Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 195 (2004) 508–527
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Kinetic description of flow past a micro-plate

Andrew J. Christlieb [a,*], W. Nicholas G. Hitchon [b], Iain D. Boyd [c], Quanhua Sun [c]

[a] Department of Mathematics, University of Michigan, 2470 East Hall, Ann Arbor, MI 48109, USA
[b] Department of Electrical and Computer Engineering, University of Wisconsin, 1415 Engineering Dr., Madison, WI 53706, USA
[c] Department of Aerospace Engineering, University of Michigan, 1320 Beal Avenue, Ann Arbor, MI 48109-2140, USA

Received 5 December 2002; received in revised form 24 September 2003; accepted 16 October 2003

Abstract

In this work we discuss a novel numerical scheme and present numerical results for the problem of 'high' (of order unity) Knudsen number, $Kn = \lambda/L$, low velocity gas flow past a micro-plate. The scheme used here is similar to one employed in the past to examine heat flow, but in order to deal with momentum transport in the vicinity of a plate, new techniques had to be developed. These include a new scheme for finding 'transition probabilities', which eliminates some forms of numerical diffusion, and a method for handling reflections off surfaces which preserves the essential properties of the flow. The purpose of the paper is to present these methods and examine their performance. The method is found to function well, but the results indicate that the collision operator which is employed here must be improved in order to obtain accurate results for the drag. Low speed neutral particle transport, in long mean free path (LMFP) environments, presents challenges for well-established techniques, such as the direct simulation Monte Carlo (DSMC) method. In particular, at low flow velocities, statistical methods suffer from noise that may render them impractical in LMFP environments. Solution of the Boltzmann equation is the alternative to these statistical methods. As computing power increases, Boltzmann-based approaches become more accessible. We discuss here a novel non-statistical (no random numbers are used) kinetic model for particle transport and explore its accuracy and sensitivity to resolution and other details of its implementation. The model is an enhanced version of the transition probability matrix (TPM) method. The results generated by the TPM are compared with the information preservation (IP) method, the Navier–Stokes (NS) slip model and when applicable to DSMC. We provide a qualitative comparison of the models and then we compare the results of the different models for various *Kn* for flow past a plate. For *Kn* in the slip flow regime, the TPM, IP and NS models exhibit similar flow features. As the *Kn* passes through the transitional regime, the TPM and IP results are quantitatively similar, while the NS model fails to adequately describe the flows. For *Kn* in the free molecular regime, we compare the TPM to an analytic approximation to flow past a flat plate; the TPM and analytic solution exhibit similar characteristics. The drag coefficient is in agreement in the two cases, although it is sensitive to the angular distribution employed in the collision operator, when particles are relaunched after collisions.

* Corresponding author. Tel.: +1-734-763-5725; fax: +1-734-763-0937.
E-mail addresses: christli@umich.edu (A.J. Christlieb), hitchon@cae.wisc.edu (W.N.G. Hitchon), iainboyd@engin.umich.edu (I.D. Boyd), qsun@engin.umich.edu (Q. Sun).

0021-9991/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2003.10.027

AMS: 65C20; 76P05; 76G25

Keywords: Boltzmann equation; High Knudsen number; Micro scale gas flows; Kinetic models; Transition probability matrix method; Information preservation method; Rarefied gas dynamics

1. Introduction

In this paper we present new methods for solving the Boltzmann equation, which permit the description of neutral gas flow and momentum transport in microscopic systems. These include new approaches to finding the probabilities for particles moving between phase space cells based on the 'convected scheme' (CS); and a philosophy for the treatment of surface interactions which preserves the critical features of the flow. We have designed the method to describe flows past micro/nanoscale objects. Of particular interest are low Mach number, $M < 0.3$, flows in high Knudsen number, $Kn > 0.05$, environments. Characterizing these types of flows will become increasingly important as micro electromechanical systems (MEMS) are reduced in size, and attempts are made to capitalize on the emergence of nano-technology-based systems (NTBS) for altering the properties of the fluid passing through the system [1,2]. For the simple system considered here, we show that the basic method works well, but for truly accurate description of momentum transport, the collision operator must be designed to faithfully reproduce the correct off-diagonal terms in the stress.

In the current paper, we will restrict our attention to the simpler problem of flow past a micro plate over the range of parameters listed above. We employ a 6D (3 space, 3 velocity) model, but with symmetry in z, to describe a flow in two spatial dimensions, for Kn between 0.05 and 10. Such flows present challenges for well-established statistical approaches, such as the direct simulation Monte Carlo (DSMC). At low Mach numbers, particle-based approaches require a large number of samples in order to reduce statistical scatter to a level where the flow can be resolved (sample size $\sim 10^5$ for a variance less than 1 m/s). The need for a large number of samples can render these methods impractical [3]. Fig. 1 illustrates the issue, showing normalized density contours for a DSMC simulation. The flow is for a Mach number of $M = 0.12$ and a Knudsen number of $Kn = 0.05$, where $Kn = \lambda/\text{Plate Length}$.

Adaptations of statistical methods have extended their range to cover these types of flows. One such adaptation is the information preservation (IP) method [3–6]. The IP method is a hybrid model that combines a particle (microscopic) description with a global (macroscopic) description. Although this ex-

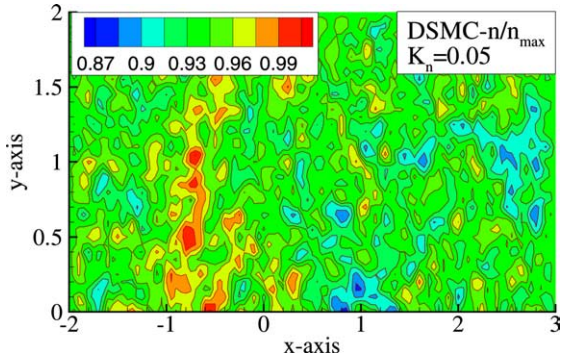

Fig. 1. Normalized density contours of DSMC results for argon flowing past a flat plate. The inlet flow velocity is (40 m/s, 0, 0), the temperature is 288 K and the Knudsen number is 0.05. The sampling size $\sim 3 \times 10^5$ particles/cell, and the simulation domain consists of 4000 cells.

tension works well for flows where $Kn < 1.0$, the continuum model used in adjusting the global information becomes of questionable validity for $Kn \geqslant 1.0$.

The classic approach to this flow problem is to solve the Boltzmann equation. This has not been practical due to lack of computational resources in the past. In addition, accurate solution of the Boltzmann equation presents multiple difficulties which were addressed in [7,8]. In this work we develop a non-statistical (no random numbers used) kinetic model suited to these types of flows. To that end, we have developed a particle transport model based on the use of one-step transition probability matrices, which we refer to as the transition probability matrix method (TPM) [9,10]. The model permits efficient, non-statistical iterative calculation of the scattering rate of particles in each cell of a 6D phase space mesh. The method uses 'propagating' directions for tracking the transport of particles throughout phase space. This transport model is similar in spirit to earlier transition probability transport codes [9,11–15]. Although earlier codes [13,14] are capable of handling arbitrary spatial meshes, they typically assume isotropic scattering and make other simplifying assumptions which reduce the computational overhead, making it possible to store large amounts of geometrical information typically needed. The model presented here does not assume isotropic scattering of particles. Another important difference from the earlier models [13,14] is that we conserve momentum, giving our model the capability to describe a flowing gas. Two momentum conserving collision operators have been employed [16], a simple monoenergetic operator and the BGK model. The monoenergetic model offers advantages in terms of speed, while the BGK model offers a more realistic description, on a 'full' phase space mesh. However, the computational overhead of storing the probabilities for a full phase space mesh is large. For example, if N_c is the number of spatial cells on the mesh, then even the geometric information required to find the probabilities of going from each initial cell to each final cell involves $(N_c)^2$ numbers. To overcome this problem, we limit the amount of information stored by efficiently computing needed information.

In previous work we applied this method to the problem of heat transport in a rarefied gas. We compared our results to those of other methods, which were capable of describing the 1D version of the problem, and found good agreement in heat fluxes, density profiles, etc. [9]. In this work we apply an enhanced form of the TPM to the problem of flow past a micro-plate. We have also applied the IP, DSMC and Navier–Stokes (NS) models to the same flows. The IP model has been extensively benchmarked versus DSMC [6]. The results are qualitatively similar and they provide some insight into the strengths and weaknesses of each of the models. We find broadly similar results for the drag coefficients for these models. The drag coefficient we find from the TPM agrees well with analytic results which are obtained using consistent assumptions. The value of the drag coefficient is sensitive to the angular distribution of the particles and in turn on the collision operator, which determines the angular distribution of the particles when they are relaunched after a collision. The operator we employ in this paper is simple to implement in such a way as to conserve critical quantities (particle numbers, energy and momentum) exactly on the mesh, but for drag calculations, higher moments of the distribution must be considered carefully as well. The form of the collision operator and improvements to increase its accuracy are discussed. In addition the IP and TPM results exhibit similar trends in velocity. As Kn increases, the domain of influence of particles coming off the plate increases in a similar manner for both the IP and TPM. The NS slip model agrees with the IP and TPM for low Kn. As Kn increases, the NS slip model fails to capture the essential physics. The density profiles for the NS slip model become more localized about the plate as Kn increases and the velocity diverges from that predicted by both the TPM and IP models. In addition to comparing the TPM to other numerical models, we compare the TPM to an analytic model for free molecular flow past a flat plate. The effects of the angular resolution employed are discussed. The TPM applied to a flow with $Kn = 10$ exhibits the same flow features as the analytic model for free molecular flow.

So far as we are aware, there have been no other applications of a kinetic theory based on calculation of scattering rates to fluid dynamics of a self-interacting rarefied gas, except our own work which focussed on heat transfer rates. This is a highly effective approach to kinetic theory of such gases which can be expected

to become important as available computational power grows. These studies indicate the issues which need to be addressed to make this technique highly effective in future.

In the following sections we will describe the new aspects of the current form of the TPM, followed by a brief discussion of the major differences between the approaches. Next, we present and compare the results generated by the different models, and finally we summarize our conclusions and discuss future work.

2. Transition probability matrix method

Our objective is to solve the Boltzmann equation for LMFP environment steady-state conditions,

$$\mathbf{v}\cdot\nabla_{\mathbf{r}}f(\mathbf{r},\mathbf{v})=\int_{-\infty}^{\infty}\int_{0}^{4\pi}|\mathbf{v}-\mu|\sigma\Big\{f(\mathbf{r},\hat{\mathbf{v}})f(\mathbf{r},\hat{\mu})-f(\mathbf{r},\mathbf{v})f(\mathbf{r},\mu)\Big\}\,\mathrm{d}\Omega\,\mathrm{d}\mathbf{v}_{\mu}, \tag{1}$$

where $f(\mathbf{r},\mathbf{v})$ is the distribution function of particles, $\mathbf{r}$ is the position in space, $\mathbf{v}$ is the velocity of the incident particles, μ is the velocity of the target particles, σ is the cross-section of the particles and $\mathrm{d}\Omega$ is the solid angle the particles are scattered through after a collision. The method is based on a one-step transition probability matrix (TPM) which describes steady-state particle transport on a phase space mesh. However, this matrix need not be formally constructed. Instead, the transport of the particles is described with two operators; a 'transport' operator and a 'collision' operator.

In the following section we provide an overview of the method, followed by a more detailed description of the two operators. It is worth noting that this approach can be extended to handle flows which evolve in time without the introduction of a costly probability history, as was the case in [17].

2.1. Overview

We present an overview of the TPM. The method is 6D, i.e., it uses a uniform 3D spatial mesh and a 3D velocity space mesh comprising a single energy mesh in combination with a 2D (Φ,Θ) directional mesh. In this paper we impose symmetry in z to obtain results in two spatial dimensions. In the past, we have not restricted our work to uniform spatial meshes [9], but the restriction is imposed here as a way for us to increase accuracy for particles reflecting off surfaces.

We calculate the collision rates in each phase space mesh cell, so information about the direction and energy of the particles is captured at each spatial location of the mesh. T is the probability that, starting after a collision at location $(\mathbf{c}',\mathbf{a}',E')$, a particle will move in space to $(\mathbf{c},\mathbf{a}',E')$, then have its next collision and be redistributed to location $(\mathbf{c},\mathbf{a},E)$. $\mathbf{c}'$ and $\mathbf{c}$ are spatial locations, $\mathbf{a}$ and $\mathbf{a}'$ represent (Φ,Θ) directions and E and E' the energy. $R(\mathbf{c}',\mathbf{a}',E')$ is the number of particles that collided in cell $\mathbf{c}'$ that were redistributed with updated direction $\mathbf{a}'$ and energy E' at the previous iteration. Then

$$R(\mathbf{c},\mathbf{a},E)=\sum_{\mathbf{c}'}\sum_{\mathbf{a}'}\sum_{E'}T(\mathbf{c},\mathbf{a},E:\mathbf{c}',\mathbf{a}',E')R(\mathbf{c}',\mathbf{a}',E'). \tag{2}$$

We break T into two operators, T_{bal} and T_{col}. T_{bal} is the one-step transition probability matrix for particle ballistic motion (that changes $\mathbf{c}$ only) and T_{col} the one-step transition probability matrix which locally (i.e., without changing $\mathbf{c}$) redistributes the particles in energy and direction after a collision. Energy, momentum and particle conservation are strictly enforced.

T_{bal} is used to compute the number of particles per second that collide in cell $\mathbf{c}$, labelled by their direction $\mathbf{a}'$ and energy E' which they had prior to the collision,

$$R(\mathbf{c},\mathbf{a}',E')=\sum_{\mathbf{c}'}R(\mathbf{c}',\mathbf{a}',E')T_{\mathrm{bal}}(\mathbf{c},\mathbf{a}',E':\mathbf{c}',\mathbf{a}',E'), \tag{3}$$

T_{col} is used to redistribute the particles after the collision [12,16],

$$R(\mathbf{c},\mathbf{a},E)=\sum_{\mathbf{a}'}\sum_{E'}R(\mathbf{c},\mathbf{a}',E')T_{\text{col}}(\mathbf{c},\mathbf{a},E:\mathbf{c},\mathbf{a}',E'). \tag{4}$$

2.2. Ballistic transport operator: constructing $R(\mathbf{c},\mathbf{a}',E')$

T_{bal} has been implemented with two different operators, the point source (PS) operator and the CS operator. Both operators make full use of symmetry [9]. The PS operator is nearly identical to the T_{bal} found in [9].

The propagator subdivides space into a finite number of directions

$$\mathbf{a}'_{i,j}=\left(\Phi_i,\Theta_j\right) \tag{5}$$

and extends a finite distance,

$$L_{\text{prop}}=\chi\langle\lambda(\mathbf{c}',E')\rangle, \tag{6}$$

where $\langle\lambda(\mathbf{c}',E')\rangle$ is the average mean free path and $\chi>0$ is a constant. Fig. 2(a) is the PS propagator and Fig. 2(b) is the CS propagator. Both propagators contain a list (in radial order) for each $\mathbf{a}'_{i,j}$, containing geometric information needed to construct T_{bal} for each mesh cell $\mathbf{c}$ that particles moving in the direction $\mathbf{a}'_{i,j}$ interact with. The PS propagator covers the entire (Φ,Θ) space with propagating rays centered about each direction $\mathbf{a}'_{i,j}$, originating from the center of the cell [9]. The geometrical information can be computed via test particles, or as in [9].

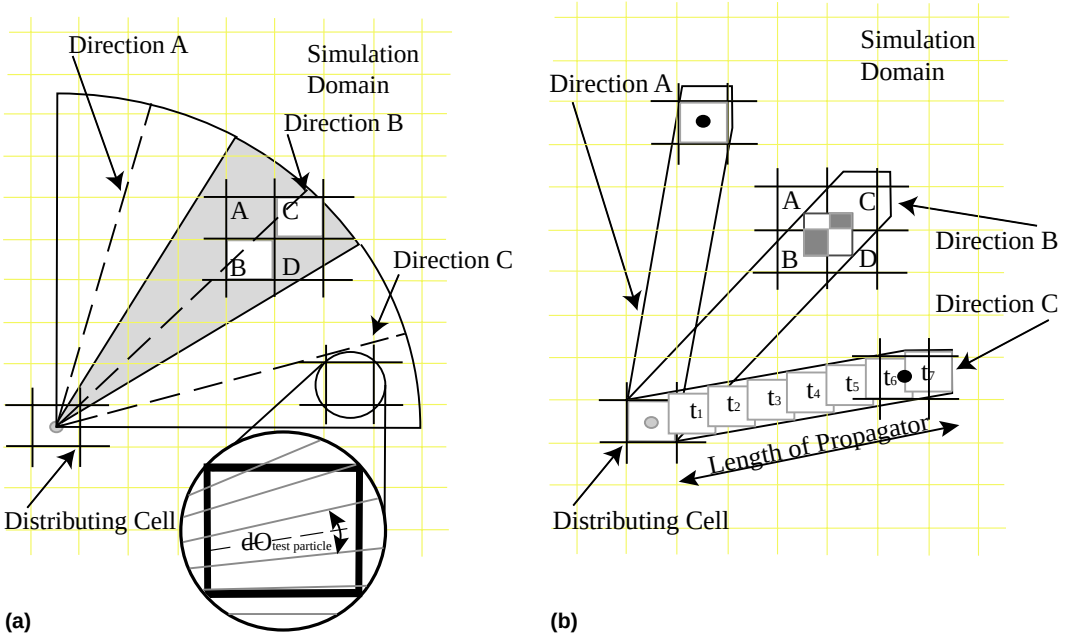

Fig. 2. Two possible propagating structures used in the transport phase, T_{bal}. (a) is a point source propagator and (b) is a convective scheme-based propagator.

The CS uses 'long-lived' moving cells (LLMC) [7,8] to compute T_{bal}. For 'angle' $\mathbf{a}'_{i,j}$, an LLMC is constructed, $\text{MC}(\mathbf{c}', \mathbf{a}'_{i,j})$, that initially overlaps the mesh cell $\mathbf{c}'$. The $\text{MC}(\mathbf{c}', \mathbf{a}'_{i,j})$ is propagated along the direction $\mathbf{a}'_{i,j}$ by time-stepping. Fig. 2(b), direction c, depicts seven time steps. After each time step, the fraction of the particles

$$f_{\text{MC}(\mathbf{c}',\mathbf{a}'_{i,j})} = n\left(1 - \mathrm{e}^{-\frac{d(t_i)}{\langle\lambda(\mathbf{c}',E')\rangle}}\right), \tag{7}$$

which collided in time step t_i is subtracted from n of $\text{MC}(\mathbf{c}', \mathbf{a}'_{i,j})$. $d(t_i)$ is the distance travelled in time t_i. $f_{\text{MC}(\mathbf{c}',\mathbf{a}'_{i,j})}$ is subdivided among the cells that $\text{MC}(\mathbf{c}', \mathbf{a}'_{i,j})$ overlapped after the time step t_i. See Fig. 2(b), direction b, where the cells marked A, B, C and D are the cells overlapped by the moving cell at a particular step. In 2D, the fractions of particles added to cells A, B, C and D, after the t_i step, are:

$$\mathbf{A}_A = \left(\frac{x_{A-\max} - x_{\text{MC}-\min}}{\Delta M}\right)\left(\frac{y_{\text{MC}-\max} - y_{A-\min}}{\Delta M}\right) f_{\text{MC}(\mathbf{c}',\mathbf{a}'_{i,j})}, \tag{8}$$

$$\mathbf{A}_B = \left(\frac{x_{B-\max} - x_{\text{MC}-\min}}{\Delta M}\right)\left(\frac{y_{B-\max} - y_{\text{MC}-\min}}{\Delta M}\right) f_{\text{MC}(\mathbf{c}',\mathbf{a}'_{i,j})}, \tag{9}$$

$$\mathbf{A}_C = \left(\frac{x_{\text{MC}-\max} - x_{C-\min}}{\Delta M}\right)\left(\frac{y_{\text{MC}-\max} - y_{C-\min}}{\Delta M}\right) f_{\text{MC}(\mathbf{c}',\mathbf{a}'_{i,j})}, \tag{10}$$

$$\mathbf{A}_D = \left(\frac{x_{\text{MC}-\max} - x_{D-\min}}{\Delta M}\right)\left(\frac{y_{D-\max} - y_{\text{MC}-\min}}{\Delta M}\right) f_{\text{MC}(\mathbf{c}',\mathbf{a}'_{i,j})}, \tag{11}$$

where $x/y_{\text{name}-\max/\min}$ are the coordinates of the mesh cell or moving cell (see [7,8] for a full description of the CS) and ΔM is the mesh spacing. This process of time stepping is continued down the propagating direction until the LLMC reaches the end of the propagator. At this point, all remaining material in $\text{MC}(\mathbf{c}', \mathbf{a}'_{i,j})$ is subdivided among the last group of cells the LLMC overlaps. For a single initial cell, the CS does not cover all of (Φ, Θ) space, as can be seen in Fig. 2(b), but the CS does redistribute particles that have collided in cell $\mathbf{c}'$ from the whole volume of $\mathbf{c}'$, not just the center [12]. Each CS ray has the same number of particles placed in it as the PS ray for the same direction, so the solid angle of the ray is properly accounted for.

2.3. Collision operator: constructing $R(\mathbf{c},\mathbf{a}',E')$

T_{col} redistributes particles on the mesh in energy while conserving momentum in particle-particle collisions [9].

For each cell, the total momentum brought into the cell by particles which collide there is computed (their initial direction being taken to be that of the center of the incoming rays). Then the distribution of outgoing particles is chosen to conserve that momentum [9,16]. The form of the distribution is

$$f(\mathbf{a}) = (1 + \alpha v_x(\mathbf{a})/v + \beta v_y(\mathbf{a})/v + \gamma v_z(\mathbf{a})/v), \tag{12}$$

where v is the speed of the outgoing particles, $v_x(\mathbf{a})$, $v_y(\mathbf{a})$ and $v_z(\mathbf{a})$ are the components of velocity of the particles moving in the direction $\mathbf{a}$ and α, β and γ are normalization factors. The distribution $f(\mathbf{a})$ must total to one, i.e., $\int f(\mathbf{a})\,\mathrm{d}\Omega_a = 1$. The integral of $f(\mathbf{a})$ places the needed constraint on Eq. (12) in order to find α, β and γ. The mean value of v_x is

$$\langle v_x \rangle = \alpha v \eta_x, \tag{13}$$

where η_x is an integration factor. Since we chose to approximate the integration using a summation, η_x is given by

$$\eta_x = \frac{\sum_a \mathrm{d}\Omega_a}{\sum_a (v_x/v)\,\mathrm{d}\Omega_a}. \tag{14}$$

Once η_x is determined, for the given set of directions, α can be found exactly, to ensure momentum conservation. η_y and η_z are found in the same manner. (The ηs are the same for all locations and energies.) This approach was outlined in [16], where particle and energy conservation were also discussed. [16] provides details, including how large values of the mean velocity are handled. Other angular distributions could be employed instead, for instance, to allow for an accurate differential cross-section. In future we shall implement a drifting Maxwellian distribution for use when relaunching particles after collisions. This has the advantage of giving a more physical distribution, but is more cumbersome to use while conserving particles, momentum and energy exactly.

2.4. Boundary conditions and surface interactions

The effect of a reflecting surface should, for a uniform density above the surface, be to give the same flux coming back off the surface as we would have for a uniform density behind the surface. Similarly, at the edge of the simulation region, where gas is introduced, the flux coming in should be exactly what we would get from a large exterior volume where we (usually) specify a uniform density. Much of our effort in setting up the simulation goes into handling boundaries so as to mimic these 'equivalent' volumes. Mathematically, since we are essentially using a method of characteristics, it is appropriate to specify the upstream boundary condition. 'Upstream' is a different spatial direction for different velocities. At an inlet, the incoming distribution obeys $f(x,v) = f^{\text{upstream}}(x^u, v^u)$. At a reflecting boundary $f^{\text{out}}(x^{\text{out}}, v^{\text{out}}) = f^{\text{in}}(x^{\text{in}}, v^{\text{in}})$ if the reflection is specular; a discussion of specular and diffuse reflection is given below. In the following sections we discuss the modelling of particle reflection off a surface, describing how to achieve uniform coverage of a surface and how to handle surfaces with an arbitrary angle of inclination, i.e., surfaces that do not lie along any of the principal (Φ, Θ) directions.

2.4.1. Particle reflections off a surface

In this section we describe how we model boundary conditions for surfaces and free space boundaries. At issue is how to approach the injection of particles along free boundaries and particle reflection off surfaces in a consistent manner for both of the transport models considered, the PS and CS propagators.

One difficulty with making a surface reproduce its equivalent volume is caused by the fact that we sometimes use the PS, instead of integrating the source over the entire initial cell, see Fig. 3. The PS uses rays (about each direction $\mathbf{a}'_{i,j}$) which fill the solid angle centered around the PS. Particles travel down these rays and strike all the cells within each ray. Rays coming from a finite number of points on a surface cannot exactly reproduce the angular and spatial distribution of particles coming from PS in an equivalent volume behind the surface, Fig. 3(b).

For the PS propagator, we achieve the equivalent volume rule by placing fictitious volume cells (ghost cells) behind the surface and use a method of images, Fig. 3(a). Numerous other versions of the boundary conditions were considered, which failed for subtle reasons which we do not have the space to describe here. This method is the simplest that we have found which satisfies the 'equivalent volume' test.

The CS provides an alternative way to handle generating probabilities, which eliminates the error in using a PS distribution off surfaces. This scheme is more convenient and easier to use than point sources, for several reasons. One advantage is that launching a CS propagator from a surface is indeed capable of generating the same angular distribution as the equivalent volume. This will be described in a later subsection. However, for consistency between the PS and CS models we have implemented the CS using ghost cells behind the surface.

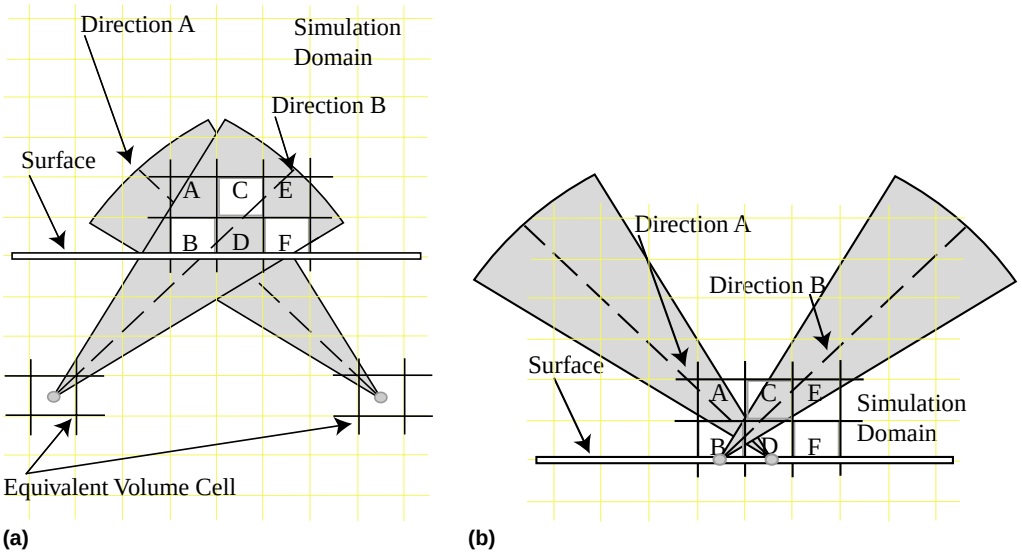

Fig. 3. Two surface 'collision' operators showing equivalent volume vs. surface point source coverage of space. (a) Equivalent volume coverage of space for two directions. Particles are put back as if from volume cells behind the surface. (b) Surface point source coverage of space for roughly the same two directions. Particles are put in rays originating at the surface.

Reflection at a surface is handled by the method of images. Particles which reflect specularly are allowed to travel down a fictitious ray, which is the continuation of the original ray behind the surface, and are placed in fictitious volume cells, see Fig. 4. After they have been propagated along the ray, they are reflected back into the real volume, i.e., particles that collide in volume reflecting (ghost) cell B are placed in cell B with the correct directional information. This would be cumbersome, but we make use of a mapping back function (see below) to decrease the length of the fictitious ray. These are then reflected back into the real volume. This procedure achieves the equivalent effect in a more efficient fashion.

Diffuse reflection is handled by finding a set of 'mirror-image' rays for an incoming ray, whose center hits a particular surface cell $\tilde{c}$ (the dot on the wall of Fig. 4(b). If the incoming ray is at angles labelled (i, j), the mirror image rays are $(-i, j)$, $(i, -j)$ and $(-i, -j)$, and come from volume cells above the surface which also strikes surface cell $\tilde{c}$, in general not at the same point in $\tilde{c}$. The cells these rays come from are mirror-image origin cells (MIOCs) for the initial cell a. Sharing the diffusely reflected particles among all four of these rays equally as if they came in equal numbers from MIOCs provides total transverse momentum loss. The shared particles encounter both the volume reflecting cell in the original direction (reflecting cell B) and the 'conjugate' volume reflecting cells, such as reflecting cell B' in Fig. 4(b). B' is the mirror image obtained when B is reflected about the center of the surface cell $\tilde{c}$. Particles placed in the conjugate cells are reflected into the real volume in the same manner as those placed in reflecting cell B. Partial sharing is done to achieve a mixture of specular and diffuse reflection; so perhaps 10% of the ray might scatter specularly and the remaining 90% be shared. In addition to this procedure providing exact momentum loss, for a uniform source above the surface it guarantees satisfaction of the equivalent volume rule, which is to say, it gives the exact numerical equivalent of the $\cos\Theta$ distribution coming off the surface.

To satisfy the 'equivalent volume' test on the free boundary of our simulation domain, we inject particles from a region of ghost cells several mean free paths deep. The density launched which reaches the interior is recorded and the recorded density is added at each iteration. Again, this would be cumbersome but we make use of our mapping back function (see below) to store the particle collision rate, of the injected particles on the mesh, in a compact form and add that rate back at each step. In addition, we can allow for

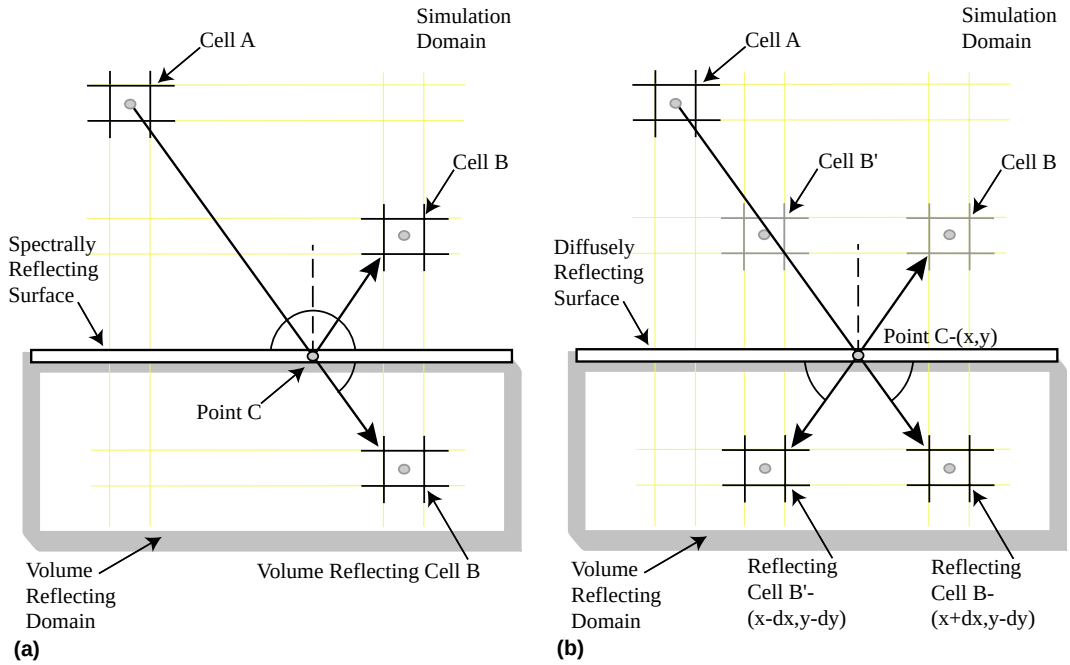

Fig. 4. Depiction of how wall collisions are handled for spectrally and diffusely reflecting particles. (a) Specularly reflecting particles and figure. (b) Diffusely reflecting particles. In (b) point C is the center of the cell that the propagating ray hits.

variations in mean free path using a combination of the CS propagator, mapping back and the null collision operator (see below).

As an alternative to using a method of images to ensure uniformity of the density of particles coming off a surface, one could set up a 'map'. The map would store where particles would be put back, if we used the method of images, for a uniform 'isotropic' distribution function placed in the phase space cells in front of the surface cells. Using the map to determine where particles coming off the surface should go will give the correct approximate $\cos\Theta$ distribution for our discrete phase space mesh. This approximate $\cos\Theta$ distribution will guarantee uniformity in density.

The 'mirror-image' approach to describing diffuse reflection can lead to significant lateral displacement (parallel to the reflecting surface) of the particles. This displacement is unphysical and may lead to numerical diffusion parallel to the surface. For this reason, the handling of diffuse reflection by construction of a 'map' equivalent to the $\cos\Theta$ distribution may be desirable. This will be the topic of future work.

2.4.2. Uniform coverage of surface

The remaining issue is how to map the material from the volume onto the surface in such a way as to ensure uniform coverage of the surface. In particular, we want uniform coverage of the surface for particles coming into corners. In the present work we are using a uniform mesh to permit us to achieve very high levels of accuracy in our algorithms. The surface cells share the symmetry of the volume cells. The easiest way to ensure uniform coverage of the surface at a corner is to 'credit' the density striking the surface to that cell where the ray center strikes. Uniform coverage is provided in part by the translational symmetry of the mesh. If one imagines a sheet of moving cells approaching a corner (which necessarily implies a direction for the sheet's motion) each surface cell which is struck, if we use this rule, is 'unaware' of the surface cells

downstream, so its density (due to *this* sheet) cannot be affected by conditions downstream. Using ray centers to determine which surface cell a moving cell goes through achieves the following:

- Makes sure that as a sheet of cells goes into a corner we will get uniform coverage since the surface cells will not 'see' the surface cells closer to the corner than them, (i.e., will not be affected by them) – and this works even for the last cell before the corner. So no surface cells should be affected by the corner.
- It will also allow the equivalent volume flux to be set up for each surface cell in a consistent and straightforward way.
- These things taken together should make the flux coming past an exterior corner smooth – in the case of isotropy, as if the corner were not there.

It is notable that, for zero flow velocity at the boundary, these algorithms give a uniform density and zero flow velocity in the interior, to very high accuracy.

2.4.3. Surfaces at arbitrary angles and resolution

An important topic for our future work on micro scale flows is the development of a consistent approach to surfaces with an angle of inclination to the direction of the flow. For flow past a flat plane, one can choose the subdivision of the principal (Φ, Θ) directions so that the surface lies along one of the principal directions. However, there are issues to be worked out regarding how to handle cells cut by the surface and how to guarantee uniformity. These issues are not insurmountable, but the problems get far more complicated when there are several surfaces that are joined together where at least one of the surfaces does not lie along any of the principal (Φ, Θ) directions. Therefore, we plan to adopt the approach that the surface can be approximated by a staircase passing through the surface where the faces of the staircase would be made up out of the faces of the mesh cells. We plan to use the effective $\cos\Theta$ distribution for each surface cell, that we get from the equivalent volume rule (discussed above), for propagating particles away from the surface. To add specular reflection to this, we could weight the approximate $\cos\Theta$ distribution with a factor $(1 + \alpha\cos\Theta)$, where α is chosen to (roughly) conserve transverse momentum for the fraction of particles reflected specularly, in the direction tangent to the TRUE surface (not the local stair surface). The way to normalize α for each of the two types of stair surface cells will be calculated once and stored.

To set up a more accurate propagator for reflection, imagine that we take a moving cell and divide each of its sides into 2; and we do the same for each surface cell. In other words, we are working on a mesh which is twice as fine. Then, we set up the reflection propagator using those small cells, as if those are the 'actual' cells. We *could* divide each moving cell into smaller cells as it approaches the surface and set up a reflection propagator for each small surface cell; then figure out (one time) how much density each surface cell's propagator puts in each large cell (if there are several surface cells, each has a different mapping onto the big cells). This should leave specular reflection unchanged, but make diffuse reflection have less numerical diffusion. We could also just take the total flux hitting any given large surface cell, but put it back using a sum of the propagators for the small surface cells.

2.5. Mapping back and the null collision operator

In this work we sometimes map back particles in a ray, to achieve greater efficiency, before the particles have travelled as far as they eventually will along a direction **a**. After they have travelled one or two mean free paths, so relatively few particles remain in the ray, there is little benefit to following the particles further. Instead, we replace them in a cell **c** of the mesh, at the exact same angle and speed. Their motion along **a** is then continued when we next propagate all of the particles in that volume cell which have the same momentum. There is some numerical diffusion in space, in this process, but none in momentum, and few particles are involved. Importantly, this process does not introduce non-uniformities into an otherwise uniform flow. Since the density variations we are studying are sometimes small, we must ensure that numerical errors do not introduce density variations.

The advantages of the CS version of the propagator follow in part from the fact that it only employs a discrete set of angles, whereas the PS version employs a set of spreading rays. The PS rays are centered on discrete angles but they spread over a range of angles. This allows the CS version to use 'mapping back' of particles, without altering the particles' angular distribution. While the mapping back is not essential, it makes several procedures a great deal more efficient. An increase in efficiency comes from the fact that the length of time an individual iteration takes is proportional to $(\chi\langle\lambda(\mathbf{c},E)\rangle/\Delta M)^2$ in 2D and $(\chi\langle\lambda(\mathbf{c},E)\rangle/\Delta M)^3$ in 3D. This is because the number of cells the propagator must loop over in a single step is proportional to the the number of cells that fit inside the area or volume the propagator sweeps out in space. The CS propagator does not fill the volume as fully as the PS, so the scaling is less detrimental than this.

The null collision operator was introduced for Monte Carlo simulations [18] and we have used it in TPM calculations. It consists of overestimating the collision rate by assuming a constant mean free path λ (or collision frequency ν) which is known to be too small (large). The fixed λ (ν) makes it easier to calculate the propagator. The overcounting of the collisions is remedied by taking some of the particles, which were removed from the ray as having collided, and mapping them back, as described above, with the identical momentum they had before their spurious collision. Both the null collision operator and the mapping back increase the numerical scattering rate and hence they increase the effective collision frequency which must be used to convert the scattering rate back to the density.

3. A comparison of the models

Generally, DSMC is one of the most successful particle methods for rarefied gas flows. It generates statistical scatter which requires a huge number of samples for low speed gas flows. The information preservation method was proposed to solve the sampling difficulty in the DSMC method. In IP, particles simulated in the DSMC method additionally preserve macroscopic information about the flow field. This dramatically reduces the statistical noise of the particle method for very low speed gas flows. It was shown [6] that the IP method, with models for updating the flow field, works very well for low speed gas flows ranging from the continuum regime to the free molecular regime.

The Navier–Stokes equations are solved using a finite volume formulation [19].

The TPM was proposed as an alternative to statistical particle transport models. Its accuracy is limited only by the accuracy of the propagators which are employed; if an accurate enough propagator is known, there is no limit to the range of mean free paths which are employed. In particular, the CS method uses short time steps to build up an overall propagator which accurately describes the fate of all the particles along a ray, extending for many mean free paths. The mesh size must be fine enough to resolve real physical phenomena, of course. The drawback to this version of the TPM is that the BGK model becomes of questionable validity for $Kn > 1.0$. However, given the differential cross-section for a desired species, the TPM could easily be altered to handle collisions exactly for high Kn flows. Another alternative to the BGK model is to use a DSMC model to determine the statistics for collisions [12].

4. Results

In the past, we have studied 1D heat transport for high Kn regimes using the TPM, [9], and obtained good agreement with previous work. The model employed here has also been applied to the Rayleigh flow. The results of the Rayleigh flow generated with the TPM were within a few percent of those of Bird [20]. The present model is intended for problems in three spatial dimensions, as opposed to some of the 1D models with which we make comparisons.

In this section, the results for the TPM, IP and NS simulations, for flow past a flat plate, are presented and discussed. All of the flows presented are for argon. The same far field boundary conditions (an inlet velocity set to ($v_x = 40$ m/s, $v_y = 0$ m/s, $v_z = 0$ m/s)) are used for all the runs presented. Although these boundary conditions are the infinity boundary conditions, they were used because they make direct comparison of the models easier. Since the speed of sound in argon is around 316 m/s, $M \sim 0.12$. The flows presented for all three cases are for Knudsen numbers (λ/L) of 0.05, 0.2 and 1.2, where L is the length of the plate, which is held constant. Table 1 summarizes the maximum and minimum values the three simulation methods computed for various Kn. A test of convergence with mesh size, based on the variation in density, is shown in Fig. 5. The method appears to be giving quadratic convergence for this Knudsen number. The main error in the method comes from the 'mapping back' operation, which itself depends on an area of overlap. This is equivalent to linear interpolation – so the method is at best second order accurate. The results shown in Table 1 correspond to Figs. 6–8. In all the figures the first columns are the normalized density contours, the second and third columns are the normalized x and y-velocity contours respectively, for the TPM, IP and (for Kn low enough) NS models. We also compare the drag coefficient, C_d, for Kn of 0.05, 0.2, 0.8 and 1.2 as predicted by these models. In addition, we present TPM results for $Kn = 10$ and compare these results to a simple analytic model for free molecular flow past a flat plate.

Table 1
Normalized minimum and maximum values predicted by the TPM, NS and IP

Model	Kn	n_{min}/n_0	n_{max}/n_0	v_x^{min}/v_0	v_x^{max}/v_0	v_y^{min}/v_0	v_y^{max}/v_0
TPM	0.05	0.981	1.019	0.13	1.15	−0.16	0.17
IP	0.05	0.980	1.020	0.13	1.07	−0.12	0.17
NS	0.05	0.973	1.034	0.16	1.14	−0.17	0.17
TPM	0.2	0.985	1.022	0.28	1.07	−0.13	0.14
IP	0.2	0.972	1.028	0.26	1.02	−0.09	0.11
NS	0.2	0.942	1.062	0.36	1.03	−0.11	0.11
TPM	1.2	0.978	1.022	0.50	1.02	−0.12	0.12
IP	1.2	0.966	1.038	0.46	1.00	−0.09	0.09

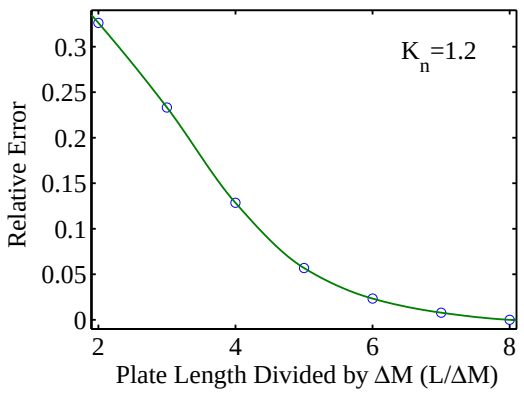

Fig. 5. Plot of error in the maximum normalized density on the mesh, assuming the finest mesh used provides the correct density. At low Knudsen number, a considerably finer mesh is necessary.

4.1. Comparison of flow fields

In this section we make a comparison of the flow fields generated by the TPM, IP and NS models.

Fig. 6 is for $Kn = 0.05$. The upstream region is $1\frac{1}{2}$ times the length of the plate, the downstream region is $2\frac{1}{2}$ times the length of the plate and the height of the simulation domain is two times the length of the plate. The simulations were run on two different meshes. The first mesh used was the same for all three simulations to make comparisons easier. This first mesh had a uniform spacing of a mean free path, i.e., $\Delta M = \langle \lambda(\mathbf{c}, E) \rangle$. The second mesh used by the IP and NS models was non-uniform with many cells packed around the plate, while the second mesh for TPM was a uniform mesh with double the resolution of the first uniform mesh. Except at the upstream boundary, where the artificial fixing of the density causes unphysical behavior in the IP and NS results, the density contours exhibit similar trends in all cases. The results indicate that the TPM and IP tend to converge as the resolution of the mesh increases.

Fig. 7 shows the density, x- and y-velocity contours for $Kn = 0.2$. The results shown in Fig. 7 are for two different meshes. The TPM was run on a uniform mesh with a mesh spacing of $\Delta M = \langle \lambda(\mathbf{c}, E) \rangle / 2$. The IP and NS results are for a non-uniform mesh that packs cells around the plate. The density contours exhibit similar behavior. The peak density computed by the TPM is slightly off the surface of the plate, whereas the IP model predicts that the maximum and minimum density should be at the surface of the plate. The

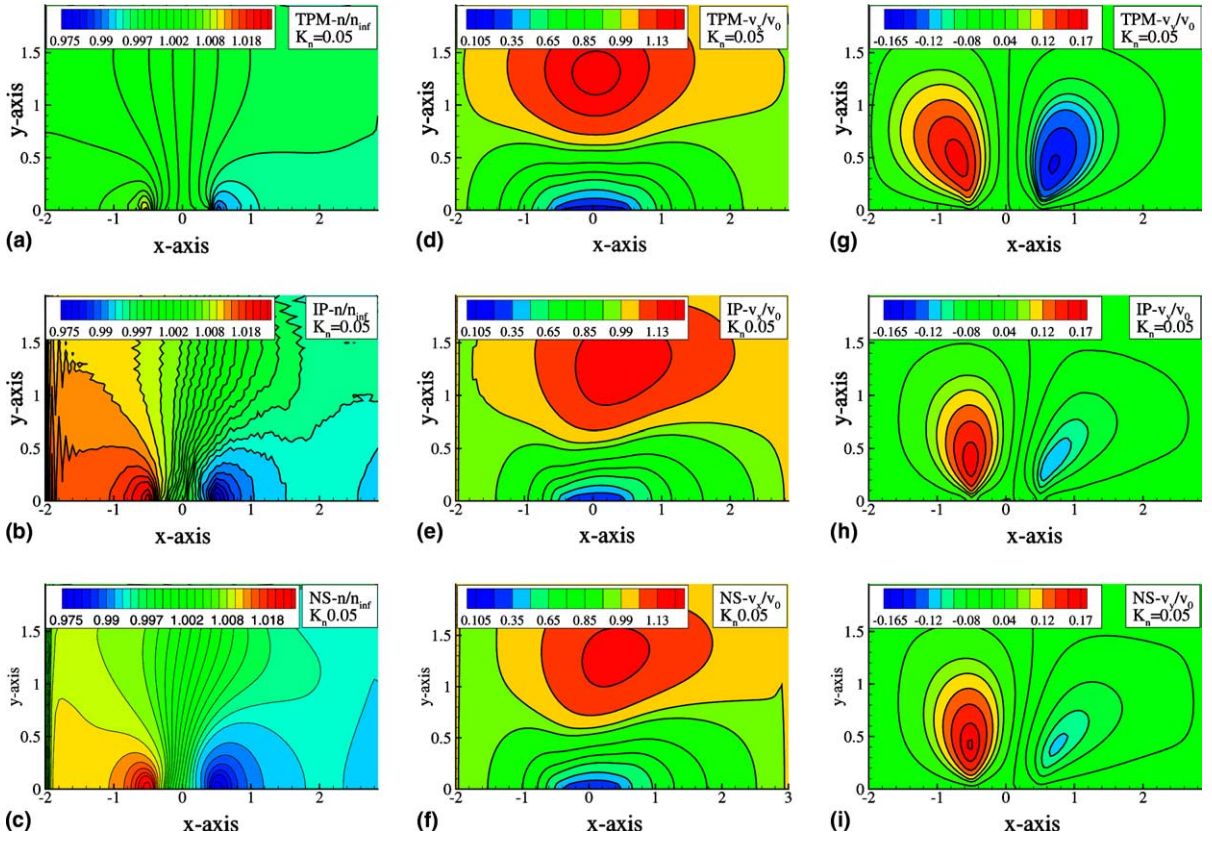

Fig. 6. Results for flow past a flat plate for the TPM, IP and NS models generated with a uniform grid. The flow is argon with an inlet velocity of (40 m/s, 0, 0) and $Kn = 0.05$. The first column is normalized density. The second column is the normalized x velocity and the third column is the normalized y velocity.

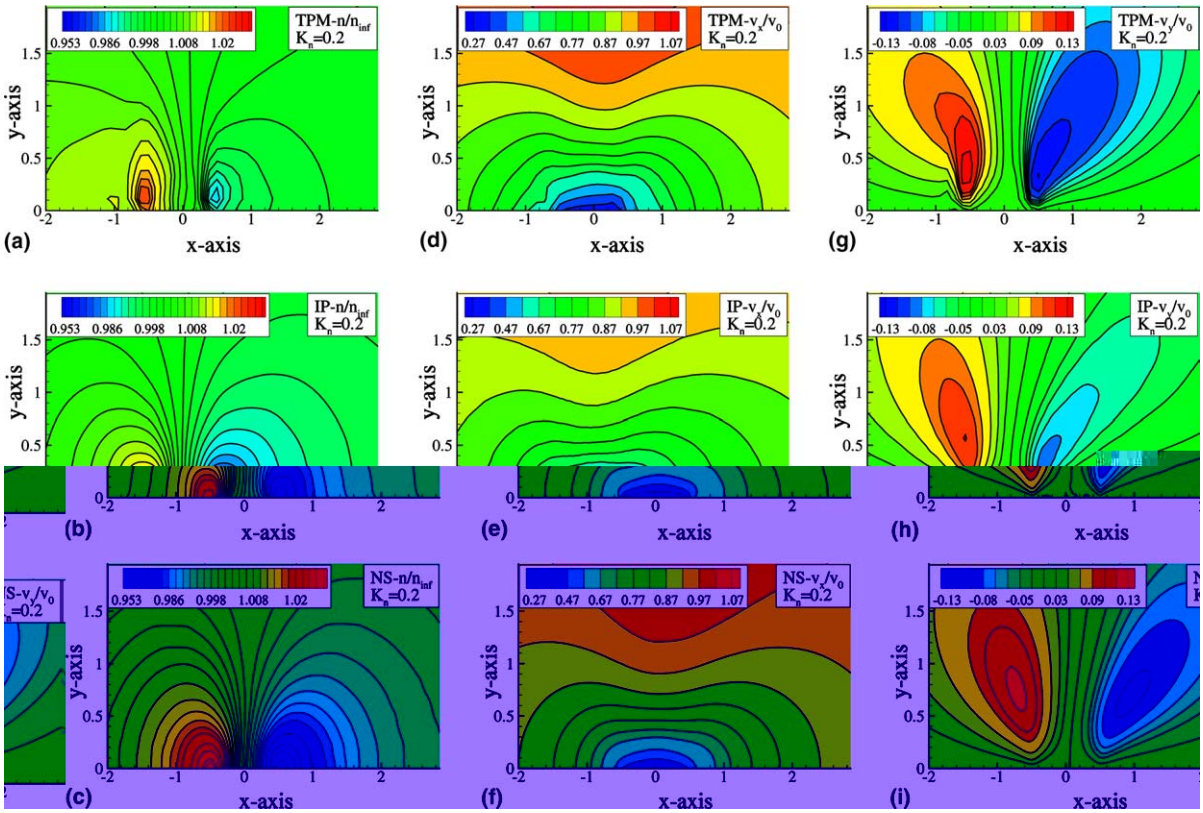

Fig. 7. Results for flow past a flat plate for the TPM, IP and NS models. The flow is argon with an inlet velocity of (40 m/s 0, 0) and $Kn = 0.2$. The first column is normalized density. The second column is the normalized x velocity and the third column is the normalized y velocity.

normalized velocities for all three models also exhibit similar behavior, as can be seen in Fig. 7. The maximum and minimum values are shown in Table 1.

Fig. 8 is for $Kn = 1.2$, where NS is inapplicable. The upstream region is five times the length of the plate, the downstream region is seven times the length of the plate and the height of the simulation domain is four times the length of the plate. The TPM results shown in Fig. 8 are for a uniform mesh while the results for the IP model shown in Fig. 8 are for a non-uniform mesh. The density contours for Kn of 1.2 are not as similar as for the lower Kn, but the models have their minimum and maximum densities in about the same location. Both the x and y velocity contours for the TPM and IP are very similar. The IP results for the same uniform mesh spacing, used for the TPM, have the same general shape but the minimum and maximum values differ from the IP or TPM results. The density range for the uniform mesh IP is (1.01, 0.99) and x- and y-velocity ranges are $(0.57, 1.0)_x$ and $(-0.016, 0.016)_y$.

There is a qualitative difference between the CS results and the other sets of results displayed here. The difference is that in the CS the density variation has its maxima and minima a little way off the plate, whereas in the other simulations the peaks are at least very close to the plate. This observation is not explained as yet, although we tentatively attribute it to numerical diffusion. The CS requires a substantial 'mesh' not only in physical space but also in velocity space. This does not at present permit us to reduce the spatial mesh size as much as is done in the other schemes. The results are otherwise apparently converged with respect to mesh size, however.

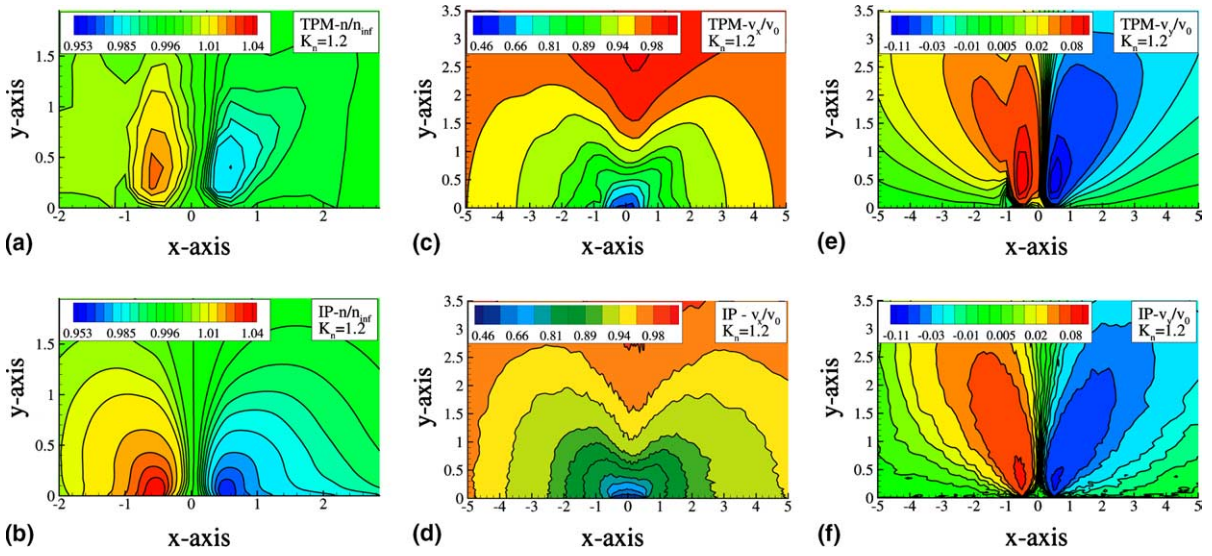

Fig. 8. Results for flow past a flat plate for the TPM and IP models. The flow is argon with an inlet velocity of (40 m/s, 0, 0) and $Kn = 1.2$. The first column is normalized density. The second column is the normalized x velocity and the third column is the normalized y velocity. The density plots show different behavior at the lower boundary, which is apparently due to the symmetry boundary conditions when applied in the TPM as opposed to the finite volume method. It is also apparent that the TPM results were obtained on a much coarser spatial mesh.

To get a better understanding of the similarities and differences between the solutions generated by the models, we compared line plots of the different domains for $y = 0$, 0.25 and 0.5. A set of plots which allow the various schemes to be compared in detail is given in Figs. 9–11. The solid line is the TPM, the dashed line is the IP (and the dot-dashed line is the NS model, included only for $Kn = 0.05$ and 0.2). Fig. 9 shows the x-velocity right at the plate, for three different Knudsen numbers, demonstrating rather close agreement. Figs. 10 and 11 show density, x and y velocities, for different heights above the plate. The plots are for $Kn = 0.2$ and 1.2, respectively. For $Kn = 0.2$, the agreement is good, except that the density variation is somewhat less than in the particle simulations – which may be related to the above point about numerical diffusion and mesh size. At $Kn = 1.2$, there appears to be some qualitative difference in the profile of the x-velocity, which seems to have a local maximum above the plate according to the particle simulation. Since

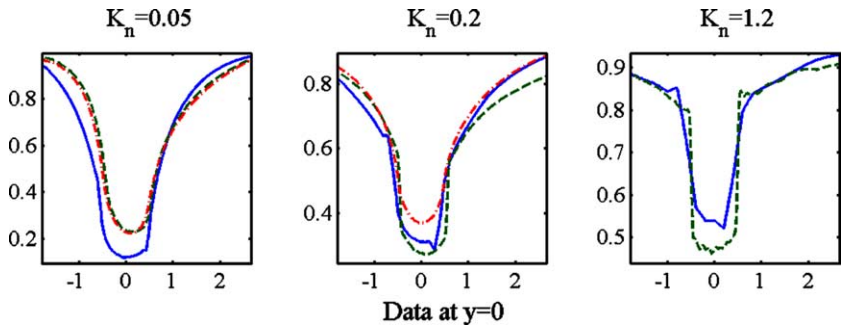

Fig. 9. Line plots of x-velocity with $y = 0.0$ for flow past a flat plate for the TPM, IP and ($Kn = 0.05$ and 0.2 only) NS models. The solid line is the TPM, the dashed line is the IP and the dot-dashed line is the NS model.

Fig. 10. Line plots of density, x and y velocity for different heights above the plate with $Kn = 0.2$. The solid line is the TPM, the dashed line is the IP and the dot-dashed line is the NS model. The top row is $y = 0.25$ and the bottom row is $y = 0.5$.

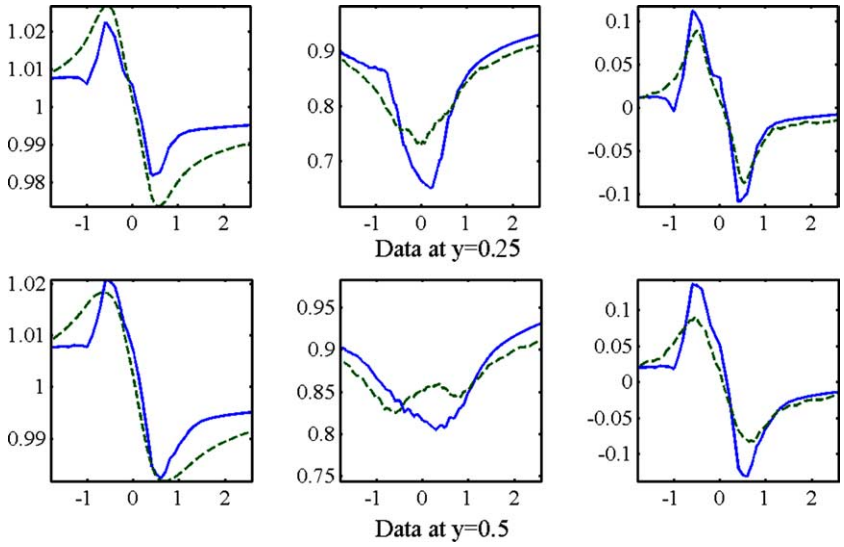

Fig. 11. Line plots of density, x and y velocity for different heights above the plate with $Kn = 1.2$. The solid line is the TPM and the dashed line is the IP model. The top row is $y = 0.25$ and the bottom row is $y = 0.5$.

this seems unlikely, this may indicate the sort of error bars we should place on these plots. Otherwise the agreement is again quite close – and certainly within expected levels of error.

Table 1 provides a summary. As Kn increases, we find that the variation in the normalized densities increases for all three models. However, the range of normalized density increases the most for the NS model and the least for the TPM. The TPM and IP x-velocities right above the plate only differ by 9%.

y-velocities are comparable. To summarize, the TPM and IP methods predict similar trends in the density in terms of how particle density profiles spread out as Kn increases and the TPM and IP methods predict very similar behavior in the velocity profiles.

4.2. Free molecular flow

For a gas in free molecular flow, one might expect the flows for $Kn = 10$ and 100 to exhibit similar characteristics. So in that spirit, we compare the results of the TPM with $Kn = 10$ to an analytic solution for flow past a flat plate with $Kn = \infty$.

Fig. 13 is for $Kn = 10$. The results were generated with the TPM. The upstream region is five times the length of the plate, the downstream region is seven times the length of the plate and the height of the simulation domain is four times the length of the plate. The maximum and minimum densities are (1.03, 0.96). The minimum and maximum velocities are $(0.55, 1.01)_x$ and $(-0.20, 0.21)_y$. All of the contours are more striated than in the previous cases, due to the absence of collisions. (Increasing the number of angular bins in the (x, y)-plane to 128 removed all visible striations, whereas in cases with any collisions, 38 bins were sufficient.)

Typical results for the analytic model are shown in Fig. 12. The minimum and maximum are comparable to the results of the TPM for $Kn = 10$. In addition, the overall behavior of the density and velocity contours is in agreement. Because the calculations for this model take only minutes, we had great flexibility to vary the mesh size and the number of angles used. This in turn allowed us to investigate the 'fingered' nature of the flow in Fig. 13. We found that for fewer than 32 subdivisions in Θ, the free molecular results become striated and exhibit the same behavior seen in TPM results for $Kn = 10$. Since the TPM results shown here all use the same number of directions, this would indicate that the number of discrete directions followed by the model becomes an important factor with increasing Kn. In short, the closer the flow gets to actual free molecular flow, the greater the need for more angular resolution.

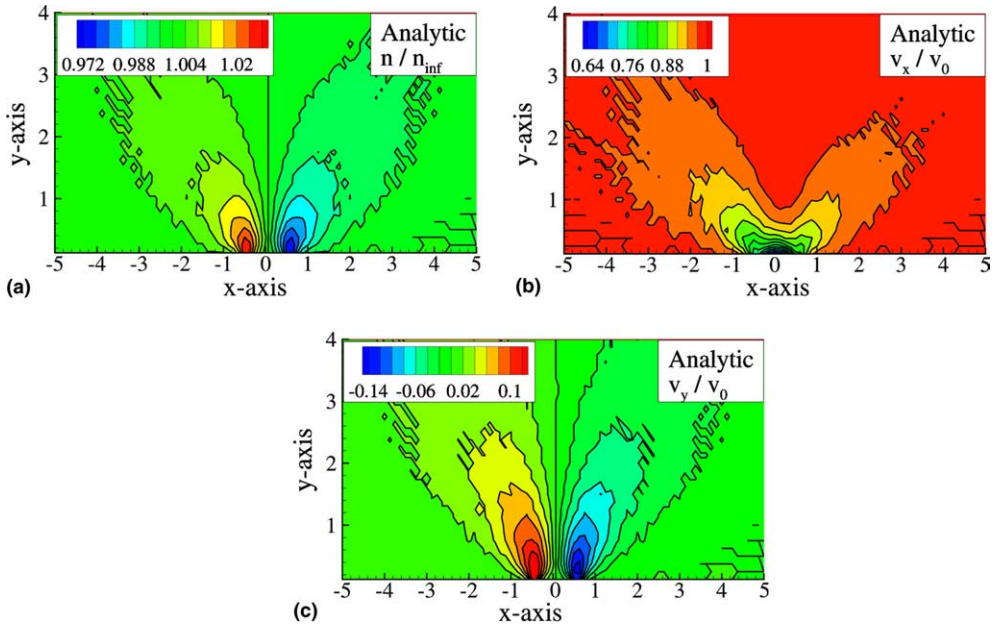

Fig. 12. Results for flow past a flat plate for the analytic model. The flow is argon with an inlet velocity of (40 m/s, 0, 0) and $Kn = \infty$. (a) The normalized density. (b) The normalized x velocity. (c) The normalized y velocity.

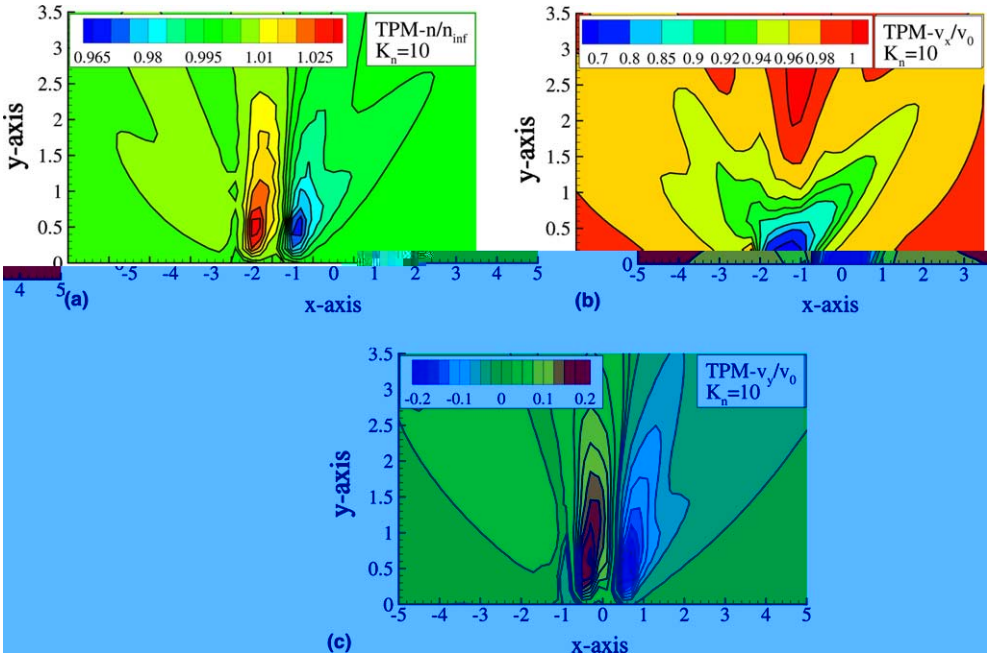

Fig. 13. Results for flow past a flat plate for the TPM model. The flow is argon with an inlet velocity of (40 m/s, 0, 0) and $Kn = 10.0$. (a) The normalized density. (b) The normalized x velocity. (c) The normalized y velocity. The effects of the relatively coarse spatial mesh are again evident.

4.3. Comparison of drag coefficients

Table 2 shows the calculated drag coefficient for the DSMC, IP, NS and TPM models. The DSMC results are given here for comparison purposes, although the statistical scatter is not small enough. The drag coefficient is somewhat sensitive to domain size and resolution – most notably angular resolution in the TPM. For a spatial mesh on which the results were converged for $Kn = 1.2$ (and mesh spacing $< \lambda/5$) we needed at least 16 rays in the Θ direction and 8 in the Φ direction (between the velocity and the z-axis), although symmetry reduces this last number to 4.

Table 2 shows the drag coefficients calculated for various Kn, in a domain at least as large as in Fig. 7(c). The free molecular flow drag coefficient for argon under these flow condition is $C_d = 4.88$. The IP and DSMC results obtained for $Kn \leqslant 1.2$ are reasonable, if a little high, since the value for $Kn = 1.2$ lies just below the free-molecular result. Comparing the TPM results to the IP and DSMC results, we see that the TPM gives a slightly higher value then the IP and DSMC for Kn of 0.05 and 0.2 and predicts a smaller drag

Table 2
Drag coefficient for flow past a flat plate for the DSMC, IP, and TPM models

Kn	TPM	IP	DSMC
0.05	1.6	1.5	1.5
0.2	3.1	3.0	3.0
0.8	3.7	3.9	4.3
1.2	4.1	4.7	4.8
10.0	4.4	–	–

coefficient for higher Kn. The TPM agrees with the analytic model but predicts a C_d less than the free molecular flow value for $Kn = 10$, which is to be expected because $f(\mathbf{a})$ (Eq. (12)) gives the correct drift velocity but underestimates the transverse momentum-flux to the surface. A collision operator which relaunches particles in the form of a drifting Maxwellian has been developed, which remedies this problem. This is more computationally intensive when implemented in a form which conserves particles, energy and momentum, but it does give a more correct transverse momentum flux, which governs drag. That is, our collision operator is designed to conserve (particles) energy and momentum. It does not exactly conserve higher moments of the distribution; in this case the rate of transport in y of x-momentum. A collision operator which also conserves these moments, so as to give a more accurate description of the stress tensor, will be the subject of future work.

5. Concluding remarks

In conclusion, we have described new techniques for solving the Boltzmann equation, including an algorithm for finding 'transition probabilities' based on the Convected Scheme, and an approach to handling surface interactions which led us to a means of implementing reflections which preserves the uniformity and other features of the flow, as appropriate. The problem of accurately simulating low Mach number high Knudsen number flows is an interesting challenge. We have presented a kinetic transport model well suited for handling low Mach number flows in high Knudsen number environments and have presented results for the model for flow past a micro-plate for various Kn. In addition, we have presented results for two other models, the IP (which is a variant of DSMC) and NS models. The results for all three models are quantitatively quite similar for density and velocity. As noted above, the exceptions in agreement took place close to the plate. There is evidence for two different contributing factors for these differences. First, the TPM collision operator underestimates the transport of linear momentum in the direction perpendicular to the momentum, which will clearly affect viscosity. What is somewhat surprising is that the operator leads to an underestimate of Cd of about 10% which persists throughout the transitional regime. The CS agrees with the two particle simulations to about the same extent that they agree with each other, with regard to the drag coefficient. Second, the diffuse reflection operators we have implemented may be contributing (in the case of diffuse reflection) to numerical diffusion parallel to the plate. There is some evidence that a finer mesh makes the density extrema occur closer to the plate, supporting this view. Future work will address improving both operators. For the particle–particle collisions, we will include higher moments of the distribution and/or place the collided particles back using a drifting Maxwellian distribution. For the particle wall diffuse reflections, we will make use of a 'map' function to better approximate the numerical $\cos\Theta$ distribution. Agreement in other regards is very good, between the TPM and the IP methods.

The TPM is perhaps 10 times slower than an NS solver, which is tackling a much smaller problem. TPM and IP take a comparable amount of time, except that the IP results presented here have to be averaged over the entire plate surface (which the TPM results do not). An IP run which gave meaningful results at each point on the surface would be an order of magnitude slower; and if it were to resolve the flow field it would be considerably slower still. A DSMC calculation which did not make use of the IP method would in turn be much slower again.

Additional directions for the work will involve extending the TPM to diatomic gas, flow past an object for any angle of attack and multiple surfaces, exploring other collision models including one which makes use of a drifting Maxwellian when relaunching particles to obtain a more exact result for the drag on surfaces and investigating how to implement irregular meshes in an efficient manner. In addition, comparisons of all methods will be made with new experimental data being generated for micro-scale airfoils being conducted in a micro-scale wind-tunnel [21].

Acknowledgements

We thank the Air Force Office of Scientific Research for supporting the research under Grant No. F49620-98-1-0433, and the referees for helpful suggestions to improve the paper.

References

[1] H.P. Tuan, H. Janssen, C. Cramers, P. Mussche, J. Lips, N. Wilson, A. Handley, Novel preconcentration technique for on-line coupling to high-speed narrow-bore capillary gas chromatography: sample enrichment by equilibrium (ab)sorption II. Coupling to a portable micro gas chromatograph 791 (1997) 187.
[2] A. Segal, T. Górecki, P. Mussche, J. Lips, J. Pawliszyn, Development of membrane extraction with a sorbent interface-micro gas chromatography system for field analysis 873 (2000) 13.
[3] I.D. Boyd, Q. Sun, Particle simulation of micro-scale gas flows, in: 39th AIAA Aerospace Science Meeting, Reno Nevada, AIAA-01-0876, January, 2001.
[4] J. Fan, C. Shen, Statistical simulation of low-speed rarefied gas flows, J. Comput. Phys. 167 (2001) 393.
[5] Q. Sun, I.D. Boyd, G.V. Candler, J. Thermophys. Heat Transfer 16 (2) (2002) 171.
[6] Q. Sun, I.D. Boyd, A direct simulation method for subsonic microscale gas flows, J. Comput. Phys. 179 (2002) 400.
[7] A. Christlieb, W.N.G. Hitchon, E.R. Keiter, A computational investigation of the effects of varying discharge geometry for an inductively coupled plasma, IEEE Trans. Plasma Sci. 28 (6) (2000) 2214.
[8] J. Feng, W.N.G. Hitchon, Self-consistent kinetic simulation of plasmas, Phys. Rev. E 61 (2000) 3160.
[9] A. Christlieb, W.N.G. Hitchon, Three-dimensional solutions of the Boltzmann equation: heat transport at long mean free paths, Phys. Rev. E 65 (2002), Article 056708.
[10] A. Christlieb, W.N.G. Hitchon, An accurate kinetic scheme for 3D solutions of the Boltzmann equation, in: Conference Paper Presented at the 23rd Symposium on Rarefied Gas Dynamics, Whistler, BC, Canada, July 20–25, 2002.
[11] R.E.P. Harvey, W.N.G. Hitchon, G.J. Parker, Plasma chemistry at long mean-free-paths, J. Appl. Phys. 75 (1994) 1940.
[12] G.J. Parker, W.N.G. Hitchon, E.R. Keiter, Transport of ions during ion implantation, Phys. Rev. E 54 (1996) 938.
[13] J.R. Askew, A general description of the lattice code wims, J. Br. Nucl. Energy Soc. 5 (1966) 564.
[14] J.L. Vujic, GTRAN2: an advanced transport theory code for advanced assembly calculations, in: Proceedings of the International Conference on Mathematical Methods and Supercomputing in Nuclear Applications, Karlsruhe, 1993.
[15] T.S. Cale, G.B. Raupp, T.H. Gandy, Free molecular-transport and deposition in long rectangular trenches, J. Appl. Phys. 68 (1990) 3645.
[16] W.N.G. Hitchon, G.J. Parker, J.E. Lawler, Accurate models of collisions in a glow discharge simulations, IEEE Trans. Plasma Sci. 22 (3) (1994) 267.
[17] W. Tan, R.J. Hoekstra, M.J. Kushner, A time dependent propagator method for long mean free path transport of neutral particles in plasma processing reactors, J. Appl. Phys. 79 (7) (1996) 3423.
[18] A.D. Boardman, W. Fawcett, S. Swain, J. Phys. Chem. 31 (1970) 1963.
[19] R.W. MacCormack, G.V. Candler, 17 (1) (1989) 135.
[20] G.A. Bird, Molecular Gas Dynamics, Clarendon press, Oxford, 1976.
[21] M.J. Martin, I.D. Boyd, L.P. Bernal, Design of a low-turbulence facility for micro-scale aerodynamics, in: Proceedings of 2002 ASME Fluids Engineering Division Summer Meeting, New York, NY, July 2002.