-dimensional flows, sorting simulators according to their distances is more complicated, so cells are used to limit the number of possible collision partners while at the same time ensuring geometrical proximity.

This basic principle is what DSMC07 aims to exploit by replacing the random selection of collision partners with a deterministic nearest-neighbor selection scheme, resulting in a smaller mean collision separation (MCS) between colliding simulators [11,12]. The nearest-neighbor selection is limited to the simulators within each cell.

The idea of performing an $O(N^2)$ operation to sort all N simulators in a cell was initially demonstrated by LeBeau et al. [14] and is termed the virtual sub-cell (VSC) scheme. The VSC scheme actually provides an efficient way of performing collisions

that minimize the MCS. There is a subtlety associated with nearest-neighbor selection. If multiple collisions take place in a cell and a simulator from a preceding collision is selected again as the first simulator, the other simulator from the preceding collision will still be the nearest simulator and will thus be selected again since no move has taken place. This would allow a highly unlikely collision because, after a collision, simulators move in opposite directions and a second collision between them should not be allowed. It is necessary therefore to keep a record of the last collision partner and, if the last collision partner is chosen again, to exclude this simulator and to choose the next-nearest simulator instead.

Since the VSC scheme is an $O(N^2)$ operation to locate the nearest neighbor of a simulator, it becomes computationally expensive when the number of simulators per cell exceeds 30–40. For such cases, Bird [12] suggests using the transient adaptive sub-cell (TASC) scheme instead. According to this scheme, computational cells are subdivided into sub-cells that exist only during the collision phase of each cell, and collision partners are selected from the same or neighboring sub-cells (instead of from anywhere in the whole cell), which reduces the distance between collision partners to a fraction of a cell width. Having fine enough sub-cells so that a maximum of one simulator is found within a sub-cell and adopting a layered search algorithm in neighboring sub-cells lead to neighboring simulators preferentially being selected for collisions. In this way, the MCS is reduced to its absolute minimum for the number of simulators used.

The VSC and TASC schemes are the two alternatives of DSMC07 collision partner selection. Since it is deterministic in selecting nearest-neighbor simulators, VSC is more accurate but more expensive computationally than TASC. Because it selects nearest-neighbor simulators in a probabilistic manner, TASC is less accurate but computationally more efficient than VSC.

The computational advantage resulting from the selection of simulators based on their spatial proximity, unlike the random selection from within a cell, depends on the dimensionality of the problem [13,14]. The average distance between points randomly selected from within a unit-size one-dimensional cell is smaller than the distance between randomly selected points from a three-dimensional unit-size cube. Therefore, the nearest-neighbor scheme (VSC or TASC) is expected to result in a smaller MCS between colliding simulators in a one-dimensional implementation than in a three-dimensional implementation. To facilitate the study of the performance of the nearest-neighbor scheme in three-dimensional space, the one-dimensional DSMC code used here is modified to use a one-dimensional linear array of virtual cubical cells in the x direction. Periodic boundary conditions are applied on the cell boundaries in the y and z directions. Whenever a simulator crosses one of these boundaries, it is reintroduced from the opposite side as a new simulator that has lost its information about its last collision partner and therefore is not excluded from further collisions. With this modification, simulators can move apart in a physically realistic manner in all three directions, facilitating the study of the three-dimensional behavior of the sophisticated algorithm.

2.2. Temporal advection

In DSMC94, a single time parameter and a single time step are used throughout the domain so that the entire flow field is advancing concurrently in time. This scheme can become computationally inefficient when large variations of density exist in the computational domain. In DSMC07, Bird proposed using multiple time parameters with variable time steps [11,12]. Instead of a single time parameter and a single time step throughout the domain, every collision cell and every simulator has its own time parameter, every collision cell has its own time step, and each simulator within a particular cell inherits the time step of that cell.

The time step for a collision cell, termed the desired time step (DTS), is set to the minimum of user-specified fractions of the MCT and the MTT for that cell. These fractions typically lie in the range of 0.1–1, with a value of 0.2 or smaller considered to be highly refined. To avoid the effect of statistical fluctuations producing extremely small or large DTS values, the DTS value of a cell is constrained to lie between minimum and maximum values that are functions of the average DTS in the domain. Typically, the minimum and maximum values are taken to be 0.1 and 10 times the average DTS, respectively.

The global time parameter is advanced in global time steps that are small compared to cell-based time steps. The global time step is set equal to a user-specified fraction of the average value of DTS in the domain, where this fraction is typically taken to be 0.2 or smaller. Practically, this results in a global time step that is about 5–10 times smaller than the cell-based time step. Because of the smallness of the global time step, only a relatively few simulators move and only a relatively few cells perform collisions during each global time step.

The cell and simulator time parameters are advanced based on the global time parameter and the DTS values within the cells. If the time parameter of a collision cell is within DTS of the global time parameter, the cell time parameter is not advanced and collisions are not performed during that global time step. Similarly, if the time parameter of a simulator is within DTS of the global time parameter, the simulator time parameter is not advanced and the simulator does not move during that global time step. When the time parameter of a collision cell falls behind the global time parameter by DTS or more, the cell's time parameter is increased by $2 \times$ DTS, and collisions are performed in that cell over that amount of time. Similarly, when the time parameter of a simulator falls behind the global time parameter by DTS or more, the simulator's time parameter is increased by $2 \times$ DTS, and the simulator is moved a distance corresponding to that amount of time. In this way, the time parameters of each cell and of each simulator remain within $\sim$DTS of the global time parameter for all global time steps.

As indicated above, the user selects the fractions of the MCT and MTT values used to determine the DTS for a cell and the fraction of the average DTS throughout the domain used to determine the global time step. Just as the selection of a smaller time step in DSMC94 produces a more accurate but more expensive simulation, the selection of smaller fractions in DSMC07

also produces a more accurate but more expensive simulation. In both situations, the user must make this selection for the simulation under consideration based on the available computational resources and the required accuracy.

2.3. Separate sampling and collision cells

Other changes involve using separate collision and sampling cells. The sampling cells usually contain approximately 4–5 collision cells. The sole use of the sampling cells is to collect molecular velocity moments and output macroscopic properties. The collision frequency and the collision partners are selected based on the population of a collision cell.

3. Spatial and temporal discretization in DSMC07

As pointed out in Section 2, computational cells in DSMC aim at minimizing the distance between collision partners. Large collision separations within the same cell would lead to physically unrealistic collisions and to a reduction of the angular momentum of the colliding pairs. This observation has been repeatedly confirmed by detailed convergence analyses [8] and by comparisons to Green-Kubo theory [6,7]. In his recent work [12], Bird recommends the use of the ratio of the MCS to the mean free path as the single most important "goodness" parameter for DSMC simulations and proposes a number of ways to optimize calculations based on this parameter. The sophisticated DSMC procedures, outlined in the previous section, aim at optimizing this parameter at the lowest possible cost.

3.1. Spatial discretization

Since in DSMC94 collision partners are selected randomly from within a cell, the MCS is a product of the cell size and a function of the dimensionality. It has been theoretically calculated that the mean separation of two points in a unit-size n-dimensional hypercube (the Robbins constant) is 0.333333, 0.521405, and 0.666667 for one, two, and three dimensions, respectively [15]. The nearest-neighbor scheme that DSMC07 uses for the selection of collision partners (VSC or TASC) leads to a reduction of the MCS, where this reduced value is a function of the number of simulators in the cell and the dimensionality of the application. LeBeau et al. [14] demonstrate that the VSC scheme reduces the MCS by factors of 8-563 for 7-120 simulators per cell, respectively. In terms of a DSMC simulation, this means that the grid of a one-dimensional DSMC simulation using the VSC scheme can be at least 8 times coarser to achieve the same spatial discretization. In three dimensions, this effect is not as pronounced, with a factor of 2–6 per coordinate direction, but it is still very important because the three-dimensional cell volume can be increased by a factor of 2^3–6^3 (i.e., a factor of 8-216) while maintaining the same MCS as DSMC94.

This benefit is offset somewhat by the higher computational cost of the VSC scheme compared to the random-selection scheme. The computational cost of the random-selection scheme scales with the number of simulators per cell as N_c in comparison with N_c^2 for the computational cost of the VSC scheme.

As the number of simulators per cell increases, the VSC scheme becomes very expensive and reaches a point beyond which it does not offer any computational advantage although it is always more accurate. The TASC scheme of Bird provides a more efficient nearest-neighbor search algorithm than the VSC scheme for large number of simulators per cell. Virtual subcells provide a deterministic detection of the nearest neighbor while transient adaptive sub-cells a probabilistic one. As such the former can be expected to be more accurate. The effect of using the TASC scheme will be examined separately once the effect and the efficiency of the VSC scheme have been established.

3.2. Temporal discretization

The effect of the variable time step is not as easy to quantify theoretically since it depends on the properties of the flow field that the method is applied to. It depends on the variation of the MCT in the flow field and on the relative extents of regions with different density. Linking the simulator time step to the local MCT introduces a computational penalty in that the algorithm cycles through all simulators and cells even though only a fraction of them will be selected for moves. This scheme introduces a computational penalty proportional to the ratio of the maximum to minimum time steps because the arrays of simulators need to be regenerated at the end of every move before collisions can begin. The ratio of the average to minimum time steps in a calculation can arbitrarily be set, but a value less than 10 can be considered adequate for most cases.

It should be pointed out that in DSMC94 the error due to temporal discretization arises from the fact that collisions are performed at discrete times between the move operations although collisions actually occur continually throughout time. DSMC07 mitigates this problem somewhat because collisions in a cell are distributed more finely over the duration of a time step, which allows for a more physically realistic simulation.

4. Fourier flow

In Fourier flow, shown in Fig. 1, the gas is motionless and confined between two infinite, parallel walls separated by a distance L at unequal temperatures ($T_1 \neq T_2$). In steady state, a uniform heat flux and a temperature gradient exist in the

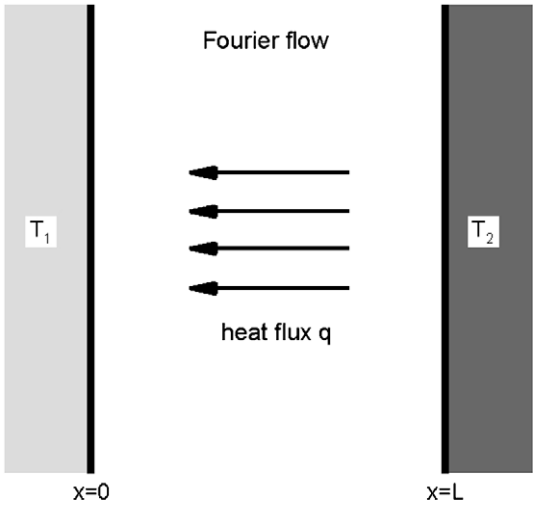

Fig. 1. Schematic diagram of Fourier flow.

domain. When both the heat-flux and the system Knudsen number are small, the heat flux is proportional to the temperature gradient in the bulk gas (i.e., several mean free paths away from the walls) according to Fourier's law, where the coefficient of proportionality is the thermal conductivity:

$$q = -K\frac{\partial T}{\partial x}. \tag{1}$$

In the limit of small temperature gradients, the CE expansion can be analytically calculated to provide an accurate prediction for the transport properties in a monatomic gas. In this case, the thermal conductivity K of the gas is of particular interest since it can be directly calculated from the DSMC code. In harmony with previous work [8], this benchmark case is used here to study the convergence of the new DSMC algorithm.

5. Convergence of DSMC94

The convergence of the original DSMC94 algorithm has been extensively studied. Green-Kubo [5–7] theory has been applied to derive expressions for the spatial and temporal discretization errors, each in the limit that the other discretization parameter vanishes, which are in excellent agreement with DSMC94 simulation results [8]. The cell size and the time step are linked to the spatial and temporal discretization errors, respectively. The derived expressions indicate second-order convergence rates as a function of the time step and the cell size and a first-order convergence rate as a function of the inverse number of simulators per cell, $1/N_c$. The apparent convergence rate of one discretization parameter is influenced by the values of the other parameters, and the theoretical convergence rate can be achieved only in the limit that the other parameters vanish.

Rader et al. [8] present a comprehensive study of the convergence behavior of the DSMC94 algorithm. The test case is pure Fourier flow with a gas that obeys the hard-sphere (HS) molecular interaction and has the molecular mass and the reference viscosity of argon. The variable soft sphere (VSS) model is used to represent this molecular interaction with $\omega = 1/2$ and $\alpha = 1$. Initially, the gas is motionless and at the reference pressure and temperature: $p_{\text{init}} = p_{\text{ref}} = 266.644$ Pa (2 Torr) and $T_{\text{init}} = T_{\text{ref}} = 273.15$ K. The domain has a length $L = 1$ mm and is bounded by two parallel solid walls that reflect all simulators diffusely (unity accommodation) at the wall temperature 223.15 K and 323.15 K.

The calculations are performed for a large number of combinations of the three discretization parameters: cell size, time step, and number of simulators per cell. For the Fourier problem, Rader et al. [8] presented a correlation for the ratio of the DSMC thermal-conductivity to the theoretical value:

$$\frac{\overline{K}_{\text{wall}}}{K} = 1.0001 + 0.0287\Delta\tilde{t}^2 + 0.0405\Delta\tilde{x}^2 - \frac{0.083}{N_c} + F(\Delta\tilde{t}, \Delta\tilde{x}, 1/N_c), \tag{2}$$

where $\Delta\tilde{x} = \Delta x/\lambda_{o,\text{HS}}$ is a dimensionless cell size, $\lambda_{o,\text{HS}} = \left(\sqrt{2}\pi d_{\text{ref}}^2 n_o\right)^{-1}$ is the hard-sphere molecular mean free path, d_{ref} is the molecular diameter, and n_o is the number density, $\Delta\tilde{t} = \Delta t/t_o$ is a dimensionless time step, $t_o = \lambda_o/c_o$ is the MCT, $c_o = \sqrt{2k_B T_o/m}$ is the most probable molecular speed at temperature T_o for a molecule of mass m, and k_B is the Boltzmann constant. The quantity N_c for the Fourier problem varies across the domain as a result of the temperature gradient. In this work and in agreement with the work of Rader et al. [8], the midpoint value is used as a reference point to be consistent with

the midpoint values of λ_o and t_o. For this problem, the midpoint value is $N_c = 0.9885N_{co}$, where N_{co} is the initial uniform value. From Eq. (2), the infinite-number-of-simulators limit ($N_c \to \infty$) of the DSMC94 correlation is given below [8]:

$$\lim_{N_c\to\infty} \frac{\overline{K}_{\mathrm{DSMC94}}}{K} = 1.0001 + 0.0287\Delta\tilde{t}^2 + 0.0405\Delta\tilde{x}^2 - 0.0009\Delta\tilde{x}^4 - 0.0016\Delta\tilde{t}^2\Delta\tilde{x}^2 + 0.0081\Delta\tilde{t}^4\Delta\tilde{x}^2. \quad (3)$$

The thermal conductivity in DSMC is calculated based on the wall heat fluxes in harmony with previous work [8]. This ratio, which approaches unity in the limit of vanishing discretization, is used as the convergence metric. This metric is not unique, and others could have been used instead. In fact, if a different metric for the heat flux is used (e.g., the cell-based heat flux), different convergence behavior would be observed.

6. Convergence of DSMC07

To evaluate the effect of the changes to the algorithm, the analysis of Rader et al. [8] is repeated with the new algorithm. By using the same test case (pure Fourier flow with a hard-sphere argon-like gas), not only can the convergence behavior of the new algorithm be derived, but also a direct comparison between DSMC07 and DSMC94 can be performed.

The sources of error in the new algorithm are expected to be the same as in the original one (spatial, temporal, and velocity-space discretization). However, if the efficiency of the new algorithm is different, the relative importance of the terms should also be different.

More than 700 simulations are performed where the basic simulation parameters, number of simulators per cell, time step, and cell size, are systematically varied. The simulations used a normalized parameter space bounded by $0.01 \leqslant 2\Delta\tilde{t} \leqslant 1$ for the time step, $0.105 \leqslant \Delta\tilde{x} \leqslant 2$ for the cell size, and $5 \leqslant N_c \leqslant 120$ for the number of simulators. The time step in DSMC07 is usually based on both the MCT and the MTT. For the purpose of this study and to allow for a fair comparison with DSMC94, the time step here is based only on the MCT. The thermal-conductivity ratio is calculated as a spatial average over the central part of the domain in a similar fashion to the work of Rader et al. [8]. Each calculation involves ensemble averaging of 100 realizations of the same problem initialized with different seeds for the random number generator.

To compare with the results of the one-dimensional DSMC94 algorithm [8], the nearest-neighbor selection (whether VSC or TASC) is based on their x distance only. A similar set of calculations, in which the full three-dimensional distance is used to better represent three-dimensional performance, is also presented.

6.1. Infinite-number-of-simulators limit

The application of Green-Kubo theory to DSMC characterizes spatial discretization in terms of the cell size, which is physically related to the MCS. As outlined above, in DSMC94, the MCS is independent of the number of simulators per cell; however, in DSMC07 the spatial discretization is strongly correlated to the number of simulators per cell. Thus, DSMC07, unlike DSMC94, is expected to exhibit a relationship between the spatial discretization and the number of simulators per cell. To isolate the effect of the cell size from the effect of the number of simulator per cell, the results are extrapolated to an infinite number of simulators per cell. The resulting DSMC07 correlation is given below:

$$\lim_{N_c\to\infty} \frac{\overline{K}_{\mathrm{DSMC07}}}{K} = 0.996503 + 0.0786099\Delta\tilde{t} + 0.028232\Delta\tilde{t}^2 + 0.00388663\Delta\tilde{x}^2 - 0.00193386\Delta\tilde{t}^2\Delta\tilde{x}^2 \quad (4)$$

The thermal-conductivity ratio is presented as a function of the cell size and the time step in Figs. 2 and 3. The solid symbols are the DSMC07 results, whereas the solid curves are the correlation given by Eq. (4).

A comparison between Eqs. (3) and (4) offers several insights into the comparative convergence behavior of the thermal-conductivity ratio with the two methods. First, the dependence of DSMC07 on cell size is much weaker than that of DSMC94. This behavior is expected because of the correlation between the number of simulators per cell and the MCS in DSMC07 and manifests itself in Fig. 2 as an almost flat DSMC07 convergence function. This weak spatial discretization error in DSMC07 may be due in part to the fact that collisions are restricted to same-cell simulators: nearest-neighbor simulators in adjacent cells do not collide. From Fig. 3, the error in DSMC07 predictions as $N_c \to \infty$ is seen to tend to zero, unlike DSMC94 predictions and regardless of the cell size as long as the cell size is smaller than one mean free path. Second, Eq. (4) has a linear term in $\Delta\tilde{t}$ that is absent from Eq. (3). The nearly linear relationship between the time step and the error of the thermal-conductivity ratio is observed in Fig. 3. This linear term indicates that DSMC07 has a much stronger dependence on time step, which is attributed to the nearest-neighbor procedures. It is worth noting that the $\Delta\tilde{t}^2$ terms in Eqs. (3) and (4) are similar although no particular meaning is attributed to this. Third, fewer higher-order terms are needed to correlate the results of DSMC07. No particular physical significance is attached to these terms since there are no theoretical predictions for them. It is noted in passing, however, that, since some of these terms have negative signs for both DSMC07 and DSMC94, they actually can spuriously reduce the observed error for coarse discretizations. Thus, the fact that DSMC07 does not generate as many negative terms is considered an improvement.

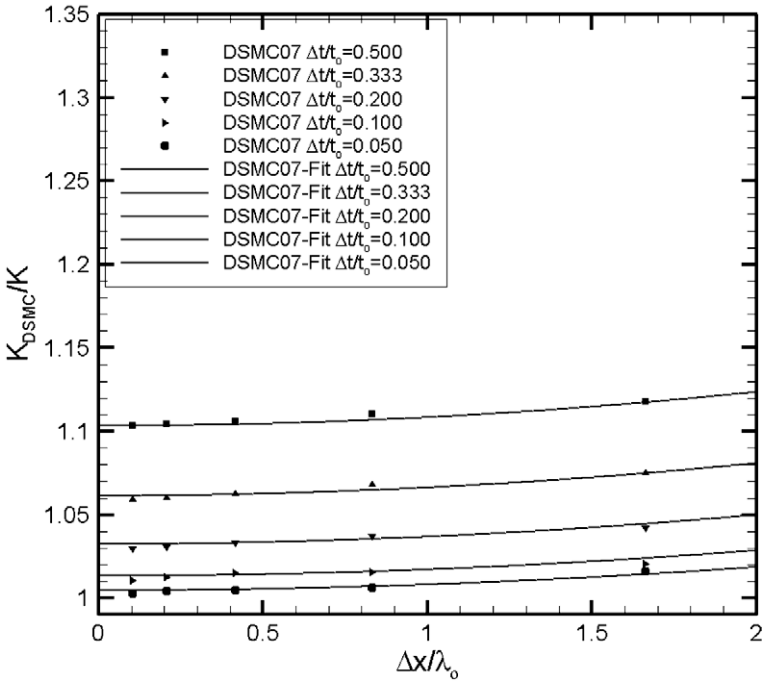

Fig. 2. Thermal-conductivity ratio for $N_c \to \infty$ as a function of $\Delta\tilde{x}$.

6.2. Comprehensive analysis of DSMC07 convergence behavior

Figs. 4 and 5 present the results for 15 and 30 simulators per cell. In these figures, the dashed curves represent the convergence behavior of DSMC94, as given by Eq. (4), the solid symbols represent DSMC07 calculations for the same cases, and the solid curves are the least-squares fit to the DSMC07 results given below:

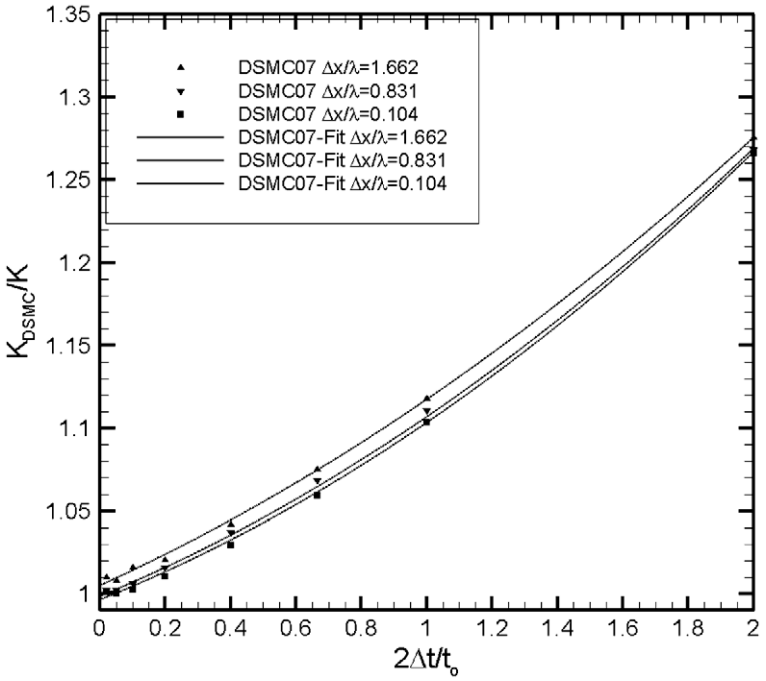

Fig. 3. Thermal-conductivity ratio for $N_c \to \infty$ as a function of $\Delta\tilde{t}$.

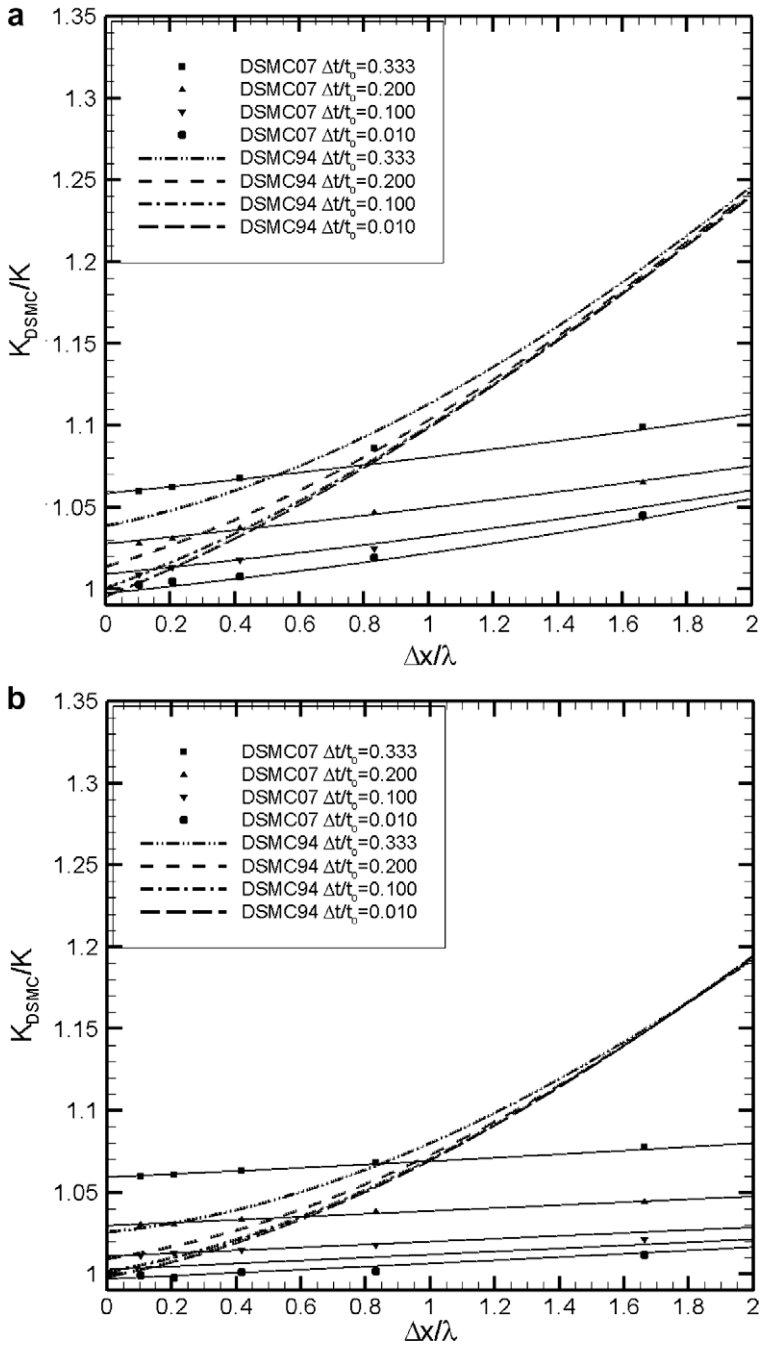

Fig. 4. (a) Thermal-conductivity ratio for $N_c = 15$ as a function of $\Delta\tilde{x}$ and (b) thermal-conductivity ratio for $N_c = 30$ as a function of $\Delta\tilde{x}$.

$$\frac{\overline{K}_{\mathrm{DSMC07}}}{K} = 0.995084 + 0.0628887\Delta\tilde{t} + 0.0477988\Delta\tilde{t}^2 + 0.00267194\Delta\tilde{x}^2 - 0.00335372\Delta\tilde{t}^4 + 0.00193386\Delta\tilde{t}^2\Delta\tilde{x}^2 - 0.106218/N_c^2 + 0.092157/N_c + F(\Delta\tilde{x}, \Delta\tilde{t}, 1/N_c) \quad (5)$$

Fig. 4(a) and (b) present the convergence behavior of the thermal-conductivity ratio as a function of cell size, whereas Fig. 5(a) and (b) present the convergence behavior as a function of time step, for 15 and 30 simulators per cell, respectively. In harmony with the observations for an infinite number of simulators per cell, the convergence rate of the thermal-conductivity ratio is less sensitive to the spatial discretization for DSMC07 than for DSMC94. The opposite is true for convergence behavior as a function of the temporal discretization because of the strong influence of the linear term in Eq. (5).

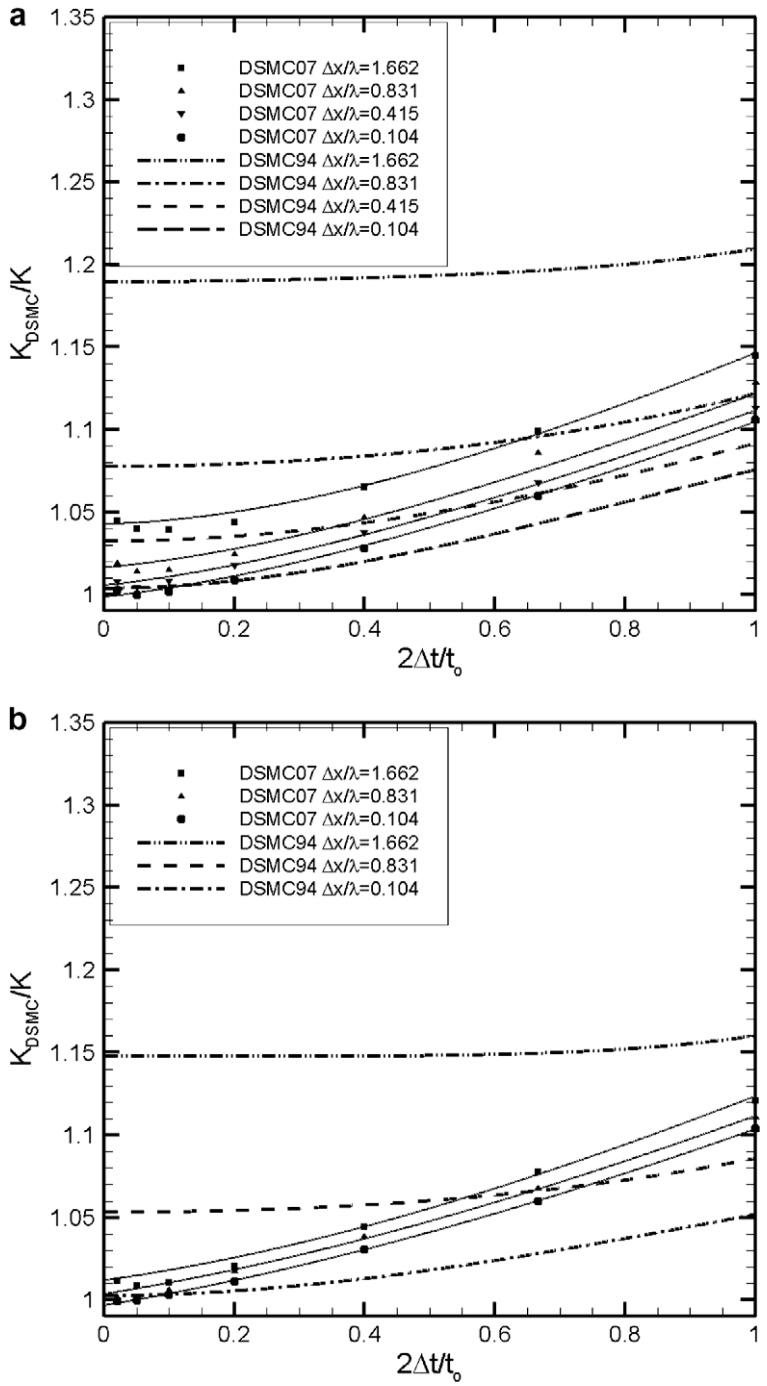

Fig. 5. (a) Thermal-conductivity ratio for $N_c = 15$ as a function of $\Delta\tilde{t}$ and (b) thermal-conductivity ratio for $N_c = 30$ as a function of $\Delta\tilde{t}$.

Comparison of these figures indicates that, to the limit of vanishing discretization error, both methods converge to the correct answer of unity to within fitting uncertainty. For coarser discretizations, DSMC07 produces smaller errors in general. The cell size clearly plays a much smaller role in determining the error in DSMC07 simulations. In fact, the DSMC07 error is always smaller when the time step is smaller than the MCT and the MTT. It appears that using time steps greater than the MCT or allowing simulators to bypass cells without examining them for potential collision partners violates the principle of the nearest-neighbor scheme. Assuming that the motion of a simulator is contained within a cell, by allowing simulators to travel further than one MCS, the nearest-neighbor scheme biases collisions to simulators in the vicinity of the final position, while ignoring all simulators between the initial and final positions. The DSMC94 algorithm, by disregarding the location of

the simulators in the cell, allows all simulators to be considered. Comparison of Fig. 4(a) and (b) shows that the simulations with 30 simulators per cell that have a finer spatial discretization require smaller time steps also. If this condition is not satisfied, then the error of DSMC07 can be greater than that of DSMC94.

To further demonstrate this point, Fig. 6 compares the results of Fig. 5(a) to additional simulations for which the time step is constrained to be no larger than the minimum of 1/4 of the MTT and a fixed fraction of the MCT, as proposed by Bird [11,12]. In this case, the DSMC07 error is always smaller than the DSMC94 error. Therefore, as pointed out by Bird, linking the time step to the local MCT and MTT is critical in reducing the discretization error of DSMC07.

A better representation of the improved performance of the new method is given in Fig. 7, which presents the convergence behavior as a function of the cell size for multiple values of the number of simulators per cell. For a particular number of simulators per cell and for small enough time steps, the DSMC07 error is smaller than the DSMC94 error. The effective time step of the calculations presented in Fig. 7 (7 ns) is small enough to prevent simulators from traveling more than one cell. Thus, the DSMC07 error is always smaller than that of DSMC94.

The above comparisons point out the stronger role that temporal discretization plays in DSMC07 than in DSMC94. This can be explained by a violation of the assumptions behind the nearest-neighbor selection scheme. The adoption of the nearest-neighbor method recognizes the importance of not overlooking neighboring simulators when collision partners are selected. A time step that is larger than the MCT would result in simulators traveling over potential collision partners without ever having the chance to be selected for collision. This behavior is described by the linear convergence rate of the thermal-conductivity ratio with time step.

For small numbers of simulators per cell, additional error sources become more important due to the $1/N_c$ term, making the convergence behavior more complicated. It is noted that DSMC07 has a positive $1/N_c$ pure term that increases the error for small number of simulators, unlike DSMC94, for which the $1/N_c$ term has a negative sign.

For coarse spatial discretizations ($\Delta x > \lambda$), very small time steps ($\Delta t < 0.025$MCT), and small numbers of simulators per cell ($N_c \leqslant 30$), the DSMC07 error is seen to increase slightly (less than 1%) with decreasing time step. Clearly observed in both Figs. 5 and 6 (and in some later figures), this counterintuitive behavior is much larger than the statistical uncertainty of the simulations and vanishes for finer spatial discretizations and large numbers of simulators per cell. It is conjectured that this effect is caused by an inaccurate value of the local collision frequency when using relatively large cells and relatively few simulators per cell. This inaccuracy becomes progressively larger as the cell size is increased. The value of the local collision frequency appears to be important when the trajectory of a simulator is very finely discretized (i.e., small time steps) and collision partners are selected on a nearest-neighbor basis (i.e., as in DSMC07). However, further investigation is needed to assess this conjecture.

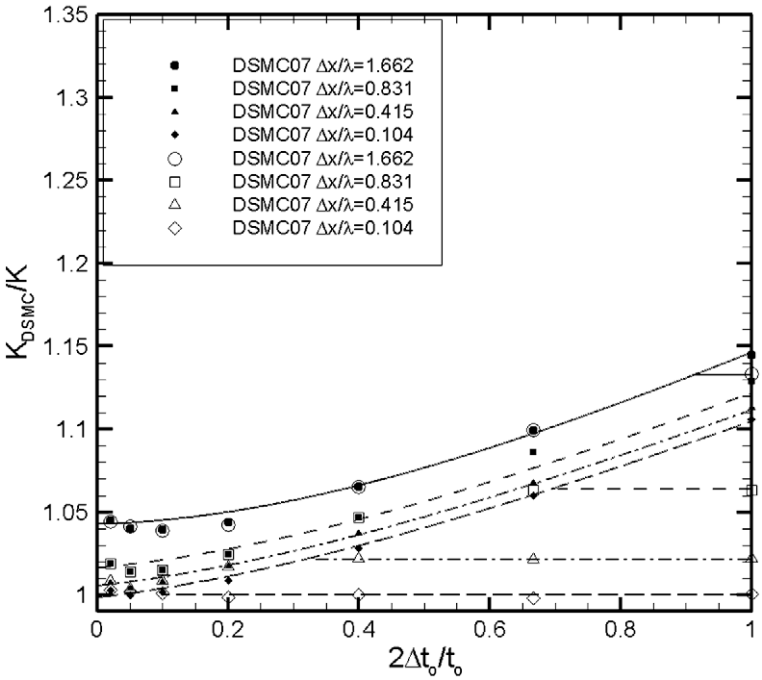

Fig. 6. DSMC07 thermal-conductivity ratio for $N_c = 15$ as a function of $\Delta\tilde{t} \leqslant 0.25$ MTT. Closed symbols: calculation where time step is a fraction of MCT; open symbols: time step is a function of both MCT and MTT.

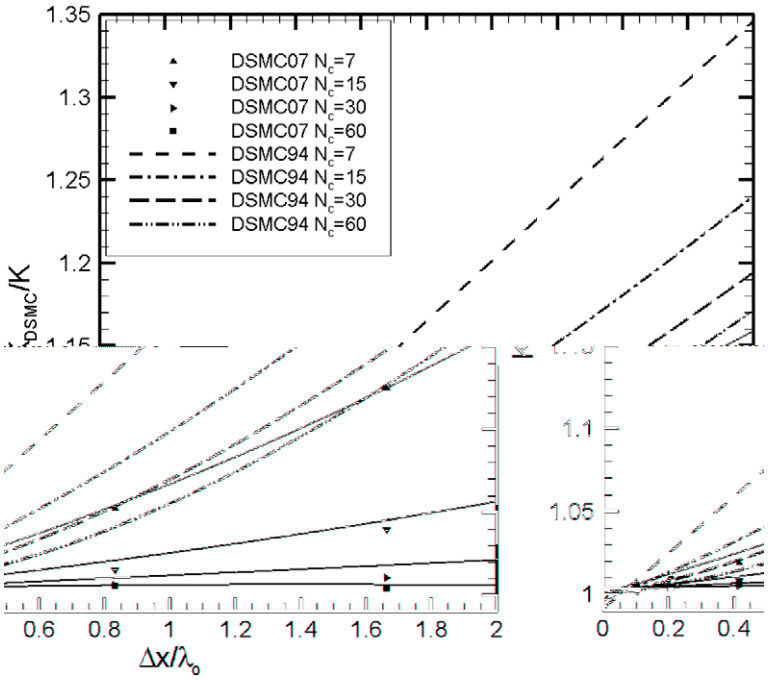

Fig. 7. Thermal-conductivity ratio for $\Delta\tilde{t} \leqslant 0.10$ MCT as a function of $\Delta\tilde{x}$ and N_c.

6.3. Three-dimensional behavior

If the three-dimensional distance between simulators is used instead of their x-distance, the performance of DSMC07 is expected to be affected since the average distance between two points in a three-dimensional unit-size cube is larger than along a one-dimensional unit-length line segment. To determine the impact on performance, the parametric analysis of the previous section is repeated using the three-dimensional distance between simulators. In a similar fashion to Fig. 5(a) and (b), Fig. 8(a) and (b) present the convergence behavior of the thermal-conductivity ratio as a function of cell size for 15 and 30 simulators per cell, respectively, and Fig. 9(a) and (b) present the convergence behavior as a function of time step for 15 and 30 simulators per cell, respectively. The convergence of the one-dimensional DSMC94 algorithm, namely Eq. (4), is presented for comparison to the least-squares fit to the 3D DSMC07 values given below:

$$\frac{\overline{K}_{\mathrm{DSMC07,3D}}}{K} = 0.998988 + 0.0494676\Delta\tilde{t} + 0.0791001\Delta\tilde{t}^2 + 0.00546296\Delta\tilde{x}^2 - 0.0268787\Delta\tilde{t}^4 + 0.0217846\Delta\tilde{t}^2\Delta\tilde{x}^2 + 1.0796/N_c^2 + 0.0728841/N_c + F(\Delta\tilde{x}, \Delta\tilde{t}, 1/N_c) \quad (6)$$

It is evident from the comparison of these figures that, as previously demonstrated [13,14], the three-dimensional implementation of the DSMC07 algorithm is not as computationally advantageous as the one-dimensional one although the improvement compared to one-dimensional DSMC94 is still considerable. This decrease in the efficiency of DSMC07 is attributed to the fact that the MCS in three-dimensional space is larger than the MCS in one-dimensional space. Thus, the three-dimensional implementation has a larger spatial discretization error than the one-dimensional one. This effect is quantitatively given by the multiplier of the $\Delta\tilde{x}^2$ error term in Eq. (6), which is about twice that of Eq. (5).

Comparing the coefficients of N_c and $1/N_c^2$ in Eqs. (5) and (6), the dependence of the error on the number of simulators per cell is also seen to be larger for the three-dimensional implementation than for the one-dimensional implementation. This is indicated by the larger value (by one order of magnitude) of the $1/N_c^2$ coefficient in Eq. (6) relative to its value in Eq. (5). The $1/N_c$ coefficient, which is negative for the one-dimensional implementation, is positive with the same order of magnitude for the three-dimensional implementation. The sizes and signs of these coefficients in the three-dimensional DSMC07 algorithm reflect the fact that N_c is strongly related to the discretization of both physical and velocity space, both of which are more demanding for three-dimensional simulations than for one-dimensional ones.

Besides these differences, all of the other arguments made previously about the one-dimensional DSMC07 implementation are still valid for the three-dimensional DSMC07 implementation. For example, when the time step is smaller than the MCT and the MTT, the error of DSMC07 is smaller than that of DSMC94. However, both methods approach the correct value in the limit of vanishing discretization.

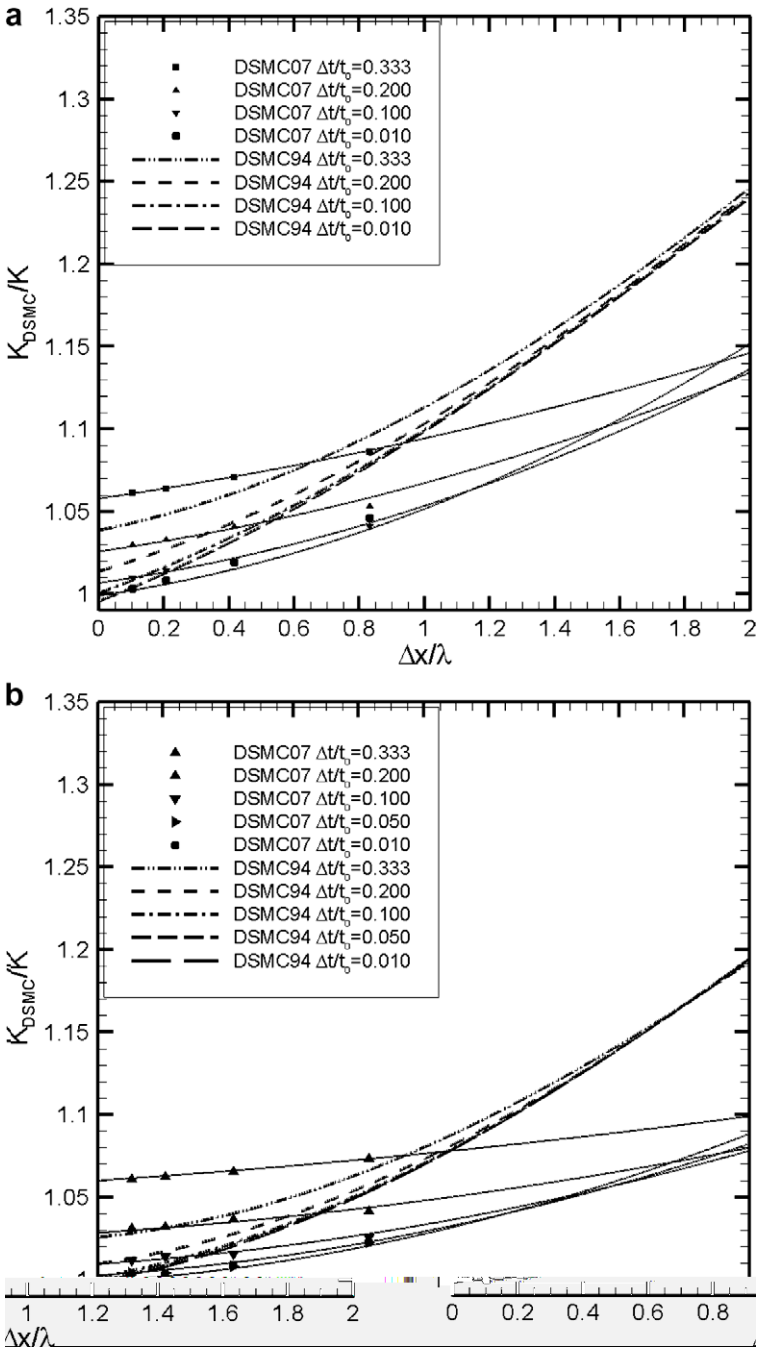

Fig. 8. (a) Thermal-conductivity ratio for $N_c = 15$ as a function of $\Delta\tilde{x}$: 3D version and (b) thermal-conductivity ratio for $N_c = 30$ as a function of $\Delta\tilde{x}$: 3D version.

Fig. 10 presents the convergence rate of the thermal-conductivity ratio for different values of the number of simulators per cell for both three-dimensional DSMC07 and one-dimensional DSMC94. When the one-dimensionality of the flow is not taken into account, the error of DSMC07 increases by about 30%. However, even in this case, DSMC07 still can offer significant error reduction over DSMC94.

6.4. DSMC07 with transient adaptive sub-cells

When the number of simulators per cell becomes large, the VSC scheme becomes prohibitively expensive to use. In this situation, Bird proposes that VSC be replaced by TASC in the collision partner selection algorithm. To examine the effects of

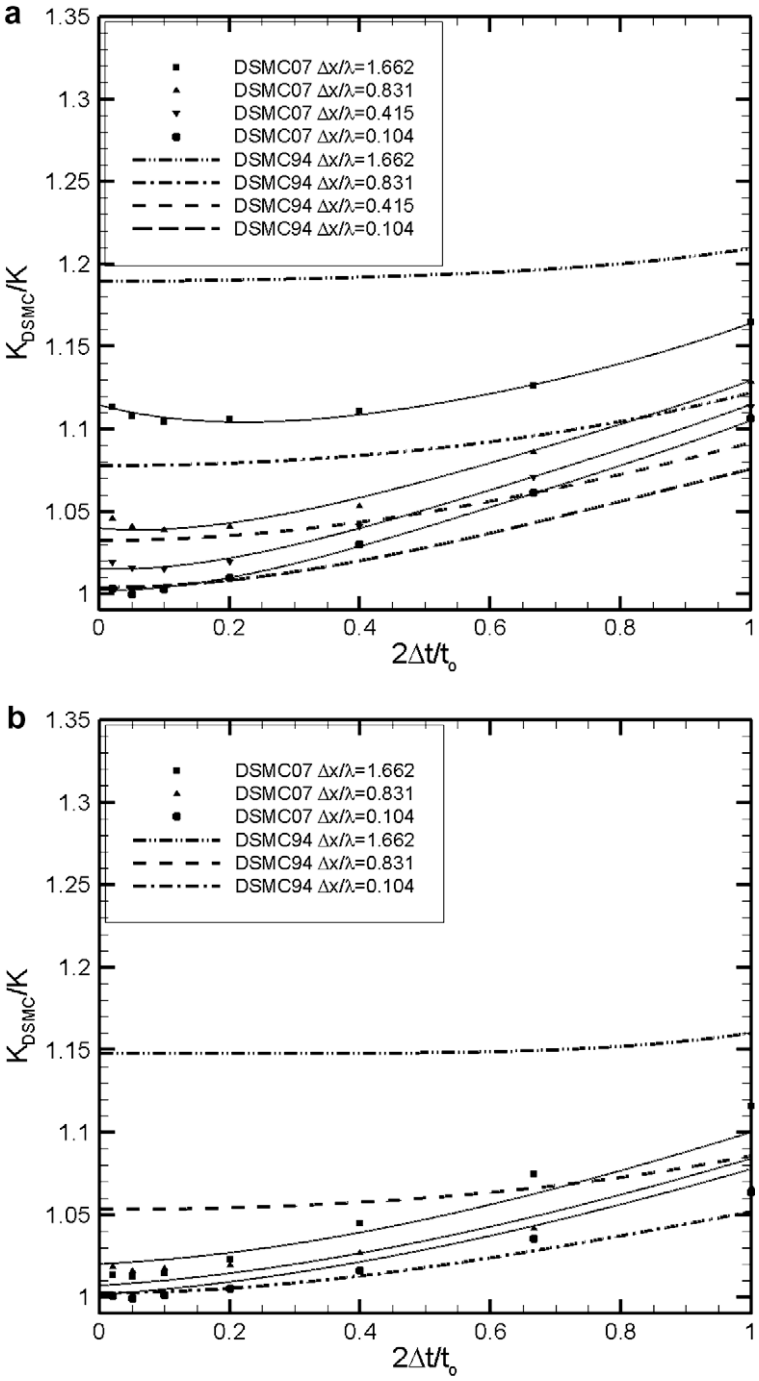

Fig. 9. (a) Thermal-conductivity ratio for $N_c = 15$ as a function of $\Delta\check{t}$: 3D version and (b) thermal-conductivity ratio for $N_c = 30$ as a function of $\Delta\check{t}$: 3D version.

this scheme on the convergence of the new algorithm, the simulations presented in Section 2 for 30, 60, and 120 simulators per cell are repeated with a total number of sub-cells that equals the number of simulators in the domain, so that, on average, one simulator per sub-cell is achieved. This scheme closely mimics the behavior of the TASC scheme. For the Fourier problem examined here, the density gradient is small enough to make the recreation of the sub-cell structure at every time step unnecessary.

Fig. 11(a) and (b) show the thermal-conductivity ratio as a function of the spatial and temporal discretization, respectively, for 30 simulators per cell. In harmony with Figs. 4(b) and 5(b), the results are compared to the one-dimensional

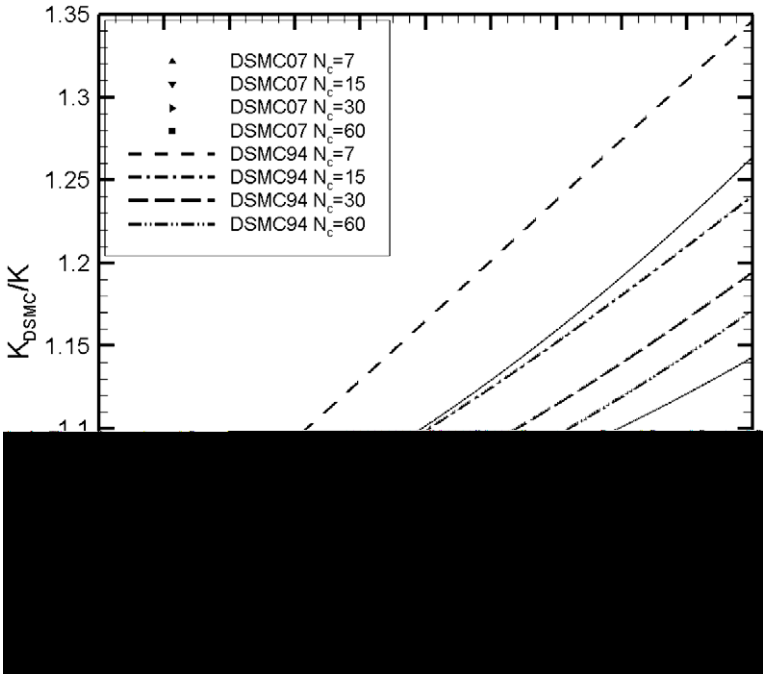

Fig. 10. Thermal-conductivity ratio for $\Delta\tilde{t} \leqslant 0.10$ MCT as a function of $\Delta\tilde{x}$ and N_c: 3D version.

DSMC94 algorithm. A comparison of Figs. 4(b) and 5(b) with Fig. 11(a) and (b) indicates that the temporal discretization error of TASC is lower than that of VSC while the opposite is true for the spatial error. To quantify this, fitting the results with the same parametric equation as before results in the following equation:

$$\frac{\overline{K}_{\mathrm{DSMC07,TASC}}}{K} = 0.994777 + 0.0318989\Delta\tilde{t} + 0.0467561\Delta\tilde{t}^2 + 0.00478986\Delta\tilde{x}^2 - 0.0066118\Delta\tilde{t}^4 + 0.0053837\Delta\tilde{t}^2\Delta\tilde{x}^2 - 0.803903/N_c^2 + 0.217767/N_c + F(\Delta\tilde{x}, \Delta\tilde{t}, 1/N_c) \quad (7)$$

Comparing the like leading terms in Eqs. (5) and (7), it is concluded that TASC has half the temporal discretization error ($\Delta\tilde{t}$ term) and double the spatial discretization error ($\Delta\tilde{x}^2$ term) of VSC. These results are not surprising. Since the nearest-neighbor selection scheme produces the smallest possible MCS, it is expected to reduce the spatial error the most. The reduction in the linear $\Delta\tilde{t}$ term can be attributed to the random collision partner selection of TASC, which is similar to the random collision partner selection of DSMC94 and therefore produces a similar temporal error.

This conclusion is further supported by comparison of Figs. 7 and 12 presenting the thermal-conductivity ratio as a function of the spatial resolution for different numbers of simulators per cell for TASC and VSC. In Fig. 12, the curves for 30 and 60 simulators per cell are calculated with TASC while the curves for 7 and 15 simulators per cell are calculated with VSC. The calculations in Fig. 7 presented earlier are all performed with VSC. It is noted that the error of the TASC-calculated profiles is approximately 10% higher than the error of the VSC-calculated profiles. This higher error is counterbalanced by the significantly less simulation time required for the TASC scheme.

7. Efficiency of the DSMC07 algorithm

In Sections 5 and 6, it has been demonstrated that DSMC07 has significantly lower discretization requirements than DSMC94 to achieve the same accuracy. However, the DSMC07 procedures are more complicated and thus require greater computational effort than the DSMC94 procedures. Therefore, when comparing the two algorithms, the question of the efficiency of the new algorithm relative to the original one becomes particularly important.

The efficiency of a DSMC algorithm can be defined as the amount of physical time for a particular physical problem that can be simulated to a prescribed level of accuracy using a fixed amount of computational resources. Three factors potentially affect the efficiency of any DSMC algorithm: the dimensionality of the implementation, the discretization needed to achieve the prescribed accuracy, and the computational effort required by the algorithmic procedures. DSMC07 and DSMC94 have significant differences in each of these areas and in the way they interact. For example, the effect of the dimensionality of the implementation on the computational effort required for collision partner selection is minimal for DSMC94 with its ran-

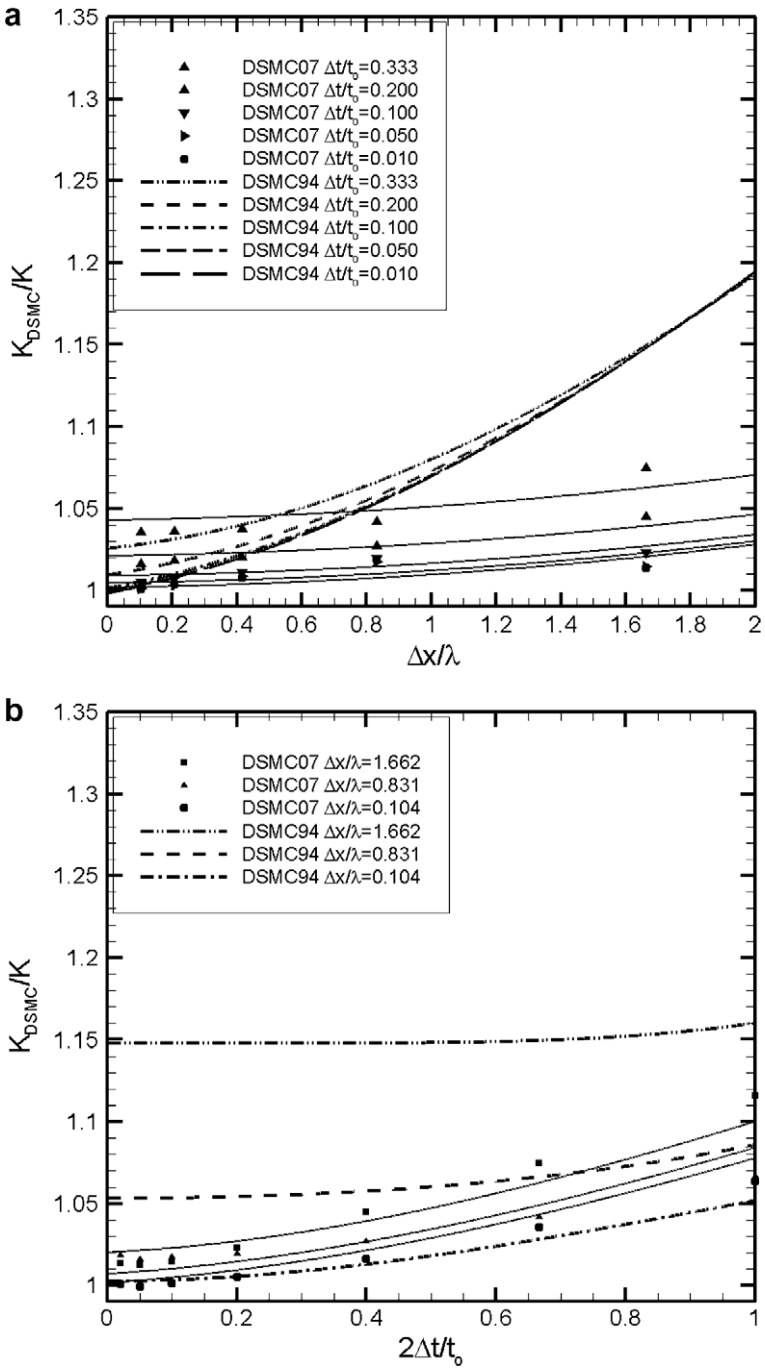

Fig. 11. (a) Thermal-conductivity ratio for $N_c = 30$ as a function of $\Delta\tilde{x}$: TASC version and (b) thermal-conductivity ratio for $N_c = 30$ as a function of $\Delta\tilde{t}$: TASC version.

dom-selection procedures but can be appreciable for DSMC07 with its nearest-neighbor search, which is more complicated in three dimensions than in one.

Because of their different dependences on spatial, temporal, and velocity-space discretization parameters, the relative efficiency of the two algorithms is problem-dependent and cannot be characterized by a single number. For example, since DSMC07 converges linearly but DSMC94 converges quadratically with time step, a reduction in time step is bound to result in a greater improvement of accuracy for DSMC07 than for DSMC94 at small time steps.

An example of the relative efficiency of the two methods for one-dimensional flows can be given based on the cases studied in Section 6. For a number of simulators per cell of $N_c = 15$ and a time step of $\Delta\tilde{t} = 0.2$, DSMC07 using the VSC technique

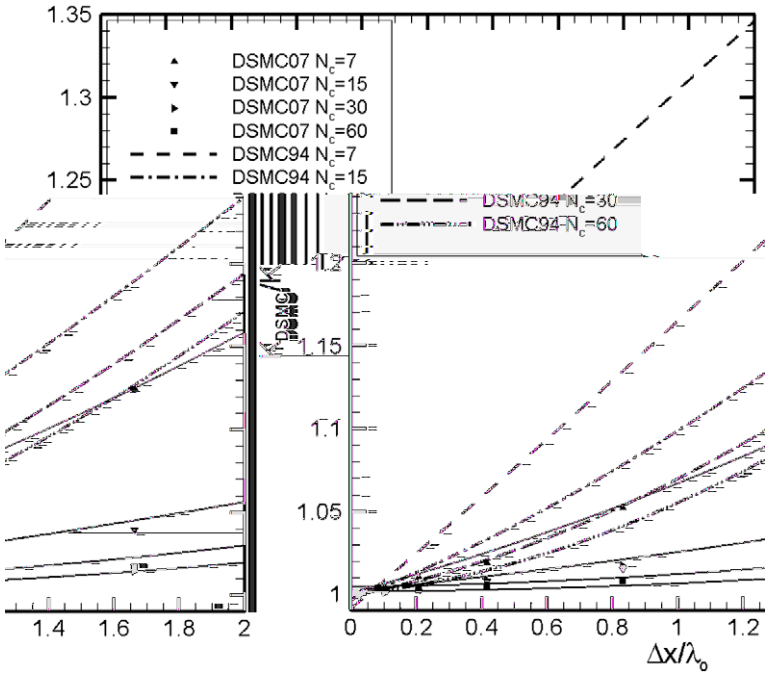

Fig. 12. Thermal-conductivity ratio for $\Delta\tilde{t} \leqslant 0.10$ MCT as a function of $\Delta\tilde{x}$ and N_c: TASC version.

achieves an accuracy of approximately 5% with a cell size of $\Delta\tilde{x} = 0.8$. For the same N_c and $\Delta\tilde{t}$ values, DSMC94 achieves the same accuracy with a cell size of $\Delta\tilde{x} = 0.4$ (i.e., twice the number of cells). Under these conditions, DSMC07 simulates about twice as much physical time as DSMC94 for the same amount of computational time. If the time step and the cell size are reduced to $\Delta\tilde{t} = 0.1$ and $\Delta\tilde{x} = 0.4$, respectively, DSMC07 achieves an error of about 1%. For the same N_c and $\Delta\tilde{t}$ values, DSMC94 achieves 1% accuracy only when the cell size is reduced to $\Delta\tilde{x} = 0.01$. Because of the different requirements on spatial resolution, DSMC07 simulates about four times as much physical time as DSMC94 for the same amount of computational time.

When DSMC07 using the TASC technique is applied to the same two test cases, the physical times simulated by DSMC07 increase to three and five times the corresponding physical times simulated by DSMC94. This ostensibly superior performance of TASC compared to VSC is related to the exceptional efficiency of sorting molecular positions in one-dimensional geometries that is not shared with higher-dimensionality geometries. Thus, for the TASC technique, the efficiency in a two-dimensional or three-dimensional geometry is not expected to be the same as in a one-dimensional geometry.

The efficiency of DSMC07 relative to DSMC94 has also been studied for a real-world subsonic two-dimensional flow [16]. In this study, DSMC07 used the VSC technique to select collision partners. The conclusion was that, when all competing factors were included, DSMC07 was 2–3 times as fast as DSMC94 and required significantly less computational resources. The larger value was attributed to the better performance of DSMC07 in a massively parallel environment because of its smaller memory requirements. The decisive factor was again found to be the ability of DSMC07 to achieve the same level of accuracy as DSMC94 using less resources. The variable time step selection was not a significant contributor in this study because the density variation in the domain was very small.

It should be pointed out that the relative efficiency of the two algorithms may vary for different test cases. For example, in a hypersonics test case, the variable time step is expected to offer some computational advantage at a relatively low cost, further improving the efficiency of DSMC07 relative to DSMC94. However, additional investigation is required to fully quantify the efficiency of the DSMC07 algorithm.

8. Conclusions

A new DSMC method recently proposed by the originator of the DSMC method has been implemented, and its convergence has been studied. The changes in the DSMC algorithm include a nearest-neighbor collision partner selection scheme and a variable adaptive local time step approach. Collisions are no longer calculated at the end of the global time step but are distributed over the duration of a time step.

The study of the convergence characteristics of the new method is performed using a benchmark Fourier flow. It is found that the new DSMC method using nearest-neighbor schemes for the selection of the collision partners achieves very high efficiency since it minimizes the mean collision separation between collision partners. The introduction of this scheme

makes the new algorithm very sensitive to the selection of the time step. However, when the same simulation parameters are used, keeping the average number of simulators per cell around 10, the time step about 20% of the minimum of the mean collision and transit times, and the cell size about one mean free path generally ensures that a higher-accuracy calculation is achieved in comparison to the original algorithm. To achieve the same level of accuracy, the new DSMC algorithm needs significantly less computational resources, which allows for more efficient calculations and extends the set of situations for which simulations can be performed.

Acknowledgments

This work was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. The authors would like to thank Dr. M.F. Barone and Dr. E.S. Piekos of Sandia National Laboratories for their critical reviews of the manuscript.

References

[1] G.A. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Clarendon Press, Oxford, 1994.
[2] W. Wagner, A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation, J. Stat. Phys. 66 (1992) 1011–1044.
[3] M.A. Gallis, J.R. Torczynski, D.J. Rader, Molecular gas dynamics observations of Chapman–Enskog behavior and departures therefrom in nonequilibrium gases, Phys. Rev. E 69 (2004) 042201.
[4] M.A. Gallis, J.R. Torczynski, D.J. Rader, M. Tij, A. Santos, Normal solutions of the Boltzmann equation for highly nonequilibrium Fourier and Couette flow, Phys. Fluids 18 (2006) 017104.
[5] F.J. Alexander, A.L. Garcia, B.A. Alder, Cell size dependence of transport coefficients in stochastic particle algorithms, Phys. Fluids 10 (1998) 1540–1542. Erratum: 'Cell size dependence of transport coefficients in stochastic particle algorithms' [Phys. Fluids 10 (1998) 1540], Phys. Fluids 12 (2000) 731.
[6] A.L. Garcia, W. Wagner, Time step error in direct simulation Monte Carlo, Phys. Fluids 12 (2000) 2621–2633.
[7] N.G. Hadjiconstantinou, Analysis of discretization in the direct simulation Monte Carlo, Phys. Fluids 12 (2000) 2634–2638.
[8] D.J. Rader, M.A. Gallis, J.R. Torczynski, W. Wagner, DSMC convergence behavior of the hard-sphere-gas thermal conductivity for Fourier heat flow, Phys. Fluids 18 (2006) 077102.
[9] M.A. Fallavollita, D. Baganoff, J.D. McDonald, Reduction of simulation cost and error for particle simulations of rarefied flows, J. Comput. Phys. 109 (1993) 30–36.
[10] N.G. Hadjiconstantinou, A.L. Garcia, M.Z. Bazant, G. He, Statistical error in particle simulations of hydrodynamic phenomena, J. Comput. Phys. 187 (2003) 274–297.
[11] G.A. Bird, The DS2V/3V program suite for DSMC calculations, in: M. Capitelli (Ed.), Rarefied Gas Dynamics: 24th International Symposium, AIP Conf. Proc. 762, New York, 2005, pp. 541–546.
[12] G.A. Bird, Visual DSMC program for two-dimensional and axially symmetric flows, The DS2V Program User's Guide, Version 3.8, GAB Consulting, Sydney, 2006.
[13] M.A. Gallis, J.R. Torczynski, D.J. Rader, G.A. Bird, Accuracy and convergence of a new DSMC algorithm, in: 40th AIAA Thermophysics Conference, AIAA 2008-2913, Seattle, Washington, June 2008.
[14] G.J. LeBeau, K.A. Boyles, F.E. Lumpkin, Virtual sub-cells for the direct simulation Monte Carlo method, in: 41st Aerospace Sciences Meeting and Exhibit, AIAA 2003-1031, Reno, Nevada, January 2003.
[15] D. Robbins, Average distance between two points in a box, Am. Math. Mon. 85 (1978) 277–278.
[16] G.A. Bird, M.A. Gallis, J.R. Torczynski, D.J. Rader, Accuracy and efficiency of the sophisticated direct simulation Monte Carlo algorithm, Phys. Fluids 21 (2009) 017103.