Three-dimensional solutions of the Boltzmann equation: Heat transport at long mean free paths

A. J. Christlieb[1,*] and W. N. G. Hitchon[2,†]
[1]*Department of Aerospace Engineering, University of Michigan, Ann Arbor, Michigan 48109-2140*
[2]*Electrical and Computer Engineering, University of Wisconsin, Madison, Wisconsin 53706*
(Received 14 November 2001; published 22 May 2002)

A kinetic transport model for arbitrary meshes is presented and the method is applied to heat transfer in a rare gas between parallel plates at different temperatures. The method uses ''propagating'' rays for tracking the transport of particles throughout the phase space in three-dimensional spatial meshes. Two collision operators are tested with the model, a simple monoenergetic operator and the Bhatnagar-Gross-Krook (BGK) model. Results are generated for several Knudsen numbers in the transition regime. The results of the kinetic simulation, which employ the BGK operator, compare favorably with those of a finite-difference solution of the Boltzmann equation using the BGK collision operator [T. Ohwada, Phys. Fluids **8**, 2153 (1996)]. In addition, the results for both collision models exhibit fair agreement with experimental data of Teagan and Springer [Phys. Fluids. **11**, 497 (1968)].

DOI: 10.1103/PhysRevE.65.056708 PACS number(s): 02.70.−c, 05.60.Cd, 47.45.−n, 44.10.+i

I. INTRODUCTION

In this paper a kinetic neutral particle transport model for long mean-free-path (LMFP) environments is discussed. The model permits efficient, nonstatistical (no random numbers are used) iterative calculation of the scattering rate of particles in each cell of a phase space mesh. The use of the scattering rate formulation is implied by the Milne version of the radiation transport equation [1]. Most implementations have been limited to rather specific applications, where they have been shown to be very accurate and computationally expensive in terms of memory requirements [2–4]. We have developed such an approach to examine self-consistent problems in long mean-free-path fluid dynamics, where force balance is achieved by means of momentum conservation during particle collisions [5,6]. We have focused on implementing the transition probabilities in a form that allows the mean free path and angular distribution of the scattered particles to vary, both in space and during the simulation (a variable angular distribution being needed to allow momentum-conserving collisions). In addition, the model has been developed for meshes constructed of arbitrary shaped elements. Most importantly, we limit the amount of information stored. The matrix of probabilities (of going from one cell to another) is very large. If N_c is the number of spatial cells on the mesh, then even the geometric information required to find all the probabilities of going from each initial cell to each final cell involves N_c^2 numbers.

As we explain in later sections, we calculate probabilities for particles going from one cell to another, based on the probabilities of going from a set of finer mesh cells to other fine mesh cells [6]. The use of fine cells is necessitated in part because the particles have a variable anisotropic angular distribution as they leave each cell. The variable angular distribution would make a single, one time, calculation of the large-cell to large-cell probability impossible. In the present work, there are N_{ray} rays leaving each initial cell, so the amount of geometric information that would have to be stored for use in finding probabilities becomes $N_c^2 N_{ray}$. Avoiding this storage problem is essential in making these calculations viable, and is a major focus of this work. We envisage problems where N_c is of the order of 10^6 and N_{ray} is typically of order 10^3.

The model presented in [6] was developed for kinetic modeling of ion implantation. The ion implantation model was developed to determine the three-dimensional (3D) profile of ions implanted in silicon and was specifically constructed for simple geometries using regular meshes. In the past [6], we used minimum stored information to construct the matrices; in this paper we find the matrix approximately, but with essentially no stored information.

In [7], a similar transport model was implemented on a 2D cylindrical mesh for the study of neutral particle transport in a low density electron cyclotron resonance plasma. The model was extended to be time dependent, and was directly compared with fluid and Monte Carlo models [7].

The LMFP transport model is applied in this paper to heat transfer, in a rare gas, between two parallel plates having a small temperature difference between them. In the limit as the radius of the plates approaches infinity, the problem becomes one-dimensional. In 1953, Chang and Uhlenbeck [8] applied the four-moment method to the problem. Gross and Ziering [9] have investigated the application of the four- and eight-moment methods to the issue of heat transfer between parallel plates. Bassanini *et al.* applied the Bhatnagar-Gross-Krook (BGK) [10] method to the problem in 1967.

In 1968 Teagan and Springer gathered experimental data on the heat flow between two plates at different temperatures for various Knudsen numbers K_n, where $K_n=\lambda/d$, λ is the mean free path, and d is a characteristic length [11]. They compared the data and the analytic models and found from 2% up to 18% variation from experimental data. Teagan and Springer presented data for both monotonic and diatomic gases, argon and nitrogen respectively. These data have been used to verify subsequent analytic and computational models.

*Electronic address : christli@engin.umich.edu
†Electronic address : hitchon@cae.wisc.edu

In 1971, Yen [12] numerically investigated the transport problem by using Nordsieck's Monte Carlo method for the evaluation of the Boltzmann collision integrals. Yen's model included one spatial axis and two velocity axes. Yen discussed results for argon at an intermediate Knudsen number and found that his results compared favorably with those of Teagan and Springer. Huang and Hwang [13] applied the method of discrete ordinates to the BGK model of a rare gas with and without internal degrees of freedom (1972). With the exception of one of the intermediate Knudsen numbers, this approach was in good agreement with the argon and N_2 data of Teagan and Springer. In order to obtain good agreement at the intermediate Knudson number, Huang and Hwang found it necessary to substantially change the accommodation coefficient of the wall. In 1985, Pazooki and Loyalka [14] applied the Hanson-Morse kinetic model to the one-dimensional problem of heat flow in a rare polyatomic gas between parallel plates, and found good agreement with the N_2 data of Teagan and Springer. Their model differed from previous models in that it included an accommodation coefficient for both internal energy and translational energy. The data of Teagan and Springer only had a bulk accommodation coefficient for N_2. Pazooki and Loyalka determined the separate accommodation coefficients by trial and error. The accommodation coefficients were determined once and used in the generation of all of their N_2 results. In 1993, Song *et al*. [15] presented a simple kinetic model for heat transfer by a rare gas in a gap. The model compared well with experimental data, including that of Teagan and Springer. Song *et al*. studied heat transport in a rarefied gas between rough surfaces and smooth surfaces. The model compared favorably with their own experimental data. In 1996, Ohwada [16] compared his nonlinear Boltzmann equation solver to the argon data gathered by Teagan and Springer. The nonlinear Boltzmann equation solver had minor discrepancies with the experimental data (no worse than other computational models), even as the Knudsen number was increased to the point where nonlinear effects become small. The nonlinear Boltzmann equation solver predicted asymmetries in the density profile, which were not seen in the experimental data. He also found that the collision operator made little difference in the solution under the condition of a small temperature difference between the two plates. Ohwada felt that the accommodation coefficient of the experiment might be incorrect for conditions other than free molecular flow. However, varying the accommodation coefficient of the plates did not have enough of an effect on the solution from the nonlinear Boltzmann equation solver to explain the discrepancy with the experimental data.

More recently, higher-dimensional computational methods, aimed at predicting flows in microelectromechanical systems (MEMS), have used the one-dimensional problem of heat transfer between parallel plates as a test bed. In 2001, Boyd and Sun [17] presented a new direct simulation Monte Carlo (DSMC) method that incorporates a fluid model to preserve global information to reduce statistical noise for low flow velocities in a long mean-free-path environment. The model has two spatial dimensions and three velocity dimensions. Their results agreed well with those of Teagan and Springer for small Knudsen numbers and in the limit of free molecular flow. Pan *et al*. [18] presented a ''molecular block'' DSMC model, which reduces statistical noise by grouping large numbers of particles into ''blocks'' while retaining the mean free path of the individual molecules. They obtained reasonable results for various different gas flows, including high Knudsen number Couette flows.

The present LMFP model describes steady state transport of particles in a complex geometry and with irregular meshes. The 6D model uses three spatial variables and two directional variables and energy to represent velocity space. This propagator (Green's function) method allows large ''steps,'' which reduce numerical diffusion [19,20]. In addition, this method is nonstatistical in nature. Partly because of this, it can handle collisions between particles in the gas phase with greater precision than statistical (or ''particle'') methods. This is an advantage for processes that involve interactions between particles—such as force or heat balance and transfer within the gas. The LMFP model conserves mass and energy. Momentum is conserved by the introduction of an angular distribution function described in [20]. We apply the LMFP model to heat transport between parallel plates and compare our results with the argon data of Teagan and Springer [11] and with numerical results of Ohwada [16]. We employ two different collision operators, a monoenergetic collision operator and the BGK model.

In Sec. II the experiment of Ref. [11] is described. In Sec. III the transport model is described. In Sec. IV, results and verification of the transport model are presented.

II. THE PHYSICAL MODEL

A. Experiment

The experiment of Teagan and Springer [11] is outlined here. The apparatus consisted of three parallel plates, 25.4 cm in diameter. The upper and lower plates were water cooled aluminum disks with argon between them. The middle plate consisted of two aluminum plates with a heating element placed between them. The ''hot'' plate (at 368 K) and either ''cold'' plate (at 288 K) were 0.13 cm apart for heat transfer measurements and 2.54 cm apart for the density profiles. Density and energy flux measurements were made for various Knudsen numbers. The accommodation factor (the percentage of particles that undergo a diffuse interaction with the surface) of the aluminum was experimentally determined for argon at densities near the free molecular flow regime to be 82.6%.

B. Simulation domain

Due to the symmetry of the experiment, the simulation only includes the lower cold plate and the hot plate (see Fig. 1). The accommodation factor was set to the experimental value.

The current form of the code is 6D, however this problem is 3D (one position variable and two velocity variables). Therefore on the vertical boundaries of the simulation domain, the accommodation factor was set to zero so that particles were spectrally reflected off the side walls. This made

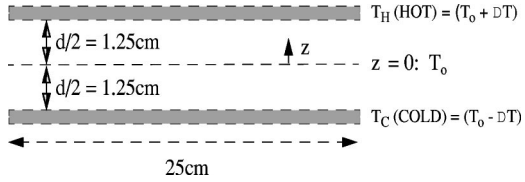

FIG. 1. The simulation region. The upper plate is the hot plate set at a temperature of 368 K and the lower plate is the cold plate set at a temperature of 288 K.

the simulation symmetric in the xy plane and thereby one-dimensional. The simulations were carried out on a rectangular domain by using meshes constructed either of triangular elements or rectangular elements. Figure 2 shows an example of the types of meshes employed in this work. It is equally easy to run the code on simulation domains constructed of triangular and/or rectangular elements.

III. LONG MEAN-FREE-PATH TRANSPORT MODEL

In this work we consider rarefied gas flows, although the basic method applies at any mean free path. In what follows,

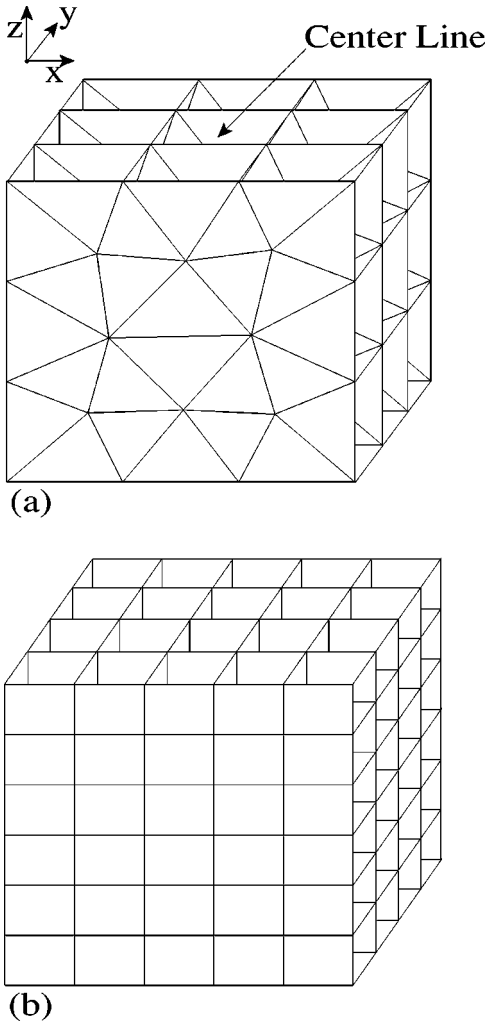

FIG. 2. Examples of the two different types of spatial meshes that were used in the simulations.

we introduce assumptions that apply to systems where the mean free path is comparable to or greater than the system size.

Our approach to the transport problem is to solve the steady-state Boltzmann equation

$$\vec{v}\cdot\vec{\nabla}_r f(\vec{r},\vec{v})+\frac{\vec{F}}{m}\cdot\vec{\nabla}_v f(\vec{r},\vec{v})=\left.\frac{\delta f}{\delta t}\right|_{collision}. \tag{1}$$

In solving Eq. (1) we employ a propagator based method, which we refer to as the transition probability matrix (TPM) method. This section describes new aspects of the TPM. We start with an overview, then present descriptions of the ''ballistic operator'' and collision operator, and end with a discussion of the numerical approach.

A. Overview

We present a model for the transport of gases when the mean free path of the particles, λ, is of the order of the characteristic length of the system L. We use a 3D arbitrary spatial mesh and a 3D velocity space mesh comprising a single energy mesh in combination with a 2D (Φ,Θ) directional mesh. The method employs a momentum-conserving collision operator [6,20]. Energy and particle conservation are strictly enforced.

To solve Eq. 1, it is only necessary to compute the collision rates of particles because it is possible to compute any other information from the collision rate. For example, the density $n(c)$ is

$$n(c)=\sum_E \frac{R(c,E)\lambda(c,E)}{|v(E)|\gamma(c)}, \tag{2}$$

where $n(c)$ is the density in cell c of the spatial mesh, $R(c,E)$ is the collision rate of particles in cell c at energy E,$\lambda(c,E)$ is the mean free path, $|v(E)|$ is the magnitude of the velocity, and $\gamma(c)$ is the volume of cell c.

To compute $R(c,E)$, the TPM divides particle behavior into two distinct phases. The first phase addresses ballistic or collisionless transport of particles. The second involves a collision operator. These operations can be performed efficiently using one-step transition probability matrices. The first transition probability matrix is used to compute the number of particles per second that collide in cell c at energy E',

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

where $R(c,E')$ is the number rate of particles that collide in cell c at energy E',$R(c',E')$ is the number rate of particles that collided in cell c' and were redistributed with energy E' in the previous iteration, and $T_{bal}(c,E':c',E')$ is the probability that a particle having started in cell c' at energy E' will have its next collision in cell c, where the sum is over all mesh cells c' at energy E'.

The second TPM is used to redistribute the particles after a collision,

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

where $T_{col}(E:E')$ is the probability that a particle, having collided in cell c at energy E' will be redistributed with energy E (in the same spatial cell c). The directional information used in the redistribution may either be directly incorporated into the TPM, $T_{bal}(c,E':c',E')$, or may be handled by assuming a form for the particle distribution function, $f(\vec{x},\vec{v})$, in cell c [6,20].

In the following sections we describe how the TPMs are set up for irregular meshes without explicitly constructing large matrices, T_{bal} or T_{col}.

B. Ballistic operator: Constructing $R(c,E')$

This section addresses the ballistic transition probability matrix T_{bal} and compares the current T_{bal} to earlier versions [6]. Some of the approximations we introduce in T_{bal} may not be appropriate at short mean free paths.

The TPM is never explicitly constructed. Instead, the TPM employs propagating "rays," $r_{i,j}$, each having a range of polar angles,

$$r_{i,j}=\left(\Phi_i\pm\frac{d\Phi_i}{2},\Theta_j\pm\frac{d\Theta_j}{2}\right). \tag{5}$$

Ballistic transport is carried out on a fixed mesh constructed of elements c'. Propagation is performed by allowing particles to move along rays from c' encountering other cells of the fixed mesh in order of increasing radius. Propagation is completed when the rays of all c' have deposited all of their particles back in the simulation domain.

The number of particles originating in cell c' at energy E' that have their next collision in cell c at E' is

$$N_c(c,E':c',E')=\sum_i \sum_j N_{r_{i,j}} f(r_{i,j})P_{i,j}(c,E'), \tag{6}$$

where $r_{i,j}$ is a ray from $c',N_c(c,E':c',E')$ is the number of particles that start at (c',E') and that collide at (c,E'), $N_{r_{i,j}}$ is the number of particles left in $r_{i,j}$ at cell c, $f(r_{i,j})$ is the fractional overlap of $r_{i,j}$ with cell c, $P_{i,j}(c,E')$ is the probability that a particle that is in $r_{i,j}$, and is crossing cell c at energy E' will undergo a collision at a distance d, the depth of cell c. The sum in Eq. (6) is over all $r_{i,j}$ that overlap cell c. $P_{i,j}(c,E')$ is given by

$$P_{i,j}(c,E')=(1-e^{-d/\lambda(c,E')}), \tag{7}$$

$\lambda(c,E')$ is the mean free path in cell c at E'. $d=D(c,r_{i,j})$ is the average distance a particle travels when passing through c along $r_{i,j}$.

The number of particles in a ray is decreased by the amount indicated in Eq. (6), as those particles are scattered in each cell they pass through. $R(c,E')$ can now be computed by summing $N_c(c,E':c',E')$ over all cells c' of the fixed mesh. Particles having scattered in a cell c', during the ballistic step, are placed in the rays of c' in such a way as to conserve momentum, as discussed in the following section.

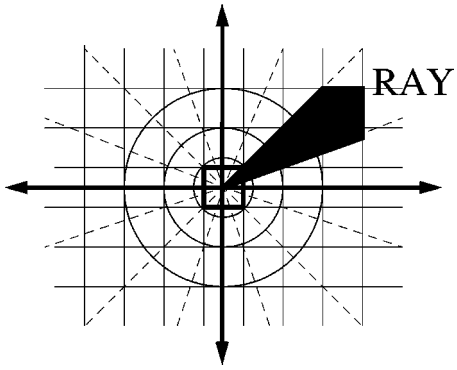

FIG. 3. Depiction of a single propagating ray as it encounters mesh elements of a regular mesh.

In the past [6], the geometric information needed to calculate the probabilities "exactly" on a Cartesian mesh was computed and stored in a table. The Cartesian mesh was referenced by an ordered triple of integers, (i,j,k). Due to the symmetry of the mesh, the geometrical information for an initial cell c' with other final cells was the same for all initial cells c', so that it was only stored once for c' at the origin, $i=j=k=0$. Geometric information was stored explicitly, so that propagation could be possible one ray at a time (see Fig. 3). Mesh information was stored, for each ray, based on the distance from c'. Particles were distributed to the mesh as the cells were encountered along the ray. The probability of a particle having a collision in cell c at energy E' was approximated from Eq. (7) as

$$P_{i,j}(c,E')\simeq\frac{D(c,r_{i,j})}{\lambda(c,E')}, \tag{8}$$

provided $D(c,r_{i,j})\ll\lambda(c,E')$. $D(c,r_{i,j})$ and $f(r_{i,j})$ were determined by numerical integration.

The convected scheme (CS) [21] finds probabilities accurately and straightforwardly. Moving cells (MCs) are launched from cells, moved along ray-centers for time Δt, and colliding particles put in cells using CS overlap rules; another step of the MC taken, etc. At walls, MCs emerge in continuous sheets [21]. Striking walls, MCs are mapped back to give proper coverage, especially in corners (e.g., by dividing the MC into many small cells).

1. The coarse mesh

If the TPM were directly implemented on an arbitrary mesh, it would be necessary to construct the propagating rays for each initial mesh element of the mesh. Storing the quantities necessary to construct the TPM would be highly memory intensive. To avoid this, a fine regular mesh underlying the arbitrary coarse mesh is introduced [21], along with a map between the two meshes. A fine regular mesh element/cell is said to overlap a coarse mesh element if the center of the fine mesh cell lies inside the coarse mesh element. A pointer to the coarse mesh element is stored in every fine regular mesh element that overlaps it. Each coarse mesh el-

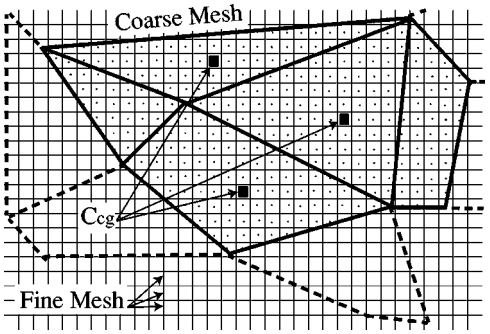

FIG. 4. A 2D example of the manner in which arbitrary coarse mesh elements are redefined in terms of fine regular mesh elements.

ement has a minimum of one fine mesh element in it. The coarse mesh is used to cluster fine regular mesh elements into cohesive units that share the same characteristics. Clustering of fine mesh elements effectively redefines the coarse mesh elements of the arbitrary mesh (see Fig. 4). After colliding in the coarse mesh element c (that is, in one of the fine mesh cells of c), particles are redistributed (i.e., relaunched) from the fine regular mesh element c_{cg} closest to the center of gravity of the cluster of fine cells representing the coarse cell c. Note that there is one c_{cg} per c, so that the number of distributing nodes is still equal to the number of coarse mesh elements.

Similarly, the boundary is represented by clusters of 2D fine regular mesh elements that lie on the boundary; these fine mesh elements are designated by c_b. The wall cell closest to the center of gravity of the cluster of wall cells is designated the distributing wall cell c_{bcg}. (It will be important to ensure that there is indeed a fine mesh cell c_{cg} or c_{bcg} with its center close to the actual center of gravity.)

The fine regular mesh introduces symmetry into the system and can be referenced by an ordered triple of integers, (i,j,k). During the ballistic move we distribute particles from each coarse element to all other coarse elements—but scattering from each coarse element is done from a fine element, c_{cg} or c_{bcg}. The distribution from c_{cg} to the coarse elements c is done by first distributing particles to each fine mesh element. Particles placed in a fine mesh element are never stored in that element, but are placed directly into the coarse mesh element c that the fine mesh overlaps. The mean free path of the particles used in calculating the number of particles that collide in a fine regular mesh element is the mean free path $\lambda(c,E')$ of the cluster of fine mesh elements representing cell c.

2. Overlap of rays with cells

Unlike the ion transport model [6], the information needed for the probabilities is not explicitly stored, but rather is computed as needed. The next cell being distributed to from cell c_{cg} is found using an appropriate ordered triple as an integer offset. The ordered triple refers to $\bar{c}$, the next closest fine regular mesh element to the origin. The fractional overlap of $\bar{c}$ and the ray, and the average distance $\bar{c}$ extending along the ray are approximated once. The propagation of the particles is then performed by looping over all the "centers of gravity" c_{cg}, distributing particles to fine mesh elements designated by

$$c_{dist}=(i_{cg}+\bar{i},j_{cg}+\bar{j},k_{cg}+\bar{k}), \tag{9}$$

where (i_{cg},j_{cg},k_{cg}) is the ordered triple of the distributing node c_{cg} and $(\bar{i},\bar{j},\bar{k})$ is the integer offset belonging to $\bar{c}$. The generation of the integer offsets is described in the Appendix.

The fractional overlap is approximated as

$$f(r_{i,j})=\begin{cases}\dfrac{\Delta\Omega_{dist}}{\Omega_{r_{i,j}}}, & r^2\Omega_{r_{i,j}}<3\Delta A_{finemesh},\\[2ex] \dfrac{\Omega_{dist}}{\Omega_{r_{i,j}}}, & r^2\Omega_{r_{i,j}}\geqslant 3\Delta A_{finemesh},\end{cases} \tag{10}$$

where $\Omega_{r_{i,j}}$ is the solid angle subtended by the ray $r_{i,j}$ with the origin of the ray located at c_{cg}, $\Delta\Omega_{dist}$ is the portion of solid angle $\Omega_{r_{i,j}}$ overlapped by cell c_{dist}, Ω_{dist} is the solid angle subtended by the faces of c_{dist} projected in the direction of $r_{i,j}$, r is the distance between c_{cg} and c_{dist}, $\Delta A_{finemesh}$ is the area of a face of c_{dist}, $r^2\Omega_{r_{i,j}}$ is the area of ray $r_{i,j}$ at cell c_{dist}, and $3\Delta A_{finemesh}$ is an overestimate for the area of the three possible exposed faces of c_{dist}. "Exposed to the ray" refers to the faces of the cell the particles enter c_{dist} through, when traveling along $r_{i,j}$. At large distances, the approximation of the fractional overlap is based on whether or not the center of c_{dist} lies on the interior of a given ray $r_{i,j}$.

The average distance a cell extends along a ray is computed using

$$D(c,r_{i,j})=\frac{V_{dist}}{\sum_k J_{r_{i,j}}(A_{finemesh}(k))}, \tag{11}$$

where V_{dist} is the volume of cell c_{dist}, $J_{r_{i,j}}(A_{finemesh}(k))$ is the area of the face of the kth face of c_{dist} projected in the direction of ray $r_{i,j}$, and the sum over k represents the sum over the faces of c_{dist} "visible" to the distributing node c_{cg} or c_{bcg} when looking in the direction of $r_{i,j}$.

3. Overlap of rays with wall cells

When a propagating ray strikes a cell c_b on the boundary, the fractional overlap of the ray with the wall cell is computed as above. This fractional overlap is the fraction of particles in the ray that has a collision with the wall cell c_b. After a ray strikes a wall cell, the area of the ray needs to be adjusted, so that the fractional overlap of the ray with other interior cells c_{dist} at greater radii than c_b is not underestimated, because the portion of the ray striking c_b is no longer propagating particles (see Fig. 5). For a simply connected region, without inclusions, the solid angle of the ray is reduced by a fractional amount $f(d\Omega)$. It is necessary to store the $\Omega_{r_{i,j}}$ of each distributing c_{cg} since the rays from each c_{cg} encounter the boundary cells c_b in a different order, hence

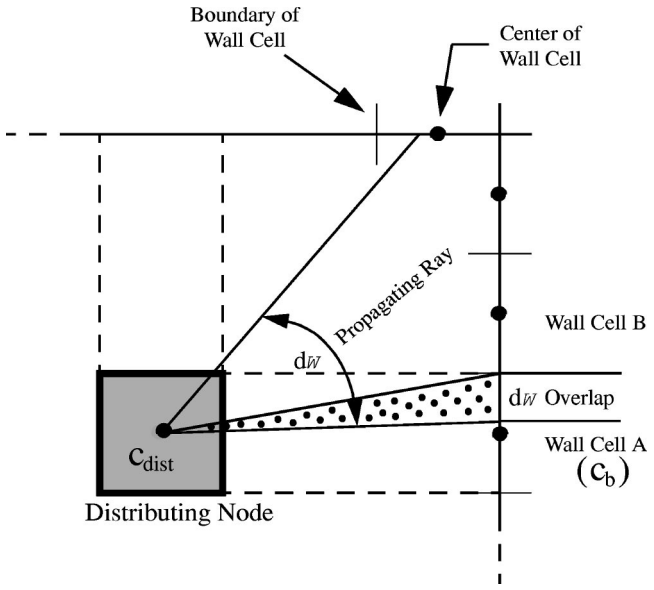

FIG. 5. Depiction of the necessity of reducing $d\Omega$. The speckled portion of the ray has the shortest distance to the wall so that this portion of the ray strikes the wall first. After the speckled portion of the ray has struck the wall, the solid angle $d\Omega$ of the ray is reduced by the amount of the speckled portion.

the $\Omega_{r_{i,j}}$ for each c_{cg} will require different fractional adjustments. An overlap rule can be used, see Ref. [21], to find the fraction of the wall cell that is in the ray or vice versa. The simplest such rule might be to find the range in Φ and Θ of the ray, from $(\Phi/\Theta)^{\text{ray}}_{min}$ to $(\Phi/\Theta)^{\text{ray}}_{max}$; and similarly $(\Phi/\Theta)^{\text{cell}}_{min}$ and $(\Phi/\Theta)^{\text{cell}}_{max}$. $\Phi^{\text{cell}}_{min/max}$ and $\Theta^{\text{cell}}_{min/max}$ are found from the coordinates of the corners of the facet. The area of a ''rectangle'' with these values defining its faces is found and compared to the actual area of the facet. The area of the rectangle is greater by a ratio f_{area}. The dimensions in Θ and Φ of the cell are temporarily reduced by a factor $\sqrt{f_{area}}$, keeping the central value the same, thus redefining $\Phi^{\text{cell}}_{min/max}$ and $\Theta^{\text{cell}}_{min/max}$ of the face. After ''correcting'' $\Phi^{\text{cell}}_{min/max}$, we find which of the minima is greater and which of the maxima is smaller, to find the range of overlap in Φ. We repeat this for Θ. The solid angle of the overlap is estimated using these two overlaps.

In a domain with inclusions, interiors of rays may strike the boundary before the edge of the rays strike the boundary, thus making the ray a complicated structure. Subdividing the ray into subrays can handle complicated ray structures, but this is memory intensive. An alternative to subdividing the ray is to subdivide the simulation region into simply connected regions without inclusions, known as convex hulls. The boundaries between convex hulls would be transparent walls. Phase space information about particles striking the boundaries between convex hulls would be stored until the completion of the current simulation step. At the start of the next step, these particles would be relaunched into the next convex hull. Subdividing into convex hulls eliminates the necessity for complicated ray subdivision routines and only requires a minimal design upgrade to the propagating structure presented here. For the current simulation of heat transfer between parallel plates, we only need a simple method of adjusting $\Omega_{r_{i,j}}$ when a ray strikes a wall. $f(d\Omega)$ is used to reduce $\Omega_{r_{i,j}}$. Because $f(d\Omega)$ is a rough approximation, $\Omega_{r_{i,j}}$ may not be equal to zero at the end of a step. If $\Omega_{r_{i,j}}$ is less than zero after adjustment by $f(d\Omega)$ (during the step), all remaining particles in $r_{i,j}$ are placed in the current wall cell and $\Omega_{r_{i,j}}$ is set to zero. If $\Omega_{r_{i,j}}$ of a ray is greater than zero at the end of the step, the small number of remaining particles is simply distributed to the boundary of the simulation domain, thereby conserving mass during the ballistic operation.

4. Mean free path

In the current model, $\lambda(c,E')$ can be determined for each spatial cell from the cross-section and the local particle distribution. However, we have made the simplifying assumptions that the gas is monatomic and has a constant collision frequency, ν, and λ is

$$\lambda(E') = \frac{|v(E')|}{\nu}. \tag{12}$$

ν is

$$\nu = \frac{|v(\tilde{E})|}{\lambda(\tilde{E})}, \tag{13}$$

where $\tilde{E}$ is the average energy. $\tilde{E} = \frac{3}{2} k_B (T_{hot} + T_{cold})/2$ where k_B is Boltzmann's constant, T is temperature and the subscripts hot and cold refer to the hot and cold plates. Choosing $\lambda(\tilde{E})$ to be equal to $K_n L$, where K_n is the Knudsen number and L is the characteristic length, yields

$$\lambda = \left(\frac{(K_n L)^2 T}{(T_{hot} + T_{cold})/2} \right)^{1/2}, \tag{14}$$

where T is the ''temperature'' of the species colliding in cell c.

In the next section we discuss the collision operators and implementation of boundary conditions.

C. Collision operator: Constructing $R(c,E)$

In this section, a discussion of the collision operator T_{col} is given. T_{col} redistributes particles on the mesh and along the boundary in energy while conserving momentum in particle-particle collisions. Application of T_{col} to $R(c,E')$ (constructed during the ballistic move) gives $R(c,E)$. We now give an overview of T_{col}, followed by the implementation of energy conservation for particles, momentum conservation for particles on the interior of the mesh, and finally, implementation of the boundary conditions.

In this work the collisions have been described using two different models. In the first, particles that undergo a collision in cell c are put back on the phase space mesh at the average energy of the particles that collided in the cell during the current simulation step. Particles that strike the wall and undergo a diffuse interaction are placed back on the phase space mesh at the energy of the wall. Momentum is con-

served using the angular distribution $f(\theta)$. The angular distribution $f(\Theta)$ provides the correct weighting for particles that undergo a diffuse interaction with the wall.

In the second collision operator the model used for the interior of the simulation domain is the BGK model, which is

$$\left.\frac{\delta f}{\delta t}\right|_{collision} = \nu n(c) f_0(E) f(\theta) - \nu f(\vec{r},\vec{v}), \tag{15}$$

and the distribution of particles coming off the wall is given by

$$f|_{wall} = \alpha\zeta\sqrt{E} f_0(E) f(\Theta) + (1-\alpha) f(\vec{r},\vec{v}), \tag{16}$$

where $f_0(E)$ is the Maxwellian distribution function, α of Eq. (16) is the bulk wall accommodation factor, and ζ is a normalization factor. $f(\theta)$ and $f(\Theta)$ carry directional information as described below. The additional factor of $\sqrt{E}$ in Eq. (16), is to account for the fact that the flux of particles coming off the wall at energy E should be the same as if there were an infinite volume behind the wall with a Maxwellian distribution of density—the flux at energy E is that density weighted with velocity $\sqrt{2E/m}\ f(\Theta)$.

Both the monoenergetic T_{col} and the T_{col} defined using Eq. (15) conserve energy and momentum, locally on average, i.e., the average of the energy and momentum of the particles in mesh cell c after redistribution by collisions is forced to equal the average energy and momentum of the particles which collide in cell c before they are redistributed. We find the energy and momentum for cell c in four steps: (1) Determine the average energy E_{av} of the particles that collide in cell c; (2) determine the average momentum of the particles that collide in cell c; (3) redistribute the particles in energy, within cell c, based on E_{av}; (4) assign an angular distribution to the particles of each energy bin, E_i, of the mesh, so that the momentum is conserved for particles that collide in the cell c. For particles that collide with a boundary, the particles either spectrally reflect or undergo a diffuse interaction with the boundary.

The redistribution of particles in energy, on the interior of the mesh, is done according to

$$\tilde{F}_i = \frac{F_i}{\sum_i F_i}, \tag{17}$$

where $\tilde{F}_i$ is the fraction of particles placed in energy bin E_i and F_i is given for the monoenergetic T_{col} by

$$F_i = \begin{cases} 0, & |\nu| > \Delta E_i \\ (E_{i+1} - E_{av}), & |\nu| < \Delta E_i \quad \text{and} \quad \nu < \Delta E_i \\ (E_{av} - E_{i-1}), & |\nu| < \Delta E_i \quad \text{and} \quad \nu > \Delta E_i, \end{cases} \tag{18}$$

where $\nu = (E_i - E_{av})$. For the BGK T_{col}, F_i is

$$F_i = \sqrt{E_i} e^{-E_i/E_{av}} \Delta E_i, \tag{19}$$

where E_i is the center of the energy bin, E_{av} is as above, and ΔE_i is the width of the energy bin E_i. E_{av} is given by

$$E_{av} = \frac{\sum_i N_c(E_i) E_i}{\sum_i N_c(E_i)}, \tag{20}$$

which is computed before F_i. Because the average energy computed using $\tilde{E}_{av} = \Sigma_i \tilde{F}_i E_i$ may not equal E_{av}, we check that E_{av} equals $\tilde{E}_{av}$ and adjust $\tilde{F}_i$, if necessary.

Conservation of momentum is achieved by choosing an angular distribution for the particles that collided in cell c, which yields the same mean velocity that they had before they collided. Since details of the differential collision cross section are usually not known, we choose a simple angular distribution. The elastic collision operator developed in [20] is employed here, with adjustments, to ensure momentum conservation. Particles are relaunched with an angular distribution

$$f(\theta) = (1 + \alpha_v \cos\theta), \tag{21}$$

where θ is measured from the direction of the mean velocity and where α_v is chosen to obtain the required velocity. Reference [20] 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.

Boundary conditions are imposed in a straightforward manner. For example, if a given surface has a porosity of $x\%$, $x\%$ of the particles striking a given c_b are subtracted from the total number of particles striking c_b. However, very little is known about how particles interact with a surface. Reflecting particles are believed to undergo diffuse or spectral reflection. Particles having a diffuse reflection off a surface come away from the surface isotropically and acclimated to the temperature of the wall. The angular distribution function that leads to uniform density for particles coming off a surface is

$$f(\Theta) = \cos\Theta, \tag{22}$$

where Θ is the angle between the outgoing particle's direction and the normal to the surface [22]. In addition, particles undergoing a diffuse interaction with the wall are redistributed in energy according to Eq. (17), where E_{av} in Eq. (18) or (19) is the mean energy of the particles at the temperature of the wall. When Eq. (19) is used for the wall, an additional factor of $\sqrt{E_i}$ must be included to make it consistent with Eq. (16). Spectrally reflected particles are perfectly reflected, i.e., the angle of incidence equals the angle of reflection. Particles that are spectrally reflected do not undergo a change in energy after a collision with a surface. Spectrally reflected particles are placed in the outgoing ray associated with the angle of reflection and are binned with the same energy the particles had prior to collision with the surface.

D. Simulation accuracy and speed

In the current implementation of this kinetic method, speed, accuracy, and memory usage are somewhat independent. In this section we give a qualitative discussion of the numerical aspects of this method.

Memory usage is primarily tied to storage of particles. In order to accurately model the distribution function, each coarse mesh element of physical space must have a set of energy bins, and each energy bin must have a set of directional (Φ,Θ) ray bins. In addition, a full set of energy bins as well as a partial set of (Φ,Θ) bins is necessary for the walls. Storing this information leads to the majority of the storage overhead associated with the current form of the method.

The number of ray bins associated with the mesh does not have much effect on the speed of the method. The overlap of all the offset fine mesh elements with the propagating rays is computed once at the beginning of the step and is used for distribution from all c_{cg} and c_{bcg} to the fine mesh elements in order of closeness. The time associated with each step is primarily due to the number of fine and coarse mesh elements, because each coarse element will usually ''hit'' each fine element once, regardless of the number of rays employed. In addition, the time it takes to compute the overlap is very small compared to the time it takes to loop over and distribute to all of the fine mesh elements from the coarse mesh elements. If the number of fine mesh elements goes up by a factor of 10, the time it takes per step also goes up by approximately a factor of 10. Increasing the number of coarse mesh elements has the same effect.

Errors are introduced into the computation through the approximation to the fractional overlap in Eq. (10). In addition, errors are introduced with the assumptions that particles are uniformly distributed in a given ray, and that particles having collided in a given coarse cell can be redistributed from c_{cg} of the coarse cell. Since the time it takes to compute the overlap of the offset fine mesh element to a distributing ray is very small compared with the time it takes to distribute particles, one effective way to improve the accuracy of the method is to generate better approximations than Eq. (10). Also, if one can assume a constant collision frequency or mean free path, one can eliminate the propagating rays from the method by using Eq. (21) to determine the number of particles that should collide in a given fine mesh element instead of using Eq. (21) to determine the number of particles in each ray. However, in the event that the differential cross section is well understood, propagating rays can offer an advantage over Eq. (21), in that we can exactly specify the angular distribution of particles. The error in approximation of the distribution from c_{cg} may be reduced either by increasing the number of coarse mesh elements or by determining an analytic model to compensate for the error in the offset. Increasing the number of coarse mesh cells, by decreasing the maximum radius of a coarse cell, will increase the accuracy by reducing the possible displacement a particle undergoes when relaunched after a collision, i.e., the distance between where the particle actually collided and the point of redistribution (c_{cg}) will be reduced if the maximum radius of the coarse cell is reduced.

The TPM is a 6D simulation with an element of phase space represented by the ordered set (x,y,z,E,Φ,Θ). As discussed earlier, the heat transport problem of Sec. II is used in the verification of the TPM. In the results presented in the following section, the coarse mesh is $7\times7\times21$, i.e., there are seven x subdivisions, seven y subdivisions, and 21 z subdivisions. In addition there are $2[7\times7+2(7\times21)]$ distributing wall bins. For each coarse distributing bin there are 21 energy bins and 14 Φ bins and 24 Θ bins, bringing the total number of bins to

$$N_{bins}=(7\times7\times21)(21\times14\times24)+\{2[7\times7+2(7\times21)]\}\times(21\times14\times24),$$

which is $1.2\approx\times10^7$ bins. If this computation was done without the rays, this number would be reduced to around 30 000 bins. There are about 10 000 fine mesh elements in the current simulation. The simulation was carried out on a 400 MHz Pentium II, processor with 512 MB of RAM (random access memory). The number of z,E, Φ, and Θ bins represents a lower limit to remove any mesh dependencies when Eqs. (15) and (16) were used as the collision operator.

The simulation is considered converged when the sum over the mesh of the difference in the density of two consecutive steps was less than 0.1%. The monoenergetic case exhibited faster convergence than its BGK counterpart. The monoenergetic model took around 2 min per step and the long mean-free-path case converged in about eight steps, whereas the short mean-free-path case converged in about 40 steps. In the BGK model, a step of the simulation was ≈4 min and the long mean-free-path simulations converged in about 30 steps while the short mean-free-path simulations took upwards of 80 steps to converge. To converge to 1% accuracy took far less time, however.

Both simulations are started by placing the particles uniformly on the mesh. The particles are assumed isotropic and are distributed according to Eq. (17), with E_{av} set to the average energy of the particles at the temperature of the cold plate. This is a poor guess at the final distribution of the particles. The long time to converge can be partially attributed to the poor initial condition. The additional time required for the BGK model to converge is related to the time it takes to heat the significant number of cold particles, which are colder than the cold plate [added everywhere to the simulation domain by the use of Eq. (17) in initializing the simulation].

In all results presented, the density profiles of the TPM exhibited peaking in the corners of the simulation domain. Peaking can be reduced by increasing the number of wall cells, which decreases the error introduced by using an ''on the fly'' approximate overlap rule for the rays with the wall cells. Since the geometry is the same as seen by each ray that travels in the same direction, it is computationally feasible to calculate overlaps accurately for each such ray. That information is used a very large number of times, so the cost of computing it is small compared to the computational load of

the overall task. Such accurate calculations are done by dividing each surface of a cell into a large number of smaller surfaces. This subdivision of the facets increases the accuracy. However, as the radius of the ray increases, the error in computing the fractional overlap, in Eq. (10), decreases like $1/r^2$. At some point the increase in accuracy is not warranted, since the errors obtained using small facets at small radii have already limited the overall accuracy of the method. When this point is reached, one can revert to using the entire face in computing the fractional overlaps. The increased accuracy greatly reduces peaking in the corners, from $\pm 5.0\%$ to less than $\pm 1.0\%$. More accurate overlap rules are discussed in [5,6], but even this rule works quite well in the examples given here.

Finite-difference-type approximations tend to underspecify some details of physical problems—and the specification of the surface we employ is an example of this. In the corners, not only are adjacent source points closer together than points on a flat surface, but there is some ambiguity about the shape of the surface in the corner between the points. If we imagine the corner between the source points as being filled with a single facet at 45° to the adjacent faces, then the angular distribution leaving the corner would be directed significantly more away from the corner than if we use two facets at 90°. To allow for a more realistic corner shape, for the given resolution provided by the source points, we have considered tilting the normals of the facets adjacent to the corner slightly away from the corner. We chose a value of 22.5° as a physically based average value. This ''correction'' does reduce peaking in the corners, but it does not appear to eliminate it.

The distributing node c_{cg} of the coarse mesh needs to be close to the actual center of gravity of the coarse mesh element. After a collision, particles are relaunched from c_{cg}. For the uniform mesh described above, an offset in c_{cg}, from the center of gravity, results in a preferred direction of motion, i.e., particles that collide in a given coarse cell will have an additional offset in their location corresponding to the offset in c_{cg}. This additional offset resembles an imposed drift velocity. This drift may result in particle distributions being skewed in the direction of the offset. On a truly arbitrary mesh, constructed of triangular elements, offsets of this nature will cancel each other out.

IV. RESULTS AND VERIFICATION

In this section we discuss the results and verification of the TPM. We begin with an overview of the simulations that were run for the verification process. Next, the results of the heat transport problem are presented and additional verification of the TPM is provided by a comparison of the current results with the existing experimental data and previous numerical work.

A. Verification

Verification of the TPM consisted of five simulations, with the fifth being the comparison between the model and the experimental data gathered by Teagan and Springer [11]

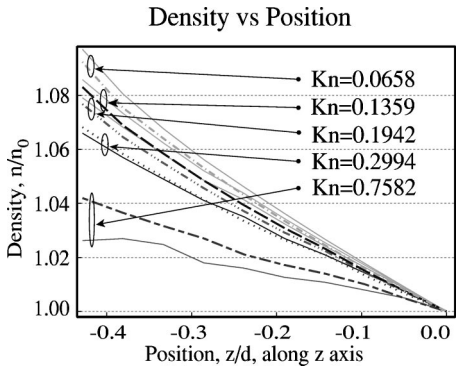

FIG. 6. Density vs position for both the BGK and monoenergetic collision operator. Density is normalized using the density at the center, n_0, between the two plates. Position is normalized using the distance between the plates. The BGK model is represented by dashed lines, while the monoenergetic model is represented with solid lines.

for the heat flow between two parallel plates at various Knudsen numbers. The first four sets of simulations of the verification process addressed the issue of the uniformity of the density profile of neutral particles generated by the TPM under the following conditions:

(1) The temperature of the simulation domain is a single value, particles undergo isotropic collisions, and the Knudsen number of the simulation region is assumed small.

(2) As in (1) except that the Knudsen number of the simulation region is assumed large.

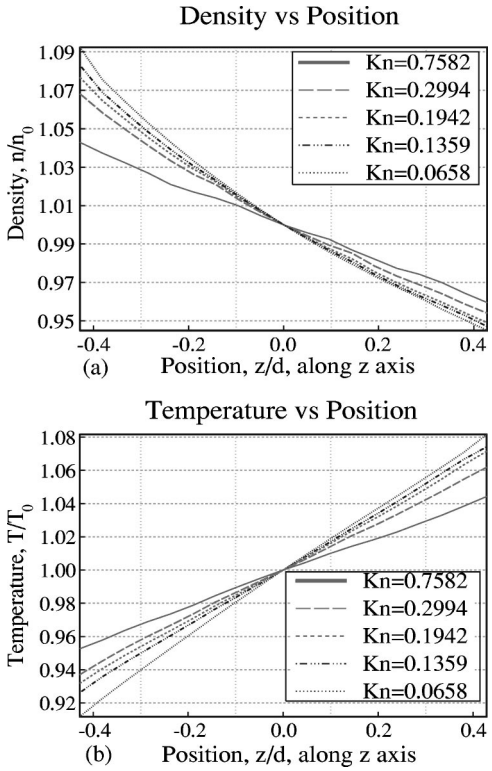

FIG. 7. Density and temperature vs position for all Knudsen numbers using the BGK collision operator. (a) Density profile and (b) temperature vs position. In both plots, the *z* axis value is normalized using the value at the midpoint between the plates.

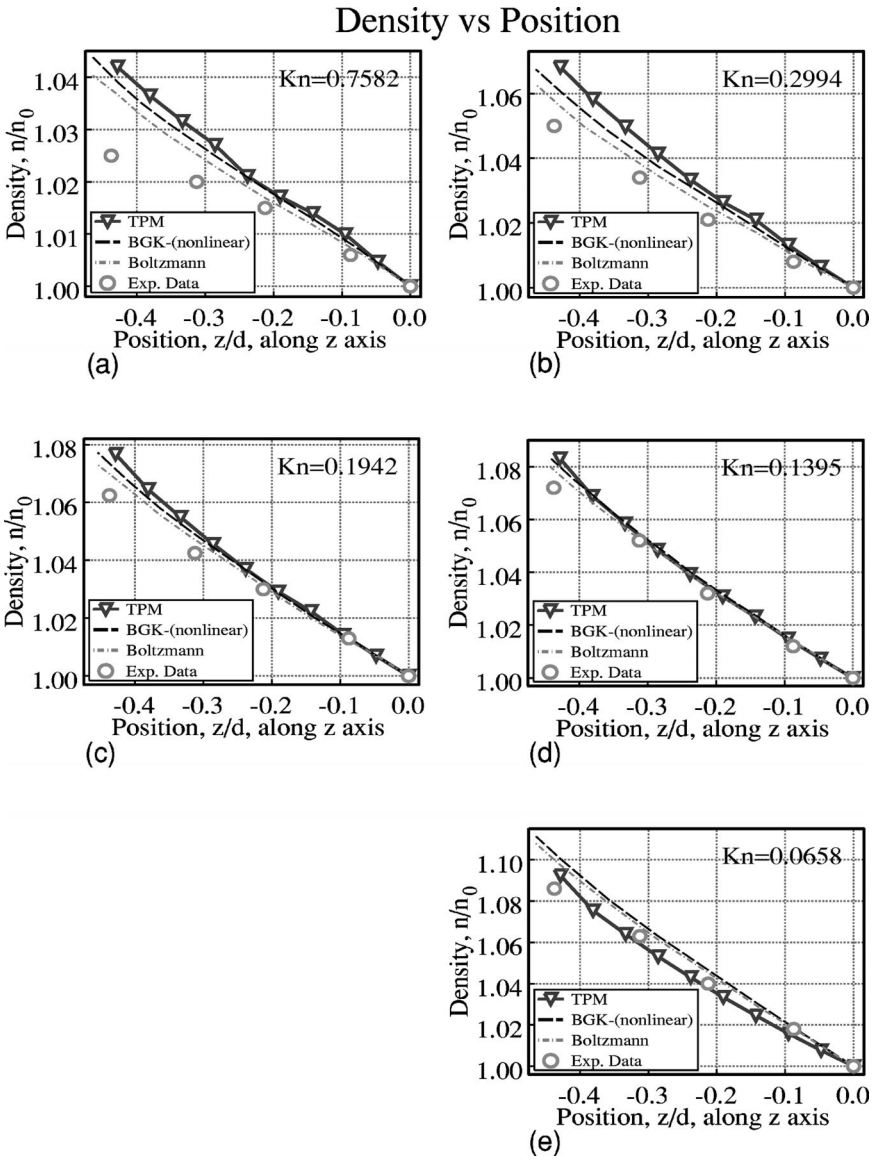

FIG. 8. Density vs position using the BGK collision operator. Density is normalized using the density at the center, n_0, between the two plates. Position is normalized using the distance between the plates. In addition, this figure includes the experimental data of Teagan and Springer as well as the numerical solutions generated by Ohwada for the nonlinear Boltzmann equation solver and the nonlinear BGK model. (a) $K_n=0.7194$, (b) $K_n=0.2994$, (c) $K_n=0.1942$, (d) $K_n=0.1395$, and (e) $K_n=0.0658$.

(3) As in (2) with the additional requirement that particles are started at a cold temperature compared to the temperature of the walls.

(4) As in (3) but the simulation includes momentum conservation.

All of the tests were started with a point distribution of particles, that is, all particles of the simulation were placed in a single initial cell. The verification was performed on a mesh similar to those of Fig. 2.

The tests of uniformity that were done first provide a rather stringent test of this type of simulation. If the particles are not launched isotropically, or various other problems occur, the density of particles will not be uniform. The density was found to be uniform to quite a high accuracy, given the rather coarse mesh that was employed.

In a second type of test, the thermalization of particles was examined by starting the particles at 160 K with the walls at 260 K. Within four steps the distribution was within 5% of the wall temperature.

The final preliminary test looked into the use of the momentum-conserving collision operator. Isotropy was again achieved using the momentum-conserving collision operator.

In the four cases presented above, the TPM generated the expected solutions. The TPM was next applied to the heat transport problem discussed in Sec. II. The results are presented in the following section.

B. Results

In this section we discuss the results of the TPM, applied to the heat transport problem, and compare these results to experiments and other simulation results. Figures 6–9 show the TPM results. In addition, Fig. 8 includes the experimental results of Teagan and Springer and the simulation results of Ohwada generated with the nonlinear Boltzmann equation solver and with the nonlinear BGK model. The nonlinear Boltzmann equation solver solutions were generated using a large integration table in combination with standard finite-difference methods in order to determine the solution to the mixed integro-differential equation. The nonlinear BGK solutions employed finite-difference methods to solve the BGK model. Figure 9 includes the experimental results of Teagan

and Springer and the results generated with the nonlinear Boltzmann equation solver but does not include the nonlinear BGK solutions, because the results of the nonlinear BGK model did not significantly differ from those of the nonlinear Boltzmann equation solver. However, Fig. 9 does include the results of Huang and Hwang, generated using the method of discrete ordinates applied to the BGK model.

The plots of density vs position (Figs. 6 and 8) essentially demonstrate that the simulations we have chosen to compare all agree with experiment and with each other at the lowest Knudsen numbers. This is to be expected, since the differences in collision operators become less significant when the collisions are very frequent—all the models give rise to a Maxwellian distribution. As the Knudsen number increases, there is some divergence between models and experiment. The models continue to agree with each other well, provided they employ the same collision operator. A collision operator that results in a full Maxwellian distribution for the scattered particles actually does less well in reproducing experiment than some other forms of the collision operator. This is true both for the two versions of the TPM shown here (comparing Figs. 6 and 8) as well as for different simulation schemes. It is interesting to note that the TPM using the monoenergetic collision operator encompasses the solutions generated using the BGK model, underscoring the sensitivity of the solution to the collision operator employed. We emphasize that the present simulation is fully 3D in space and velocity, whereas those we compare to are 1D in space.

Figure 7 shows density and temperature versus position, for different Knudsen numbers. The temperature does not reach that of the plates, as we go to a long mean-free-path. This is to be expected—at infinite mean-free-path the temperature would not vary with position. The density variation shows the inverse trend ($p=nk_BT$ holds to reasonable accuracy), so the density varies less, with position, at long mean-free-path. In addition, Fig. 7(a) shows an asymmetry in the density profile that was not seen in the experiment but has been observed in the solutions generated using the nonlinear Boltzmann equation solver. These asymmetries are larger in the solutions generated with the TPM than those observed by Ohwada. The increase may be due to the particular collision operator (and its implementation). Ohwada went to some lengths to resolve discrepancies between the experimental data and the theoretical solutions, and being unable to do so, he argued that the experimental results may be incorrect [16].

Figure 9 shows the heat flux, normalized to the infinite mean-free-path value, and also expressed normalized to the density of particles. While this point is not stated clearly in the various papers, some plot the total heat flux and some plot the heat flux per particle. The normalization to the infinite-λ case makes the comparison less clear, but the agreement is generally good. An improved collision operator might rectify the discrepancies that arise.

Overall, the TPM appears to perform well in this simple test case, giving support to the contention that it will be appropriate in complex 3D geometries. A more accurate collision operator would appear to be the remedy for the modest discrepancies that are observed.

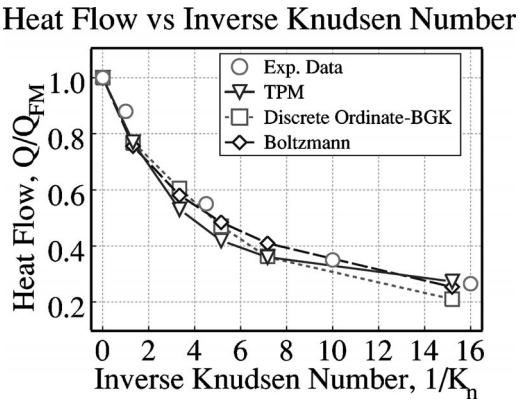

FIG. 9. Heat flux vs inverse Knudsen number using the BGK collision operator. Heat flux has been normalized using the free molecular flow value, Q_{FM}. Also included are the experimental data of Teagan and Springer, the numerical solutions generated by Ohwada for the nonlinear Boltzmann equation solver, and the numerical solution of Huang and Hwang generated by the method of discrete ordinates applied to the BGK transport model.

V. CONCLUDING REMARKS

The transport model presented has extended the transition probability matrix method (method of propagators) to arbitrary meshes in three spatial dimensions, allowing momentum-conserving collisions and keeping memory requirements modest. The simulation was successfully applied to the problem of heat transfer between two parallel plates at different temperatures with a rare gas between them. The results of the kinetic model were in good agreement with the experimental data of Teagan and Springer [11]. Issues surrounding generalizing the method as well as reducing the memory needs were discussed. Future work will involve the development of a convective scheme propagator for computing probabilities, the introduction of convex hulls, more accurate methods for determining the fractional overlap of fine mesh elements with propagating rays, and an investigation of the advantages and tradeoffs of using a rayless method. In addition, the application of this method to heat transfer and flows in MEMS will be investigated. In summary, the method performed very well in describing the behavior of a rare gas between two parallel plates.

ACKNOWLEDGMENT

The authors would like to express our sincere thanks to Professor Iain Boyd of the Aerospace Department at the University of Michigan for his input regarding high Knudsen number Couette flows and for pointing out existing experimental data in this area.

APPENDIX: RADIAL COUNTING ALGORITHM

An efficient radial counting algorithm has been developed for the determination of the integer offsets, in order of increasing radius. Since the ordered triple (i,j,k) lies on the same radius as $(\pm i,\pm j,\pm k)$, we only need to generate the ordered triples in the first quadrant. A further simplifying

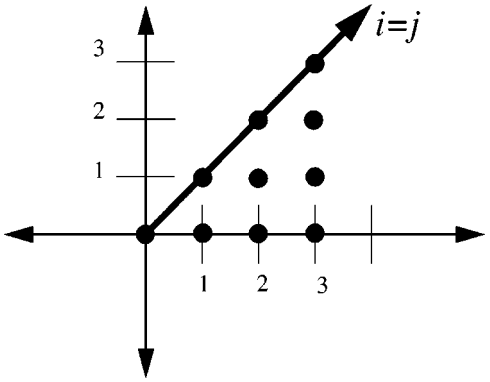

FIG. 10. A 2D visualization of the slice of the first quadrant for which the counting algorithm generates ordered triples in order of increasing radius ($i \geq j \geq 0$) for 2D.

observation is that the ordered triples given by

$$(\hat{i},\hat{j},\hat{k}) \in \chi = \{(i,j,k) |\ i \geq j \geq k \geq 0\} \tag{A1}$$

are unique and permutations of $(\hat{i},\hat{j},\hat{k})$ generate the ordered triples of the first quadrant. Therefore, generation of the ordered triples by increasing the radius can be reduced to the problem of generating the elements of the set χ in order of increasing radius.

The algorithm that generates the ordered triples of set χ by increasing the radius can be described as (a 2D visualization can be seen in Fig. 10) follows.

Let L be a set of ordered triples (initially empty), $L=\{\}$.
Let o be an ordered triple such that
$o = \min_r \{p |\ p \in L \text{ and } r = |p|\}$.
Let l be an ordered triple, $l=(i,j,k)$, where
$i,j,k \in \{0,1,2,\ldots\}$.
Let $m=(\tilde{i},0,0)$, where $\tilde{i} \in \{1,2,3,\ldots\}$.
Let $n=(i',j',0)$, where $i',j' \in \{1,2,3,\ldots\}$ and $i'>j'$.
On the first iteration, set $l=(0,0,0)$.
(1) If $l=m$,
  (a) $m=(\tilde{i}+1,0,0)$.
  (b) Append m to L.
  (2) If $l=n$,
  (a) $n=(i',j'+1,0)$.
  (b) Append n to L.
(3) Increment l using one of the following rules.
  (a) If $i=j=k$, then $l=(i+1,j,k)$.
  (b) If $i>j=k$, then $l=(i,j+1,k)$.
  (c) If $i>j>k$, then $l=(i,j,k+1)$.
(4) Determine o of L.
(5) If $|l|=|o|$, then
  (a) Determine if $l \in L$.
  (b) If $l \in L$, then remove l from L.
(6) If $|l|>|o|$, then
  (a) Determine if $l \in L$.
  (b) If l not in L, then place l in L.
  (c) Let $l=o$.

At the end of an iteration of the counting algorithm, l is the next smallest integer offset of the set χ. Consider the two ordered triples $(7,1,0)$ and $(5,5,0)$ with the same radius $r=\sqrt{50}$. Although the counting algorithm will not store these two elements in L at the same time, they demonstrate the potential of having more than one element in the set χ with the same radius. If there is more than one element in L with the same radius, they are stored consecutively in L so that they may be accessed as a group. After the next element/elements of the set χ has been determined, the unique permutations of the element/elements are generated. These permutations are all ordered triples in the first quadrant with the same radius. Next, the ordered triples for the remaining seven quadrants are generated, giving a complete list of the ordered triples at the next largest radius.

[1] P. M. Morse and H. Feshbach, *Methods of Theoretical Physics* (McGraw-Hill, New York, 1953).
[2] J. Askew, J. Br. Nucl. Energy Soc. **5**, 564 (1966).
[3] T. Cale, G. Raupp, and T. Gandy, J. Appl. Phys. **68**, 3645 (1990).
[4] T. Cale, T. Gandy, and G. Raupp, J. Vac. Sci. Technol. A **9**, 524 (1991).
[5] R. Harvey, W. Hitchon, and G. Parker, J. Appl. Phys. **75**, 1940 (1994).
[6] G. Parker, W. Hitchon, and E. Keiter, Phys. Rev. E **54**, 938 (1996).
[7] W. Tan, R. Hoekstra, and M. Kushner, J. Appl. Phys. **79**, 3423 (1996).
[8] C. W. Chang and G. Uhlenbeck, University of Michigan Report No. M999, 1953 (unpublished).
[9] E. Gross and S. Ziering, Phys. Fluids **2**, 701 (1959).
[10] P. Bassanini, C. Cercignani, and C. Pagani, Int. J. Heat Mass Transf. **10**, 447 (1967).
[11] W. Teagan and G. Springer, Phys. Fluids **11**, 497 (1968).
[12] S.M. Yen, Int. J. Heat Mass Transf. **14**, 1865 (1972).
[13] A.B. Huang and P.F. Hwang, Phys. Fluids **16**, 466 (1972).
[14] N. Pazooki and S.K. Loyalka, Int. J. Heat Mass Transf. **28**, 2019 (1985).
[15] S. Song, M. Yovanovich, and F. Goodman, J. Heat Transfer **1993**, 533 (115).
[16] T. Ohwada, Phys. Fluids **8**, 2153 (1996).
[17] I. Boyd and Q. Sun, in 39th AIAA Aerospace Science Meeting, Reno, Nevada, 2001.
[18] L.S. Pan, T.Y. Ng, D. Xu, and K.Y. Lam, J. Micromech. Microeng. **11**, 181 (2001).
[19] W. Hitchon, G. Parker, and J. Lawler, IEEE Trans. Plasma Sci. **21**, 228 (1993).
[20] W. Hitchon, G. Parker, and J. Lawler, IEEE Trans. Plasma Sci. **22/3**, 267 (1994).
[21] A. Christlieb, W. Hitchon, and E. Keiter, IEEE Trans. Plasma Sci. **28:6**, 2214 (2000).
[22] W. Hitchon, *Plasma Processes for Semiconductor Fabrication* (Cambridge University Press, Cambridge, England, 1999).