SANDIA REPORT
SAND2004-0089
Unlimited Release
Printed January 2004

Radiation-Transport Method to Simulate Noncontinuum Gas Flows for MEMS Devices

John R. Torczynski and Michail A. Gallis

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2004-0089
Unlimited Release
Printed January 2004

Radiation-Transport Method to Simulate Noncontinuum Gas Flows for MEMS Devices

John R. Torczynski and Michail A. Gallis
Engineering Sciences Center
Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185-0826

Abstract

A Micro Electro Mechanical System (MEMS) typically consists of micron-scale parts that move through a gas at atmospheric or reduced pressure. In this situation, the gas-molecule mean free path is comparable to the geometric features of the microsystem, so the gas flow is noncontinuum. When mean-free-path effects cannot be neglected, the Boltzmann equation must be used to describe the gas flow. Solution of the Boltzmann equation is difficult even for the simplest case because of its sevenfold dimensionality (one temporal dimension, three spatial dimensions, and three velocity dimensions) and because of the integral nature of the collision term. The Direct Simulation Monte Carlo (DSMC) method is the method of choice to simulate high-speed noncontinuum flows. However, since DSMC uses computational molecules to represent the gas, the inherent statistical noise must be minimized by sampling large numbers of molecules. Since typical microsystem velocities are low (< 1 m/s) compared to molecular velocities (~400 m/s), the number of molecular samples required to achieve 1% precision can exceed 10^{10} per cell. The Discrete Velocity Gas (DVG) method, an approach motivated by radiation transport, provides another way to simulate noncontinuum gas flows. Unlike DSMC, the DVG method restricts molecular velocities to have only certain discrete values. The transport of the number density of a velocity state is governed by a discrete Boltzmann equation that has one temporal dimension and three spatial dimensions and a polynomial collision term. Specification and implementation of DVG models are discussed, and DVG models are applied to Couette flow and to Fourier flow. While the DVG results for these benchmark problems are qualitatively correct, the errors in the shear stress and the heat flux can be order-unity even for DVG models with 88 velocity states. It is concluded that the DVG method, as described herein, is not sufficiently accurate to simulate the low-speed gas flows that occur in microsystems.

Acknowledgment

This project, LDRD 03-0037, “Radiation-Transport Method to Simulate Noncontinuum Gas Flows for MEMS Devices,” was funded by the Laboratory Directed Research and Development (LDRD) program at Sandia National Laboratories. The authors thank Daniel J. Rader of Sandia National Laboratories for many helpful discussions and for performing the ensemble-averaged DSMC simulations of Fourier and Couette flow.

Table of Contents

Table of Contents 5
List of Figures 6
List of Tables 7
Nomenclature 8
1. Introduction 11
1.1. Overview 11
1.2. Microscale Noncontinuum Flows 11
2. Modeling Approaches 14
2.1. Overview 14
2.2. Navier-Stokes Slip/Jump (NSSJ) Method 14
2.3. Direct Simulation Monte Carlo (DSMC) Method 16
2.4. Discrete Velocity Gas (DVG) Method 20
3. Implementation of DVG Method 24
3.1. Overview 24
3.2. Selection of Velocity States 24
3.3. Determination of Collision Terms 26
3.4. Numerical Approach 32
3.5. Computational Implementation 36
4. Couette-Flow Benchmark Problem 39
4.1. Overview 39
4.2. Problem Description 39
4.3. Results 40
5. Fourier-Flow Benchmark Problem 48
5.1. Overview 48
5.2. Problem Description 48
5.3. Results 49
6. Conclusions 57
References 58
Distribution 62

List of Figures

Figure 1. Cantilevered microbeams fabricated at Sandia using the SUMMiT V^{TM} process. 13
Figure 2. Schematic diagram of an 8-state Discrete Velocity Gas (DVG) model. 23
Figure 3. Example of FORTRAN 77 subroutine encoding DVG collision terms. 30
Figure 4. Geometry for DVG numerical implementation. 31
Figure 5. DSMC velocity profiles. 42
Figure 6. NSSJ velocity profiles. 42
Figure 7. DVG velocity profiles: 32 states, $(c_0/c_m)^2 = 0.166666$. 43
Figure 8. DVG velocity profiles: 32 states, $(c_0/c_m)^2 = 0.425443$. 43
Figure 9. DVG velocity profiles: 88 states, $(c_0/c_m)^2 = 0.166666$. 44
Figure 10. DVG velocity profiles: 88 states, $(c_0/c_m)^2 = 0.425443$. 44
Figure 11. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 0.133322$ Pa. 45
Figure 12. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 1.33322$ Pa. 45
Figure 13. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 13.3322$ Pa. 46
Figure 14. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 133.322$ Pa. 46
Figure 15. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 266.644$ Pa. 47
Figure 16. DSMC, NSSJ, and DVG shear stress τ vs. pressure p. 47
Figure 17. DSMC temperature profiles. 51
Figure 18. NSSJ temperature profiles. 51
Figure 19. DVG temperature profiles: 32 states, $(c_0/c_m)^2 = 0.166666$. 52
Figure 20. DVG temperature profiles: 32 states, $(c_0/c_m)^2 = 0.425443$. 52
Figure 21. DVG temperature profiles: 88 states, $(c_0/c_m)^2 = 0.166666$. 53
Figure 22. DVG temperature profiles: 88 states, $(c_0/c_m)^2 = 0.425443$. 53
Figure 23. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 0.133322$ Pa. 54
Figure 24. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 1.33322$ Pa. 54
Figure 25. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 13.3322$ Pa. 55
Figure 26. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 133.322$ Pa. 55
Figure 27. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 266.644$ Pa. 56
Figure 28. DSMC, NSSJ, and DVG heat flux q vs. pressure p. 56

List of Tables

Table 1. VSS, IPL, and CE parameters for DSMC. 19
Table 2. DVG models: generator states and speed scales. 25
Table 3. Relevant physical parameters in Couette-flow simulations. 40
Table 4. Relevant physical parameters in Fourier-flow simulations. 49

Nomenclature

Roman Variables

$\mathbf{a}$	velocity-independent body force per unit mass, acceleration [m/s^2]
$a_{s,\,s_1}^{(s_2,\,s_3)}$	collision probability for states s and s_1 to produce allowed state pair (s_2, s_3) [1]
A_1	function tabulated by Chapman and Cowling (1970) [1]
A_2	function tabulated by Chapman and Cowling (1970) [1]
A	parameter in equilibrium distribution [1]
B	parameter in equilibrium distribution [1]
C_s	collision term for velocity state s [1/(m^3·s)]
c_p	specific heat at constant pressure, $(\gamma/(\gamma-1))(k_B/m)$ [J/(kg·K)]
c_v	specific heat at constant volume, $(1/(\gamma-1))(k_B/m)$ [J/(kg·K)]
c_m	most probable speed of a Maxwellian distribution, $\sqrt{2k_BT/m}$ [m/s]
$\bar{c}$	average speed of a Maxwellian distribution, $\sqrt{8k_BT/\pi m}$ [m/s]
$\mathbf{c}_s$	velocity of velocity state s, $c_0(i\hat{\mathbf{e}}_x + j\hat{\mathbf{e}}_y + k\hat{\mathbf{e}}_z)$ [m/s]
c_0	DVG speed scale [m/s]
D	self-diffusion coefficient [m^2/s]
d	molecular hard-sphere diameter [m]
e	thermal energy per unit mass [J/kg]
$\hat{\mathbf{e}}_x$	unit normal vector in x-direction [1]
$\hat{\mathbf{e}}_y$	unit normal vector in y-direction [1]
$\hat{\mathbf{e}}_z$	unit normal vector in z-direction [1]
$\mathbf{f}$	body force per unit volume [N/m^3]
f	molecular velocity probability distribution [s^3/m^3]
F	molecular number density distribution [s^3/m^6]
F_s	number flux of velocity state s [1/(m^2·s)]
f_s	fraction of total outgoing flux reflected into incoming velocity state s [1]
g	body-force acceleration in y-direction [m/s^2]
I	identity tensor [1]
i, j, k	indices denoting a velocity state [integer]
K	thermal conductivity [W/(m·K)]
k_B	Boltzmann constant [1.380658×10^{-23} J/K]
L	domain length [m]
l	mesh cell index [integer]
m	molecular mass [kg]
N_{ODE}	number of ODE time steps per time step [integer]
N_s	number of velocity states [integer]
n_s	number density of velocity state s [1/m^3]
N_t	number of time steps [integer]
N_x	number of mesh cells in domain [integer]
$\hat{\mathbf{n}}$	unit normal vector pointing out of the solid and into the gas [1]
n	number density [1/m^3]
Pr	Prandtl number, $\mu c_p/K$ [1]
p	pressure [Pa]

$\mathbf{q}$	heat flux vector [W/m^2]
q	heat flux magnitude [W/m^2]
s	velocity state, (i, j, k) [integer]
s^2	magnitude of velocity state, $i^2 + j^2 + k^2$ [integer]
T	temperature [K]
T_1	temperature of wall at $x = 0$ [K]
T_2	temperature of wall at $x = L$ [K]
t	time [s]
Δt	time step [s]
Δt_{ODE}	ODE time step, $\Delta t / N_{\mathrm{ODE}}$ [s]
$\mathbf{u}$	velocity of gas, solid, or molecule [m/s]
u	velocity component in x-direction [m/s]
V	velocity component in y-direction [m/s]
V_1	velocity component in y-direction of wall at $x = 0$ [m/s]
V_2	velocity component in y-direction of wall at $x = L$ [m/s]
$\mathbf{v}$	velocity of gas, solid, or molecule [m/s]
v	velocity component in y-direction [m/s]
w	velocity component in z-direction [m/s]
Δx	mesh cell width, L / N_x [m]
x	Cartesian coordinate across domain [m]
y	Cartesian coordinate in flow direction [m]
z	Cartesian coordinate perpendicular to x and y [m]

Greek Variables

α	VSS angular scattering parameter [1]
α_1	fraction of reflections that are antisymmetric (bounce-back) for wall at $x = 0$ [1]
α_2	fraction of reflections that are antisymmetric (bounce-back) for wall at $x = L$ [1]
δ	Dirac delta function [inverse of units of its argument]
γ	specific-heat ratio, c_p / c_v, 5/3 for monatomic gas, 7/5 for diatomic gas [1]
ε_1	fraction of reflections that are symmetric (specular) for wall at $x = 0$ [1]
ε_2	fraction of reflections that are symmetric (specular) for wall at $x = L$ [1]
ζ	bulk viscosity, 0 for monatomic gas [Pa·s]
Λ	slip length, $((2 - \sigma)/\sigma)\lambda$ [m]
λ	mean free path, $(2\mu)/(\rho\bar{c})$ [m]
μ	absolute viscosity [Pa·s]
ν	exponent of intermolecular separation in IPL force law [1]
ρ	mass density [kg/m^3]
σ	accommodation coefficient [1]
σ_c	cross section per unit solid angle [m^2]
τ	shear stress [Pa]
Ω	solid angle [steradian]
ω	VSS viscosity temperature exponent [1]

Superscripts

A	bounce-back (antisymmetric) reflection
eq	equilibrium at T and U
$init$	initial value
I	diffuse-isothermal reflection at T and U
S	specular (symmetric) reflection
T	transpose of a matrix

Subscripts

C	continuum
FM	free-molecular
in	incoming quantity at a boundary
min	minimum
max	maximum
out	outgoing quantity at a boundary
ref	reference value
wall	quantity at a solid wall boundary
1	boundary or wall at $x = 0$, or state index
2	boundary or wall at $x = L$, or state index
3	state index

Annotations

$(\dots)'$	prime denotes postcollision quantity
$\langle\dots\rangle$	angle brackets denote an average or expectation
$[\dots]$	square brackets denote arguments to a function
$(\dots)_\infty/(\dots)_1$	ratio of Chapman-Enskog infinite-approximation to first-approximation values
$(\dots, \dots)$	postcollision state pair conserving precollision mass, momentum, and energy

Acronyms

CE	Chapman-Enskog
DSMC	Direct Simulation Monte Carlo
DVG	Discrete Velocity Gas
HS	Hard Sphere
IPL	Inverse Power Law
MP	Massively Parallel
MX	Maxwellian
NSSJ	Navier-Stokes Slip/Jump
ODE	Ordinary Differential Equation
VSS	Variable Soft Sphere

1. Introduction

1.1. Overview

The phenomena that are particularly important for modeling microscale gas flows are described. Microscale gas flows are noncontinuum: the mean free path of gas molecules is comparable to geometric length scales. Microscale gas flows are subsonic: the gas velocity is small compared to the molecule speed. Microscale gas flows are creeping: inertial effects are small compared to viscous effects. Microscale gas flows have varying density: significant pressure variations caused by flow through long, thin gaps and/or significant temperature variations induced by adjacent solid boundaries produce significant density variations. Microscale gas flows encounter time-varying geometries: the motion can be in-plane (sliding) or out-of-plane (opening and closing) and steady or transient (e.g., periodic).

1.2. Microscale Noncontinuum Flows

A Micro Electro Mechanical System (MEMS) has micron-scale geometric features. Figure 1 shows a typical example: cantilevered microbeams fabricated by Sandia's SUMMiT V^{TM} process (Sniegowski and de Boer, 2000). These microbeams have a thickness of 2 μm, a width of 20 μm, a length of 100 μm, and a gap height with respect to the substrate of 2 μm. Dimensions like these are typical of microstructures, which commonly have a vertical extent of 1-10 μm and a horizontal extent of 10-1000 μm (Sniegowski and de Boer, 2000).

Although vacuum-packaging is an option, microsystems are usually operated in a gas at ambient or reduced pressure. The mean free path for common gases (air, nitrogen, oxygen, and argon) is around 0.06 μm at atmospheric pressure and room temperature (Karniadakis and Beskok, 2002). Since the mean free path is not negligibly small compared to the geometric length scales, the gas flow exhibits noncontinuum features. Since the mean free path is inversely proportional to the gas density, microsystems operated at reduced pressures experience larger noncontinuum gas effects. If helium is chosen instead of air or nitrogen as the packaging gas, noncontinuum effects are enhanced because the mean free path of helium is approximately triple that of air at the same pressure and temperature (Karniadakis and Beskok, 2002).

Because of the small geometric length scales, the velocities of moving microstructures and of the corresponding gas flows are usually subsonic. As an example, consider one of the cantilevered microbeams in Figure 1. The maximum amplitude for out-of-plane motion of the microbeam is 2 μm, the gap height between the microbeam and the substrate. Typical oscillation frequencies are generally less than 100 kHz (Torczynski et al., 2002; Gallis and Torczynski, 2003). Under these conditions, the maximum velocity at the tip of the microbeam is around 1 m/s, whereas the speed of sound in common gases at ambient conditions is 300-400 m/s and for helium is around 1000 m/s (Thompson, 1972). The velocity of the gas entering or exiting the gap along the edge of the microbeam can be an order of magnitude larger than the microbeam velocity because of the aspect ratio of the geometry (a gap height of 2 μm versus a gap width of 20 μm). Even allowing for this factor, the gas velocity is still subsonic.

Again, because of the small geometric length scales, microscale gas flows are typically creeping: inertial effects are much smaller than viscous effects. The cantilevered microbeam is a good example of this (Torczynski et al., 2002; Gallis and Torczynski, 2003). The Reynolds number is the ratio of inertial to viscous effects and is given by the product of the gas mass density, the gas velocity, and the length scale divided by the absolute viscosity (Karniadakis and Beskok, 2002). For common gases at ambient conditions, the mass density is around 1 kg/m^3 or less, and the absolute viscosity is around $2\times10^{-5}\ \mathrm{Pa}\cdot\mathrm{s}$ (Thompson, 1972). For the cantilevered microbeam, the gas velocity is typically below 10 m/s, and the gap height of 2 μm is the length scale over which viscosity operates. Under these conditions, the Reynolds number is at most order-unity and is often one or more orders of magnitude smaller than unity, indicating that inertial effects are small compared with viscous effects and that microscale gas flows are typically creeping flows.

Although microscale gas flows are too subsonic to have large velocity-induced density variations, frictional and thermal effects can produce significant density variations. As indicated above, microscale surface topography is typically 1-10 μm high but 10-1000 μm across. Gas flowing through a microchannel with a comparable diameter-to-length aspect ratio can be essentially isothermal but can have a substantial pressure variation, and therefore a substantial density variation, along the length of the microchannel because of wall friction (Karniadakis and Beskok, 2002). A large temperature difference between adjacent microstructures, such as produced by Joule heating in one of the microstructures, can produce a large temperature variation and a correspondingly large density variation in the gas around these microstructures (Torczynski and Gallis, 2003).

Microscale gas flows often encounter a moving geometry. The motion of the geometry can be out-of-plane or in-plane and can be transient or steady. In the case of the cantilevered microbeams shown in Figure 1, the microbeam oscillates periodically, producing transient, out-of-plane motion. In the more common case of rotating gears, the motion is in-plane and steady (or periodic in the vicinity of gear teeth, holes, dimples, or other nonaxisymmetric structures). In fact, the gas flows of interest are usually produced by and affect the motion of the microsystem.

Methods for simulating microscale gas flows should be applicable to the following conditions. First, noncontinuum effects should be incorporated. Second, only subsonic, creeping gas flow with frictionally-induced or thermally-induced density variations needs to be treated: inertial and transonic effects can typically be neglected. Third, the method should admit the possibility of extension to moving geometries, with in-plane or out-of-plane and steady or transient motion. Herein, only the special case of in-plane motion that does not alter the geometry is considered.

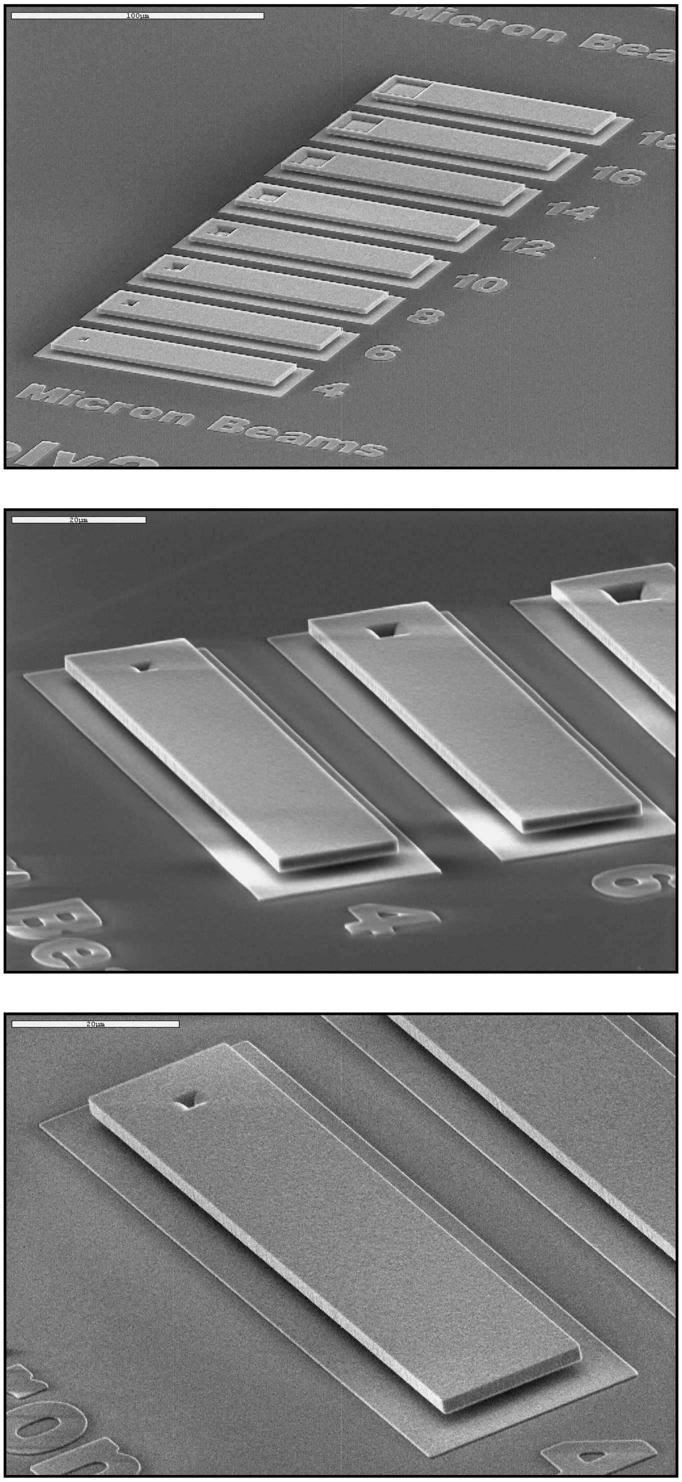

Figure 1. Cantilevered microbeams fabricated at Sandia using the SUMMiT V^{TM} process.

2. Modeling Approaches

2.1. Overview

Three methods of modeling microscale gas flows that meet the requirements delineated above are described. The Navier-Stokes slip/jump method (NSSJ) uses the conventional Navier-Stokes equations and incorporates noncontinuum effects through modified boundary conditions that allow velocity slips and temperature jumps at solid surfaces. The Direct Simulation Monte Carlo (DSMC) method uses molecular gas dynamics with computational molecules that move, collide, and interact with solid surfaces to solve the Boltzmann equation. The Discrete Velocity Method (DVG) restricts molecular velocities to a finite set of discrete values and solves discrete-Boltzmann transport equations for the number densities of these velocity states.

2.2. Navier-Stokes Slip/Jump (NSSJ) Method

The Navier-Stokes slip/jump (NSSJ) method is one approach for modeling microscale gas flows (Karniadakis and Beskok, 2002). The behavior of the gas away from solid surfaces is described by the Navier-Stokes (NS) equations, the ideal gas equation of state, and the Newtonian and Fourier constitutive relations for the shear stress and heat flux (Karniadakis and Beskok, 2002):

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0\,, \tag{1}$$

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mathbf{f} + \nabla\cdot\tau\,, \tag{2}$$

$$\rho\left(\frac{\partial e}{\partial t} + \mathbf{u}\cdot\nabla e\right) = -p\nabla\cdot\mathbf{u} - \nabla\cdot\mathbf{q} + \tau\cdot\nabla\mathbf{u}\,, \tag{3}$$

$$e = \frac{p}{\rho(\gamma-1)} = \frac{k_B T}{m(\gamma-1)}\,, \tag{4}$$

$$\tau = \mu\{(\nabla\mathbf{u}) + (\nabla\mathbf{u})^T\} + \zeta(\nabla\cdot\mathbf{u})I\,, \tag{5}$$

$$\mathbf{q} = -K\nabla T\,. \tag{6}$$

Here, $k_B = 1.380658\times10^{-23}$ J/K is the Boltzmann constant, ρ is the mass density, $\mathbf{u}$ is the velocity vector, p is the pressure, $\mathbf{f}$ is the volumetric-force vector, τ is the shear-stress tensor, e is the thermal energy per unit mass, $\mathbf{q}$ is the heat-flux vector, γ is the specific-heat ratio, T is the temperature, m is the molecular mass, μ is the absolute viscosity, ζ is the bulk viscosity, and K is the thermal conductivity, where μ, ζ, and K are assumed to depend on p and T alone.

Boundary conditions are required to complete the specification of a problem. While important in their own right, inflow and outflow boundary conditions are not discussed here because they relate to computational convenience rather than to physical realism: in principle, the computational domain can always be made large enough to remove accuracy issues associated with the proximity of inflow and outflow boundaries. Although the subject of boundary conditions at solid surfaces remains a research topic, the slip/jump (SJ) formulation has proved popular (Karniadakis and Beskok, 2002). In this formulation, differences between gas and wall quantities at a solid surface are proportional to normal gradients of these quantities.

In the absence of temperature gradients, the SJ velocity boundary condition takes the following form (Karniadakis and Beskok, 2002):

$$\hat{\mathbf{n}} \cdot (\mathbf{u} - \mathbf{u}_{\text{wall}}) = 0, \tag{7}$$

$$\hat{\mathbf{n}} \times \{(\mathbf{u} - \mathbf{u}_{\text{wall}}) - \Lambda(\hat{\mathbf{n}} \cdot \nabla)(\mathbf{u} - \mathbf{u}_{\text{wall}})\} = 0. \tag{8}$$

Similarly, in the absence of velocity gradients, the SJ temperature boundary condition takes the following form (Kennard, 1938):

$$(T - T_{\text{wall}}) - \left(\frac{1}{Pr}\right)\left(\frac{2\gamma}{\gamma + 1}\right)\Lambda(\hat{\mathbf{n}} \cdot \nabla)(T - T_{\text{wall}}) = 0. \tag{9}$$

Here, $\hat{\mathbf{n}}$ is the unit normal vector pointing out of the solid and into the gas, variables with and without the subscript "wall" refer to values associated with the solid surface and with the gas adjacent to the solid surface, respectively, and Λ is the slip length (Kennard, 1938):

$$\Lambda = \left(\frac{2 - \sigma}{\sigma}\right)\lambda, \tag{10}$$

$$\lambda = \frac{2\mu}{\rho\bar{c}}, \tag{11}$$

$$\bar{c} = \sqrt{\frac{8k_BT}{\pi m}}, \tag{12}$$

$$Pr = \frac{\mu c_p}{K} = \frac{\mu\gamma k_B}{K(\gamma - 1)m} \approx \frac{4\gamma}{9\gamma - 5}. \tag{13}$$

Here, σ is an accommodation coefficient of the surface that satisfies $0 \le \sigma \le 1$, λ is the mean free path (this definition is one of many in the literature and is used exclusively herein because it is consistent with the above SJ boundary conditions), $\bar{c}$ is the mean speed of molecules in an equilibrium gas, and Pr is the Prandtl number, exhibited here with the approximation of Eucken (Thompson, 1972). When both velocity and temperature effects exist simultaneously, more complicated SJ boundary conditions are needed to describe phenomena such as “thermal creep” (Karniadakis and Beskok, 2002). Moreover, different accommodation coefficients may be needed to describe the different processes associated with the accommodation of normal momentum, tangential momentum, translational energy, and internal energy.

To reiterate, the NSSJ approach to simulating microscale gas flows treats the entire gas flow as continuum flow: noncontinuum effects are introduced only through boundary conditions. This is tantamount to assuming that noncontinuum effects are confined to thin regions at the boundaries (i.e., Knudsen layers) and that these Knudsen layers are much thinner than geometric length scales. Nevertheless, as discussed in later sections, the NSSJ approach does yield correct results for certain classes of flows in the free-molecular limit.

2.3. Direct Simulation Monte Carlo (DSMC) Method

The Direct Simulation Monte Carlo (DSMC) method is another approach to simulating noncontinuum gas flows. Developed by Bird and discussed extensively in monographs and review articles (Bird, 1970, 1976, 1978, 1998; Cercignani, 2000), DSMC provides approximate solutions to the Boltzmann equation.

The Boltzmann equation describes the behavior of a noncontinuum gas under the following conditions (Gombosi, 1994).

1. The gas is dilute. Only binary molecular collisions are considered.

2. “Molecular chaos” applies. The states of any two colliding molecules (their velocities and internal energies) are uncorrelated, and the position and velocity (and angular position and angular velocity) of any particular molecule are also uncorrelated.

3. The molecules possess spherically symmetric potentials through which they interact. Although obviously untrue for diatomic molecules, etc., through application of the previous assumption, a nonspherical potential can be replaced with an effective spherical potential because all possible angular orientations of molecules during collisions are equally likely.

4. Classical mechanics applies. Velocities are nonrelativistic, and quantum effects enter only in determining the collision cross section, which is subsequently treated as a material parameter.

5. The extent of the intermolecular potential is small compared to other length scales, such as intermolecular separation, mean free path, and gradients of statistical quantities.

6. External forces, such as gravity, do not affect collision dynamics.

7. Statistical quantities can be treated as unchanging over the duration of and in the vicinity of a collision.

The Boltzmann equation for a single-species monatomic gas is given below (Vincenti and Kruger, 1965; Bird, 1976, 1998; Cercignani, 1988, 2000; Gombosi, 1994; Garzo and Santos, 2003):

$$\frac{\partial F}{\partial t} + \mathbf{u} \cdot \frac{\partial F}{\partial \mathbf{x}} + \mathbf{a} \cdot \frac{\partial F}{\partial \mathbf{u}} = \int_{-\infty}^{\infty} \int_{0}^{4\pi} (F[\mathbf{v}']F[\mathbf{u}'] - F[\mathbf{v}]F[\mathbf{u}])|\mathbf{v} - \mathbf{u}|\sigma_c d\Omega d\mathbf{v}, \tag{14}$$

$$F = nf, \int f[\mathbf{u}]d\mathbf{u} = 1. \tag{15}$$

Here, $Fd\mathbf{x}d\mathbf{u}$ is the number of molecules at time t with positions within $d\mathbf{x}$ around $\mathbf{x}$ and velocities within $d\mathbf{u}$ around $\mathbf{u}$, and f is the corresponding probability density in velocity space. The quantity $\mathbf{a}$ on the left side of the equation is the body force per unit mass (i.e., acceleration), here taken to be independent of the molecular velocity $\mathbf{u}$ (e.g., gravity). The term on the right side of the equation is the collision integral, where the unprimed and primed quantities refer to the precollision and postcollision velocities, respectively. The quantity $\sigma_c d\Omega$ is the differential cross section for collisions, where, for elastic hard-sphere (HS) molecules of diameter d, the quantity σ_c has a value of $d^2/4$ (Bird, 1976), and, for Maxwell (MX) molecules, the quantity $|\mathbf{v} - \mathbf{u}|\sigma_c$ is independent of the relative speed $|\mathbf{v} - \mathbf{u}|$ (Bird, 1976). Generalization to gases with multiple monatomic species is straightforward (Vincenti and Kruger, 1965; Bird, 1998); however, generalization to molecules with internal energy is more difficult (Hirschfelder et al., 1964).

Macroscopic quantities are determined from the Boltzmann equation by computing the corresponding moments of the molecular velocity distribution. For example, the average number density and velocity are given by the following moments.

$$n = \int_{-\infty}^{\infty} F[\mathbf{u}]d\mathbf{u}, \tag{16}$$

$$\langle \mathbf{u} \rangle = \int_{-\infty}^{\infty} \mathbf{u}f[\mathbf{u}]d\mathbf{u} = \frac{1}{n}\int_{-\infty}^{\infty} \mathbf{u}F[\mathbf{u}]d\mathbf{u}. \tag{17}$$

Closed-form analytical solutions of the Boltzmann equation are known for only a few situations. Free-molecular solutions (i.e., when the collision integral is set equal to zero) are known in certain simple geometries, such as one-dimensional momentum transfer (Couette flow) or heat transfer (Fourier flow) between infinite parallel walls (Bird, 1976; Gombosi, 1994; Bird, 1998). Solutions in the continuum nonequilibrium regime are provided by Chapman-Enskog theory (Chapman and Cowling, 1970). This approach involves a Taylor-series expansion that relies on the smallness of the mean free path compared to length scales associated with gradients of macroscopic quantities like temperature. The principal successes of Chapman-Enskog theory are the derivation of the Newtonian and Fourier constitutive relations for shear stress and heat flux, the determination of the values of the associated transport coefficients, and the prediction of noncontinuum gas effects like thermal diffusion in gas mixtures (Chapman and Cowling, 1970). Thus, in the continuum nonequilibrium limit, the Boltzmann equation reduces to the NS equations.

The DSMC method uses computational molecules to provide an approximate solution to the Boltzmann equation (Bird, 1998). Each computational molecule typically represents a large number of real molecules and undergoes the same types of processes during a time step that real molecules experience: movement from one location to another, interaction with boundaries, and collision with other molecules. Computational molecules move in a ballistic fashion, traveling at constant velocity along straight-line trajectories during a time step. Following movement, boundary conditions are applied. In the case of an inlet or outlet, computational molecules leaving the domain by crossing the boundary may be reflected or deleted, and new computational molecules may be created and allowed to enter the domain, with the relative importance of each process determined by the boundary condition. In the case of a solid surface, a computational molecule crossing this boundary is reflected back into the domain by finding the crossing point, assigning a new velocity to the computational molecule, and allowing it to move from this point at the new velocity for the remainder of the time step. The reflected velocity assigned to the computational molecule is selected stochastically from a prescribed distribution, such as a linear combination of specular reflection, diffuse-isothermal reflection, and diffuse-adiabatic reflection (Gallis et al., 2001). Following movement and reflection from boundaries, computational molecules are allowed to collide in a binary fashion. Pairs of molecules are selected randomly at the appropriate rate and are allowed to collide, by which it is meant that they are assigned postcollision velocities and internal energies (but not new positions). These postcollision values conserve mass, momentum, and energy and are selected stochastically so as to maintain the prescribed collision statistics, which determine important macroscopic properties like the temperature dependence of the transport coefficients.

The Variable Soft Sphere (VSS) model is typically used to specify postcollision velocities (Bird, 1998). The VSS model requires four parameters per collision: the reference temperature T_{ref}, the reference diameter d_{ref}, the viscosity temperature exponent ω, and the angular scattering parameter α. The viscosity temperature exponent usually lies in the range $1/2 \le \omega \le 1$ and produces a temperature dependence for the viscosity, thermal conductivity, and self-diffusion coefficient of $(T/T_{\mathrm{ref}})^{\omega}$. The values $\omega = 1/2$ and $\alpha = 1$ correspond to hard-sphere molecules, and the values $\omega = 1$ and $\alpha = 2.139862$ correspond to Maxwell molecules. For general inverse-power-law (IPL) molecules in which the intermolecular force varies inversely with the intermolecular separation to the power ν, the VSS parameters satisfy the following relations (Torczynski et al., 2003; Gallis et al., 2004):

$$d_{\mathrm{ref}} = \left(\frac{5(\alpha+1)(\alpha+2)(mk_BT_{\mathrm{ref}}/\pi)^{1/2}}{4\alpha(5-2\omega)(7-2\omega)(\mu_1/\mu_\infty)\mu_{\mathrm{ref}}}\right)^{1/2}, \tag{18}$$

$$K_{\mathrm{ref}} = \left(\frac{K_\infty}{K_1}\right)\left(\frac{\mu_1}{\mu_\infty}\right)\left(\frac{15}{4}\right)\left(\frac{k_B}{m}\right)\mu_{\mathrm{ref}}, \tag{19}$$

$$\omega = \frac{1}{2} + \frac{2}{\nu-1}, \ \alpha = \frac{2A_2[\nu]}{2A_1[\nu]-A_2[\nu]}. \tag{20}$$

Here, the subscript “ref” denotes a quantity at the reference temperature, $(\dots)_\infty/(\dots)_1$ denotes the ratio of the infinite-approximation and first-approximation values from Chapman-Enskog (CE) theory (see Table 1), and A_1 and A_2 are tabulated functions of ν (Chapman and Cowling, 1970; Torczynski et al., 2003; Gallis et al., 2004).

Table 1: VSS, IPL, and CE parameters for DSMC.

ν	ω	α	μ_∞/μ_1	K_∞/K_1	D_∞/D_1	Comment
∞	0.5	1.000000	1.016034	1.025218	1.018954	Hard-sphere
21	0.6	1.209042	1.010193	1.016019	1.011708	
11	0.7	1.422476	1.005702	1.008959	1.006411	
23/3	0.8	1.645558	1.002524	1.003965	1.002792	
6	0.9	1.883125	1.000629	1.000988	1.000687	
5	1.0	2.139862	1.000000	1.000000	1.000000	Maxwell

The computational mesh in a DSMC simulation performs two functions. First, computational molecules must be in the same mesh cell in order to collide with each other. Second, the computational mesh is used to accumulate moments of the molecular velocity distribution, such as the number density and the velocity. These molecular properties are sampled before and after the collisions are performed. Although conserved moments such as momentum are unchanged by collisions, higher-order moments such as shear stress and heat flux can be significantly affected. The above double-sample strategy significantly improves the accuracy of these quantities (Gallis et al., 2004). The uncertainties in the moments that result from molecular fluctuations can be reduced by sampling many computational molecules. For a stationary (statistically steady) flow, the ergodic hypothesis is employed: large numbers of molecules are sampled by averaging over large numbers of time steps. For an unsteady flow, either large numbers of molecules per cell must be used, or an ensemble of many simulations must be sampled, where each simulation takes a different path through phase space (usually by starting with a different random seed).

The accuracy of a DSMC simulation is governed by the choice of the mesh cell size, the time step, and the average number of molecules per cell. Reasonable accuracy is obtained when the cell size is one third or less of a mean free path, the time step is one third or less of the time required by molecules to cross a cell, and the number of computational molecules per cell exceeds about 30 (Bird, 1998). Some results are available that quantify the effect of the numerical parameters on the absolute viscosity and the thermal conductivity (Alexander et al., 1998; Alexander et al., 2000; Hadjiconstantinou, 2000; Garcia and Wagner, 2000). Theoretical investigations have led to a formal proof that, as the numerical parameters approach their limits, DSMC converges to a solution of the Boltzmann equation (Bird, 1970; Nanbu, 1986; Wagner, 1992; Cercignani, 2000).

2.4. Discrete Velocity Gas (DVG) Method

The Discrete Velocity Gas (DVG) method is an approach based on radiation-transport ideas for simulating noncontinuum gas flow. Like the DSMC method and in contradistinction to the NSSJ method, the DVG method treats the gas flow as noncontinuum throughout the computational domain, rather than incorporating noncontinuum effects only via boundary conditions. However, unlike DSMC but like discrete-ordinate radiation-transport methods (Siegel and Howell, 1992), molecules in the DVG method are constrained to have only certain discrete velocity values. In discrete-ordinate radiation-transport methods, the direction and the frequency of the radiation are discretized, but the speed is not since the speed of light is essentially constant in a gas (implementation in a transparent liquid or solid does not allow this simplification). In the DVG method, typically each component of the velocity vector is discretized.

Although the idea behind the DVG method may be traced back to the origins of kinetic theory, the first applications of the DVG method are the studies of Broadwell (1964a, 1964b). In the former study, Couette flow is investigated, whereas the structure of a shock wave is examined in the latter study. In both cases, qualitatively correct behavior is observed. Further studies of these steady flows (Hamel and Wachman, 1965; Gatignol, 1975; Cabannes, 1976; Gatignol, 1977) and of transient Rayleigh flow (Longo and Monaco, 1985) confirm the observation that the DVG method qualitatively reproduces known features of noncontinuum gas flow.

Systematic developments and investigations of the DVG method are presented by several researchers (Inamuro and Sturtevant, 1990; Goldstein, 1990; Goldstein, 1991; Monaco and Preziosi, 1991; Nadiga, 1992). The presentation below draws extensively from the development of Inamuro and Sturtevant (1990).

In the DVG method, molecules are constrained to be in discrete velocity “states” $\{s\}$, where $s = (i, j, k)$ denotes the velocity $\mathbf{c}_s = \mathbf{c}_{i,\,j,\,k} = c_0\mathbf{s}$, c_0 is a constant speed scale governed by constraints discussed below, and the i, j, k are integers (see Figure 2). A particular set $\{s\}$ and a particular c_0 value specify a particular DVG model. The number density corresponding to a velocity state s is denoted by $n_s = n_{i,\,j,\,k}$. Heuristically, a DVG model can be thought of as a particular choice for the distribution F in the Boltzmann equation:

$$F[t, \mathbf{x}, \mathbf{c}] \to \sum_s n_s[t, \mathbf{x}]\delta[\mathbf{c} - \mathbf{c}_s]\,. \tag{21}$$

Here, δ is the three-dimensional Dirac delta function. Each number density is governed by a discrete Boltzmann equation of the form given below (the body acceleration term is not included):

$$\frac{\partial n_s}{\partial t} + \mathbf{c}_s \cdot \frac{\partial n_s}{\partial \mathbf{x}} = C_s = \sum_{s_1}\sum_{(s_2,\,s_3)} \pi d^2 \left|\mathbf{c}_s - \mathbf{c}_{s_1}\right| a_{s,\,s_1}^{(s_2,\,s_3)}(n_{s_2}n_{s_3} - n_s n_{s_1})\,, \tag{22}$$

$$\sum_{(s_2,\,s_3)} a_{s,\,s_1}^{(s_2,\,s_3)} = 1\,. \tag{23}$$

The above form is appropriate for monatomic hard-sphere molecules with diameter d. The transition coefficient $a_{s,\,s_1}^{(s_2,\,s_3)}$ represents the loss from state s when states s and s_1 collide to produce states s_2 and s_3 and the gain to state s when states s_2 and s_3 collide in the inverse collision to produce states s and s_1. The quantity (s_2, s_3) represents a pair of postcollision states that conserves precollision values of mass, momentum, and energy. Here, $\mathbf{c}_s + \mathbf{c}_{s_1} = \mathbf{c}_{s_2} + \mathbf{c}_{s_3}$ and $|\mathbf{c}_s - \mathbf{c}_{s_1}| = |\mathbf{c}_{s_2} - \mathbf{c}_{s_3}|$. For hard-sphere molecules, the transition coefficients are independent of the quantity (s_2, s_3), so each transmission coefficient is just the inverse of the total number of postcollision pairs corresponding to the prescribed precollision states s and s_1.

A few observations can be made about the DVG equations.

1. The seven-dimensional Boltzmann equation is replaced by a set of four-dimensional equations (one temporal dimension and three spatial dimensions). The infinite-extent velocity dimensions are removed.

2. The number of equations in this set depends on the number of velocity states in the particular DVG model.

3. The left sides of these equations are strictly linear in the number densities. The differential operator represents linear advection at constant velocity. This is molecular movement.

4. The right sides of these equations are quadratic polynomials in the number densities. The coefficients in these polynomials are constants. This is molecular collision.

Boundary conditions for the DVG method are exactly analogous to the DSMC method but have some special issues. Since the molecular velocities have discrete directions, it is straightforward to specify boundary conditions when the boundaries "conform" to these preferred directions. A specular reflection from a solid plane perpendicular to the z direction just converts state (i, j, k) into state $(i, j, -k)$. Similarly, a "bounce-back" reflection from this boundary converts state (i, j, k) into state $(-i, -j, -k)$. Diffuse adiabatic reflection converts state (i, j, k) into a linear combination of states including states $(i, j, -k)$, $(-i, j, -k)$, $(i, -j, -k)$, and $(-i, -j, -k)$. Diffuse isothermal reflection converts outgoing state (i, j, k) into a linear combination of all incoming states so as to produce their equilibrium distribution. For such processes to be allowed, all of the desired reflected states must be included in the set of states. Boundaries that do not conform to the preferred directions are more complicated and are not considered herein.

In the DVG method, macroscopic quantities are determined as in the DSMC method, by taking moments of the distribution. As examples, the number density and velocity are given below:

$$n = \sum_s n_s , \tag{24}$$

$$\mathbf{u} = \frac{1}{n}\sum_s n_s \mathbf{c}_s . \tag{25}$$

There are three reasons that the DVG method has not been more widely used in the past. First, most past efforts to simulate noncontinuum gas flows have focused on hypersonic applications. The large velocities and the large thermodynamic variations that occur in hypersonic flows produce a large range of molecular velocities, which necessitates the use of a correspondingly large number of discrete velocities. For example, Inamuro and Sturtevant (1990) used on the order of 10^3 velocity states. Second, numerically solving DVG models with large numbers of velocity states is computationally intense. The advent of massively parallel (MP) computers in the last decade has eased this difficulty somewhat. Third, deriving the collision terms for DVG models with more than a few velocity states is analytically intense. As the number of velocity states increases, the probability of error-free hand derivation and implementation decreases rapidly. Fortunately, the availability of symbol-manipulation computer software such as Mathematica (Wolfram, 1996) enables much of the derivation and implementation processes to be automated.

The unique properties of microscale gas flows make it reasonable to reconsider DVG methods. The low-speed nature of microscale gas flows makes them computationally intense for DSMC. Very large numbers of samples of computational molecules per cell are required to achieve acceptable precision, which results in very long simulation times even for simple geometries. However, the low velocities and small thermodynamic variations typical of microscale gas flows is hypothesized to work to the advantage of the DVG method. Throughout the domain, the molecular velocity distribution function is close to a Maxwellian distribution with zero velocity and at some representative temperature, so the range of molecular velocities is known and is the same at all locations in the flow. This feature of microscale gas flow, together with advances in MP computers and symbol-manipulation software, make it reasonable to investigate the possibility of using the DVG method to simulate microscale gas flows. The outstanding issue is the degree to which DVG models can accurately represent relevant gas flows and how the accuracy varies with the parameters used to specify the model.

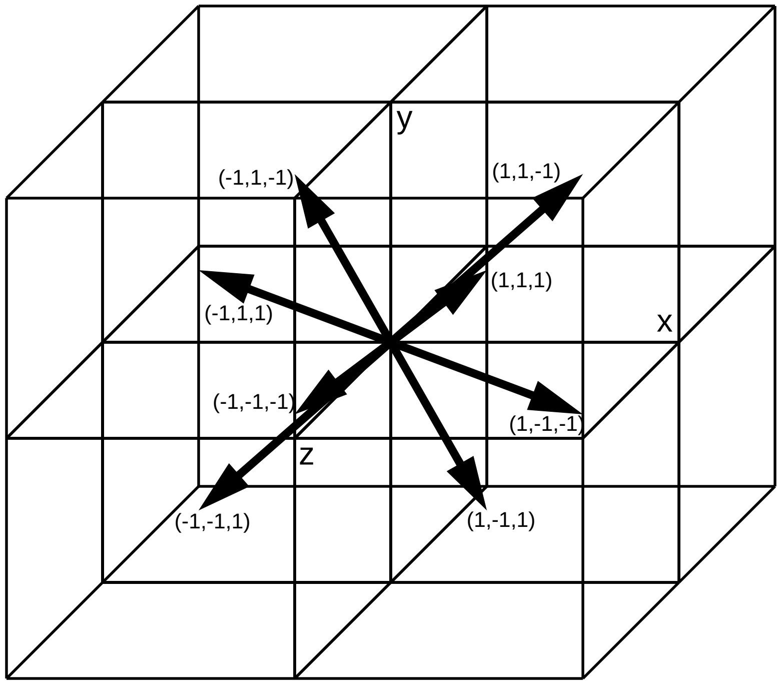

Figure 2. Schematic diagram of an 8-state Discrete Velocity Gas (DVG) model.

3. Implementation of DVG Method

3.1. Overview

The Discrete Velocity Gas (DVG) method is an approximate method for solving the Boltzmann equation. Molecules are restricted to have velocities lying within a set of discrete velocity states. Discrete-Boltzmann transport equations govern the evolution of the number densities of these discrete velocity states. The collision term for each equation is a quadratic polynomial in the number densities of the velocity states, rather than a quadratic integral operator on the distribution function. Even for relatively few velocity states, the algebra associated with determining the collision terms is significant, so symbol-manipulation software is employed to perform this algebra. Macroscopic quantities are determined by taking moments, sums where the quantity of interest is weighted by the number densities of the velocity states. A numerical implementation for a one-dimensional geometry is discussed. The discrete Boltzmann equations are solved using a time-accurate finite-difference method with upwinding. Allowed boundary conditions include various combinations of specular, bounce-back, and diffuse-isothermal processes.

3.2. Selection of Velocity States

The first step in prescribing a DVG model is the specification of the velocity states. Herein, DVG models akin to the "even" models of Inamuro and Sturtevant (1990) are used. In these models, the integers i, j, k are restricted to odd values. The apparent contradiction between the name "even" and the use of odd integers is resolved by the fact that a cubical region of velocity space centered on the origin always has an even number of odd velocity states contained within it (Inamuro and Sturtevant, 1990). This observation holds for any "symmetric" region of velocity space (i.e., one that is invariant under any combination of the transformations $x \to -x$, $y \to -y$, and $z \to -z$). In the "odd" models of Inamuro and Sturtevant (1990), the integers i, j, k have both odd and even values. Since the state $(0, 0, 0)$ is contained in any symmetric region, odd models on symmetric regions always have odd numbers of velocity states. Odd models are not considered herein because of their inability to simulate free-molecular flow: the state $(0, 0, 0)$ does not interact with boundaries or, for free-molecular flow, with other velocity states, so its number density remains unchanged from its initial specification.

Some even DVG models are indicated in Table 2. The models implemented herein range from the simplest model (the 8-state model shown in Figure 2) up to the 88-state model. These models are specified by starting from the 8-state unispeed model (all 8 velocities have the same speed) and progressively adding all states corresponding to the next "generator" state. These states are constructed from all permutations of the generator index values and $\pm$ signs, as discussed below. The bounds on the speed scale c_0 are also included, as discussed below. Here, c_m is the most probable molecular speed for a Maxwellian distribution:

$$c_m = \sqrt{\frac{2k_BT}{m}}. \qquad (26)$$

Table 2: DVG models: generator states and speed scales.

Model	Generator States	State Magnitudes	Speed-Scale Bounds
8-state	$(1, 1, 1)$, see Figure 2	$\sqrt{3}$, see Figure 2	$(1/2) \le (c_0/c_m)^2 \le (1/2)$
32-state	$(3, 1, 1)$ and all above	$\sqrt{11}$ and all above	$(3/22) \le (c_0/c_m)^2 \le (1/2)$
56-state	$(3, 3, 1)$ and all above	$\sqrt{19}$ and all above	$(3/38) \le (c_0/c_m)^2 \le (1/2)$
64-state	$(3, 3, 3)$ and all above	$\sqrt{27}$ and all above	$(3/54) \le (c_0/c_m)^2 \le (1/2)$
88-state	$(5, 1, 1)$ and all above	$\sqrt{27}$ and all above	$(3/54) \le (c_0/c_m)^2 \le (1/2)$
136-state	$(5, 3, 1)$ and all above	$\sqrt{35}$ and all above	$(3/70) \le (c_0/c_m)^2 \le (1/2)$

The speed-scale bounds are determined in the following manner. The gas is taken to have a representative temperature of T and a representative velocity of zero (compared to molecular speeds) throughout the domain. For a monatomic gas, the energy per unit mass associated with these conditions is $(3/2)(k_BT/m) = (3/4)c_m^2$. In a DVG model, the energy per unit mass has minimum and maximum values, denoted by $(c_0s_{\text{min}})^2/2$ and $(c_0s_{\text{max}})^2/2$, where s_{min} and s_{max} are the minimum and maximum state magnitudes, as in Table 2. Recognizing that the actual energy per unit mass must lie between these limiting values, the following constraint is found, where all DVG models shown in Table 2 have the value $s_{\text{min}} = \sqrt{3}$:

$$\frac{3}{2s_{\text{max}}^2} \le \left(\frac{c_0}{c_m}\right)^2 \le \frac{3}{2s_{\text{min}}^2}. \tag{27}$$

Several observations can be made about the DVG models shown in Table 2. First, the upper bound for the speed scale c_0 is always $c_m/\sqrt{2}$. This corresponds to the discretization of velocity space. Second, as more and more velocity states are included in the DVG model, the lower bound of the speed scale tends to zero (albeit rather slowly). Third, prescribing a speed-scale value close to the lower bound produces a “population inversion”: higher-energy velocity states have larger number densities than lower-energy velocity states. This type of distribution does not closely resemble a Maxwellian distribution. Fourth, prescribing a speed-scale value close to the upper bound effectively reduces the DVG model to the 8-state model because the number densities of states with larger velocities become very small. Fifth, the 8-state model is inherently incapable of simulating nonisothermal flows because it is a unispeed model. Thus, from the energy point of view, it is desirable to select a speed-scale value well away from the bounding values; nevertheless, the speed scale is a free parameter of the model, at least within the above bounds.

3.3. Determination of Collision Terms

Once a DVG model has been specified by selecting the velocity states and the speed scale, the collision terms that appear on the right sides of the discrete Boltzmann equations must be determined. The form of the collision terms has already been prescribed, where the only unknown quantities are the transition coefficients $a_{s,\,s_1}^{(s_2,\,s_3)}$ that correspond to precollision states s and s_1 and allowed pairs of postcollision states (s_2, s_3). Allowed pairs of postcollision states conserve the precollision values for mass, momentum, and energy. For a single-species monatomic gas, as considered herein, the following constraints ensure conservation of these quantities:

$$\mathbf{c}_s + \mathbf{c}_{s_1} = \mathbf{c}_{s_2} + \mathbf{c}_{s_3}, \tag{28}$$

$$\left|\mathbf{c}_s - \mathbf{c}_{s_1}\right| = \left|\mathbf{c}_{s_2} - \mathbf{c}_{s_3}\right|. \tag{29}$$

As indicated earlier, for hard-sphere molecules of diameter d, all allowed collisions are equally probable. This includes the trivial collision, in which the precollision and postcollision states are identical. Since the transition coefficients are equal and sum to unity over the allowed postcollision states, the determination of the collision terms reduces to the task of determining all of the allowed postcollision states for each combination of precollision states. When the number of velocity states becomes large, this is not a trivial task, and neither is encoding these collision terms after they have been determined.

The symbol-manipulation program `Mathematica` (Wolfram, 1996) is applied to determine the collision terms and encode them in FORTRAN 77. Several previously-developed `Mathematica` functions facilitate these operations (Torczynski, 1993). These functions are documented below.

`stategen[state_]`: This function operates on the generator state `state` to generate all other "similar" states by permuting the order of the i, j, k triple and/or changing signs. The output is a sorted list of all states equivalent to the generator state, including the generator state itself.

```
stategen[state_]:=Union[
Permutations[state*{1,1,1}],
Permutations[state*{1,1,-1}],
Permutations[state*{1,-1,1}],
Permutations[state*{-1,1,1}],
Permutations[state*{1,-1,-1}],
Permutations[state*{-1,1,-1}],
Permutations[state*{-1,-1,1}],
Permutations[state*{-1,-1,-1}]]
```

`statesgen[stateg_]`: This function operates on a list of generator states `stateg` by using the function `stategen` to generate all of the states corresponding to each generator in the list. The output is a sorted list of all states in the model.

```
statesgen[stateg_]:=
Union[Flatten[Table[stategen[stateg[[s]]],{s,1,Length[stateg]}],1]]
```

`coll[state0_,states_]`: This function calculates the collision term for the state `state0` colliding with the states in the list `states`. It loops through all possible momentum-conserving collisions and adds the collision to the collision term if energy conservation is also satisfied. The list `states` is usually generated by the function `statesgen`, and the state `state0` is usually one entry from this list. The output is the discrete Boltzmann equation for state `state0` with only the temporal derivative on the left side and with the collision term on the right side lacking the scale factor $\pi d^2 c_0$.

```
coll[state0_,states_]:=
Block[{state1,state2,state3,plus03,smag03,sterm,scoll,s3},
For[s3=1;scoll=0,s3<=Length[states],s3=s3+1,
state3=Part[states,s3];
plus03=state0+state3;
smag03=Apply[Plus,(state0-state3)^2];
sterm={0,0};
Do[
state1=Part[states,s1];
state2=plus03-state1;
If[And[smag03==Apply[Plus,(state1-state2)^2],
Length[Intersection[states,{state2}]]>0],
sterm=sterm+{n[state1][t]*n[state2][t],1}],
{s1,1,Length[states]}
];
scoll=scoll+Sqrt[smag03]*
(Part[sterm,1]/Part[sterm,2]-n[state0][t]*n[state3][t])
];
Expand[scoll]
]
```

`feqngen[states_]`: This function applies the function `coll` successively to each state in the state list `states` to determine all the discrete Boltzmann equations for these states. The output is a list of discrete Boltzmann equations in the same format as output from `coll`. This system of equations can be solved by `Mathematica` using its built-in function `NDSolve`.

```
feqngen[states_]:=
Table[n[states[[s]]]’[t]==coll[states[[s]],states],
{s,1,Length[states]}]
```

`numeqn[eqn_]`: This function converts symbolic numerical constants to decimal numerical constants in the equation `eqn`, usually one of the entries in the equation list output by `feqngen`. This is necessary prior to inputting the equation list output by `feqngen` to the function `fortsub` to output FORTRAN 77 source code.

```
numeqn[eqn_]:=Block[{a,b,c,d,i,j,k,eq},
eq=N[eqn,16] /. {i_,j_,k_}->Round[{i,j,k}];
eq=eq //. {1.*a_->1*a, -1.*b_->-1*b, 0.*c_->0*c, 0.+d_->0+d};
eq]
```

`fortsub[subname_String,eqn_List]`: This function takes the list of equations `eqn_List` output by `feqngen`, processes them using `numeqn`, and outputs a FORTRAN 77 subroutine named `subname_String.for` that encodes these equations, where `subname_String` is a string.

```
fortsub[subname_String,eqn_List]:=
Block[{ls,fr,fs,ss,teststr,indstr,writestr,icount,vec,state,neqn,c1,c2},
ls=Length[eqn];
sls=ToString[ls];
fr={};
Do[
state=eqn[[ss,1,0,1,1]];
AppendTo[fr,n[state][t]->
ToExpression["s["<>ToString[ss]<>"]"]];
AppendTo[fr,(n[state])’[t]->
ToExpression["c["<>ToString[ss]<>"]"]],
{ss,1,ls}
];
fs=OpenAppend[subname<>".for"];
WriteString[fs,"      ","subroutine ",subname,"(s,c)","\n"];
WriteString[fs,"c","\n"];
WriteString[fs,"      ","dimension s(",sls,"), c(",sls,")","\n"];
For[c1=0;ss=1,ss<=ls,ss=ss+1,
state=eqn[[ss,1,0,1,1]];
vec=StringJoin["(",ToString[state[[1]]],",",
ToString[state[[2]]],",",ToString[state[[3]]],")"];
WriteString[fs,"c","\n"];
WriteString[fs,"c state ",ToString[ss],
", state vector ",vec,"\n"];
neqn=numeqn[eqn[[ss]]] /. fr;
teststr=StringReplace[ToString[InputForm[neqn]],
{" "->"","["->"(","]"->")","=="->"="}]<>"$";
icount=0;
c2=0;
While[StringLength[teststr]>1,
c2=c2+1;
indstr=Last[Union[Flatten[StringPosition[StringTake[
teststr,Min[StringLength[teststr],65]],{"+","*","-","$"}]]]];
writestr=StringReplace[StringTake[teststr,indstr],"$"->""];
If[icount==0,WriteString[fs,"      ",writestr,"\n"]];
If[icount==1,WriteString[fs,"     $ ",writestr,"\n"]];
icount=1;
teststr=StringDrop[teststr,indstr]
];
c1=Max[c1,c2]
];
WriteString[fs,"c","\n"];
WriteString[fs,"c continuation lines ",ToString[c1],"\n"];
WriteString[fs,"c","\n"];
WriteString[fs,"      ","return","\n"];
WriteString[fs,"      ","end","\n"];
Close[fs]
]
```

These Mathematica functions are used in the following manner, presented below in the context of the 8-state DVG model (see Table 2 and Figures 2-3).

```
   stateg008={{1,1,1}}
   states008=statesgen[stateg008]
   eqn008=feqngen[states008]
   fortsub["pcl008",eqn008]
```

stateg008: List containing the generator states (only 1) for the 8-state model.

states008: List containing all the states (exactly 8) for the 8-state model.

eqn008: List containing all the discrete Boltzmann equations for the 8-state model.

pcl008.for: File containing a FORTRAN 77 subroutine encoding the collision terms (exactly 8) for the 8-state model. This file is displayed in Figure 3, where implicit double precision (a-h,o-z), parameter, dimension, and comment statements have been added or modified.

The above Mathematica functions can be applied to any of the DVG models described in Table 2. A FORTRAN 77 subroutine produced in this manner returns the collision terms for the velocity states delineated in the comments. The array argument s(ns) contains the number densities of the states, and the array argument c(ns) contains the corresponding collision terms. These collision terms must be multiplied in the calling program by the factor $\pi d^2 c_0$. As discussed below, a subroutine produced in this manner can be linked with the FORTRAN 77 code dvg1d to simulate flow in the geometry shown in Figure 4.

```
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      subroutine pcl008(s,c)
c
      implicit double precision (a-h,o-z)
c
c *** Collision term, dimensionless, for 8-state model.
c
      parameter (ns=8)
c
      dimension s(ns)
      dimension c(ns)
c
c state 1, state vector (-1,-1,-1)
      c(1)=1.4142135623730951*s(2)*s(3)-1.4142135623730951*s(1)*s(4)+
     $ 1.4142135623730951*s(2)*s(5)+1.4142135623730951*s(3)*s(5)+
     $ 0.8660254037844386*s(4)*s(5)-1.4142135623730951*s(1)*s(6)+
     $ 0.8660254037844386*s(3)*s(6)-1.4142135623730951*s(1)*s(7)+
     $ 0.8660254037844386*s(2)*s(7)-2.598076211353316*s(1)*s(8)
c
c state 2, state vector (-1,-1,1)
      c(2)=-1.4142135623730951*s(2)*s(3)+1.4142135623730951*s(1)*s(4)-
     $ 1.4142135623730951*s(2)*s(5)+0.8660254037844386*s(4)*s(5)+
     $ 1.4142135623730951*s(1)*s(6)+0.8660254037844386*s(3)*s(6)+
     $ 1.4142135623730951*s(4)*s(6)-2.598076211353316*s(2)*s(7)+
     $ 0.8660254037844386*s(1)*s(8)-1.4142135623730951*s(2)*s(8)
c
c state 3, state vector (-1,1,-1)
      c(3)=-1.4142135623730951*s(2)*s(3)+1.4142135623730951*s(1)*s(4)-
     $ 1.4142135623730951*s(3)*s(5)+0.8660254037844386*s(4)*s(5)-
     $ 2.598076211353316*s(3)*s(6)+1.4142135623730951*s(1)*s(7)+
     $ 0.8660254037844386*s(2)*s(7)+1.4142135623730951*s(4)*s(7)+
     $ 0.8660254037844386*s(1)*s(8)-1.4142135623730951*s(3)*s(8)
c
c state 4, state vector (-1,1,1)
      c(4)=1.4142135623730951*s(2)*s(3)-1.4142135623730951*s(1)*s(4)-
     $ 2.598076211353316*s(4)*s(5)+0.8660254037844386*s(3)*s(6)-
     $ 1.4142135623730951*s(4)*s(6)+0.8660254037844386*s(2)*s(7)-
     $ 1.4142135623730951*s(4)*s(7)+0.8660254037844386*s(1)*s(8)+
     $ 1.4142135623730951*s(2)*s(8)+1.4142135623730951*s(3)*s(8)
c
c state 5, state vector (1,-1,-1)
      c(5)=-1.4142135623730951*s(2)*s(5)-1.4142135623730951*s(3)*s(5)-
     $ 2.598076211353316*s(4)*s(5)+1.4142135623730951*s(1)*s(6)+
     $ 0.8660254037844386*s(3)*s(6)+1.4142135623730951*s(1)*s(7)+
     $ 0.8660254037844386*s(2)*s(7)+1.4142135623730951*s(6)*s(7)+
     $ 0.8660254037844386*s(1)*s(8)-1.4142135623730951*s(5)*s(8)
c
c state 6, state vector (1,-1,1)
      c(6)=1.4142135623730951*s(2)*s(5)+0.8660254037844386*s(4)*s(5)-
     $ 1.4142135623730951*s(1)*s(6)-2.598076211353316*s(3)*s(6)-
     $ 1.4142135623730951*s(4)*s(6)+0.8660254037844386*s(2)*s(7)-
     $ 1.4142135623730951*s(6)*s(7)+0.8660254037844386*s(1)*s(8)+
     $ 1.4142135623730951*s(2)*s(8)+1.4142135623730951*s(5)*s(8)
c
c state 7, state vector (1,1,-1)
      c(7)=1.4142135623730951*s(3)*s(5)+0.8660254037844386*s(4)*s(5)+
     $ 0.8660254037844386*s(3)*s(6)-1.4142135623730951*s(1)*s(7)-
     $ 2.598076211353316*s(2)*s(7)-1.4142135623730951*s(4)*s(7)-
     $ 1.4142135623730951*s(6)*s(7)+0.8660254037844386*s(1)*s(8)+
     $ 1.4142135623730951*s(3)*s(8)+1.4142135623730951*s(5)*s(8)
c
c state 8, state vector (1,1,1)
      c(8)=0.8660254037844386*s(4)*s(5)+0.8660254037844386*s(3)*s(6)+
     $ 1.4142135623730951*s(4)*s(6)+0.8660254037844386*s(2)*s(7)+
     $ 1.4142135623730951*s(4)*s(7)+1.4142135623730951*s(6)*s(7)-
     $ 2.598076211353316*s(1)*s(8)-1.4142135623730951*s(2)*s(8)-
     $ 1.4142135623730951*s(3)*s(8)-1.4142135623730951*s(5)*s(8)
c
c continuation lines 5
c
      return
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
```

Figure 3. Example of FORTRAN 77 subroutine encoding DVG collision terms.

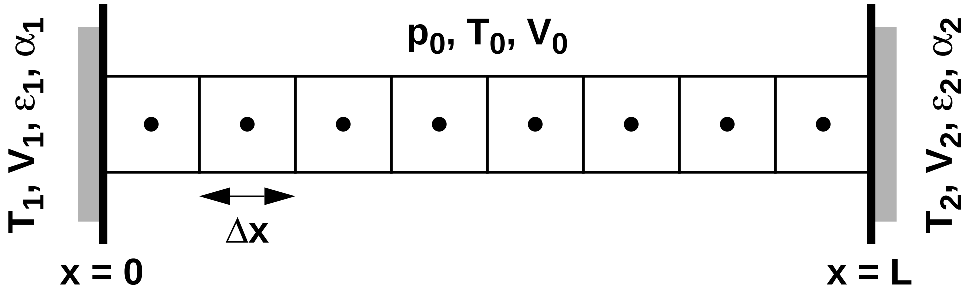
p_0, T_0, V_0
$T_1, V_1, \varepsilon_1, \alpha_1$
$T_2, V_2, \varepsilon_2, \alpha_2$
Δx
x = 0
x = L

Figure 4. Geometry for DVG numerical implementation.

3.4. Numerical Approach

The DVG models discussed previously are implemented in the one-dimensional geometry shown in Figure 4. The domain is bounded by two impermeable walls, where walls 1 and 2 are located at $x = 0$ and $x = L$, have temperatures T_1 and T_2, have tangential velocities V_1 and V_2, have specular-reflection fractions ε_1 and ε_2 between 0 and 1, have bounce-back-reflection fractions α_1 and α_2 between 0 and 1, and have diffuse-isothermal-reflection fractions of $1 - \alpha_1 - \varepsilon_1$ and $1 - \alpha_2 - \varepsilon_2$ between 0 and 1, respectively. No variations are permitted in the y and z directions.

In this domain, the N_s discrete Boltzmann equations for a DVG model with states $\{s\}$ and speed scale c_0 are discretized using a one-dimensional, time-accurate, finite-difference approach with operator splitting and upwinding. This method is used only because of its simple implementation. The one-dimensional domain is divided into N_x uniform cells of width $\Delta x = L/N_x$. Initially, the number densities $\{n_s\}$ are assigned in all cells to the equilibrium values $\{n_s^{eq}\}$ corresponding to a single-species monatomic gas of molecular mass m with pressure p_0, temperature T_0, and tangential velocity V_0. The method of determining these equilibrium values is discussed below. A constant time step $\Delta t < s_{\max} c_0 \Delta x$ is used, and a simulation is run for N_t time steps, where this parameter is usually chosen large enough to achieve steady behavior.

Heuristically, the operator splitting is performed in the following manner:

$$\frac{\partial n_s}{\partial t} = -c_{s,x} \frac{\partial n_s}{\partial x} \text{ (movement),} \tag{30}$$

$$\frac{\partial n_s}{\partial t} = C_s \text{ (collision).} \tag{31}$$

The above movement and collision operators are applied sequentially throughout the simulation. The N_s movement equations do not couple to each other except through the boundary conditions. Although the N_s collision equations are fully coupled, their finite-difference discretization decouples the quantities in each cell from quantities in neighboring cells.

The movement operator is implemented in the following manner. The cell-centered number densities and their updates are denoted by $n_{s,l}$ and $\Delta n_{s,l}$, respectively, where the cell index is $l = 1, \ldots, N_x$. For each time step, the number-density updates are given by

$$\Delta n_{s,l} = (\Delta t / \Delta x)(F_{s,l-1} - F_{s,l}), \tag{32}$$

where the rightward flux at the right boundary of cell l is $F_{s,l}$. At interior right cell boundaries, for which $1 \le l \le (N_x - 1)$, upwinding is used to prescribe the fluxes:

$$F_{s,l} = c_{s,x} \left(\begin{array}{ll} n_{s,l} & \text{for } c_{s,x} > 0 \\ n_{s,l+1} & \text{for } c_{s,x} < 0 \end{array} \right. . \tag{33}$$

The above expression is also used for outgoing fluxes at the two boundaries. Outgoing fluxes at the left boundary have $c_{s,x} < 0$ and $l = 0$, and outgoing fluxes at the right boundary have $c_{s,x} > 0$ and $l = N_x$, so the above expression is well defined.

Boundary conditions are required for incoming fluxes at the boundaries. The flux F_s for an incoming state s at a boundary is formed from a linear combination of the fluxes corresponding to three reflection processes (others like diffuse-adiabatic reflection could be included if desired): bounce-back reflection, specular reflection, and diffuse-isothermal reflection, denoted by F_s^A, F_s^S, and F_s^I, respectively:

$$F_s = \alpha F_s^A + \varepsilon F_s^S + (1 - \alpha - \varepsilon) F_s^I . \qquad (34)$$

The fluxes for bounce-back and specular reflections are easily specified. Let state $s = (i, j, k)$ denote an incoming state. Then the corresponding bounce-back and specular states that reflect into s are outgoing and are given by $s^A = (-i, -j, -k)$ and $s^S = (-i, j, k)$, respectively. Conservation of mass is used to prescribe the corresponding fluxes at the boundaries:

$$F_s^A = -F_{s^A} , \qquad (35)$$

$$F_s^S = -F_{s^S} . \qquad (36)$$

In other words, a bounce-back reflection converts the number density from state s^A into state s, and a specular reflection converts the number density from state s^S into state s. The fluxes on the right sides of the above two equations are well defined by upwinding since they are outgoing.

The fluxes for diffuse-isothermal reflections are specified in terms of the equilibrium distribution at the wall conditions. Let F_{in} and F_{out} be the total rightward incoming and outgoing fluxes at a boundary, respectively:

$$F_{\text{in}} = \sum_{s \text{ incoming}} F_s , \qquad (37)$$

$$F_{\text{out}} = \sum_{s \text{ outgoing}} F_s . \qquad (38)$$

Then, by conservation of mass at an impermeable wall, these two quantities must sum to zero:

$$F_{\text{in}} + F_{\text{out}} = 0 . \qquad (39)$$

Let f_s be the fraction of the total incoming flux that is transported by an incoming state s:

$$F_s = f_s F_{\text{in}} . \qquad (40)$$

Clearly the $\{f_s\}$ all satisfy $0 \le f_s \le 1$ and all sum to unity:

$$\sum_{s \text{ incoming}} f_s = 1 . \tag{41}$$

For diffuse-isothermal reflection from a wall, f_s is related to the equilibrium distribution at the wall conditions, where the proportionality coefficient is independent of s:

$$f_s \propto |c_{s,x}| n_s^{eq} . \tag{42}$$

When combined, these two equations determine the quantity f_s:

$$f_s = \frac{c_{s,x} n_s^{eq}}{\sum_{s_1 \text{ incoming}} c_{s_1,x} n_{s_1}^{eq}} . \tag{43}$$

The absolute values are removed because all incoming states have the same sign for $c_{s,x}$. Thus, the diffuse-isothermal-reflection flux F_s^I for an incoming state s is determined:

$$F_s^I = -\left(\frac{c_{s,x} n_s^{eq}}{\sum_{s_1 \text{ incoming}} c_{s_1,x} n_{s_1}^{eq}} \right) \left(\sum_{s_2 \text{ outgoing}} F_{s_2} \right), \tag{44}$$

where the equilibrium number densities $\{n_s^{eq}\}$ are evaluated at the wall conditions.

Thus, Equations (34), (35), (36), and (44) prescribe the flux of an incoming state at a boundary, where an incoming state has velocity $\mathbf{c}_s$ pointing into the gas region and out of the solid region. These equations, along with Equations (32) and (33), completely specify the movement operator. Determination of the equilibrium number densities is discussed below.

The collision operator is implemented in the following manner. Since the cell-centered finite-difference discretization discussed above decouples the collision operators of adjacent cells (i.e., the quantities in Equation (31) are replaced by their cell-centered quantities), the resulting ordinary differential equations (ODEs) are integrated forward in time on a cell-by-cell basis. For simplicity, a purely explicit integration scheme that enforces conservation is implemented. The time step Δt is subdivided into N_{ODE} smaller time steps of $\Delta t_{\text{ODE}} = \Delta t / N_{ODE}$. The explicit update is given below, where the integer index $0 \le r \le N_{\text{ODE}}$ represents a time $t + r\Delta t_{\text{ODE}}$, $r = 0$ represents the precollision states, and $r = N_{\text{ODE}}$ represents the postcollision states:

$$n_s^{(r)} = n_s^{(r-1)} + \Delta t_{ODE} C_s^{(r-1)} . \tag{45}$$

The equilibrium number densities $\{n_s^{eq}\}$, needed to specify the initial conditions and the diffuse-isothermal-reflection boundary condition, are determined simply by integrating the collision operator forward in time until equilibrium is obtained (i.e., the number densities stop changing). However, initial conditions are required that satisfy the prescribed macroscopic quantities: a number density of n, a temperature of T, a velocity x-component of 0, a velocity y-component of V, and a velocity z-component of 0. These constraints are expressed by the equations below:

$$\sum_s n_s^{init} = n\,, \tag{46}$$

$$\sum_s n_s^{init}\mathbf{c}_s = nV\hat{\mathbf{e}}_y\,, \tag{47}$$

$$\sum_s n_s^{init}\frac{1}{2}m\left|\mathbf{c}_s - V\hat{\mathbf{e}}_y\right|^2 = n\frac{3}{2}k_BT\,. \tag{48}$$

With some algebra, these expressions can be manipulated into more convenient forms:

$$\sum_s\left(\frac{n_s^{init}}{n}\right)\mathbf{s} = \left(\frac{V}{c_0}\right)\hat{\mathbf{e}}_y\,, \tag{49}$$

$$\sum_s\left(\frac{n_s^{init}}{n}\right)s^2 = \frac{3}{2}\left(\frac{c_m}{c_0}\right)^2 + \left(\frac{V}{c_0}\right)^2. \tag{50}$$

Here, $(\mathbf{c}_s/c_0) = \mathbf{s} = i\hat{\mathbf{e}}_x + j\hat{\mathbf{e}}_y + k\hat{\mathbf{e}}_z$, $(c_s/c_0)^2 = s^2 = i^2 + j^2 + k^2$, and $c_m = \sqrt{2k_BT/m}$. Thus, the fraction of the total number density in each state depends only on the states themselves and the two dimensionless parameters (c_m/c_0) and (V/c_0). These equations clearly do not have a unique solution when there are more than four states. One generic solution for symmetric sets of states that is close to the equilibrium solution is presented below:

$$(n_s^{init}/n) = (B^{s^2}(1 + Aj))/\left(\sum_{s_1}B^{s_1^2}\right), \tag{51}$$

$$A = \left((V/c_0)\left(\sum_s B^{s^2}\right)\right)/\left(\sum_s B^{s^2}j^2\right), \tag{52}$$

$$\sum_s B^{s^2}s^2 = \{(3/2)(c_m/c_0)^2 + (V/c_0)^2\}\sum_s B^{s^2}\,. \tag{53}$$

When c_0, c_m, and V are given, it is necessary to solve the above equation numerically for B. This polynomial equation has exactly one finite positive solution when the following constraint is satisfied:

$$s_{\min}^2 < \left(\frac{3}{2}\right)\left(\frac{c_m}{c_0}\right)^2 + \left(\frac{V}{c_0}\right)^2 < s_{\max}^2 . \qquad (54)$$

When the gas is motionless (i.e., $V = 0$), this is identical to the constraint in Table 2. A standard Newton-Raphson technique is used to solve the polynomial equation for B. Alternatively, if c_0 is viewed as unknown and B is viewed as known, the polynomial equation for B is easily solved in closed form for c_0.

The above approach for determining the equilibrium distribution is applicable only to multispeed models. For a unispeed model, such as the 8-state model in Table 2, the speed scale c_0 is determined directly through the equation

$$c_0^2 s^2 = \frac{3k_B T}{m} + V^2 . \qquad (55)$$

This limits the applicability of the 8-state model to isothermal, extremely subsonic flows.

3.5. Computational Implementation

The above numerical approach is implemented in the FORTRAN 77 code dvg1d. This code reads two input files and writes two output files. The first input file is dvg1d_inp.dat. This file contains the basic physical and numerical parameters to set up and run a dvg1d simulation. The second input file is dvg1d_sta.dat. This file contains a list of the velocity states (see Figure 2 and Table 2). Two numerical parameters for a simulation are not included in these files but are included in the parameter statements in dvg1d: ns, the number of velocity states; and nx, the number of cells. These parameter statements must be checked for consistency prior to any run. In particular, the number of velocity states ns must be consistent with the file dvg1d_sta.dat. Examples of these two files are provided below.

dvg1d_sta.dat: This input file provides the velocity states for dvg1d (see Figure 2 and Table 2). The file for the 8-state model is shown below. The first column is a reference index, and the next three columns are the i, j, k components of the velocity state s.

```
1  -1  -1  -1
2  -1  -1   1
3  -1   1  -1
4  -1   1   1
5   1  -1  -1
6   1  -1   1
7   1   1  -1
8   1   1   1
```

`dvg1d_inp.dat`: This input file provides the inputs for a `dvg1d` simulation (see Figure 4). The lines alternate between an explanatory comment and the corresponding numerical value. The values below are in MKS units; however, any consistent set of units is acceptable. If more than one value appears on a line (say, for reference), only the first value is used.

```
Boltzmann constant
1.380658D-023
Collision flag (0 = no, 1 = yes)
1.
Viscosity ratio, hard sphere
1.016034
Thermal-conductivity ratio, hard sphere
1.025218
Molecular mass
66.3D-027
Temperature reference
273.15
Viscosity reference
2.117D-005
Pressure initial
266.644 133.322 13.3322 1.33322 .133322
Temperature initial
273.15
Velocity_y initial
0.
Temperature wall 1
253.15
Temperature wall 2
293.15
Velocity_y wall 1
0.
Velocity_y wall 2
0.
Symmetric fraction wall 1
0.
Symmetric fraction wall 2
0.
Antisymmetric fraction wall 1
0.
Antisymmetric fraction wall 2
0.
Domain length
0.001
Gravitational acceleration in y-direction
0.
Time step for move
1.0D-009
Number of move time steps
1000000
Number of ODE time steps per move time step
1
Speed scale of states
180.
```

The two output files `dvg1d_ini.dat` and `dvg1d_out.dat` have the same format and differ only in that `dvg1d_ini.dat` reports the initial values, whereas `dvg1d_out.dat` reports the final values. This latter file is updated throughout the simulation so that intermediate results can be examined. Quantities in these files correspond to the averages of the precollision and postcollision values. The rows in these files report values at successive positions across the domain. The quantities in the columns are delineated below.

```
Column  1:  index of x cell
Column  2:  x position
Column  3:  number density
Column  4:  mass density
Column  5:  temperature
Column  6:  pressure
Column  7:  x component of velocity
Column  8:  y component of velocity
Column  9:  z component of velocity
Column 10:  x component of energy flux
Column 11:  y component of energy flux
Column 12:  z component of energy flux
Column 13:  xy component of stress
Column 14:  xz component of stress
Column 15:  yz component of stress
Column 16:  x temperature
Column 17:  y temperature
Column 18:  z temperature
Column 19:  x pressure (xx component of stress)
Column 20:  y pressure (yy component of stress)
Column 21:  z pressure (zz component of stress)
```

The code `dvg1d` resides in the file `dvg1d.for` and can be compiled with standard FORTRAN 77 compilers. It is important to select a compiler option ensuring double precision of numerical constants in the code. One example is provided below in which the `pgf77` compiler is used with the `-r8` flag to enforce double precision. The file `pcl.for` contains the collision subroutine, an example of which is shown in Figure 3 for the 8-state model.

```
% pgf77 -r8 -o dvg1d dvg1d.for pcl.for
```

4. Couette-Flow Benchmark Problem

4.1. Overview

The benchmark problem of Couette flow is examined. This mathematically one-dimensional steady-state situation has a hard-sphere monatomic gas confined between a plane of antisymmetry and a diffuse-isothermal moving wall. Of particular interest are the velocity profile and the shear stress (constant across the domain) for near-continuum to free-molecular conditions. The DSMC, NSSJ, and DVG methods are applied to this problem. DSMC is considered to provide the "exact" solution to within the uncertainties arising from discretization and statistical effects. For all conditions, the NSSJ results differ by at most several percent from the DSMC results. DVG results are provided for several sets of velocity states and speed scales. For each set of flow conditions, the velocity profile depends only modestly on the model, but the shear stress (the quantity of interest for predicting structural response) depends strongly on the model.

4.2. Problem Description

Couette flow is a special case of the more general flow shown in Figure 4. A monatomic gas is confined between two parallel planes. The right plane is a diffuse-isothermal wall with prescribed temperature and tangential velocity, here taken to be upward, and the left plane is taken to be a bounce-back wall (i.e., antisymmetric reflection, as discussed above). In most other studies, these two planes are diffuse-isothermal walls with equal temperatures but opposite tangential velocities. With that choice, the plane halfway between the two walls is a plane of antisymmetry, so the use of a boundary with antisymmetric reflection reduces the domain by half.

For the nonturbulent flows considered here, the long-time behavior of this flow problem is steady. This steady flow depends only on the molecular mass m, the hard-sphere diameter d (or, equivalently, the absolute viscosity μ_{ref} at a reference temperature T_{ref}), the average number density n in the domain (or, equivalently, the initial pressure p_0 and the initial temperature T_0, here taken to be the wall value), the wall temperature T_2, the wall tangential velocity V_2, and the domain length L. Of particular interest are the x-profile of the y-component of velocity v and the wall shear stress τ (the shear stress is constant across the domain).

In the limit of small velocities, for which the flow is isobaric ($p = p_0$) and isothermal ($T = T_0 = T_2$) and for which the DVG model can be applied, the NSSJ model yields a closed-form solution (Vincenti and Kruger, 1965):

$$v = \frac{V_2 x}{L+\lambda},\ \tau = \frac{\mu V_2}{L+\lambda},\ \lambda = \frac{2\mu}{\rho\bar{c}},\ \rho = mn = \frac{mp}{k_B T},\ \bar{c} = \sqrt{\frac{8k_B T}{\pi m}}. \tag{56}$$

Here, λ is the mean free path, ρ is the mass density, and $\bar{c}$ is the average molecular speed. This result is correct in the continuum and near-continuum regimes, for which $\lambda/L \ll 1$. Interestingly, this result also is correct in free-molecular limit, for which $\lambda/L \to \infty$.

4.3. Results

The DSMC, NSSJ, and DVG methods are applied to the Couette-flow benchmark problem discussed above. DSMC simulations are performed using the code `DSMC1` (Bird, 1998), which is modified to incorporate precollision and postcollision sampling (Torczynski et al., 2003). This code accurately predicts noncontinuum gas flows in the above geometry (Gallis et al., 2004). NSSJ results are generated by evaluation of the closed-form expressions above. DVG simulations are performed using the code `dvg1d`, for which the numerical approach and the computational implementation are discussed extensively in previous sections. The relevant physical parameters for the simulations are shown in the table below and follow the order in the file `dvg1d_inp.dat`.

Table 3: Relevant physical parameters in Couette-flow simulations.

Quantity	Symbol	Value	Comment
Boltzmann constant	k_B	1.380658×10^{-23} J/K	Bird (1998)
Viscosity ratio	μ_∞ / μ_1	1.016034	CE hard-sphere
Thermal conductivity ratio	K_∞ / K_1	1.025218	CE hard-sphere
Molecular mass	m	66.3×10^{-27} kg	argon
Temperature reference	T_{ref}	273.15 K = 0 ˚C	standard
Viscosity reference	μ_{ref}	2.117×10^{-5} Pa · s	argon
Pressure, initial	p_0	266.644 Pa = 2000 mtorr 133.322 Pa = 1000 mtorr 13.3322 Pa = 100 mtorr 1.33322 Pa = 10 mtorr 0.13322 Pa = 1 mtorr	near- continuum near-free-molecular
Temperature, initial	T_0	273.15 K = 0 ˚C	standard
Temperature, wall 2	T_2	273.15 K = 0 ˚C	standard
Tangential velocity, wall 2	V_2	10 m/s	much less than $\bar{c}$
Domain length	L	0.001 m = 1 mm	Gallis et al., (2004)

The following numerical parameters are used in the `DSMC1` simulations. Hard-sphere molecules are used, which have $\omega = 1/2$ and $\alpha = 1$. The computational domain is divided into 200 uniform cells, each having a size of 5 μm. A time step of 7 ns is used. The average number of computational molecules per cell is 30. The gas is initially motionless at the given conditions. Usually, 32 independent simulations are performed, which enables about 2 billion precollision and postcollision steady-state samples to be acquired for each cell. These parameters easily satisfy Bird's criteria (Bird, 1998) and are known to produce steady-state results that are accurate to well within 1% (Gallis et al., 2004).

The following numerical parameters are used in the `dvg1d` simulations and are specified either in `dvg1d` itself or in the input file `dvg1d_inp.dat`. Hard-sphere molecules are used inherently by the DVG method. The computational domain is divided into $N_x = 100$ uniform cells, each having a size of $\Delta x = 10\ \mu\text{m}$. An overall time step of $\Delta t = 1$ ns is used, with $N_{\text{ODE}} = 1$ ODE time step per overall time step, so that the time step to integrate the collision terms is $\Delta t_{\text{ODE}} = 1$ ns. The gas is initially moving with the wall velocity at the indicated temperature (also the wall value) and pressure. The flow properties are reported after $N_t = 1{,}000{,}000$ time steps. Simulations using other values of these parameters indicate that the above choices produce numerical uncertainties of well less than 1% in the results.

DVG simulations are performed with several sets of velocity states $\{s\}$ and with several values of the speed scale c_0. As indicated above, these quantities completely specify a DVG model. Four sets of velocity states are used (see Table 2): the 8-state, 32-state, 56-state, and 88-state models. As indicated in Table 2, the square of the ratio of the speed scale c_0 to the Maxwellian most probable speed $c_m = \sqrt{2k_BT/m}$ is constrained to lie in a certain range for a given DVG model; otherwise, the DVG model cannot represent the state of the gas. Eight values of this quantity are selected that satisfy the constraint in Table 2 for the 88-state model: $(c_0/c_m)^2 = 0.108739$, 0.126578, 0.166666, 0.225028, 0.284801, 0.351606, 0.425443, and 0.500000. Simulations with one of the other models are performed using only the values that satisfy that model's constraint. For example, simulations with the 8-state model use only the value 0.500000 (see Table 2).

Figures 5-10 show DSMC, NSSJ, and DVG velocity profiles for the above conditions. Each plot shows the simulation method used and the profiles at the 5 pressures that are examined. All six sets of profiles show the same qualitative behavior. The velocity at the left boundary is zero, the velocity rises linearly or slightly more rapidly across the domain, and the gas velocity near the wall differs by a finite amount from the wall value. This velocity slip is very small at near-continuum conditions (the highest pressure) but becomes comparable to the wall velocity at near-free-molecular conditions (the lowest pressure). The NSSJ and DVG profiles have less curvature and larger slips at the walls than the corresponding DSMC profiles. Only a selected subset of the DVG results are displayed in these figures. This is because the DVG profiles exhibit only minor variations for all of the different models and the different values of $(c_0/c_m)^2$ that are examined.

Figures 11-16 show the corresponding wall shear stresses from the three methods. The first five figures exhibit the shear stress at one of the prescribed pressures (from lowest to highest) as a function of the parameter $(c_0/c_m)^2$. Since the DSMC and NSSJ values are independent of this parameter, they appear as horizontal lines. Although the NSSJ values are always within a few percent of the DSMC values, the DVG values differ significantly from the DSMC values as the parameter $(c_0/c_m)^2$ is varied, especially at lower pressures (i.e., near-free-molecular flow). These differences are reduced slightly as the number of velocity states is increased from 8 to 88. Figure 16 summarizes the shear-stress results for all cases examined. Although the DSMC and NSSJ values are in close agreement at all pressures and tend to the correct limits, the DVG results at each pressure exhibit a substantial variation about the DSMC value (a log scale is used).

Thus, the DVG method produces comparably accurate velocity profiles but far less accurate shear-stress values than the NSSJ method does, even at near-free-molecular conditions, despite the much greater computational effort required by the DVG method.

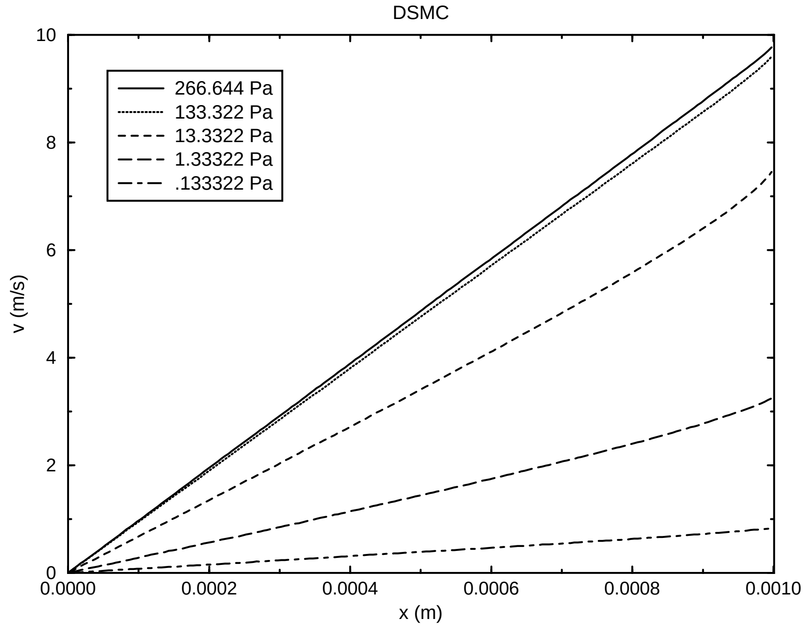

Figure 5. DSMC velocity profiles.

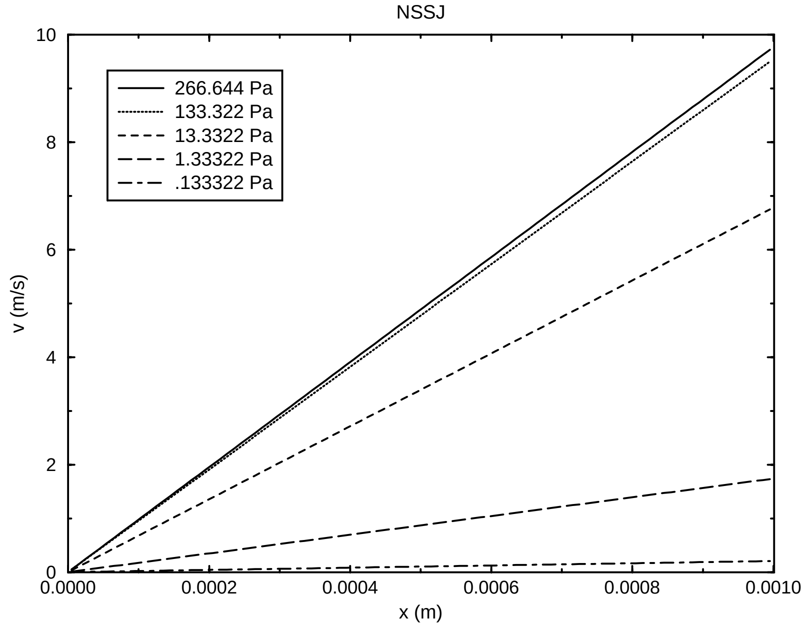

Figure 6. NSSJ velocity profiles.

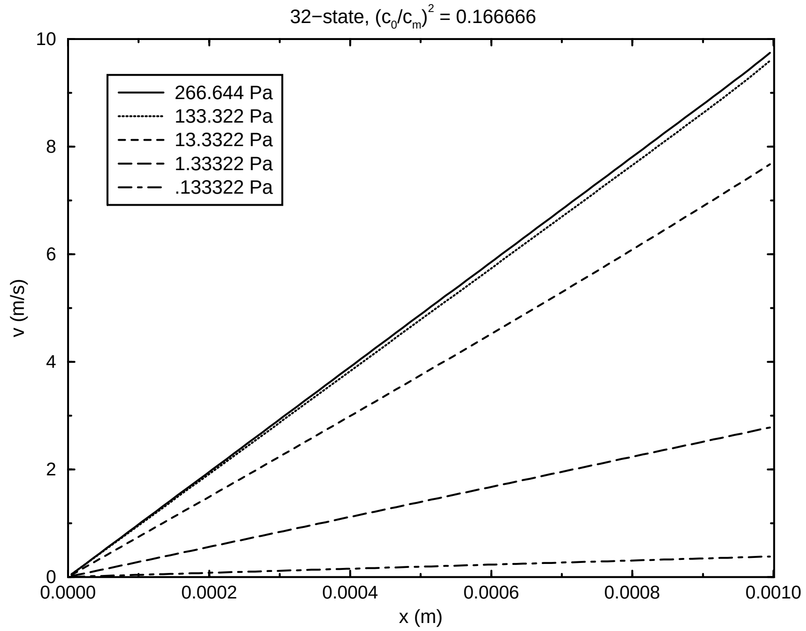

Figure 7. DVG velocity profiles: 32 states, $(c_0/c_m)^2 = 0.166666$.

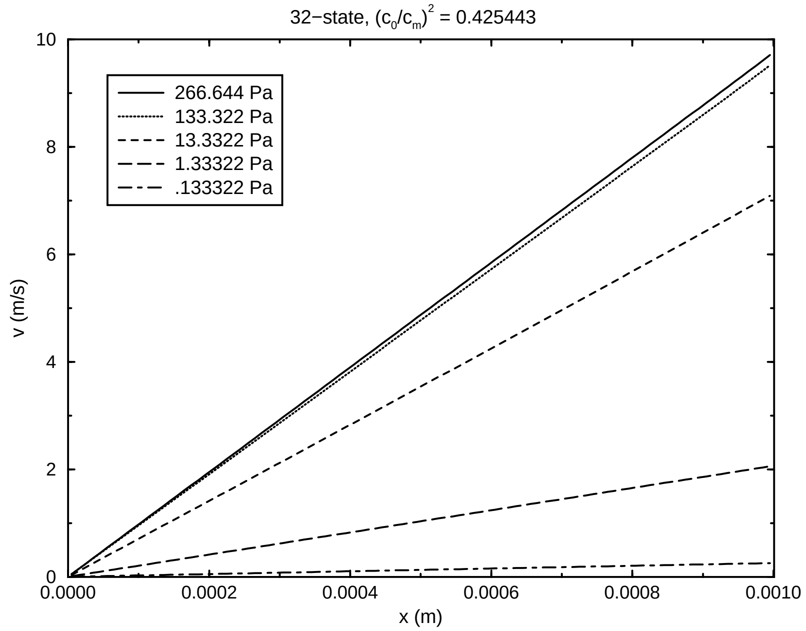

Figure 8. DVG velocity profiles: 32 states, $(c_0/c_m)^2 = 0.425443$.

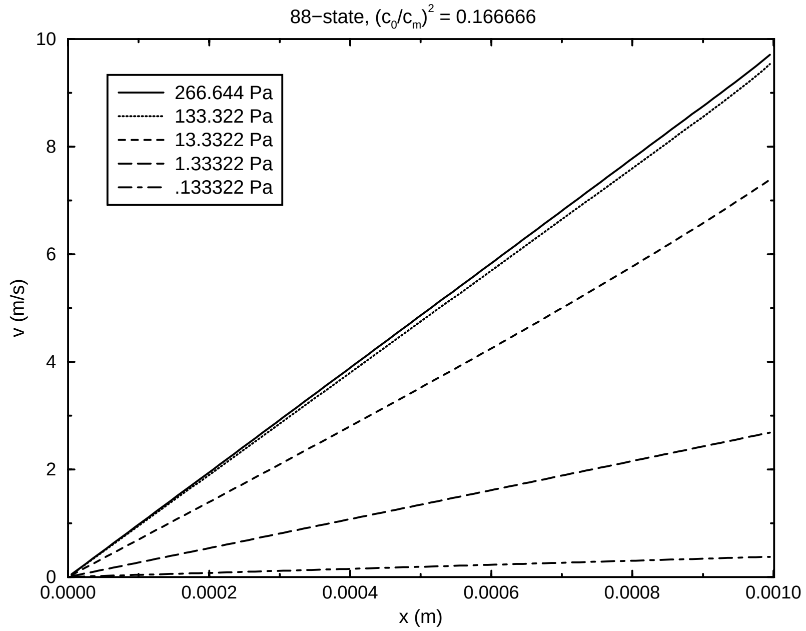

Figure 9. DVG velocity profiles: 88 states, $(c_0/c_m)^2 = 0.166666$.

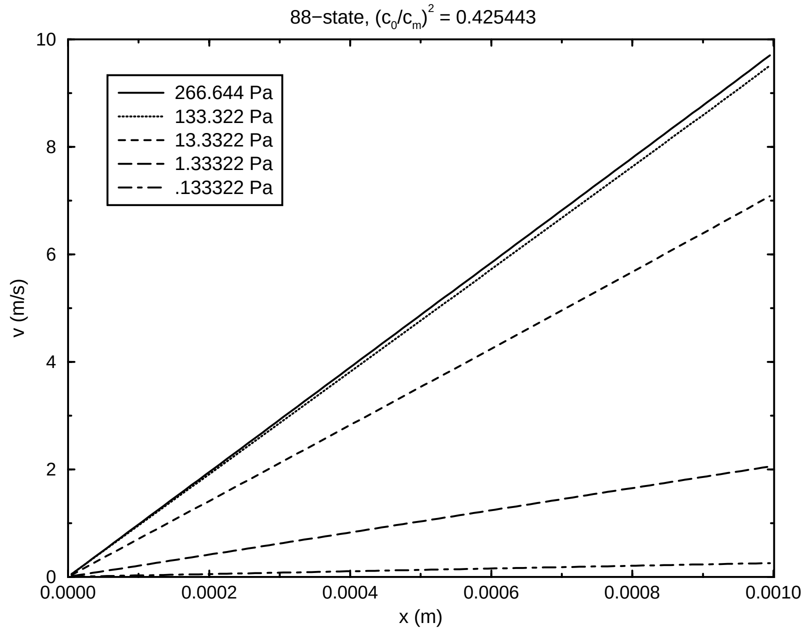

Figure 10. DVG velocity profiles: 88 states, $(c_0/c_m)^2 = 0.425443$.

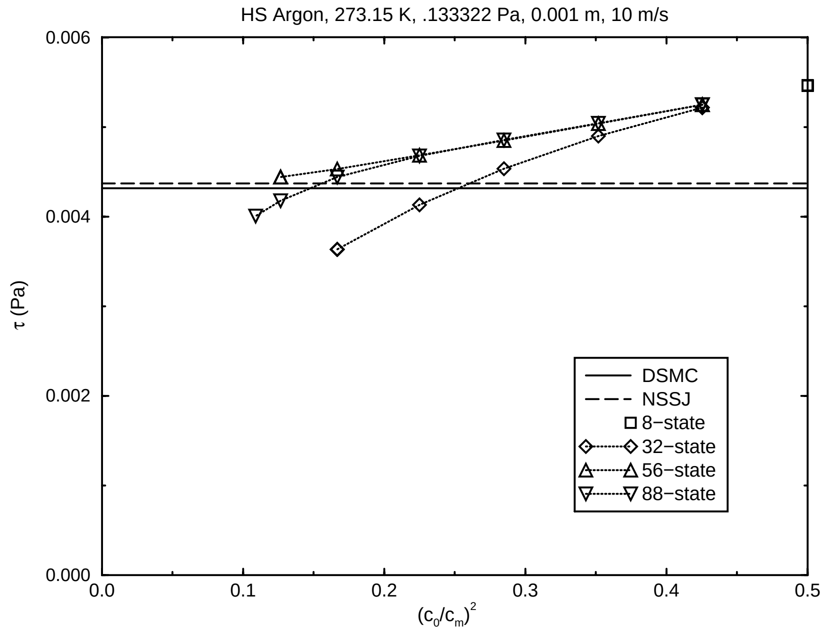

Figure 11. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 0.133322$ Pa.

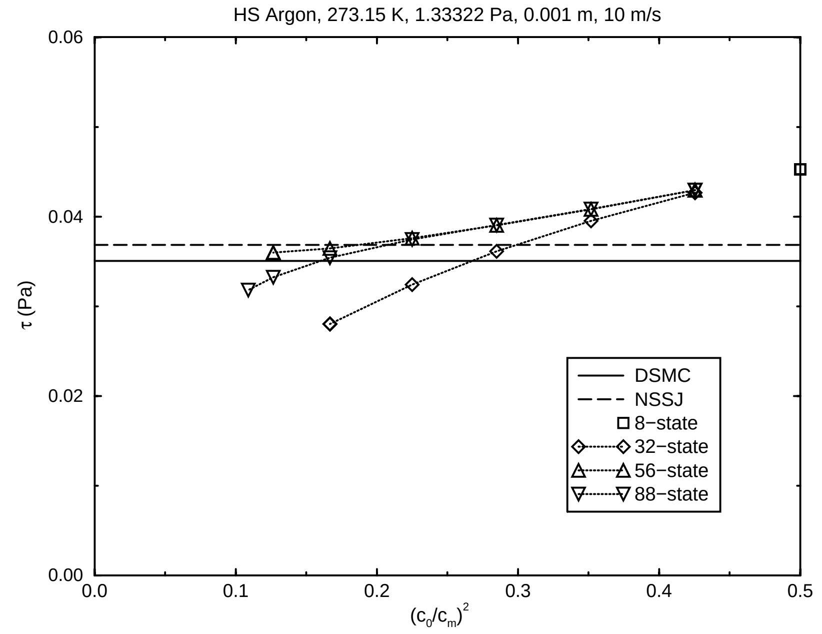

Figure 12. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 1.33322$ Pa.

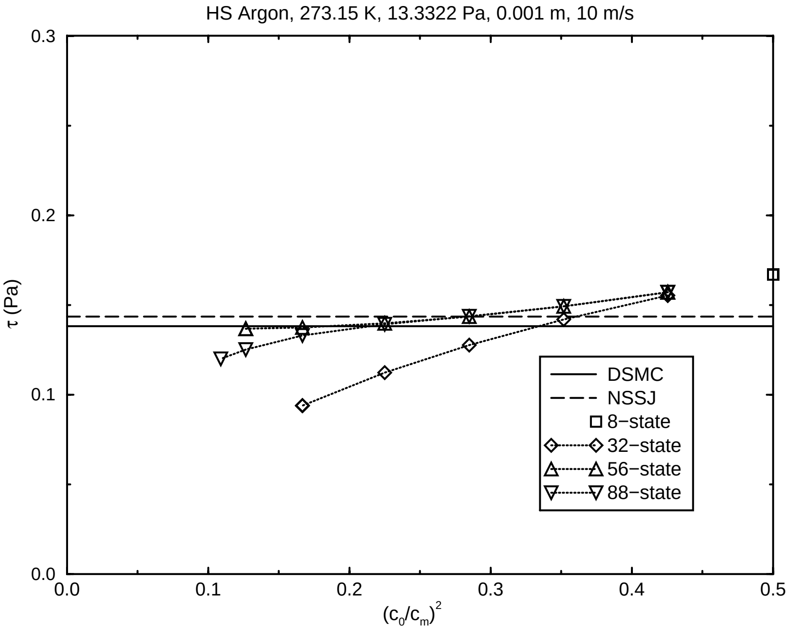

Figure 13. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 13.3322$ Pa.

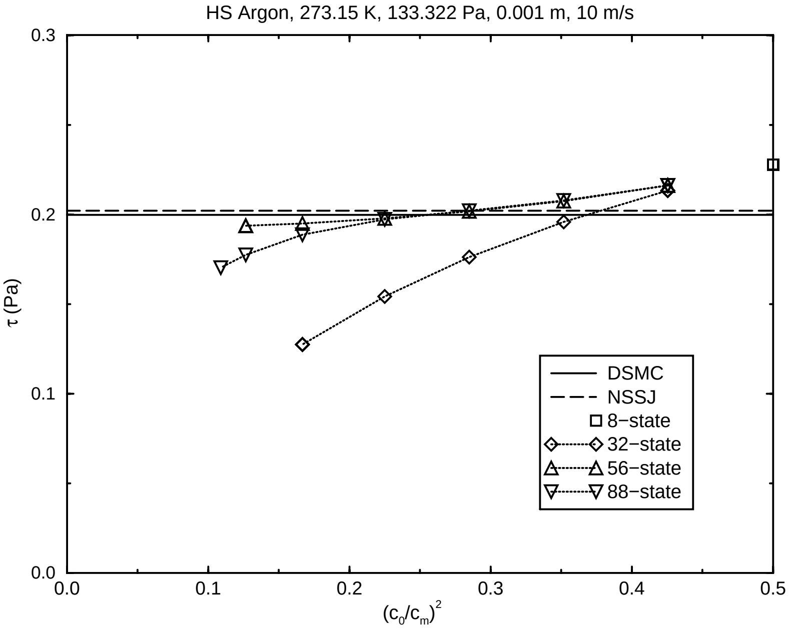

Figure 14. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 133.322$ Pa.

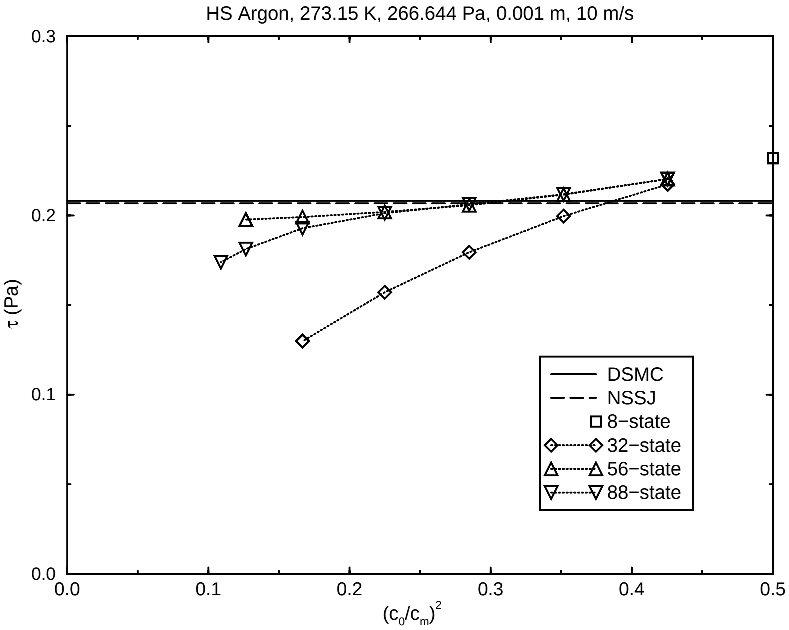

Figure 15. DSMC, NSSJ, and DVG shear stress τ vs. $(c_0/c_m)^2$ at $p = 266.644$ Pa.

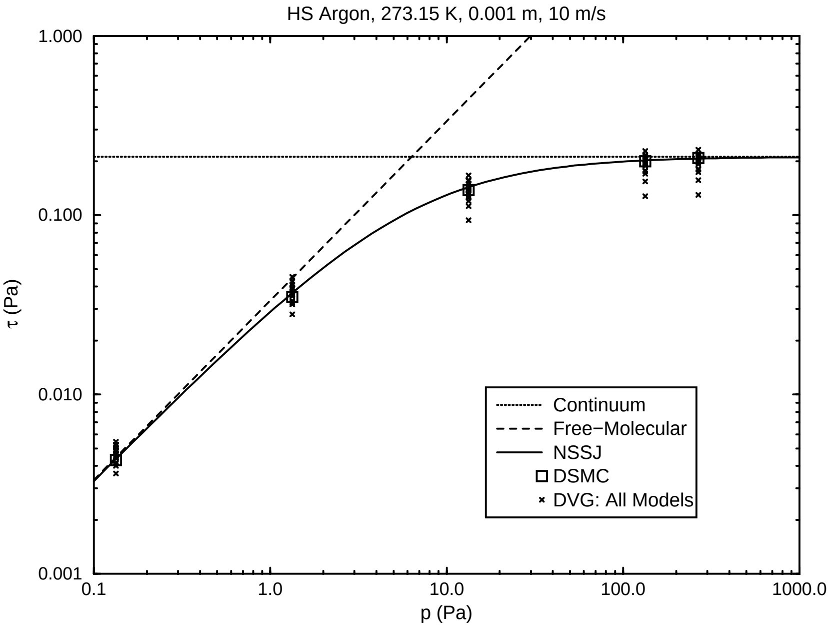

Figure 16. DSMC, NSSJ, and DVG shear stress τ vs. pressure p.

5. Fourier-Flow Benchmark Problem

5.1. Overview

The benchmark problem of Fourier flow is examined. This mathematically one-dimensional steady-state situation has a hard-sphere monatomic gas confined between two motionless diffuse-isothermal walls at different temperatures. Of particular interest are the temperature profile and the heat flux (constant across the domain) for near-continuum to free-molecular conditions. The DSMC, NSSJ, and DVG methods are applied to this problem. DSMC is considered to provide the "exact" solution to within the uncertainties arising from discretization and statistical effects. For all conditions, the NSSJ results differ by at most several percent from the DSMC results. DVG results are provided for several sets of velocity states and speed scales. For each set of flow conditions, the temperature profile depends only modestly on the model, but the heat flux (the quantity of interest for predicting structural response) depends strongly on the model.

5.2. Problem Description

Fourier flow is a special case of the more general flow shown in Figure 4. A monatomic gas is confined between two parallel planes. The left and right planes are motionless diffuse-isothermal walls with prescribed temperatures, where for convenience the left wall is colder and the right wall is hotter. Unlike the situation of Couette flow, no plane of antisymmetry exists, so the extent of the domain cannot be reduced.

Over long times, this flow is steady. This flow depends only on the molecular mass m, the hard-sphere diameter d (or, equivalently, the thermal conductivity K_{ref} at a reference temperature T_{ref}), the average number density n (or, equivalently, the initial pressure p_0 and temperature T_0), the wall temperatures T_1 and T_2, and the domain length L. Of particular interest are the x-profile of the temperature T and the wall heat flux q (constant across the domain).

In the limit of small temperature differences, for which the temperature dependence of all other thermodynamics quantities can be ignored and for which the DVG model can be applied, the NSSJ model yields a closed-form solution (Vincenti and Kruger, 1965):

$$T = \left(\frac{T_1 + T_2}{2}\right) + \left(\frac{T_2 - T_1}{2}\right)\left(\frac{2x - L}{L + (15/4)\lambda}\right),\ q = \frac{K(T_2 - T_1)}{L + (15/4)\lambda},\ \lambda = \frac{2\mu}{\rho\bar{c}},\ \rho = mn = \frac{mp}{k_B T},$$

$$\bar{c} = \sqrt{\frac{8k_B T}{\pi m}}. \tag{57}$$

Here, λ is the mean free path, ρ is the mass density, and $\bar{c}$ is the average molecular speed. This result is correct in the continuum and near-continuum regimes, for which $\lambda/L \ll 1$. Interestingly, this result also is correct in free-molecular limit, for which $\lambda/L \to \infty$.

5.3. Results

The DSMC, NSSJ, and DVG methods are applied to the Fourier-flow benchmark problem discussed above. DSMC simulations are performed using the code DSMC1 (Bird, 1998), which is modified to incorporate precollision and postcollision sampling (Torczynski et al., 2003). This code accurately predicts noncontinuum gas flows in the above geometry (Gallis et al., 2004). NSSJ results are generated by evaluation of the closed-form expressions above. DVG simulations are performed using the code dvg1d, for which the numerical approach and the computational implementation are discussed extensively in previous sections. The relevant physical parameters for the simulations are shown in the table below and follow the order in the file dvg1d_inp.dat.

Table 4: Relevant physical parameters in Fourier-flow simulations.

Quantity	Symbol	Value	Comment
Boltzmann constant	k_B	1.380658×10^{-23} J/K	Bird (1998)
Viscosity ratio	μ_∞/μ_1	1.016034	CE hard-sphere
Thermal conductivity ratio	K_∞/K_1	1.025218	CE hard-sphere
Molecular mass	m	66.3×10^{-27} kg	argon
Temperature reference	T_{ref}	273.15 K = 0 ˚C	standard
Viscosity reference	μ_{ref}	2.117×10^{-5} Pa · s	argon
Pressure, initial	p_0	266.644 Pa = 2000 mtorr 133.322 Pa = 1000 mtorr 13.3322 Pa = 100 mtorr 1.33322 Pa = 10 mtorr 0.13322 Pa = 1 mtorr	near- continuum near-free-molecular
Temperature, initial	T_0	273.15 K = 0 ˚C	standard
Temperature, wall 1	T_1	253.15 K = -20 ˚C	colder
Temperature, wall 2	T_2	293.15 K = 20 ˚C	hotter
Domain length	L	0.001 m = 1 mm	Gallis et al., (2004)

The following numerical parameters are used in the DSMC1 simulations. Hard-sphere molecules are used, which have $\omega = 1/2$ and $\alpha = 1$. The computational domain is divided into 200 uniform cells, each having a size of 5 µm. A time step of 7 ns is used. The average number of computational molecules per cell is 30. The gas is initially motionless at the given conditions. Usually, 32 distinct simulations are performed, which enables about 2 billion precollision and postcollision steady-state samples to be acquired for each cell. These parameters easily satisfy Bird’s criteria (Bird, 1998) and are known to produce steady-state results that are accurate to well within 1% (Gallis et al., 2004).

The following numerical parameters are used in the `dvg1d` simulations and are specified either in `dvg1d` itself or in the input file `dvg1d_inp.dat`. Hard-sphere molecules are used inherently by the DVG method. The computational domain is divided into $N_x = 100$ uniform cells, each having a size of $\Delta x = 10\ \mu\mathrm{m}$. An overall time step of $\Delta t = 1\ \mathrm{ns}$ is used, with $N_{\mathrm{ODE}} = 1$ ODE time step per overall time step, so that the time step to integrate the collision terms is $\Delta t_{\mathrm{ODE}} = 1\ \mathrm{ns}$. The gas is initially motionless at the indicated temperature and pressure. The flow properties are reported after $N_t = 1{,}000{,}000$ time steps. Simulations using other values of these parameters indicate that the above choices produce numerical uncertainties of well less than 1% in the results.

DVG simulations are performed with several sets of velocity states $\{s\}$ and with several values of the speed scale c_0. As indicated above, these quantities completely specify a DVG model. Three sets of velocity states are used (see Table 2): the 32-state, 56-state, and 88-state models (the 8-state model cannot represent nonisothermal flows). As indicated in Table 2, the square of the ratio of the speed scale c_0 to the Maxwellian most probable speed $c_m = \sqrt{2k_BT/m}$ is constrained to lie in a certain range for a given DVG model; otherwise, the DVG model cannot represent the state of the gas. Eight values of this quantity are selected that satisfy the constraint in Table 2 for the 88-state model: $(c_0/c_m)^2 = 0.108739$, 0.126578, 0.166666, 0.225028, 0.284801, 0.351606, 0.425443, and 0.500000. Simulations with one of the other models are performed using only the values that satisfy that model's constraint. For example, simulations with the 8-state model use only the value 0.500000 (see Table 2).

Figures 17-22 show DSMC, NSSJ, and DVG temperature profiles for the above conditions. Each plot shows the simulation method used and the profiles at the 5 pressures that are examined. All six sets of profiles show the same qualitative behavior. The temperature rises fairly linearly across the domain, and the temperatures near the walls differ by a finite amount from the wall values. These temperature jumps are very small at near-continuum conditions (the highest pressure) but become comparable to the wall temperatures at near-free-molecular conditions (the lowest pressure). The NSSJ and DVG profiles have less curvature and larger jumps at the walls than the corresponding DSMC profiles. Only a selected subset of the DVG results are displayed in these figures. This is because the DVG profiles exhibit only minor variations for all of the different models and the different values of $(c_0/c_m)^2$ that are examined.

Figures 23-28 show the corresponding wall heat flux from the three methods. The first five figures exhibit the heat flux at one of the prescribed pressures (from lowest to highest) as a function of the parameter $(c_0/c_m)^2$. Since the DSMC and NSSJ values are independent of this parameter, they appear as horizontal lines. Although the NSSJ values are always within a few percent of the DSMC values, the DVG values at all pressures differ tremendously from the DSMC values as the parameter $(c_0/c_m)^2$ is varied. These differences are reduced slightly as the number of velocity states is increased from 32 to 88. Figure 28 summarizes the heat-flux results for all cases examined. Although the DSMC and NSSJ values are in close agreement at all pressures and tend to the correct limits, the DVG results at each pressure exhibit a substantial variation about the DSMC value (a log scale is used).

Thus, the DVG method produces comparably accurate temperature profiles but far less accurate heat-flux values than the NSSJ method does, even at near-free-molecular conditions, despite the much greater computational effort required by the DVG method.

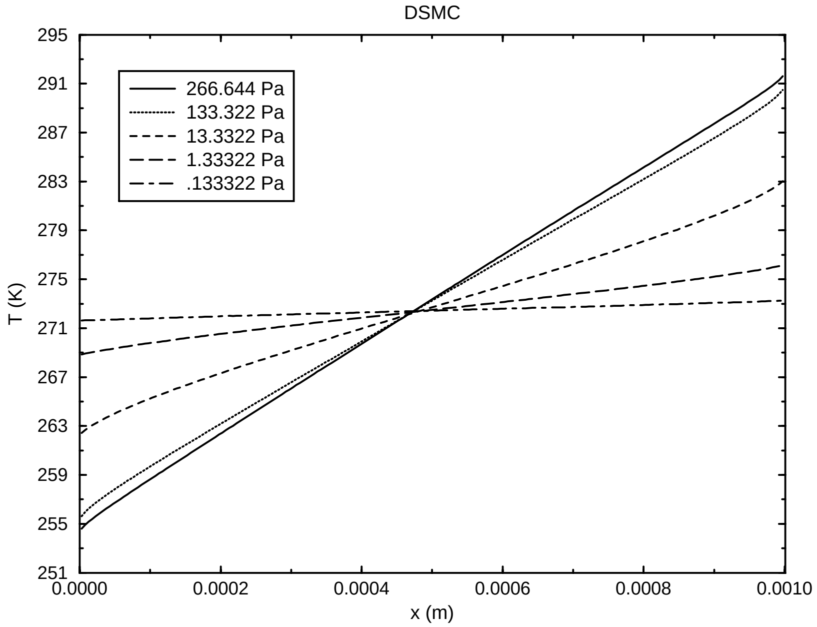

Figure 17. DSMC temperature profiles.

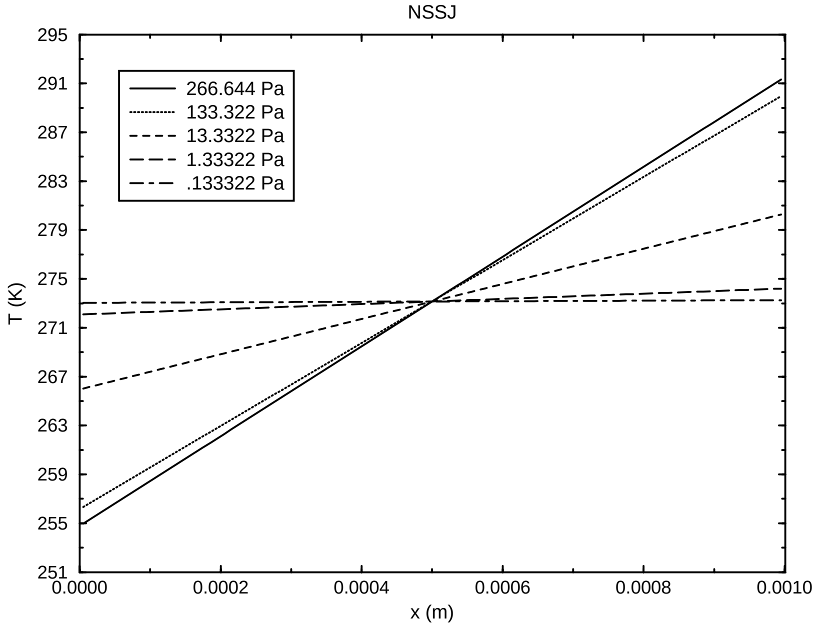

Figure 18. NSSJ temperature profiles.

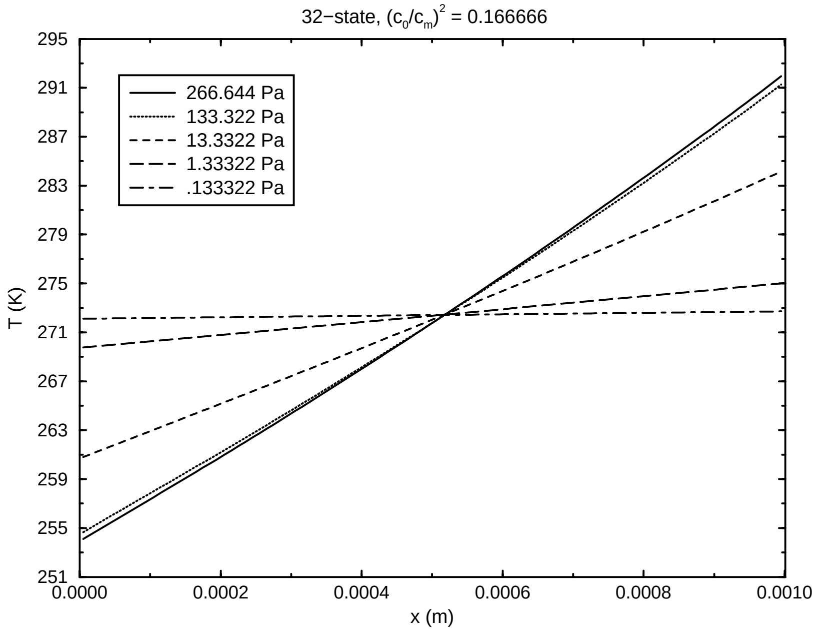

Figure 19. DVG temperature profiles: 32 states, $(c_0/c_m)^2 = 0.166666$.

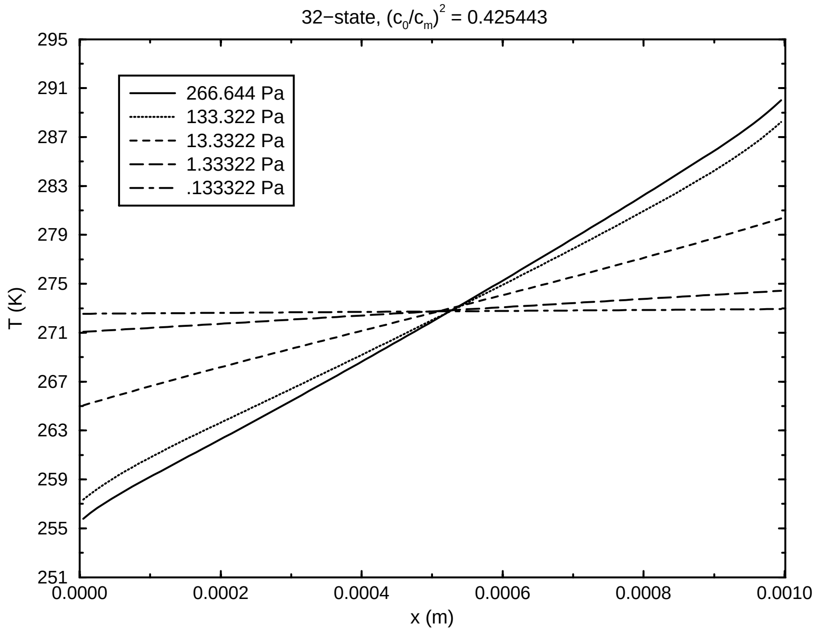

Figure 20. DVG temperature profiles: 32 states, $(c_0/c_m)^2 = 0.425443$.

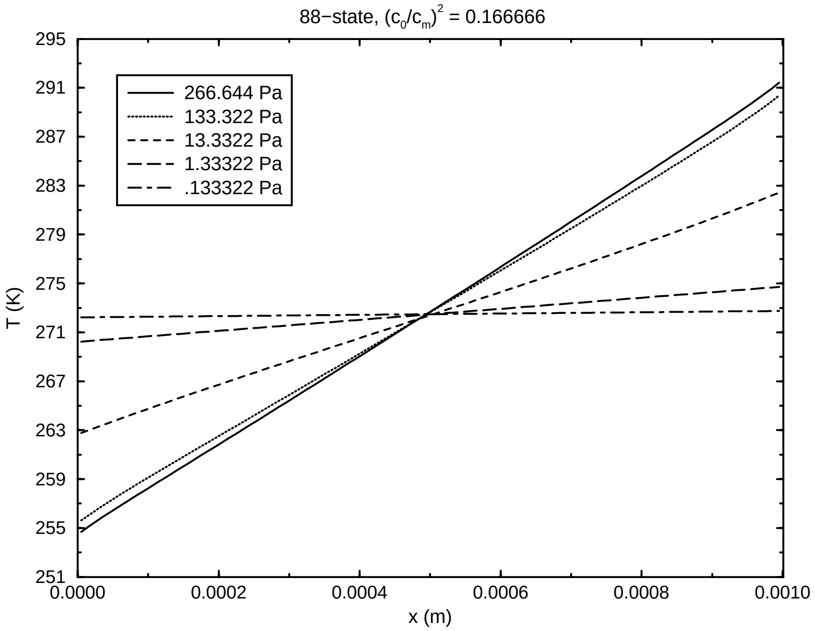

Figure 21. DVG temperature profiles: 88 states, $(c_0/c_m)^2 = 0.166666$.

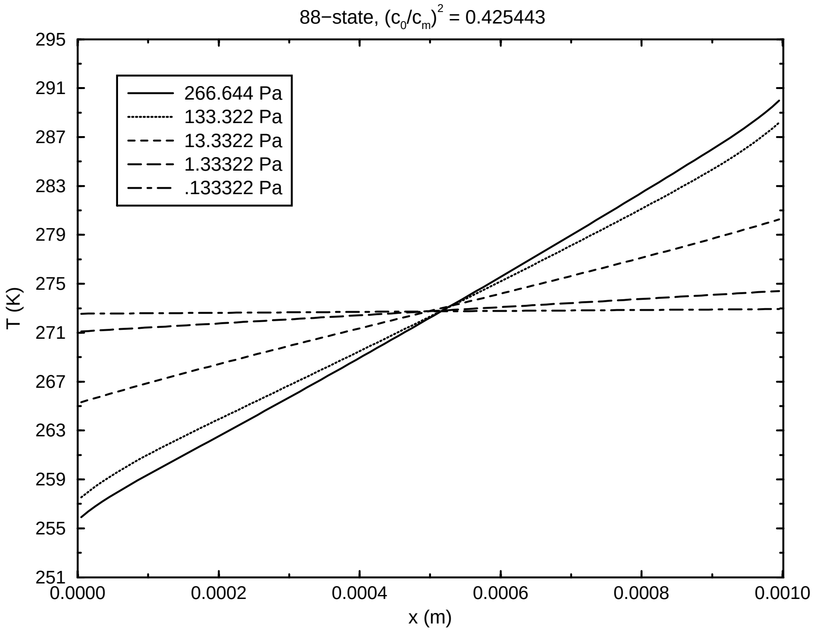

Figure 22. DVG temperature profiles: 88 states, $(c_0/c_m)^2 = 0.425443$.

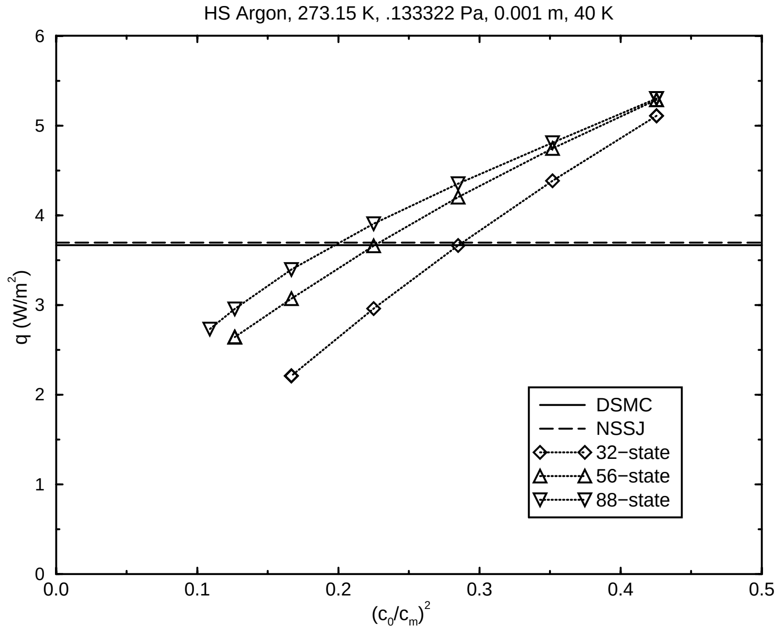

Figure 23. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 0.133322$ Pa.

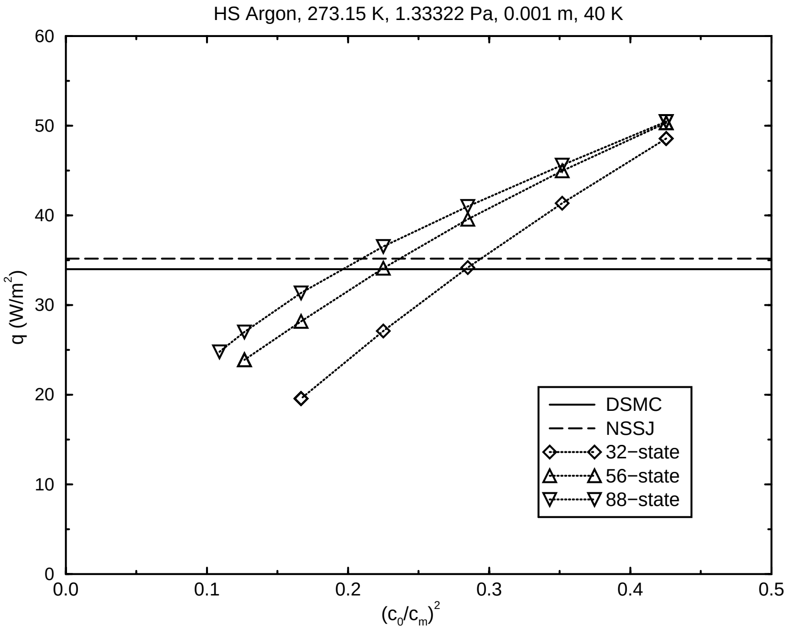

Figure 24. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 1.33322$ Pa.

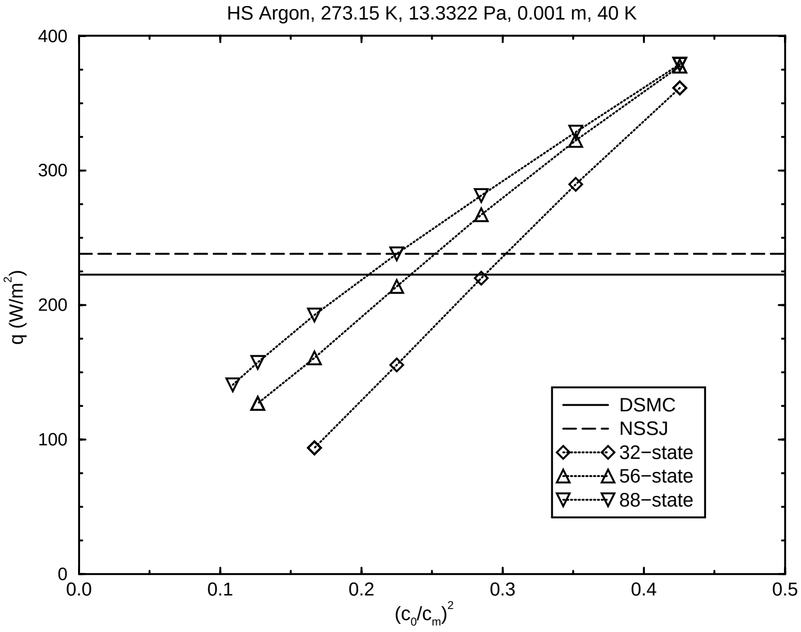

Figure 25. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 13.3322$ Pa.

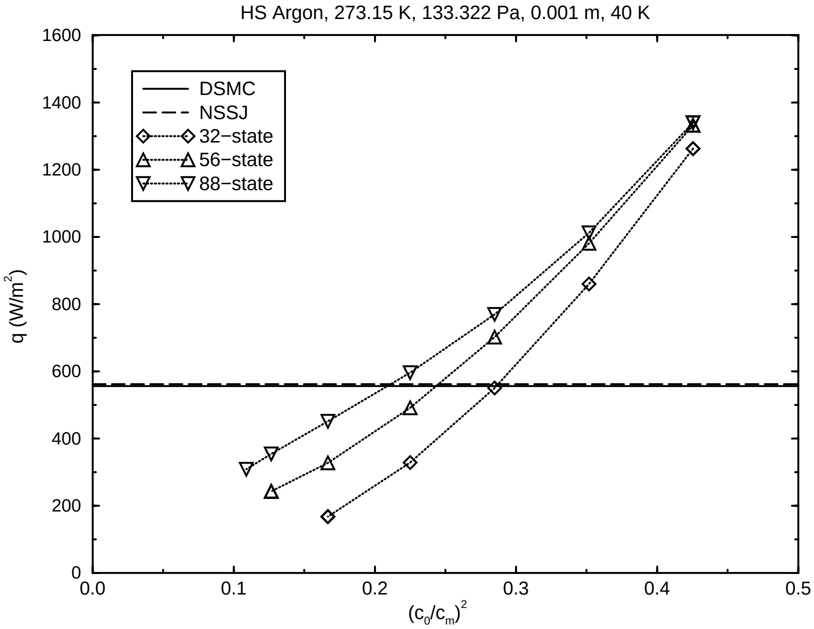

Figure 26. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 133.322$ Pa.

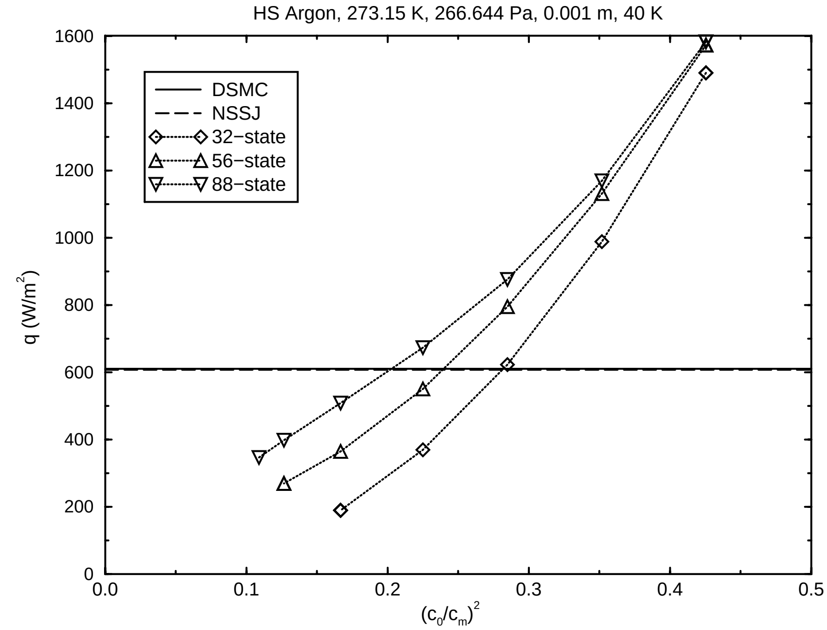

Figure 27. DSMC, NSSJ, and DVG heat flux q vs. $(c_0/c_m)^2$ at $p = 266.644$ Pa.

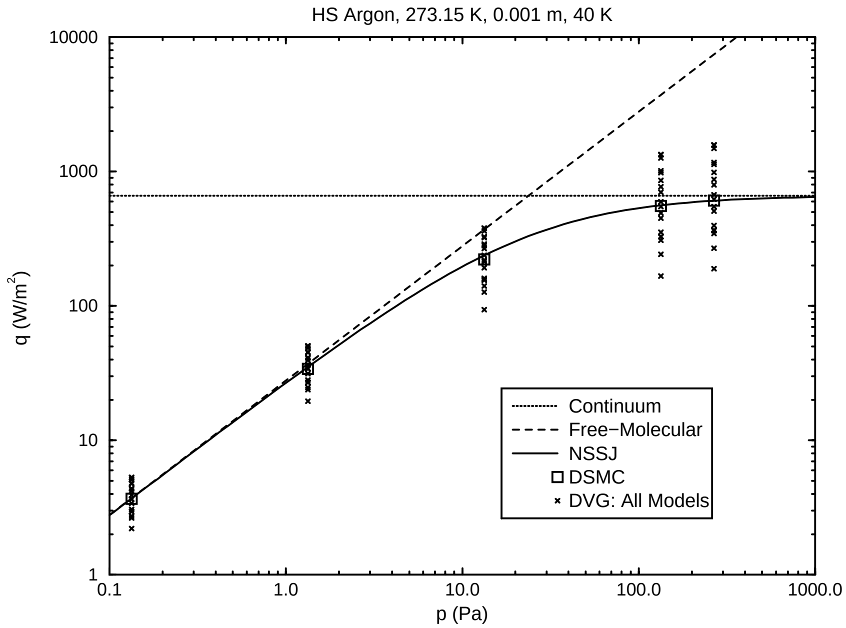

Figure 28. DSMC, NSSJ, and DVG heat flux q vs. pressure p.

6. Conclusions

A method to simulate noncontinuum microscale gas based on radiation-transport methods has been investigated. The Discrete Velocity Gas (DVG) method restricts the velocities of gas molecules to a finite set of values. These values are described by a speed scale and a set of nondimensional direction vectors referred to here as velocity states. `Mathematica` functions are developed that allow the collision terms for these velocity states to be determined and encoded in FORTRAN 77. Boundary conditions are formulated that represent the transfer of momentum and heat to a solid surface under fairly general conditions (bounce-back reflection, specular reflection, diffuse-isothermal reflection, and linear combinations thereof). The discrete Boltzmann equations describing the temporal and spatial evolution of the number densities corresponding to the velocity states, as well as the collision terms and the boundary conditions above, are encoded in a FORTRAN 77 code called `dvg1d`. This code simulates the flow in a one-dimensional domain bounded by two parallel planes.

To assess the accuracy of the DVG method, two benchmark problems are examined: Couette flow, which has isothermal sliding walls, and Fourier flow, which has nonisothermal motionless walls. The Direct Simulation Monte Carlo (DSMC) method, as embodied in the code `DSMC1`, is applied to these benchmark problems to generate highly accurate results. These results are compared to closed-form expressions derived for the Navier-Stokes Slip/Jump (NSSJ) method in the limit of small departures from uniform conditions and to the results from DVG simulations using `dvg1d`. DVG simulations are performed of these flows using the 8-state, 32-state, 56-state, and 88-state models with up to 8 values of the speed scale for each model, as appropriate.

For both benchmark problems and for all sets of conditions, the NSSJ shear-stress and heat-flux results are at most only a few percent different from the DSMC results, whereas the DVG results are always significantly different from the DSMC results. Moreover, for quantities of interest to structural-dynamics and heat-transfer simulations, such as the shear stress and the heat flux, the DVG results can differ from the DSMC results by order-unity amounts. In fact, the agreement between the DVG and DSMC results apparently becomes worse as the order of the moment that determines the physical quantity increases and is worse for odd moments than for even moments. Velocity, a first-order odd moment, is reasonably accurate, whereas temperature, a second-order even moment, is less accurate. Shear stress, a second-order odd moment, is rather inaccurate, and heat flux, a third-order odd moment is highly inaccurate.

Thus, the hypothesis motivating this study, namely that the DVG method might offer advantages for the small velocities and temperature differences found in microsystems, appears to be false. When the DSMC method cannot be used, the NSSJ method appears to offer advantages compared to the DVG method in terms of accuracy and computational efficiency. While the causes for the observed inaccuracy of the DVG method have not been identified, it is conjectured that the coarseness with which the molecular velocity distribution function is represented, the anisotropy inherent in molecular movement, and the inclusion of only discrete collisions in the molecular collision terms may be responsible.

References

Alexander, F. J., Garcia, A. L., and Alder, B. J., 1998, "Cell Size Dependence of Transport Coefficients in Stochastic Particle Algorithms," *Physics of Fluids*, Vol. 10, No. 6, pp. 1540-1542.

Alexander, F. J., Garcia, A. L., and Alder, B. J., 2000, "Erratum: 'Cell Size Dependence of Transport Coefficients in Stochastic Particle Algorithms,'" *Physics of Fluids*, Vol. 12, No. 3, p. 731.

Bird, G. A., 1970, "Direct Simulation and the Boltzmann Equation," *Physics of Fluids*, Vol. 13, No. 11, pp. 2676-2681.

Bird, G. A., 1976, *Molecular Gas Dynamics*, Clarendon Press, Oxford, UK.

Bird, G. A., 1978, "Monte Carlo Simulation of Gas Flows," *Annual Review of Fluid Mechanics*, Vol. 10, pp. 11-31.

Bird, G. A., 1998, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, Oxford, UK.

Broadwell, J. E., 1964a, "Study of Rarefied Shear Flow by the Discrete Velocity Method," *Journal of Fluid Mechanics*, Vol. 19, pp. 401-414.

Broadwell, J. E., 1964b, "Shock Structure in a Simple Discrete Velocity Gas," *Physics of Fluids*, Vol. 7, No. 8, pp. 1243-1247.

Cabannes, H., 1976, "Couette Flow for a Gas with a Discrete Velocity Distribution," *Journal of Fluid Mechanics*, Vol. 76, Part 2, pp. 273-287.

Cercignani, C., 1988, *The Boltzmann Equation and Its Applications*, Springer-Verlag, New York, NY.

Cercignani, C., 2000, *Rarefied Gas Dynamics: From Basic Concepts to Actual Calculations*, Cambridge University Press, Cambridge, UK.

Chapman, S., and Cowling, T. G., 1970, *The Mathematical Theory of Non-Uniform Gases*, third edition, Cambridge University Press, Cambridge, UK.

Gallis, M. A., Torczynski, J. R., and Rader, D. J., 2001, "An Approach for Simulating the Transport of Spherical Particles in a Rarefied Gas Flow via the Direct Simulation Monte Carlo (DSMC) Method," *Physics of Fluids*, Vol. 13, No. 11, pp. 3482-3492.

Gallis, M. A., and Torczynski, J. R., 2003, "Simulation of Moving Microbeams With the Direct Simulation Monte Carlo Method," paper AIAA-2003-4013, American Institute of Aeronautics and Astronautics, Reston, VA.

Gallis, M. A., Torczynski, J. R., and Rader, D. J., 2004, "Molecular Gas Dynamics Observations of Chapman-Enskog Behavior and Departures Therefrom in Nonequilibrium Gases," *Physical Review E*, submitted.

Garcia, A. L., and Wagner, W., 2000, "Time Step Truncation Error in Direct Simulation Monte Carlo," *Physics of Fluids*, Vol. 12, No. 10, pp. 2621-2633.

Garzo, V., and Santos, A., 2003, *Kinetic Theory of Gases in Shear Flows*, Kluwer, Dordrecht, Netherlands.

Gatignol, R., 1975, "Kinetic Theory for a Discrete Velocity Gas and Application to the Shock Structure," *Physics of Fluids*, Vol. 18, No. 2, pp. 153-161.

Gatignol, R., 1977, "Kinetic Theory Boundary Conditions for Discrete Velocity Gases," *Physics of Fluids*, Vol. 20, No. 12, pp. 2022-2030.

Goldstein, D. B., 1990, *Investigations of a Discrete Velocity Gas*, Ph. D. thesis, California Institute of Technology, Pasadena, CA.

Goldstein, D., 1991, "Near-Continuum Applications of a Discrete-Velocity Gas Model," in *Rarefied Gas Dynamics*, Volume 19, edited by A. E. Beylich, VCH Verlagsgesellschaft mbH, Weinheim, Germany, pp. 846-853.

Gombosi, T. I., 1994, *Gaskinetic Theory*, Cambridge University Press, Cambridge, UK.

Hadjiconstantinou, N. G., 2000, "Analysis of Discretization in the Direct Simulation Monte Carlo," *Physics of Fluids*, Vol. 12, No. 10, pp. 2634-2638.

Hamel, B., and Wachman, M., 1965, "A Discrete Ordinate Technique for the Linearized Boltzmann Equation with Application to Couette Flow," in *Rarefied Gas Dynamics*, Volume I, edited by J. H. de Leeuw, Academic Press, New York, NY, pp. 370-393.

Hirschfelder, J. O., Curtiss, C. F., and Bird, R. B., 1964, *Molecular Theory of Gases and Liquids*, corrected printing, John Wiley and Sons, New York, NY.

Inamuro, T., and Sturtevant, B., 1990, "Numerical Study of Discrete-Velocity Gases," *Physics of Fluids A*, Vol. 2, No. 12, pp. 2196-2203.

Karniadakis, G. E., and Beskok, A., 2002, *Micro Flows: Fundamentals and Simulations*, Springer-Verlag, New York, NY.

Kennard, E. H., 1938, *Kinetic Theory of Gases*, McGraw-Hill, New York, NY.

Longo, E., and Monaco, R., 1985, "Analytical Solutions of the Discrete Boltzmann Equation for the Rayleigh Flow Problem for Gas Mixtures," *Physics of Fluids*, Vol. 28, No. 9, pp. 2730-2734.

Monaco, R., and Preziosi, L., 1991, *Fluid Dynamic Applications of the Discrete Boltzmann Equation*, World Scientific, Singapore.

Nadiga, B. T., 1992, *A Study of Multi-Speed Discrete-Velocity Gases*, Ph. D. thesis, California Institute of Technology, Pasadena, CA.

Nanbu, K., 1986, "Theoretical Basis of the Direct Simulation Monte Carlo Method," in *Proceedings of the 15th International Symposium on Rarefied Gas Dynamics*," edited by V. Boffi and C. Cercignani, B. G. Teubner, Stuttgart, Germany, pp. 369-383.

Siegel, R., and Howell, J. R., 1992, *Thermal Radiation Heat Transfer*, third edition, Taylor and Francis, Washington, DC.

Sniegowski, J. J., and de Boer, M. P., 2000, "IC-Compatible Polysilicon Surface Micromachining," *Annual Reviews of Material Science*, Vol. 30, pp. 299-333.

Thompson, P. A., 1972, *Compressible-Fluid Dynamics*, McGraw-Hill, New York, NY.

Torczynski, J. R., 1993, "Development of the Collision Terms for Discrete Velocity Gases," Sandia internal memorandum, June 29.

Torczynski, J. R., Gallis, M. A., and Piekos, E. S., 2002, "Comparison of Methods for Simulating Gas Forces on Moving Microbeams," paper IMECE2002-33788, American Society of Mechanical Engineers, New York, NY.

Torczynski, J. R., Gallis, M. A., and Rader, D. J., 2003, "Using DSMC to Compute the Force on a Particle in a Rarefied Gas Flow," in *Rarefied Gas Dynamics: 23rd International Symposium*, edited by A. D. Ketsdever and E. P. Muntz, American Institute of Physics, Melville, NY.

Torczynski, J. R., and Gallis, M. A., 2003, "Gas Force on Heated Microbeams in Thermal Actuators," Sandia internal memorandum, December 4.

Vincenti, W. G., and Kruger, Jr., C. H., 1965, *Introduction to Physical Gas Dynamics*, John Wiley and Sons, Inc., New York, NY.

Wagner, W., 1992, "A Convergence Proof for Bird's Direct Simulation Monte Carlo Method for the Boltzmann Equation," *Journal of Statistical Physics*, Vol. 66, Nos. 3/4, pp. 1011-1044.

Wolfram, S., 1996, *The Mathematica Book* (version 3.0), Cambridge University Press, Cambridge, UK.

Distribution

1	MS 0841	Org. 9100	T.	C.	Bickel
1	MS 0841	Org. 9100	D.	K.	Gartling
1	MS 0824	Org. 9110	W.	L.	Hermina
1	MS 0834	Org. 9112	M.	R.	Prairie
1	MS 0834	Org. 9112	C.	J.	Bourdon
1	MS 0834	Org. 9112	D.	J.	Rader
1	MS 0826	Org. 9113	S.	N.	Kempka
4	MS 0826	Org. 9113	M.	A.	Gallis
1	MS 0826	Org. 9113	A.	S.	Geller
1	MS 0826	Org. 9113	D.	R.	Noble
1	MS 0826	Org. 9113	E.	S.	Piekos
4	MS 0826	Org. 9113	J.	R.	Torczynski
1	MS 0826	Org. 9113	C.	C.	Wong
1	MS 0834	Org. 9114	J.	E.	Johannes
1	MS 0834	Org. 9114	R.	C.	Givler
1	MS 0836	Org. 9117	R.	O.	Griffith
1	MS 0836	Org. 9117	R.	J.	Buss
1	MS 0847	Org. 9124	J.	M.	Redmond
1	MS 0847	Org. 9124	O.	B.	Ozdoganlar
1	MS 0323	Org. 1011	D.	L.	Chavez (LDRD Office)
1	MS 9018	Org. 8945-1	Central Technical Files		
2	MS 0899	Org. 9616	Technical Library		