THE DIRECT SIMULATION MONTE CARLO METHOD

Francis J. Alexander
and Alejandro L. Garcia

Department Editors:
Harvey Gould
hgould@clarku.edu
Jan Tobochnik
jant@kzoo.edu

The primary concern of computational fluid dynamics is the development and application of numerical methods to solve the partial differential equations of hydrodynamics.[1] These equations, most notably the Navier-Stokes and Euler equations, describe Newtonian fluids, that is, gases and simple liquids, over a wide range of conditions. Although very useful, the continuum description of a fluid has its limits. For example, the flow of a dilute gas requires a kinetic-theory description.[2] As a result, various specialized methods for simulating such flows have been developed. This column presents one such method, direct simulation Monte Carlo (DSMC), developed by Bird in the 1960s.[3]

Flows are characterized by a variety of dimensionless quantities. The most useful one for our purposes is the Knudsen number Kn. This dimensionless quantity is defined as $\mathrm{Kn}=\lambda/L$, where L is the characteristic length scale of the physical system, and λ is the molecular mean free path. In general, a continuum description is not accurate when $\mathrm{Kn}>1/10$.

The mean free path λ of a molecule in a dilute gas is given by

$$\lambda=\frac{1}{\sqrt{2}\pi\sigma^2 n}, \tag{1}$$

where n is the number density and σ is the effective diameter of the molecule. For example, in air at atmospheric pressure, $\lambda\approx 50$ nm (about the wavelength of visible light), whereas in the rarefied upper atmosphere at an altitude of 120 km, the mean free path is several meters. The Knudsen number for air flow through the 50-nm-wide gap between the head and platter in a disk drive is $\mathrm{Kn}\approx 1$, and Kn of the bow shock off the nose of the space shuttle is also of order unity, because the characteristic size of the nose is meters.

A more general criterion that indicates the breakdown of the continuum description is the appearance of anisotropic pressure effects. To quantify this criterion, we define a local Knudsen number as $\mathrm{Kn}_{\ell}=\lambda|\nabla n|/n$. Extensive studies done by Boyd, Chen, and Candler[4] indicate that the appropriate criterion for the failure of the continuum description is $\mathrm{Kn}_{\ell}>0.05$.

For cases such as those discussed above, the continuum description based on partial differential equations is inadequate, and a particle-based approach is needed. For physicists, the best known particle-based algorithm is molecular dynamics.[5] In molecular dynamics, the trajectory of every particle in the fluid is computed from Newton's equations, given an empirically determined interparticle potential. Although molecular dynamics is a useful technique in statistical mechanics, its application is limited to simple hydrodynamic flows due to its enormous computational overhead.[6] Molecular-dynamics simulations of a dilute gas are extremely time-consuming even when run on the most powerful computers. To simulate one cubic millimeter of air at STP would require 10^{16} molecules and on the order of 10^{23} floating-point operations to evolve the system for a mean free time of 10^{-10} s. This calculation would take about 10^2 years, even on a teraflops machine.

Fortunately, the DSMC method is an efficient alternative for simulating a dilute gas. The method can be viewed as either a simplified molecular-dynamics (DSMC is several orders of magnitude faster) or a Monte Carlo method for solving the time-dependent nonlinear Boltzmann equation (which describes the evolution of a dilute gas at the level of the single-particle distribution function). Rather than exactly calculating collisions as in molecular dynamics, the DSMC method generates collisions stochastically with scattering rates and postcollision velocity distributions determined from the kinetic theory of a dilute gas. Although DSMC simulations are not correct at a length scale as short as that of an atomic diameter, they are accurate at scales smaller than that of a mean free path.

The method has been thoroughly tested in high-Knudsen-number flows over the past 25 years and found to

Frank Alexander is a research assistant professor at the Center for Computational Science, Boston University, 3 Cummington Street, Boston, MA 02215. E-mail: fja@bu.edu

Alej Garcia is a professor of physics at San Jose State University, San Jose, CA 95192-0106, and a member of the Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA 94720. E-mail: algarcia@wenet.net

© 1997 AMERICAN INSTITUTE OF PHYSICS S0894-1866(97)00706-2

be in excellent agreement with both experimental data[7] and molecular-dynamics computations.[8,9] The DSMC method also has been used successfully for several decades in the study of rarefied gas flows, and has recently found new and exciting applications in chemistry and physics.[17] Despite this history, the method is still not widely known (compared with, say, molecular dynamics) even among computational physicists.

DSMC simulations employ various types of boundaries (for example, specular surfaces, periodic boundaries, and thermal walls).

In this column, we present a brief description of DSMC. Detailed expositions can be found in Ref. 3 or Ref. 10. As a concrete example, we describe how to simulate the Rayleigh problem, that is, flow in a semi-infinite gas acted on by an impulsively started, flat plate.

In the Rayleigh problem the system consists of a spatial domain bounded in the vertical direction by a flat thermal wall on the bottom, its temperature fixed at T_w, and a flat, specularly reflecting plate on the top. The gas between the plates is initially at rest in thermodynamic equilibrium with a temperature $T_0 = T_w$. At time $t=0$, the lower plate's tangential velocity is impulsively set to u_w. The top plate is stationary. In the horizontal direction there are periodic boundary conditions, so that a particle that moves off the right reappears at the left. The system is partitioned into a regular array of cells (see Fig. 1), each of which is less than a mean free path across ($\Delta z < \lambda$).

Initially the N particles are randomly distributed with uniform density throughout the system, such that there are at least 50 particles in each cell. Each particle in the simulation represents N_e effective molecules in the physical system. In this sense, the DSMC method solves the Boltzmann equation using a representative random sample drawn from the actual velocity distribution. This representation allows us to model many systems of interest using only $10^4 - 10^5$ particles (although simulations using over 10^8 particles are not uncommon). If the number of particles used in the simulation is too small, fewer than about 20 particles per cubic mean free path, the DSMC method is not accurate.[11]

In addition to its location $\mathbf{r}_i$, each particle is initialized with a velocity $\mathbf{v}_i$, typically chosen from a Maxwellian distribution. Therefore, the DSMC algorithm is like molecular dynamics in that the state of the system is given by the positions and velocities of the particles $\{\mathbf{r}_i, \mathbf{v}_i\}$ for $i = 1, \ldots, N$.

The particle evolution is integrated in time steps of duration Δt and consists of two distinct phases: advection and collision. This splitting of the evolution between streaming and collisions is accurate only when Δt is a fraction of the mean free time.

In the advection phase, the particles are moved as if they did not interact, that is, their positions are updated to $\mathbf{r}_i + \mathbf{v}_i \Delta t$. Particles that reach a boundary are processed according to the appropriate boundary condition, as described below. If a particle strikes a wall, the time of the collision is determined by tracing the straight-line trajectory from the initial location $\mathbf{r}_i$ to the point of impact, $\mathbf{r}_w$. The time of flight from the particle's initial position to the point of impact is $\Delta t_w = (\mathbf{r}_w - \mathbf{r}_i) \cdot \hat{\mathbf{n}} / (\mathbf{v}_i \cdot \hat{\mathbf{n}})$, where $\hat{\mathbf{n}}$ is the unit normal to the surface. After striking the surface, the particle moves off with a new velocity as dictated by the boundary conditions and with the remaining time $\Delta t - \Delta t_w$.

What can happen to the particle when it reaches an interface? DSMC simulations employ various types of boundaries (for example, specular surfaces, periodic boundaries, and thermal walls). When a particle strikes a specular surface, its component of velocity normal to the surface is reversed. This condition is the one satisfied by a particle that reflects off the top wall. When a particle strikes a perfect thermal wall at temperature T_w, all three components of the velocity are reset according to a biased Maxwellian distribution. The component normal to the wall is distributed as

$$P_\perp(v_\perp) = \frac{m}{kT_w} v_\perp \, e^{-mv_\perp^2/2kT_w}, \tag{2}$$

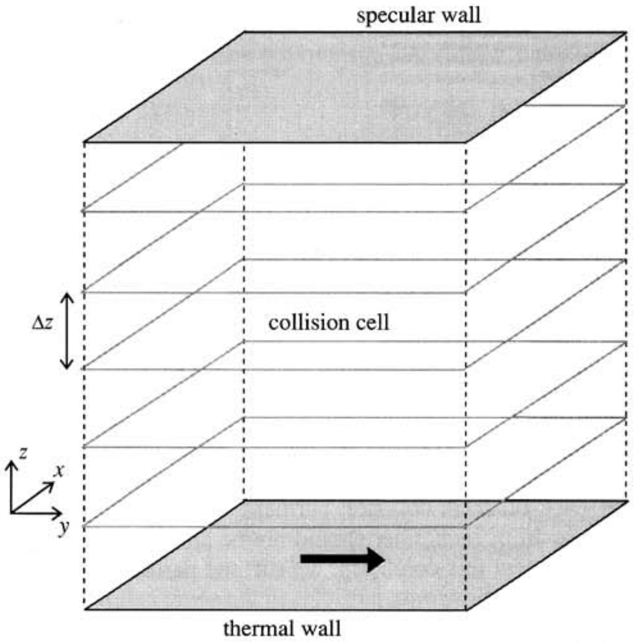

Figure 1. Schematic of the Rayleigh problem. The system consists of a fluid bounded in the z direction by a flat thermal wall on the bottom with its temperature fixed at T_w, and a flat, specularly reflecting plate on the top. At time $t=0$, the bottom velocity of the wall is set equal to u_w in the y direction.

and each parallel component is distributed as

$$P_{\parallel}(v_{\parallel}) = \sqrt{\frac{m}{2\pi k T_{\mathrm{w}}}}\, e^{-mv_{\parallel}^2/2kT_{\mathrm{w}}}, \tag{3}$$

where m is the particle's mass and k is Boltzmann's constant. We shall treat particles striking the bottom wall in this way.

Note that the distributions (2) and (3) are biased because the faster particles strike the wall more frequently. The distribution for particles leaving the thermal wall must reflect this bias. In addition to the tangential component generated thermally by the wall, we need to add an additional velocity to account for the uniform translational motion of the wall in the Rayleigh problem. In other words, (3) is the distribution in the rest frame of the thermal wall.

Because the wall is moving in the lab frame, we need to add u_{w} to the y component of velocity for particles scattering off the wall. Using (2) and (3), the components of the velocity of a particle leaving the lower thermal wall become

$$v_x = \sqrt{\frac{kT_{\mathrm{w}}}{m}}\, r_{\mathrm{G}}, \tag{4}$$

$$v_y = \sqrt{\frac{kT_{\mathrm{w}}}{m}}\, r'_{\mathrm{G}} + u_{\mathrm{w}}, \tag{5}$$

$$v_z = \sqrt{-\frac{2kT_{\mathrm{w}}}{m} \ln r}, \tag{6}$$

where r is a uniformly distributed random number in (0,1) and $r_{\mathrm{G}}, r'_{\mathrm{G}}$ are Gaussian-distributed random numbers with zero mean and unit variance. In reality, gas-surface scattering is more complex, and these additional complexities can be included within the gas-surface scattering routine (see Ref. 17).

In each cell, a set of representative collisions is processed at each time step.

After all the particles have been moved, a given number are randomly selected for collisions. The rules for this random selection process are obtained from kinetic theory. Our intuition tells us to select only those particles that are near each other as collision partners. In other words, particles far from each other should not be allowed to interact. To implement this condition, we sort the particles into spatial cells and allow only particles in the same cell to collide. We could invent more complicated methods, but this one works well as long as Δz, the linear dimension of a cell, is no larger than a mean free path.

The concept of a collision implies that the interaction potential between particles is short-ranged. For simplicity, the DSMC algorithm is formulated for a dilute gas of hard-sphere particles with diameter σ. For engineering applications, more realistic representations of the molecular interaction may be used to give accurate transport properties[3] and equations of state for air and other gases.[12]

In each cell, a set of representative collisions is processed at each time step. All pairs of particles in a cell are considered to be candidate collision partners, regardless of their positions within the cell. Because only the magnitude of the relative velocity between particles is used in determining their collision probability, even particles that are moving away from each other may collide. This condition allows two particles to collide by simply being located within the same cell; no other positional information is used in the evaluation of collisions. Remember that collisions in the DSMC algorithm are only statistically correct because the particles' Newtonian trajectories are *not* calculated. (If they were, we would be doing molecular dynamics.)

The collision probability for a pair of hard spheres, i and j, is proportional to their relative speed,

$$P_{\mathrm{coll}}(i,j) = \frac{|\mathbf{v}_i - \mathbf{v}_j|}{\sum_{m=1}^{N_{\mathrm{c}}} \sum_{n=1}^{m-1} |\mathbf{v}_m - \mathbf{v}_n|}, \tag{7}$$

where N_{c} is the number of particles in the cell. Note that the denominator normalizes this discrete probability distribution. The sums in (7) are over all the particles in a given cell.

It would be computationally expensive to use (7) directly, because of the double sum in the denominator. Instead, the following acceptance-rejection scheme is used to select collision pairs in each cell.

(1) A pair of potential collision partners, i and j, is chosen at random from the particles within the cell.

(2) The pair is accepted as collision partners if

$$\frac{|\mathbf{v}_i - \mathbf{v}_j|}{v_{\mathrm{r,max}}} > r, \tag{8}$$

where $v_{\mathrm{r,max}}$ is the maximum relative speed in the cell and r is a uniform deviate in (0, 1).

(3) If the pair is accepted, the collision is processed and the velocities of the particles are reset as discussed below.

(4) After the collision is processed or if the pair is rejected, the routine moves again to step (1) until the required number of candidate pairs M_{coll} in the cell has been processed. The value of M_{coll} is discussed in the following.

This acceptance-rejection procedure exactly selects collision pairs according to (7). The method is also exact if we overestimate the value of $v_{\mathrm{r,\,max}}$, although it is less efficient in the sense that more candidates are rejected. It is computationally cheaper to make an intelligent guess that overestimates $v_{\mathrm{r,\,max}}$ rather than recompute it at each time step, because the explicit calculation of $v_{\mathrm{r,\,max}}$ is as costly as the calculation of the denominator in (7).

After the collision pair is chosen, their postcollision velocities, $\mathbf{v}'_i$ and $\mathbf{v}'_j$, need to be evaluated. Conservation of linear momentum tells us that the center-of-mass (cm) velocity remains unchanged by the collision,

$$\mathbf{v}_{\rm cm}=\tfrac{1}{2}(\mathbf{v}_i+\mathbf{v}_j)=\tfrac{1}{2}(\mathbf{v}_i'+\mathbf{v}_j')=\mathbf{v}_{\rm cm}'. \tag{9}$$

From conservation of energy, the magnitude of the relative velocity is also unchanged by the collision,

$$v_{\rm r}=|\mathbf{v}_i-\mathbf{v}_j|=|\mathbf{v}_i'-\mathbf{v}_j'|=v_{\rm r}'. \tag{10}$$

Equations (9) and (10) give us four constraints for the six unknowns in $\mathbf{v}_i'$ and $\mathbf{v}_j'$.

The two remaining unknowns are fixed by the angles, θ and ϕ, for the relative velocity

$$\mathbf{v}_{\rm r}'=v_{\rm r}\,[(\sin\theta\cos\phi)\hat{\mathbf{x}}+(\sin\theta\sin\phi)\hat{\mathbf{y}}+\cos\theta\,\hat{\mathbf{z}}]. \tag{11}$$

For the hard-sphere model, these angles are uniformly distributed over the unit sphere. The azimuthal angle ϕ is uniformly distributed between 0 and 2π, and so it is selected as $\phi=2\pi r$. The θ angle is distributed according to the probability,

$$P(\theta)\,d\theta=\tfrac{1}{2}\sin\theta\,d\theta. \tag{12}$$

Using the change of variable $q=\cos\theta$, we have $P(q)dq=\frac{1}{2}dq$, and so q is uniformly distributed in the interval $[-1,1]$. We do not really need to find θ; instead we compute

$$\begin{aligned} q&=2r-1,\\ \cos\theta&=q,\\ \sin\theta&=\sqrt{1-q^2}. \end{aligned} \tag{13}$$

The values of $\sin\theta$ and $\cos\theta$ generated in this way are used in (11). The postcollision velocities are given by

$$\begin{aligned} \mathbf{v}_i'&=\mathbf{v}_{\rm cm}'+\tfrac{1}{2}\mathbf{v}_{\rm r}'\\ \mathbf{v}_j'&=\mathbf{v}_{\rm cm}'-\tfrac{1}{2}\mathbf{v}_{\rm r}'. \end{aligned} \tag{14}$$

The total number of collisions that should take place in a cell during a time step is given by

$$M_{\rm coll}=\frac{N_{\rm c}^2\pi\sigma^2\langle v_{\rm r}\rangle N_{\rm e}\Delta t}{2V_{\rm c}}, \tag{15}$$

where $V_{\rm c}$ is the volume of the cell and $\langle v_{\rm r}\rangle$ is the average relative velocity in the cell. However, we do not really want to compute $\langle v_{\rm r}\rangle$, because that involves doing a sum over all $\frac{1}{2}N_{\rm c}^2$ pairs of particles in the cell. Let us recall that collision candidates go through an acceptance-rejection procedure. The ratio of total accepted to total candidates is

$$\frac{M_{\rm coll}}{M_{\rm cand}}=\frac{\langle v_{\rm r}\rangle}{v_{\rm r,max}}, \tag{16}$$

because the probability of accepting a pair is proportional to their relative velocity. Using (15) and (16), we have

$$M_{\rm cand}=\frac{N_{\rm c}^2\pi\sigma^2 v_{\rm r,max}N_{\rm e}\Delta t}{2V_{\rm c}}, \tag{17}$$

which is the number of candidate pairs we should select over a time step Δt. Note that $M_{\rm coll}$ will equal on average the acceptance probability (8) multiplied by (17) and is independent of $v_{\rm r,max}$. However, if we set $v_{\rm r,max}$ too high, we still process the same number of collisions on the average, but the program is inefficient, because the acceptance probability is low and thus many candidates are rejected.

Because the DSMC method is inherently stochastic, most physical quantities of interest are computed as averages.

Because the DSMC method is inherently stochastic, most physical quantities of interest are computed as averages. The values of the mass density $\tilde{\rho}(z,t)$, momentum density $\tilde{\mathbf{p}}(z,t)$, and energy density $\tilde{e}(z,t)$ are periodically measured as

$$\begin{Bmatrix} \tilde{\rho}(z,t)\\ \tilde{\mathbf{p}}(z,t)\\ \tilde{e}(z,t) \end{Bmatrix}=\frac{1}{V_s}\sum_i^{\text{cell at }z}\begin{Bmatrix} m\\ m\mathbf{v}_i\\ \frac{1}{2}m|\mathbf{v}_i|^2 \end{Bmatrix}, \tag{18}$$

where the sum is over particles in the statistics cell (of volume V_s) located at position z. The tilde indicates that these quantities are instantaneous, fluctuating values. For convenience, we often use collision cells as statistics cells.

Unless the number of particles in a statistics cell is very large, there will be significant fluctuations in the average quantities. In steady flows, we can take statistics over a long run after the system has reached its steady state. For time-dependent problems, we can average over many realizations of the same experiment (that is, average over many runs, varying the random-number seed between runs). The fluid velocity is computed as $\mathbf{u}(z,t)=\mathbf{p}(z,t)/\rho(z,t)$, using the averaged values of the mass and momentum density. The temperature is computed from the equipartition theorem as

$$T(z,t)=\frac{2m}{3k}\left[\frac{e(z,t)}{\rho(z,t)}-\frac{1}{2}|\mathbf{u}(z,t)|^2\right]. \tag{19}$$

The local pressure $P(z,t)$ may be evaluated from the ideal-gas law or by explicitly evaluating the pressure tensor.[2] The two results may not be equivalent because under some conditions (for example, high Mach number) the pressure tensor will not be isotropic. The normal and tangential forces on a wall (that is, pressure and drag) may be directly computed from the time-averaged change in the momentum of particles striking the wall; similarly the heat flux at a surface may be obtained from the particles' change in energy.[10]

Now that all the elements of the DSMC algorithm have been presented, we use it to study the Rayleigh problem. In particular, we consider the parallel component of fluid velocity for the gas near the moving wall. Conventional hydrodynamics predicts that $u_y(z=0,t)=u_{\mathrm{w}}$ for no-slip boundary conditions. In the collisionless limit, $u_y(z=0,t)=\frac{1}{2}u_{\mathrm{w}}$, because particles approaching the wall have zero average y velocity. The Boltzmann equation, in the BGK approximation, gives[18]

$$u_y(z=0,t)=\frac{u_{\mathrm{w}}}{2}\left(1+\int_0^{t/\tau} x^{-1}e^{-x}I_1(x)\,dx\right), \quad (20)$$

where $\tau=\langle v\rangle/\lambda$ is the mean free time and $\langle v\rangle$ is the average particle speed. $I_1(x)$ is the modified Bessel function of the first kind. Notice that (20) gives the collisionless result for short times and the no-slip hydrodynamic result when $t\gg\tau$.

The DSMC simulation is formulated in dimensionless units, taking $m=\sigma=kT_0=1$. The initial density is $n_0=10^{-3}$, and so in these units the mean free path is $\lambda=225$. The distance between the walls is $L_z=2250.8$, and the system is partitioned into 50 DSMC collision cells $(\Delta z=0.2\lambda)$, each initially containing 1000 particles, so that there is one cell in the x-y direction for each bin of z. To obtain accurate statistics, 500 realizations were computed. Each realization ran for 5 mean free times using 125 time steps. The entire calculation was performed in several hours on a personal computer.

In the past decade the DSMC method has been used in a variety of physics and chemistry applications that require a kinetic formulation.

The wall speed is $u_{\mathrm{w}}=0.2$, which is about one-sixth of the speed of sound. The gas speed parallel to the wall is obtained by extrapolating the measured velocity profile to $z=0$. Figure 2 shows that the BGK prediction (20) is in good agreement with the DSMC results.

DSMC was originally developed by aerospace engineers for the simulation of rarefied gas flows. In the past decade the method has been used in a variety of physics and chemistry applications (for example, nonequilibrium fluctuations, nanoscale flows, and chemical vapor deposition) that require a kinetic formulation. Even the limitation to dilute systems has been lifted with the introduction of the consistent Boltzmann algorithm[12] and Enskog simulation Monte Carlo.[19]

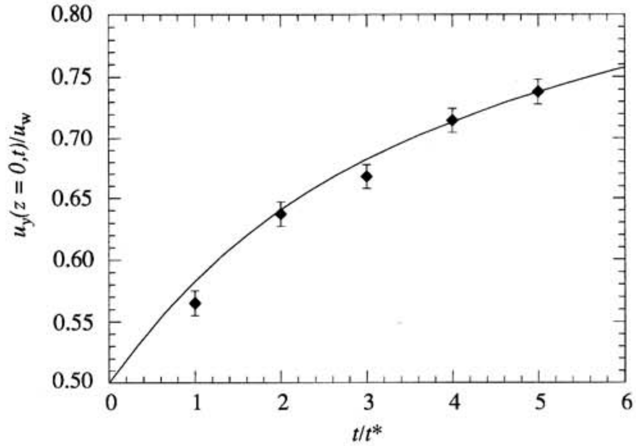

Figure 2. Fluid velocity of the gas near the wall as a function of time. The solid line is the BGK prediction from Eq. (20); the data points are from the DSMC simulation.

For reasons of computational expense, the continuum methods of computational fluid dynamics will never entirely be replaced by particle schemes. Sometimes, continuum methods are the only way to go.

Suggestions for further study

(1) Implement the DSMC method for the geometry shown in Fig. 1. Fix the velocity of the lower wall at zero, that is, consider the case in which the system is in thermodynamic equilibrium. Record the velocity of each particle striking the specular upper surface. From the histogram of this data confirm Eqs. (2) and (3). Measure the average change in momentum for particles striking the upper surface and use the impulse-momentum theorem to find the pressure on the surface. Remember that each particle represents N_{e} molecules. Confirm that for various initial densities and temperatures your measurement of P agrees with the ideal-gas law. Repeat the above for the lower, thermal wall.

(2) Write a DSMC program to simulate the Rayleigh problem described in the text. Confirm the result shown in Fig. 2 for the fluid velocity extrapolated to the thermal wall. Run your program for various values of L, the separation between the walls, and examine its effect on the results. From the average change in momentum for particles striking the thermal wall, compute and plot the pressure and drag force per unit area as a function of time. Measure and plot the profiles of the density $\rho(z,t)$, the fluid velocity $\mathbf{u}(z,t)$, and the temperature $T(z,t)$ as functions of position and time. Is it accurate to assume that ρ and T remain constant? What effect does the system length L have on each of these profiles?

(3) Set up a DSMC program for the geometry shown in Fig. 1, but instead of impulsively moving the lower

wall at $t=0$, change its temperature to $T_w \neq T_0$. Separately consider the cases in which the wall temperature is hotter than the gas and colder than the gas. Compute and plot $T(z=0,t)$, the gas temperature extrapolated to the surface of the thermal wall. From the average change in energy for particles striking the thermal wall, compute and plot the heat flux as a function of time. Measure and plot the profiles for the pressure $P(z,t)$ (as given by the ideal-gas law) and the fluid velocity $\mathbf{u}(z,t)$. Does the temperature change cause the fluid to move?

(4) Set up a DSMC program for the geometry shown in Fig. 1, but choose the upper surface to be a thermal wall at fixed temperature $T_w = T_0$ moving with a tangential velocity opposite to that of the lower wall (that is, shear flow). Run your program until the system reaches a steady state with a linear velocity profile. Show that the steady-state fluid velocity extrapolated to a wall does not match the wall's velocity. Find the distance inside the wall at which the extrapolated fluid velocity would equal the wall's velocity; this distance is called the slip length. Show that this slip length is approximately equal to the mean free path λ when $L \gg \lambda$. How does the slip length vary with L? Measure the steady-state temperature profile and show that it is parabolic with a maximum in the center of the system (due to viscous heating). How does the maximum temperature vary with the velocity gradient?

(5) Set up a DSMC program for a two-dimensional rectangular geometry with a thermal lower wall; the three other surfaces are specular. Your program should use a lattice of computational cells with cell dimensions $\Delta y, \Delta z < \lambda$. Repeat the Rayleigh problem (impulsively started lower wall) and examine the gas velocity at the wall both at the center of the wall and near the corners. Compute the total angular momentum in the gas and plot it as a function of time. Does the angular momentum reach a limiting value? Repeat this calculation with the upper surface being a stationary thermal wall.

(6) As in problem (5), set up a DSMC program for a two-dimensional rectangular geometry. Make the top and bottom surfaces specular and the other two surfaces thermal walls at different temperatures. Run the program until the system reaches its steady state and plot the profiles of the temperature $T(y,z)$ and the fluid velocity $\mathbf{u}(y,z)$. There should be nothing interesting—temperature varies linearly from one wall to the other and the average velocity is zero. Next, modify the program to include a constant force on the particles in the downward direction (that is, gravity). Show that the fluid forms a convection roll with the gas rising at the hot wall and falling at the cold wall. In houses, this convection can be a major source of heat loss, and fiberglass insulation is used to reduce it.

Acknowledgments

The authors thank B. J. Alder, F. Baras, and M. Malek-Mansour for many useful discussions. This work was supported in part by the National Science Foundation's Fluid, Particulate and Hydraulic Systems Program.

References

1. J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics* (Springer, Berlin, 1996).
2. S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases* (Cambridge University Press, Cambridge, 1970).
3. G. A. Bird, *Molecular Gas Dynamics* (Clarendon, Oxford, 1976); G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows* (Clarendon, Oxford, 1994).
4. I. Boyd, G. Chen, and G. Candler, Phys. Fluids **7**, 210 (1995).
5. B. J. Alder and T. E. Wainwright, J. Chem. Phys. **27**, 1208 (1957); M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon, Oxford, 1987); D. C. Rapaport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 1996).
6. *Microscopic Simulations of Complex Hydrodynamic Phenomena*, edited by M. Mareschal and B. L. Holian, NATO ASI Series Vol. 292 (Plenum, New York, 1992).
7. E. P. Muntz, Annu. Rev. Fluid Mech. **21**, 387 (1989); D. A. Erwin, G. C. Pham-Van-Diep, and E. P. Muntz, Phys. Fluids A **3**, 697 (1991).
8. D. L. Morris, L. Hannon, and A. L. Garcia, Phys. Rev. A **46**, 5279 (1992).
9. E. Salomons and M. Mareschal, Phys. Rev. Lett. **69**, 269 (1992).
10. A. L. Garcia, *Numerical Methods for Physics* (Prentice-Hall, Englewood Cliffs, NJ, 1994).
11. M. Fallavollita, D. Baganoff, and J. McDonald, J. Comput. Phys. **109**, 30 (1993); G. Chen and I. Boyd, J. Comput. Phys. **126**, 434 (1996).
12. F. Alexander, A. L. Garcia, and B. Alder, Phys. Rev. Lett. **74**, 5212 (1995); F. Alexander, A. L. Garcia, and B. Alder, in *25 Years of Non-Equilibrium Statistical Mechanics*, edited by J. J. Brey, J. Marco, J. M. Rubi, and M. San Miguel, (Springer, Berlin, 1995).
13. G. A. Bird, in *Microscopic Simulations of Complex Hydrodynamic Phenomena*, in Ref. 6.
14. J. C. Maxwell, Philos. Trans. R. Soc. London **170**, 231 (1867).
15. L. D. Landau and E. M. Lifshitz, *Statistical Mechanics* (Pergamon, Oxford, 1980).
16. P. Resibois and M. De Leener, *Classical Kinetic Theory of Fluids* (Wiley, New York, 1977).
17. A. L. Garcia and F. Baras, Proceedings of the Third Workshop on Modeling of Chemical Reaction Systems, Heidelberg, 1997. (CD-ROM only.)
18. C. Cercignani, *Mathematical Methods în Kinetic Theory* (Plenum, New York, 1990), p. 190.
19. J. M. Montanero and A. Santos, Phys. Rev. E **54**, 438 (1996); J. M. Montanero and A. Santos, Phys. Fluids **9**, 2057 (1997).