NASA Contractor Report 195463

Molecular Dynamics Simulation of a Piston Driven Shock Wave in a Hard Sphere Gas

Myeung-Jouh Woo and Isaac Greber
Case Western Reserve University
Cleveland, Ohio

April 1995

Prepared for
Lewis Research Center
Under Grant NAG3–795

National Aeronautics and
Space Administration

MOLECULAR DYNAMICS SIMULATION OF A PISTON DRIVEN SHOCK WAVE IN A HARD SPHERE GAS

by

MYEUNG-JOUH WOO

and

ISAAC GREBER

Contractor's Report Submitted to NASA Lewis Research Center

June, 1994

NASA Program Monitor: **Dale C. Ferguson**

MOLECULAR DYNAMICS SIMULATION OF A PISTON DRIVEN SHOCK WAVE IN A HARD SPHERE GAS

Abstract
by

MYEUNG-JOUH WOO

Molecular dynamics simulation is used to study the piston driven shock wave at Mach 1.5, 3, and 10. A shock tube, whose shape is a circular cylinder, is filled with hard sphere molecules having a Maxwellian thermal velocity distribution and zero mean velocity. The piston moves and a shock wave is generated. All collisions are specular, including those between the molecules and the computational boundaries, so that the shock development is entirely causal, with no imposed statistics. The structure of the generated shock is examined in detail, and the wave speed, profiles of density, velocity, and temperature, and shock thickness are determined. The results are compared with published results of other methods, especially the direct simulation Monte-Carlo method. Property profiles are similar to those generated by direct simulation

Monte-Carlo method. The shock wave thicknesses are smaller than the direct simulation Monte-Carlo results, but larger than those of the other methods.

Simulation of a shock wave, which is 1-dimensional, is a severe test of the molecular dynamics method, which is always 3-dimensional. A major challenge of the thesis is to examine the capability of the molecular dynamics methods by choosing a difficult task.

This work is essentially the doctoral dissertation of Myeung-Jouh Woo, performed with Isaac Greber as Faculty Advisor. The work is supported by NASA Lewis Research Center under grant NAG 3-795; the NASA program monitor was Dale C. Ferguson.

Acknowledgments

I thank Dr. Isaac Greber, not only for providing me with this research opportunity under his guidance, but also for teaching me to observe problems in an engineer's point of view. Because of his support and encouragement, I am where I am now.

I thank Dr. Dale Ferguson, NASA Lewis, for his continuous support of this project.

I thank NASA, Lewis for funding the research and granting computing time on the CRAY YMP.

I thank Paul Barnhart of Sverdrup Technology, Inc. for providing me with his results of the direct simulation Monte-Carlo method which our results of simulation are compared.

I thank NASA, Ames for granting computing time on the CRAY YMP and YMP C90. Without this computing time grant, this project would have never finished. The user support group in NASA Ames has been very helpful in improving the execution speed of the final production runs.

Table of contents

List of figures

List of tables

List of symbols

A	area under a graph
A_c	area under a graph to point c
B	area under a graph
B_c	area under a graph to point c
b_1	location of upstream boundary in x
c	a random number within a range
c'	a random number between 0 and 1
c_{mp}	most probable speed, $=\sqrt{\frac{2kT}{m}}$
D	the diameter of the computational region
d	diameter of a particle
$\bar{E}_{l_k}$	averaged thermal energy in the l^{th} cell during the k^{th} interval
e	base of natural logarithms
F	Maxwellian distribution function
Kn	Knudsen number, ratio of mean free path to a characteristics body length
k	Boltzmann constant
L	length of the computational region
M_1	upstream Mach number
M_2	downstream Mach number
$\dot{m}$	mass flux,
N	total number of particles in a simulation
N_{coll}	collision counter
n	number of particles
n_{l_k}	number of particles in the l^{th} cell during the k^{th} interval
$\bar{n}_{l_k}$	density ratio, n_{l_k}/n_o, subscript k represents x when k=1, y when k=2, and z when k=3
n_o	initial number of particles per cell
r	radius of a particle
x_p	location of the piston
$\vec{r}_i$	$=(x_i, y_i, z_i)$, vector position of the i^{th} particle
$\vec{r}_{c_k}$	vector position of center of the computational cross section, k=2, 3.
s	spacing; center to center distance between neighboring particles
s/d	spacing ratio
T	overall temperature

T_x axial temperature, x direction

T_y lateral temperature, y direction

T_z lateral temperature, z direction

$\bar{t}$ time scale in the simulation, $= \frac{t}{\lambda / c_{mp}}$

t_1 time to collide with wall 1

t_2 time to collide with wall 2, the piston

U_1 upstream streamwise velocity

U_2 downstream streamwise velocity

U_p piston speed

u x direction velocity in a Cartesian system

u_{cut1} cutoff velocity for the upstream region

u_{cut2} cutoff velocity for the downstream region

$u_{i,l}$ u velocity of the i^{th} particle in the l^{th} cell

$\bar{u}_{l_k}$ averaged velocity in the l^{th} cell during the k^{th} interval

u_{n1} streamwise thermal component of velocity for upstream region

u_{n2} streamwise thermal component of velocity for downstream region

$\vec{V}_i$ $=(u_i, v_i, w_i)$, vector velocity of the i^{th} particle

v y direction velocity in a Cartesian system

$v_{i,l}$ v velocity of the i^{th} particle in the l^{th} cell

w z direction velocity in a Cartesian system

$w_{i,l}$ w velocity of the i^{th} particle in the l^{th} cell

x x coordinate in a Cartesian system

x_l center coordinate of a cell

y y coordinate in a Cartesian system

z z coordinate in a Cartesian system

Δ shock thickness from the maximum slope definition

$\delta\tau_{i,l}$ time occupancy of the i^{th} particle in the l^{th} cell

ε_n Streamwise thermal component of velocity

γ specific heat ratio, 5/3 for hard sphere molecules

λ mean free path

λ_1 upstream mean free path

μ viscosity coefficient

π pi, 3.141519.....

ρ density

1. Introduction

1.1 Historical background

The calculation of the thickness of a shock wave in a gas, and of distribution of density and velocity within it, is an interesting and challenging problem. The continuum description of the transition region between two well defined regions, i.e. the regions before and after the shock, has been a difficult problem. This is an old, traditional problem, and has played an important role in the development of fluid mechanics and in the development of kinetic theory computations.

Early investigations were restricted to the perfect gas equation of state and the Navier-Stokes relations with constant coefficients of heat conduction and viscosity. It was recognized separately by Rankine,[1] Lord Rayleigh,[2] and Taylor[3] that the effects of viscosity and heat conduction must be considered to properly describe the shock. Rankine gave a solution considering heat conduction but not viscosity. Taylor gave a solution for shock thickness considering the effects of viscosity, but not heat conduction. He also gave an explicit formula valid for weak shocks with both viscosity and heat conduction present. Essential results of these early works using the

continuum equations, are that the shock thickness increases with decreasing Mach number and the thickness is of the order of a mean free path in the region before the shock.

More refined solutions of the Navier-Stokes equation including both heat conduction and viscosity were obtained by Becker and later improved by Thomas.[4] Becker's work used constant transport coefficients whereas Thomas' work used the hydrodynamic theory of Becker but including the fact that the viscosity and the thermal conductivity have a temperature dependence given by the kinetic theory of gases for hard sphere molecules. Works on the shock thickness up to that date suggested that the shock thickness for all but the weakest shocks is of the order of a few upstream mean free paths. Also, the focus was on the weak shock, with the upstream Mach number below 2. A rigorous mathematical proof of the existence and uniqueness of the solution for a steady one-dimensional flow of a viscous heat conducting fluid including the effect of small viscosity and heat conductivity the was given by Gilbarg.[5] A simple method of estimating the upper and lower bounds of the shock thickness in a perfect gas has been given by von Mises.[6]

Departing from the Navier-Stokes equation, attempts made to obtain shock wave solutions include Zoller's solution to the Burnett equations,[7] the Chapman-Enskog-Burnett[8] iteration method, and the Grad[9] 13-moment method.[10] Both the Chapman-Enskog-Burnett iteration method (CEB in short) and the Grad 13-moment method are approximation methods for solving the Boltzmann equation in which the Navier-Stokes equation appears as a low order approximation in a perturbation scheme. The CEB gives a solution in the form of a series expansion for the velocity distribution function for Mach numbers less than 1.2. Zoller's solution to the Burnett equations gives predictions similar to the Navier-Stokes prediction at low Mach numbers but does not give solutions for Mach numbers above 2.35. Grad's 13 moment method for shock thickness also fails to yield solutions above Mach number 1.65.

Mott-Smith[11] took a different approach to solve the problem by satisfying moments of the Boltzmann equation rather than solving the Boltzmann equation directly, by first suggesting an approximate distribution function for the strong shock wave problem. What Mott-Smith suggested was the "Bimodal Model." According to the bimodal model, the form of the distribution function for the region between

the upstream region before the shock and the downstream region after the shock is a linear combination of the upstream and downstream Maxwellian distribution functions. Each distribution function is a function of number density and temperature. It is a better description than a skewed Maxwellian, which is the form of the resulting distribution function both from the CEB and the Grad 13-moment methods. Mott-Smith showed that his result matched the solution of the Navier-stokes equation for weak shock but there were substantial deviations from it for strong shocks. His results for strong shocks showed broader shocks than the results of the Navier-Stokes solutions. To that date, Mott-Smith's results were the closest to experimental results, and the method is applicable to wide range of Mach number.

Many other efforts have been made to improve Mott-Smith's method over the years. Muckenfuss[12] made calculations of shock thickness of argon and helium for several realistic intermolecular force law such as the Lennard-Jones 6-12, the modified Buckingham exp-6, and power law and exponential repulsive potentials. Salwen, Grosch, and Ziering[13] have developed a method of adding an arbitrary number of additional terms to the two-term Mott-Smith

distribution function for a one-dimensional shock wave and carried out a calculation for a monatomic gas of Maxwellian molecules with a three-term distribution function. It was found that the additional distribution function produced results even closer to the result of the Navier-Stokes equation for weak shocks. Radin and Mintzer[14] used the Mott-Smith bimodal distribution function as a weighting function to generate an orthogonal polynomial expansion for a solution of the Boltzmann equation for a strong shock for Mach numbers greater than 2.14. Rode and Tanenbaum[15] generalized the Mott-Smith shock thickness computation by obtaining a general solution in which the order of moment appeared as a variable. They showed a strong dependence of results on the choice of the moment to be satisfied. When second and third moment are used as Mott-Smith has done, results are essentially the same. When a higher moment is selected such as the fourth moment or higher, the results deviate dramatically from the results of lower moments.

Gilbarg and Paolucci[16] reworked the Navier-Stokes approach arguing that Mott-Smith and Zoller, each using his own method, are based, respectively, on the hard sphere molecule, for which kinetic theory gives $\mu \propto T^{1/2}$ and on the Maxwellian molecule, for which $\mu \propto T^{1}$.

The Navier-Stokes values, using empirical values of the viscosity and Prandtl number as required by the continuum theory, for helium and argon at larger Mach numbers fall between the two kinetic theory values. The Navier-Stokes equations predict for these gases a shock thickness larger than Mott-Smith's but smaller than Zoller's. They also point out that Becker's and Thomas's results are misquoted by many authors and criticisms on the shock thickness prediction of the Navier-Stokes equation stand on shaky grounds. Becker and Thomas intended to compute the thickness of the shock in air, which is composed mainly of non-monatomic gas. One of the criticisms is that the continuum theory cannot be used to predict the shock thickness since the shock transition is only a few mean free paths long. But results of experiments[17, 18, 19] on the problem of ultrasonic absorption on monatomic gases and the Navier-Stokes equation predicted the result correctly down to wavelengths of two to three mean free paths. It supports that the Navier-Stokes equation can make valid predictions even if predictions are made on magnitude as small as two to three mean free paths. Another objection stems from the fact that the Navier-Stokes equation appears as a low order approximation when starting from Boltzmann equation and Mott-

Smith and Grad attribute the "breakdown" of the Navier-Stokes equation to it.

The first wave of experiments[20, 21, 22] on the determination of shock thickness were done by a reflectivity method, as first suggested by Hornig.[23] The optical reflectivity of a shock front is sensitive to the shock thickness. The method is to introduce a plane shock in a cylindrical shock tube, which propagates down the tube until it intersects a beam of light. The optical reflectivity is measured and from it the shock thickness is estimated. These experiments produced shock thickness measurements up to Mach numbers 4.85 for argon and 3.72 for nitrogen. Shock thickness measurements using the electron beam fluorescence method were made by Robben and Talbot[24] at Mach numbers up to 17.4 for helium, argon and nitrogen. In this method, the visible radiation emitted by atoms or molecules excited by a high energy electron beam is observed. For a constant current, the emitted radiation at a point on the beam path is directly proportional to the local number density of atoms (or molecules) in the ground state. Using this method, Schmidt[25] did not stop at just measuring the thickness of the shock in argon but proceeded to obtain density profiles as well, up to Mach number 8.

Schmidt also noted that the maximum slope thickness of density was not sufficient for a detailed description of the shock structure. Using the electron fluorescence method, Muntz and Harnett[26] experimentally measured the random molecular motions in the direction perpendicular and parallel to the flow of helium at Mach number 1.59. By integration of these experimental distributions, they showed the peaking of the axial temperature within the shock. Holtz, Muntz, and Yen[27] computed velocity distribution functions within the shock wave and compared them with the measured velocity distribution function of Muntz and Harnett and found general agreement between the two. Gilbarg and Paolucci[16] state that they are doubtful on the accuracy of results of some of the published experiments.[21, 23]

As the above narration of work on shocks shows, there is agreement between theoretical works and experimental works are only for weak shocks below Mach 2. All theoretical works, be they numerical or analytical, have a limited range of applicable Mach number, usually below Mach 3. One notable exception was the work of Mott-Smith. Note that experimental work on weak shocks are in abundance but are rare for strong shocks. Departing from

theoretical and experimental approaches, there have been attempts to simulate the shock problem. Two major simulation techniques are the direct simulation Monte-Carlo method[28] and the molecular dynamics method.[29] Both methods are discussed further in a later section.

At an early stage of the development of the direct simulation Monte-Carlo method (DSMC in short), Bird studied the shock structure in a rigid sphere gas.[30] Shock structures at Mach numbers 1.5, 3, 10, and 30 were compared with the Navier-Stokes and the Mott-Smith predictions. Later, Bird used an improved DSMC to generate profiles of shocks at Mach numbers 1.5, 3, and 10,[31] and showed not only the profile of density, velocity and temperature but also that the axial temperature profiles for both Mach 3 and 10 had peaks whereas density and axial velocity vary monotonically. The Mach 1.5 results agreed well with the Navier-Stokes but the shock thickness increases beyond the Navier-Stokes values as the simulation Mach number increases and at Mach 10 the thickness is greater than that of the Mott-Smith thickness. The axial temperature peaks at Mach 3 and 10 were in good agreement with the theory of Yen.[32] Shock profiles for strong shocks have been studied further by

Bird with the DSMC method employing different intermolecular power force laws.[33] Mach number of 8, 25, and 100 were used and profiles of the shocks were compared at each Mach number to show the effect of different force laws. The closest agreement for a strong shock between an experimental result and a computed one appears to be between the experiment of Schmidt[25] and Bird's DSMC.[33, 34] Schmidt's experimentally obtained density profile for argon at Mach 8 was practically duplicated by Bird with his DSMC method that used a model of gas molecules that have an inverse 12th power intermolecular force law.

The MD method has been applied almost exclusively in the simulation of shocks in liquids or solids. In these computations, Maxwellian molecule or Lennard-Jones molecules are typically used as the choice for intermolecular force laws. Tsai and Trevino[35] studied the propagation of a planar shock in a dense Lennard-Jones fluid. Hoover[36] simulated the structure of a shock wave front in a dense Lennard-Jones fluid, and found a good agreement with the solutions of the Navier-Stokes equation for strong shock. Holian, Hoover, Moran and Straub[37] used 4,800 particles to simulate a dense-fluid shock wave and found differences to be relatively small

when compared with the Navier-Stokes continuum mechanics. Barker, Fisher, and Watts[38] compared thermodynamics properties of liquid argon obtained by MD and Monte-Carlo methods and concluded that both results are in good agreement. They suggest that agreement among the results of simulations of argon assuming Lennard-Jones intermolecular force law and experimental data might have been accidental since works of Guggenheim and McGlashan[39] and McGlashan[40] showed that the intermolecular force between the argon atoms was not Lennard-Jones 6-12 potential. Fiscko and Chapman[41] studied comparisons of shock structure using independent continuum and kinetic theory approaches and simulation Mach numbers ranged from 1.4 to 35. Methods used are DSMC, Steady state Navier-Stokes equation, and Burnett equation that was determined by relaxation to a steady state of time dependent continuum equation. DSMC results showed excellent agreement with published experimental results.

It appears that the only previous attempt to study the shock structure in a gas using MD was that of Niki and Ono.[42] They used 135 hard sphere particles in a computational region length of 40 to 60 times the particle's diameter with unspecified width of the

computational region to simulate the shock at Mach numbers ranging from 1.73 to 11.3. The density profiles presented show large scatter and the shock thickness increases with increasing Mach number and the differences in shock thickness are too big since the simulation Mach number changed only from 5.12 to 7.32. This is contrary to known behaviors of shock thickness since the thickness of strong shock does not change much with Mach number for hard sphere molecules and shock thickness should decrease with increasing Mach number, if any change in thickness should occur.

There are other simulation techniques such as the test particle method,[43] Hicks-Yen-Nordisieck method.[44] The test particle method studies a path of a particle introduced one at a time, therefore it is suitable for study of the free molecular flow where intermolecular interaction does not occur. The Hicks-Yen-Nordisieck method (HYN for short) is a simulation technique using a finite difference technique of solving the Boltzmann equation and Monte-Carlo sampling to evaluate the Boltzmann collision integral. An advantage of the HYN method is that because it is a numerical solution of Boltzmann equation rather than direct simulation of the gas, it is possible to employ the various models of intermolecular force law easily. But,

the method has a serious restriction in the sense that it requires a good initial estimate of the flow in order to achieve convergence on a subsequent iteration. The best known of this method is the BGK model named after Bhatnagar, Gross, and Krook.[45]

As stated earlier, DSMC appears to be the only technique that provides good agreement with experimental results for a wide range of Mach numbers. It has been voiced strongly that MD may have limited capabilities since its applicability is limited to the dense state of matter such as liquid or solid.[46, 47] Survey of literature on the application of MD shows that most applications have dealt with liquids or solids.[48]

Recently Greber and Wachman[49] used MD successfully to study various fluid flows such as Couette flow, and heat transfer between two flat plates to show the velocity slip and the temperature jump. Using the scaling rules and the time averaging technique of Greber and Wachman, test computations on supersonic flow past blunt bodies[50, 51] showed that MD can be used to simulate dilute gases. The examination of shock structure using the molecular dynamics method is a severe test of the capability of the method, and

it has been suggested that it is a test that molecular dynamics should meet.

1.2 Simulation techniques

As stated, there are two major simulation techniques, DSMC and MD. The use of random numbers is the distinctive feature of Monte-Carlo procedures and the simultaneous following of trajectories of a large number of simulated molecules within a region of simulated physical space is the distinctive feature of the molecular dynamics method.

Probably the first Monte Carlo simulation of molecular motion is by William Anderson as reported by Kelvin[52] in 1901. Since there were no computers then, he managed to generate the random numbers by shuffling decks of numbered cards and used them to calculate a total of five thousand molecular impacts with surfaces and three hundred intermolecular collisions.[47] In Monte-Carlo methods, probability is computed to determine whether two molecules should interact. Probabilities are computed to determine velocities after the interaction of two molecules. Monte-Carlo hopes

that differences between exact values and its probabilistic answers are a statistical deviation which will disappear when averaged.

Bird suggests that his DSMC method solves the Boltzmann equations directly by uncoupling the molecular motion and the intermolecular collisions over a small time interval, and by suitable choice of time scale, and by the manner of selecting colliding particles. Bird describes his DSMC technique as follows.[28]

"(I) All molecules are moved through distances appropriate to their velocity components and small time increment. Appropriate action is taken if the molecules cross the boundaries representing solid surfaces, lines or surfaces of symmetry, or the outer boundary of the flow. New molecules are generated at boundaries across which there is an inward flux.

(II) A representative set of collisions, appropriate to the small time increment, is computed among the molecules. The pre-collision velocity components of the molecules involved in the collision are replaced by the post-collision values. Since the change in flow variables across a cell is small, the molecules in a cell at any instant are regarded as a sample of the molecules at the location of the cell.

This enables the relative positions of the molecules within a cell to be disregarded when choosing a collision pair."

Bird has produced many papers from the study of the rate of phase transition[53] to the simulation of a dissociating diatomic gas.[54] His method has been embraced by many and the list is too long to repeat here. An overview of the Monte Carlo simulation of gas flow is given by Bird.[47]

The first MD simulation attempt using a computer appears to be the work of Alder and Wainwright[29] on the phase transition for a hard sphere system. Basically they were trying to compute the rate of approach to the Maxwellian equilibrium when all molecules started at the same speed. They attempted using 32, 108, 256 and 500 particles in a rectangular computational domain with periodic boundary conditions but only the results for 32 and 108 particles were reported. In general, MD deals with a set of particles with given initial conditions and force laws. Force laws can include long and short range forces. The basic approach in MD calculations is that of kinetic theory, in which collisions between gas molecules determine the average behavior of a gas. Further discussion of the MD method is given in a later section where its implementation for our problem is

discussed in detail. A review of the status of MD was first given by Hoover[55] in 1983 then updated in 1986 by Evans and Hoover.[56] Both cite an extensive literature covering a broad spectrum of engineering and will serve as good starting points to grasp the current state of MD.

There has not been a serious attempt to generate shock structure by MD using a hard sphere model of a gas except that of Niki and Ono,[42] probably because until recently there have not been sufficient computing resources available. Recently there have been rapid developments in computing speed and storage capacity such that a reasonable simulation by MD is feasible. As evidence of the trend, there have been successful applications of MD for various fluid flows as stated. Therefore, the method has grown up to a point where its general validity should be examined by applying it to a difficult problem that had previously been thought to be beyond the capability of MD. It has been said that the MD calculations are generally unable to simulate dilute gases, let alone the simulation of shock structure, due to its shortcomings, as Bird has pointed out.[47, 57] The shock structures generated by Bird using DSMC have been successfully duplicated by Barnhardt[58] for Mach numbers 1.5, 3,

and 10. Results of MD simulation of the same set of Mach numbers are directly compared with the DSMC results. Results of direct comparisons should settle questions on the applicability of the MD method for problems related to dilute gases.

2. Statement of the problem and approaches

The structure of a normal shock wave is computed using the MD technique for Mach numbers 1.5, 3, and 10. The purpose of the simulation is to obtain density profiles, axial velocity profiles, and temperature profiles for each Mach number. To examine the results of MD, they are compared with published results including those of DSMC, Mott-Smith and Navier-Stokes.

There are two possible approaches in trying to simulate a shock. One is to generate a stationary shock and the other is to generate a moving shock within the computational region. Each has advantages and disadvantages. In principle, the generation of the stationary shock is natural because the collection and interpretation of data is easy. Serious attempts have been made for some time to generate the stationary shock and the time restraint has prevented further examinations of the problem since the generation of a moving shock produced satisfactory results. But the methodology is described in detail in the following section as a stepping stone for those who may wish to further study the generation of stationary shocks. The method of generating a moving shock is chosen after some test computations and it leads to the results presented.

2.1 Generation of stationary shock

The basic idea is to create a simulation where a stationary shock develops and is completely contained within the computational region.

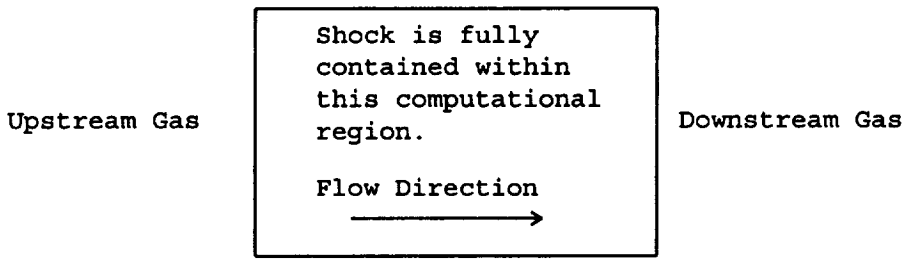

Figure 1 Stationary shock

An advantage of generating a stationary shock is the ease of data reduction from the simulation as stated. Disadvantages stem from having to set rather complicated boundary conditions in order to keep the shock stationary.

Since a finite computational region is used, boundary conditions must be imposed on the lateral boundaries such that the finite region mimics the infinite region. This can be achieved by specifying either specular reflection or periodic boundary condition on the lateral boundaries. Since the mass flux must be satisfied for the two axial boundaries, molecules that leave the computational

region must be somehow replaced. If Maxwellian velocity distributions are assumed for the two regions axially outside the computational region, one can compute what the ratio is between the number of molecules entering into the computational region through the upstream boundary to number of molecules entering through the downstream boundary.

Particles in the upstream region have a mean velocity that is the upstream velocity and deviations from the mean appropriate to the upstream temperature. A particle can enter the computational region through the upstream boundary if its inward normal component is between $-U_1$ and $+\infty$. A particle can enter the computational region from the downstream boundary if its inward normal component is between $-U_2$ and $-\infty$. Flux of particles, Φ, is given by:

$$\Phi = \int u_n \cdot F \, dc_1 dc_2 dc_3$$

The probability of a particle with a normal component of velocity, u_n, crossing a surface is proportional to $u_n F$, not to F, and the normal component of velocity should be selected from a distribution function $u_n F$. Therefore the inward normal component of the velocity is selected from a Maxwellian weighted by the appropriate normal

component. This is done so that the particle flux across the surface will be the same as that for the fictitious equilibrium gas outside the surface. Appropriate weighted Maxwellians of the normal components of velocity at the upstream and downstream boundaries have the following forms:

$$u_n \cdot F_{upstream} \propto (U_1 + \varepsilon_n) \cdot e^{-\varepsilon_n^2}$$
$$u_n \cdot F_{downstream} \propto -(U_2 + \varepsilon_n) \cdot e^{-\varepsilon_n^2}$$

In order to select a value randomly within a distribution function, one use two approaches; one is to use the rejection technique and the other is to use the one-step method.

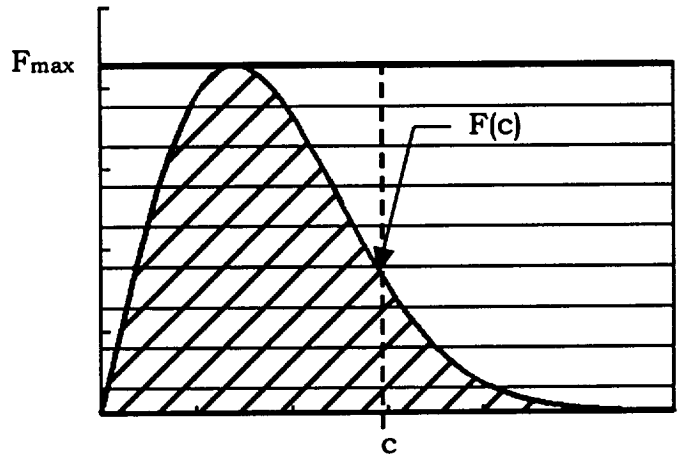

Figure 2 Rejection Technique

In Figure 2 and the following discussion, *F(c)* refers to any normalized distribution functions, for example the Maxwellian or the weighted Maxwellian. In the rejection technique, a value is accepted

or rejected according to the following procedure. If the distribution function exists for $0 < c < \infty$, then a cutoff must be selected to restrict c to a finite range. First, a random number, c, is selected within the region considered and $F(c)$ is computed. A second random number, c', between 0 and F_{max} is selected. If c' is less than $F(c)$, the value c is then accepted, otherwise the procedure is repeated. In essence, the selection is made in proportion to the value of $F(c)$, that is in proportion to the "height" of the curve.

An alternative technique, that avoids the need for establishing cutoffs in the range of c, selects values in proportion to the area under the $F(c)$ diagram, that is, proportional to the accumulated probability.

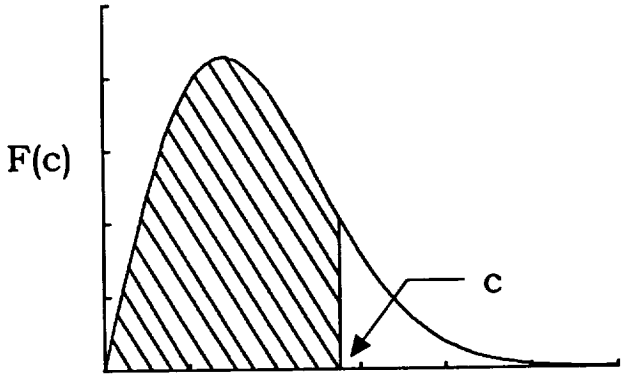

Figure 3 One step method

In the one step method, a random number, *g*, selected between 0 and 1. Then one finds *c* such that

$$g = \int_0^c F(y)dy$$

If the integral has an analytical form, the *c* can be chosen in one step! In our problem the lack of an analytical form of the integral forced a numerical approach, and the rejection method is used exclusively.

The weighted Maxwellian distribution function exponentially decreases. One must select cutoffs that cover a range large enough to capture most of the particles of interest. However the range must be kept small enough to avoid excessive rejections, and correspondingly excessive computational time. Note that the number of rejections is proportional to *(1 - F)*, so that one must avoid very small values of *F*. The range of values of ε_n for the upstream and downstream regions should be compatible. They should contain the same fraction of the total flux of particles crossing the surface. The weighted Maxwellian distribution functions with and without cutoffs are sketched in Figure 4, where "A" and "B" refers to the upstream and downstream boundaries, respectively.

$$A = \int_{-U_1}^{\infty} [(U_1 + \varepsilon_{n_1}) \cdot e^{-\varepsilon_{n_1}^2}] du_{n_1} \qquad A_c = \int_{-U_1}^{U_{cut1}} [(U_1 + \varepsilon_{n1}) \cdot e^{-\varepsilon_{n_1}^2}] d\varepsilon_{n_1}$$

$$B = -\int_{-\infty}^{-U_2} [(U_2 + \varepsilon_{n_2}) \cdot e^{-\varepsilon_{n_2}^2}] d\varepsilon_{n_2} \qquad B_c = -\int_{-u_{cut2}}^{-U_2} [(U_2 + \varepsilon_{n_2}) \cdot e^{-\varepsilon_{n_2}^2}] d\varepsilon_{n_2}$$

Then the cutoffs are chosen such that $B_c/B = A_c/A$.

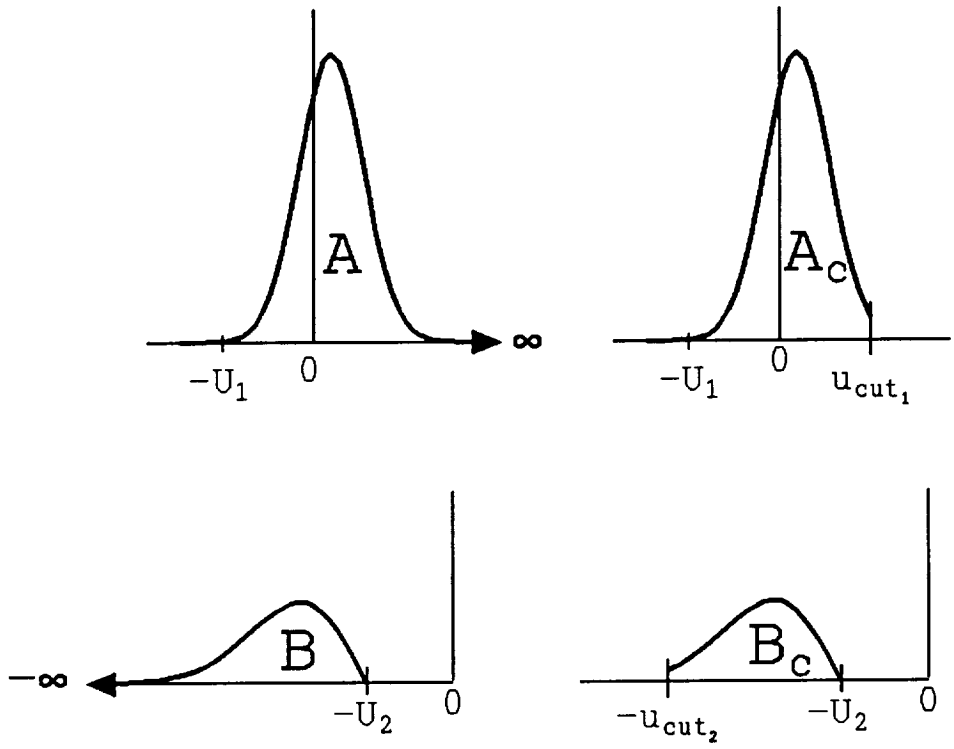

Figure 4 Finding cutoffs

When a particle leaves the computational region, a replacement is made at the boundary. The determination of which boundary can be made in 2 ways for the case of fixed boundary conditions at downstream boundary. One is to select a boundary giving them an equal chance and letting the distribution function reject the unlikely event. The other is to choose the boundary proportional to the flux

ratio and insist that a particle be re-emitted from that boundary. If the temperature, density and the velocity at the boundaries are held fixed then either method can be used. If any of the boundary quantities are allowed to be free, one must use the first method. One then must re-compute at each selection of an entering particles.

The generation of random numbers can be a problem. Test computations show that the correct flux ratio is not achieved immediately for a small sample and the rate of approach to the correct ratio is dependent on the initial seed chosen. This is a serious problem. First the results depends on the initial seed for the random number generator. Secondly, almost all particles leave the computational region through the downstream boundary and they end up being replaced by particles that enter the computational region from the upstream boundary. Since a small number of particles enters the computational region from the downstream boundary and some form of a shock profile exists within the computational region, one must conclude that this small number of particles that enter the computational region are the cause of the shock. Since the rate of insertion is not constant, the position of the shock oscillates. Therefore sensitivity of the position of the shock to

the small fluctuation in number flux must be addressed and resolved.

Even though the simulation assumes the existence of an upstream gas just outside the computational upstream boundary, there is no clearly defined upstream region which has constant values for density, velocity and temperature. Since the ratio of density to an unknown upstream reference is defined, the density ratio for the downstream region will shift by a large magnitude when the reference density is shifting by a small magnitude, even though both shifts may be the same percentage. This is because of the constant number of particle condition imposed and partially on the sensitivity of the shock position within the computational region. As the shock oscillates due to the fluctuation in the mass flux ratio, the upstream density may rise or fall depending on the direction the shock moves. For example, if the shock moves towards the upstream boundary, the physical size of the high density region increases. Particles somehow have to redistribute themselves to accommodate the shift in the position of the shock such that the density ratio is maintained as well as the fixed number of particles conditions. Likewise, if the shock moves towards the downstream boundary,

particles must redistribute themselves to accommodate the change of shock location.

Several preliminary computations were performed using a rectangular cylindrical and a circular cylindrical computational region. Initial configurations included the step profiles in flow properties in which the upstream and downstream values are used and the step was placed at various locations within the computational region, even at the boundaries of the computational region. Overall results showed that the shock wave was not stable but would move slowly in most cases and some would even begin to head towards an uniform profile. For the cases when motion of shock was slow, some intermediate results showed that values obtained for the downstream region were off by as much as 20% from the theoretical downstream values. It appears that the computational condition of using constant number of particles within the computational region at all time and the problems associated obtaining a correct flux ratio with the random number generator took its toll.

2.2 Shock tube simulation

The basic idea is to simulate a piston driven shock wave as in the schematic diagram shows in Figure 5. The piston is given a constant speed impulsively and moves into a region containing particles. As the result, a shock wave is generated. Boundary conditions are specular everywhere.

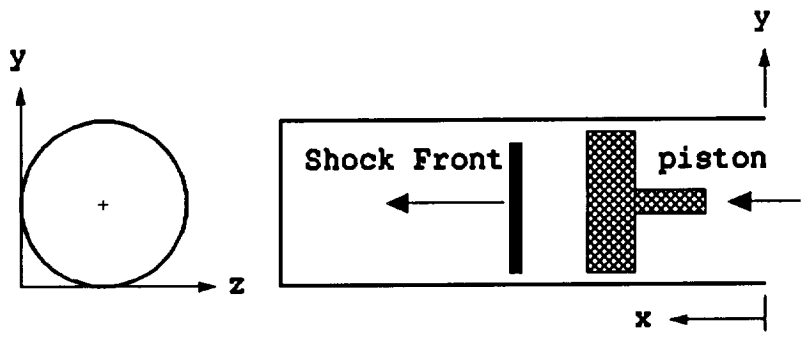

Figure 5 Piston driven shock wave and coordinate system

The simulation of a piston driven shock wave has many advantages. The initial number of particles in a cell remains relatively constant throughout the simulation until the shock wave nears. This gives a constant value of the density in the upstream region, and correspondingly the upstream mean free path is also a constant value. The constant value of density for the upstream region is an important feature for various purposes. One can only

show a profile of the density as a ratio of a local density to the upstream density for a comparison purpose and the constant upstream density makes analysis simple and clear.

Secondly, because of specular boundary conditions, there are no statistics involved; the use of probability or the random number generation is not required at all during the simulation. Random numbers are used only in the initializing stage of the simulation to distribute particles evenly within the computational region and to assign initial velocities according to the Maxwellian velocity distribution function. Therefore the system is deterministic at all time, and naturally evolves from the uniform initial configuration of particles to a shock profile. This brings an interesting observation. The whole process is entirely reversible at any point during the simulation. At any desired moment, all velocities including that of the piston can be reversed to get back to any configuration of particles in the past. Actually there is nothing that stops one from going past the initial stage.

Thirdly, the simulation of the piston driven shock wave closely resembles the corresponding physical experiment. Even though assumptions are made on the nature of interactions among particles,

and between particles and computational boundaries, the model can simulate otherwise difficult experiments. For example, no heat transfer condition is imposed easily just by assigning specular conditions at all boundaries. A physical experiment with Mach number in excess of 100 may be practically impossible whereas the simulation can be done with just a change of one line in the source code.

The shock can be examined as it develops and if there is a sufficient distance between the shock wave and the piston at the time of the shock reflection, the shock reflection phenomena can also be studied in detail.

Barring leakage, an experiment with piston driven shock wave has a fixed number of molecules throughout the experiment. Having a fixed number of particles in the computational region during the simulation, allows for a direct comparison with experiments.

3. Methodology

Our computational region can be described as an imaginary circular cylinder containing many moving particles. Particles collide among each other and with boundaries. The basic procedure of the MD technique used in the current computations is as follows.

1. Initialization (See section 3.1)

    Select the computational region

    Select the number of particles and a spacing ratio

    Assign positions and velocities to particles

2. Main Loop (See section 3.2)

    Find the shortest time to collide

    Solve the collision dynamics for the pair

    Repeat the loop

3. Post processing (See section 3.3)

    Extraction of data

3.1 Initialization

Each particle is taken as a sphere with a spherically symmetric mass distribution. The simulation allows variations in the diameters and the masses of particles, but the present computations are based

on identical particles. The number of particles within the computational region remains fixed throughout the simulation.

3.1.1 Size and number of particles

The number of particles is directly connected to the total computing time and also storage requirements. A small number of particles is desirable in order to finish the simulation within a reasonable amount of time and storage. An important criterion in determining the total number of particles as well as their diameter is the spacing ratio. The spacing ratio is defined as a ratio of the center to center distance between particles to the particle diameter. Following formula relates the number of particles and the spacing ratio.

$$n^{1/3} = \frac{1}{\sqrt{2} \cdot \pi} \cdot (s / d)^2 \qquad (1)$$

where
s = distance between neighboring particles' centers.
n = number of particles in a cube whose side is a mean free path, λ. $\lambda = 1/(\sqrt{2}\,\pi\, n\, d^2)$
d = diameter of a particle.

One tends to expect that an inordinately large number of particles would be needed to perform computations that provide

reasonable results for a gas. Fortunately, this not so. Greber and Wachman[49] have shown that remarkably small numbers of particles are sufficient, and that one can estimate the particle density in the simulation that is needed to achieve desired levels of agreement with gas behavior. The basic idea is that if the ratio of the distance between particle centers to the diameter is sufficiently large, then this ratio is no longer an important scaling parameter, and that the Knudsen number, *Kn*, (ratio of mean free path to a characteristic body length) becomes the sufficient gas length parameter to achieve approximate dynamic similarity. For a fixed body length, equal *Kn* of a prototype and simulation implies equal surface areas of particles, and also equal ratio of gas-gas to gas-body collision frequency. A small number of large diameter particles can then be used to simulate the behavior of a large number of small diameter particles, using equal surface areas of the simulation and prototype particles. As an example error estimate, a spacing ratio of 3 results in collision frequency error of approximately 8% as compared with the infinite spacing ratio value.[49] For our simulation, the smallest spacing ratio is chosen to be 3, and the spacing ratio is smallest in the high density area.

Since the lateral boundary cannot be infinite, it is necessary to select the computational Knudsen number which is the ratio of the upstream mean free path, λ_1, to the diameter, D, of our computational region. MD must specify an enclosed computational region because MD is 3-D in nature. DSMC can treat the shock problem as 1-D problem so that the reduction in dimensionalities help DSMC to compute the shock in less time and storage. DSMC keeps and tracks the axial coordinate of particles, but does not even keep two lateral coordinates. Since MD is 3-D in nature, there is no computational advantage for MD dealing with 2-D or 1-D problem. This is why the ratio of the width of the computational region to the upstream mean free path becomes a parameter in MD but not in DSMC.

The theoretical density ratio is used in order to estimate the number of particles needed for the simulation which satisfies the spacing ratio and Kn requirement. It should be emphasized that the theoretical density ratio is used only for this particle number estimate; the simulation does not force the density ratio in any way. The density ratio naturally evolves by itself. The theoretical density

ratio between the upstream and the downstream gas is a function of upstream Mach number, M_1, and the specific heat ratio, γ:

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma+1)\cdot M_1^2}{(\gamma-1)\cdot M_1^2+2} \qquad (2)$$

where $\gamma = 5 / 3$ for hard spheres.

The following table shows results of applying equations (1) and (2) with the *Kn* requirement for simulations at different Mach numbers with 3,000 particles. The spacing ratio in the downstream region, $(s/d)_2$, is fixed at 3.

Mach number	n_2/n_1	$(s/d)_1$	Kn_1	d/λ_1
1.5	1.714	3.590	0.556	0.0960
3	3.000	4.327	0.812	0.0549
10	3.883	4.716	0.967	0.0424

Table 1 Sample parameters used in MD simulation

For our simulations, the total number of particles and the smallest spacing ratio are chosen first. The Knudsen number is then determined by the chosen conditions. The resulting Knudsen numbers are not round figures. The effect of computational Knudsen

number is examined by increasing *Kn* to about 1 at Mach 1.5 and decreasing *Kn* to about 0.5 at Mach 3.

3.1.2 Computational region

Our computational region is a circular cylinder shown as a sketch in Figure 5. The diameter of the computational region is about 1 to 2 in term of the upstream mean free paths.

The length of the circular cylinder is determined basically by trial and error, to achieve a fully developed shock profile. The length of the cylinder must be longer for high Mach number simulation, since the generated shock will travel at a higher speed giving less time to observe and less time for the downstream region to develop before the shock reflection occurs. The distance between the shock wave and the piston must be sufficiently long. The unit of measurement of time is defined as the time it takes for a particle traveling at the most probable speed to travel an upstream mean free path. Assuming that the shock forms the moment the piston starts to move, and for the computational region length of 54, 49 and 41 in units of the mean free paths, the shock will take about 15, 15, and 4

time units for Mach 1.5, 3, and 10 respectively before the shock reflection occurs.

One end of the computational region acts as a piston. The piston starts moving from the beginning of the simulation at a constant speed until the simulation is over. The piston speed is chosen from the theoretical upstream and downstream velocity in anticipation that the generated shock will travel at the theoretical speed. This is done purely for the convenience of comparison with published results.

3.1.3 Assigning the locations and velocities of particles

Using a Cartesian coordinate system, each particle center location is chosen such that the particle positions are randomly selected within the computational region.

The three components of velocity of the particles are selected randomly within a Maxwellian velocity distribution. A rejection technique is used to select the velocities within the Maxwellian. The cutoffs are set at ±3 times the most probable speed where the Maxwellian is already on the order of 10^{-4}. The previously mentioned one step method is not used here due to the lack of inverse function

of the Error function. Since velocities of the particles are assigned as a part of the initialization only, computing time is not an issue.

3.2 Main loop

There are no long-range forces involved. Thus, the particles move in straight lines at constant speed between instantaneous collisions. Time to collide is computed for each pair which can be a particle to particle or a particle to a boundary. A row by row search is made to find the first collision among all possible collision pairs. The appropriate collision dynamics are applied to the colliding pair and the time to collide is updated for any pair that has the current colliding pair as its member. The process of finding the colliding pair and solving the collision dynamics is the main loop and the most time consuming.

3.2.1 On particle to particle collision

3.2.1.1 Time to collide between particles

The time to collide is calculated between the i^{th} particle and the j^{th} particle. The time to collide between two particles is computed

from the condition that the distance between centers of two particles is the sum of the radii of two particles at a collision. $\vec{r}$ is defined a vector position of a particle, $\vec{V}$ is a vector velocity of a particle, r_i is the radius of the i^{th} particle, and k has values of 1, 2, 3 corresponding to the Cartesian coordinate. The collision condition is then given by:

$$\begin{aligned} \vec{r}_{k_{collision}} &= \vec{r}_{k_{initial}} + \vec{V}_k \cdot t \\ \left|\vec{r}_i - \vec{r}_j\right|_{collision} &= r_i + r_j \\ &= 2 \cdot r \end{aligned}$$

Since each particle travels at constant velocity between collisions, the time of collision t is given by the solutions of the following quadratic equation:

$$At^2 + Bt + c = 0 \qquad (3)$$

$$\text{where } A = \sum_{k=1}^{3} (\vec{V}_{i_k} - \vec{V}_{j_k})^2$$

$$B = 2\sum_{k=1}^{3} (\vec{r}_{i_k} - \vec{r}_{j_k}) \cdot (\vec{V}_{i_k} - \vec{V}_{j_k})$$

$$C = \sum_{k=1}^{3} (\vec{r}_{i_k} - \vec{r}_{j_k})^2 - (r_i + r_j)^2$$

3.2.1.2 Velocities of the particles after collision

All collisions between a particle and another particle are taken to be elastic. Incident velocities of the colliding pairs and their physical locations at the impact make computing the velocities for

the colliding pairs after collision simple arithmetic. If a plane is drawn through the point which two particles meet, the component of velocity parallel to the plane remain unchanged whereas for identical particles, the normal velocities are interchanged. For two identical particles, velocities after the collision are then simply computed as;

$$\vec{V}_{i_{k_{after}}} = \vec{V}_{i_k} + (\vec{r}_{j_k} - \vec{r}_{i_k}) \cdot (CJ - CI)$$

$$\vec{V}_{j_{k_{after}}} = \vec{V}_{j_k} + (\vec{r}_{j_k} - \vec{r}_{i_k}) \cdot (CI - CJ)$$

$$\text{where} \quad CI = \frac{\sum_{k=1}^{3} [\vec{V}_{i_k} \cdot (\vec{r}_{j_k} - \vec{r}_{i_k})]}{r_i + r_j}$$

$$CJ = \frac{\sum_{k=1}^{3} [\vec{V}_{j_k} \cdot (\vec{r}_{j_k} - \vec{r}_{i_k})]}{r_i + r_j}$$

One assumption is made in setting the condition for the particle to particle collision, that is only binary collisions are allowed. As a direct consequence of the assumption, the simulation treats a simultaneous collision of three or more particles as sequential binary collisions. Simultaneous collision of many particles is anticipated as being rare. Since only binary collisions are allowed, the colliding pair will be decided by the search method used in the case of such

collision. The results will be different from actual results of a collision involving three or more particles.

3.2.2 Particle to boundary collision

Particles are not allowed to leave the computational region. A particle is in a collision position with the boundary of the computational region when the center of the particle is exactly a particle radius away from the boundary of the computational region. Therefore the time of collision of a particle with either the upstream or the downstream boundaries is a solution to a linear equation. The equation is simply a distance divided by the relative approach velocity. If t_1 is the time to collide with the upstream boundary which is at b_1, t_2 is the time to collide with downstream boundary (This is the piston.) which is at x_p and moving at U_p, we can write for t_1 and t_2 as follows:

$$t_1 = \frac{(b_1 - \vec{r}_1) - r_i}{\vec{V}_1} \qquad t_2 = \frac{r_i - \vec{r}_1 + x_p}{\vec{V}_1 + U_p}$$

Since a Cartesian coordinate system is used exclusively, the time to collide between a particle and the lateral boundary of the computational region involves the solution of a quadratic equation,

not a linear equation as would result with a rectangular region. This is due to the shape of our computational region. Finding the time to collide with the lateral boundary follows an almost identical procedure as solving for the time to collide between two particles.

$$At^2 + Bt + c = 0 \tag{4}$$

$$\text{where } A = \sum_{k=2}^{3} \vec{V}_k^2$$

$$B = 2 \cdot \sum_{k=2}^{3} [\vec{V}_k \cdot (\vec{r}_k - \vec{r}_{c_k})]$$

$$C = \sum_{k=2}^{3} (\vec{r}_k - \vec{r}_{c_k})^2 - (\frac{D}{2} - r_1)^2$$

It can be thought of as applying the procedure used in computing the time to collide between two particles to a 2-dimensional problem as the axial direction plays no role. The distance at collision is no longer the sum of two radii as in the particle collision but the difference between the two, the radius of the cross section of the computational region and the radius of the particle. Since the lateral computational region is stationary with respect to the particles within it, the equation (4) can be obtained by assuming zero velocity for a particle in the quadratic equation (3). For a particle which is not touching the boundary, the equation gives two real roots. One is a positive root representing the collision time

and the other is a negative value, representing the collision time with the reversed velocity. For a particle that is touching the lateral boundary of the circular cylindrical computational region, the term C drops out of the equation and time to collide is simply $-B/A$ in equation (4).

For our simulation, specular reflection is imposed at all boundaries of the computational region. During the specular reflection, the incident normal component of relative velocity is reversed whereas the tangential components are preserved. For our lateral boundary which has a circular cross section, a coordinate transformation of velocities is required. Since only the normal component of the velocity reverses sign for the specular reflection condition, the axis of the coordinate system is rotated such that the normal coincides with the line joining the center of the particle and the center of the circular cylinder's cross section containing the particle center. Since the rest of the computations are done in the Cartesian coordinate system, another rotation of coordinate is required to return to the same coordinate system.

3.3 Post processing

During the computation, the computational region that particles can occupy shrinks because the piston is continuously moving. The shock is moving ahead of the piston getting farther and farther away from the piston. A data collecting scheme with fixed data collecting cells is used to describe the system which is comparable to setting up data collecting stations in a physical experiment. Since the instantaneous location and the speed of the shock wave are unknown, data cells cannot be moved following the shock wave every step of the way. Instantaneous shock wave speed cannot really be detected because the simulation is dealing with a discrete system.

For the purpose of data collection, the computational region is cut into slabs or cells of equal width which lie perpendicular to the direction of the flow. The location of a particle is regarded as the location of its center. Due to the physical size of the particle, the center can not be located closer than 1 radius away from the computational region. Therefore, there are regions within the computational region that particles cannot occupy fully due to the physical size of particles as shown in Figure 6 as the region between

the computational boundary and the dotted lines. These regions are excluded from the data collection cells.

The width of the cells are approximately one upstream mean free path. Note that width of the data collecting cells dictate the accuracy of the simulation. The cells can be regarded as measuring stations in an experiment. In theory, one measuring station should be able to describe the shock wave as it goes by the measuring station. But our system is discrete system and there are not enough particles to produce a smooth shock profile from a measuring station. Table 2 shows the width in terms of the upstream mean free path used in the various simulation runs.

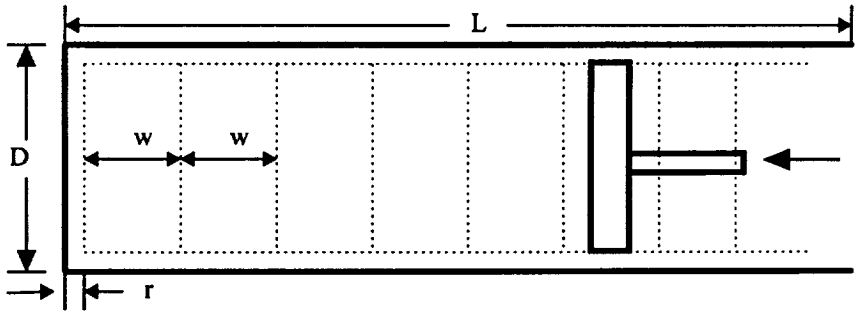

Figure 6 Data Collecting Cells

For a fixed computational region length, there is a trade-off between having many narrow cells and having few wide cells. If the width is smaller than a mean free path, the chance of a collision

occurring outside the concerned data collecting cell by a particle currently residing within the cell increases. Scatter in the resulting data offsets the gain in the number of data points. Too wide a width will result in blurring of the profile and peaks may get hidden. This may offset the advantage of having smoother profiles. Therefore the smoothness of the final results involves more than just a displayed value at a point. The exact manner of how results are obtained needs to be revealed, a practice rarely seen in the field of the rarefied gas flow simulation. The fact is that some form of averaging is a must for simulation. To describe the discrete system in terms of density, velocity, and temperature, a small region of interest must be defined and variables are determined for that region. The resulting values can be sensitive to the size and location of the region.

Mach number	w/λ_1	λ_1/D	N	L/λ_1
1.5	1.80	0.56	3,000	54.
1.5, test run	1.02	0.98	500	31.
3	1.23	0.81	4,000	49.
3, test run	0.97	0.52	4,000	19.
10	1.03	0.97	4,000	41.

Table 2 Cell width and computational Knudsen number

For the purpose of data reduction, snap shots of various variables listed later are saved at a regular time interval. Using the information, density, velocity and temperature are extracted. The time interval used is obtained, for convenience, from the theoretical wave speed prior to the simulation and equals the time for the shock wave to move about one upstream mean free path. If a longer interval is used such that the piston is able to cross more than 1 data cell, there will be significantly fewer data points as the cells the piston has crossed during an interval must be discarded as these cells will have incomplete information since sizes of these cells is time dependent. If the next collision occurs at a later time than the time for the shock wave to move one upstream mean free path since the information was recorded, following information is recorded. Summation variables continuously collect data from the beginning to end of the simulation. Note that outputs are not averaged values so far but instantaneous values at the time τ_k.

For the k^{th} time, τ_k,

1. Current collision count, $(N_{coll.})_{\tau_k}$
2. Time the data collection begun, $t_b = \tau_{k-1}$
3. Time that the data collection ended, $t_e = \tau_k$

4. Location of the piston, $\left(x_p\right)_{\tau_k}$

and for each cell,

1. The center coordinate of the l^{th} cell, x_l
2. Sum of the time of occupancy in the l^{th} cell, $\left(\sum_i \delta\tau_{i,l}\right)_{\tau_k}$
3. Total number of particles in the l^{th} cell, $\left(\sum_i i_l\right)_{\tau_k}$
4. Sum of the axial velocity, u, in the l^{th} cell, $\left(\sum_i u_{i,l}\right)_{\tau_k}$
5. Sum of the first lateral velocity, v, in the l^{th} cell, $\left(\sum_i v_{i,l}\right)_{\tau_k}$
6. Sum of the second lateral velocity, w, in the l^{th} cell, $\left(\sum_i w_{i,l}\right)_{\tau_k}$
7. Sum of the u squared in the l^{th} cell, $\left(\sum_i (u^2)_{i,l}\right)_{\tau_k}$
8. Sum of the v squared in the l^{th} cell, $\left(\sum_i (v^2)_{i,l}\right)_{\tau_k}$
9. Sum of the w squared in the l^{th} cell, $\left(\sum_i (w^2)_{i,l}\right)_{\tau_k}$
10. Time that the cell stops collecting information, t_{stop}

The information is then used to compute the differences from τ_1 to τ_0, τ_2 to τ_1, τ_3 to τ_2, and so forth. k snap shots give k sets of values. The intention is to obtain time averaged values for density, velocity and temperature once they are defined. For convenience, each set is referred to as the k^{th} set and the values represents the interval of time and will be averaged.

3.3.1 Density

For any k^{th} set, the number of particles in the l^{th} cell is obtained by dividing the sum of time occupancy by the time of observation for the set. All particles that have been located within the cell during the time of observation contribute to the time occupancy for the cell. In term of the variables collected, the number of particles in the l^{th} cell for the k^{th} set is then;

$$n_{l_k} = \frac{\left(\sum_i \delta\tau_{i,l}\right)_{\tau_k} - \left(\sum_i \delta\tau_{i,l}\right)_{\tau_{k-1}}}{\tau_k - \tau_{k-1}}$$

If particles are uniformly distributed within the computational region before the piston starts to move, then we can define a nominal number of particles, n_0, in a data cell. This is the initial number of

particles per cell; this number density remains almost constant in the upstream region during the simulation until the shock wave moves into it. The nominal number of particles is known from the total number of particles, N, and the number of data collecting cells, n_{cell}.

$$n_0 = \frac{N}{n_{cell}}$$

The density ratio, $\bar{n}_{l_k}$, is then obtained as the ratio of the local number of particles, n_{l_k}, to n_o. $\bar{n}_{l_k}$ is placed at x_l and represented at the time t_m where $t_m = (\tau_k + \tau_{k-1})/\ 2$ and $m = k$. $\bar{n}_{l_k}$ are used to plot a contour plot of equal density ratio with horizontal axis representing the distance from the origin in units of mean free path and the vertical axis representing the time. In order to convert the discrete data into continuous lines representing the equal density, following procedure is used.

For every point on the grid, the procedure searches for the nearby raw data points and then estimates the value of the density ratio for that point on the grid. Our search type looks for 2 nearest neighbors in each of 8 directions, i.e., 0° - 45°, 45° - 90°, 90° - 135° etc. The weighting function determines how all of the points found by the search is weighted. The weighting function used is scaled

$1/(H/H_{max})^2$, where H is the distance between two points. It assigns a weight equal to the square of the inverse of the proportion of the distance of this neighbor to the furthest neighbor and gives more weight to close neighbors than to distant neighbors. The result is that the farthest neighbor receives a weight of 0.[59]

As stated, each cell is equivalent to a sensor or a measuring station in a physical experiment. A time history of a station then should produce the shock profile for our simulation. Being a discrete system, the result obtained from a single station shows a great deal of scatter. The speed of the shock wave can be computed by observing how far the shock wave has moved from a set to another set. This speed is then used to superimpose all sets into one set in order to increase the number of data points for averaging, a simple Galilean transformation. Since the speed of the shock wave from the simulation is found to be in an excellent agreement with the theoretical shock speed, the theoretical speed is used for the transformation. Again, considering the density ratio, $\bar{n}_{l_m}$, at the l^{th} cell at time, τ_m, the transformation with a known wave speed, u_w, is as follows;

$$(x_l,\ \bar{n}_{l_m}) \rightarrow (x_l + u_w \cdot \tau_m,\ \bar{n}_{l_m})$$

The data collecting scheme in which the coordinate system is moving at the same speed as the shock wave is achieved by shifting the results obtained from a fixed coordinate system by the wave speed. Once shifted, the x coordinate no longer has a clear interpretation. For the purpose of comparison with other results, the origin is declared at the location where the density ratio is exactly half way between the upstream and downstream values. Since each cell collects not only the density information but also the velocity and temperature information at the same time, the x coordinate system is defined from the density profile once and for all and the same x coordinate system is used to plot the velocity and temperature profiles as well.

Once shifted, data points form a narrow band showing the shock profile. To get a value at a location, resulting data points are regrouped such that points within a mean free path distance forms a group and each data points belongs to only one group. These group of data points are then averaged and their value is noted at the midpoint.

This process of averaging can be summarized as, first, finding time averaged information for a small time interval for each cell.

Next, when superposition of all data points are performed with the shifted location, each set in the specified width is treated as a snap shot and the ensemble average is performed for a given cell width.

DSMC does exactly the same averaging but in the reverse order. Ensemble averaging is performed first during the simulation, i.e., during a given time interval and a specific cell, many snap shots of the cell are taken and arithmetic average is obtained. Results are then used as the representative value of the cell for the time interval and long time average is obtained finally to produce a profile.

3.3.2 Velocity and temperature

For any k^{th} set, the mean velocity is defined as the sum of velocity divided by the number of particles observed in a region for an interval of time. Therefore the mean velocity for the l^{th} cell over the k^{th} time interval can be written as Equation (3) where summation over i is the number of particles observed in the region.

$$\bar{u}_k = \frac{\left(\sum_i u_{i,l}\right)_{\tau_k} - \left(\sum_i u_{i,l}\right)_{\tau_{k-1}}}{\left(\sum_i i_l\right)_{\tau_k} - \left(\sum_i i_l\right)_{\tau_{k-1}}} \qquad (3)$$

The same procedure as described in the extraction of the profile of the density ratio is used to shift data points and to obtain a profile of velocity ratio. Note that the origin of the x coordinate for velocity is at where the density is midway between the upstream and downstream density.

The velocity of any particle in the l^{th} cell can now be written in the form of the mean velocity plus the deviation from the mean which is the thermal component of the velocity:

$$u_l = \bar{u}_l + \varepsilon_i$$

Since the thermal energy, E, is directly related to the thermal component of the velocity, we can write the thermal energy for the l^{th} cell during an interval as;

$$E_l \propto \frac{\sum \varepsilon_i^2}{\sum i} \tag{4}$$

Since

$$\begin{aligned} \varepsilon_i^2 &= (u_l - \bar{u})^2 \\ &= u_l^2 - 2 \cdot \bar{u} \cdot u_l + \bar{u}^2 \end{aligned}$$

we can expand the right hand side of Equation (4) to finally obtain an expression for the thermal energy in the term of the variables collected;

$$\bar{E}_k \propto \frac{\left(\sum_I (u^2)_{I,I}\right)_{\tau_k} - \left(\sum_I (u^2)_{I,I}\right)_{\tau_k}}{\left(\sum_I i_I\right)_{\tau_k} - \left(\sum_I i_I\right)_{\tau_{k-1}}} - \bar{u}_k^{\,2}$$

The constant of proportionality disappears when the ratio of the local thermal energy to the thermal energy of the undisturbed upstream region is computed. The ratio obtained is equal to the temperature ratio. The profiles of the temperature ratio are then obtained following the same procedure outlined in obtaining the density ratio.

3.4 Computer and computational error

3.4.1 Computer used and execution speed

The CRAY YMP at NASA Lewis was used during the code development and test runs of the generation of the stationary shock wave. The CRAY YMP C90 at NASA Ames was used for additional code development of the stationary shock wave and the moving shock wave. Final production runs were all done on the CRAY YMP C90. The execution speed has reached 160 Mflops for the final production run without any special effort to vectorize the source code. A typical

run of 4,000 particles with 100,000 interactions takes 4 CPU Hours to complete the dynamics portion of the simulation on the CRAY YMP C90. About 400 CPU Seconds is then needed to analyze the output file to produce results.

3.4.2 Error and Machine accuracy

The molecular dynamics technique is not a numerical analysis of a partial differential equation, therefore the computational error discussion is somewhat different from the traditional one.

The first type of error is due to the Machine accuracy. The CRAY YMP carries out computations to the 16th decimal place in single precision mode.

Since the particles are followed at all times by the coordinates of their center locations, the error in positioning of particles is the first concern. The position of particles is continuously updated throughout the simulation by moving each particle from its initial positions to the next position obtained from its velocities and the shortest time to collide. This repeated process will gradually add errors in the position of particles. Since the algorithm uses geometrical information to find the exact collision time, computer

accuracy can be a nightmare. Suppose a particle center is and should be at the collision position with a boundary but due to the numerical accuracy, the particle center may get placed just outside the boundary, one may experience a floating point error. Extreme care must be taken since this type of error is extremely difficult to find and the existence of such error gets apparent only after extensive testing.

One way of circumventing the problem is to use an error margin such that the absolute value of a difference is compared with the specified error margin rather than with zero. In our simulation, the error margin is set at 10^{-15} when geometrical information has to be processed to determine a collision.

As another remedy, when a particle collides with a boundary, the position of a particle is then reset using the known position of the computational boundary. This has an effect of removing the buildup of error due to the Machine accuracy. In doing so however, the system is no longer a reversible system in a strict sense. In order to compensate for a possible blunder resulting from the re-positioning of particles, particles positions are analyzed from time to time to see if any particle has left the computational region.

The width of the data collecting cells determines the resolution of results presented. The width of data collecting cells is set at 1 upstream mean free path for all simulations, and the resolution of results is the same order. After the Galilean transformation is perform to form a single profile, data points are simply averaged in a band of 1 upstream mean free path width and no special weighting function is used to reflect the position or magnitude of each data points within a band because of the cell width. The cell width for DSMC is also 1 upstream mean free path, therefore the resolution of the results obtained are equivalent for both methods.

There are some inherent errors in data collection. Since MD is a collision marching scheme, not a time marching scheme, the time interval has a built in error of order of the shortest time to collide. But the shortest time to collide, which is a measure of a collision frequency, decreases with the decreasing distance between the piston and the end wall, increasing the collision frequency. Therefore the error also decreases as the piston near the end wall. Currently observed error in the interval time is in the fifth decimal place. The actual time interval thus is not absolutely constant, but an averaged value for an interval is computed based on the actual interval of time.

3.5 Some finer points on computational strategy

3.5.1 Separation of dynamics and data collection

Hard sphere model simulation is not a time-marching scheme but a naturally collision-marching scheme because the scheme searches for the shortest time to collide. The time is not a constant value. A simulation ends after some specified number of collisions. A simulation starts from a file which has the location of particles and their velocities. When the simulation is over, the position and velocity of each particle are recorded so that simulation can be continued if desired. After each interaction, quantities that have changed and cannot be deduced are written to an output file, which is used to extract the spatial information once the simulation is over. Though the size of the output file may become very large, the most time-consuming portion is the simulation itself, not the data extraction. One could do the simulation and collect the spatial information simultaneously, but this method puts so much overhead on the computer (CRAY YMP C90) that the identical simulation that

included the data extraction took four times as long as doing them separately.

3.5.2 Use of subroutine and function statement

Use of user-defined subroutines and function calls is strongly discouraged on CRAY YMP. The execution speed went from 90 Mflops to 160 Mflops when subroutines and functions calls were avoided. This improvement in the execution speed has almost doubled the number of collisions that can be computed in the same amount of CPU time.

If the program is too complicated to avoid the use of subroutines and functions, one may opt to use the in-lining option of the CRAY FORTRAN compiler. Each subroutine and function is then written to the main program by the optimizing compiler as if there were no subroutines and functions calls.

3.5.3 Non-repetition of computations

The most time consuming feature of our computational process is the computation of new time-to-collide after each collision. For a

system of N particles, this requires the solution of N(N-1)/2 quadratic equations for the particle to particle collisions, 2N linear equations for the particle to 2 end walls collisions, and N quadratic equations for the particle to the lateral boundary collisions.

However we do not need to compute the time to collide for the pairs of particles that have not participated in the current collision. This reduces the time to collide computations to 2(N-2) quadratic equations and 4 linear equation plus 2 quadratic equations. Clearly only half this number of computations is needed if the collision is with a boundary, because only one particle is involved in the collision.

4. Results and discussion

4.1 Shock development

The life cycle of a shock wave can be seen most easily in a contour plot of equi-density lines (or equi-velocity lines, or equi-temperature lines) in a time-distance graph as done in Figure 7. The life cycle of a piston driven shock wave involves the formation, the Galilean transformation of its profiles, and the shock reflection. The horizontal axis corresponds to the distance from the origin measured in units of the upstream mean free path and the distance scale is identical in all three graphs. The vertical axis represents the time in units of the upstream mean free path divided by the most probable upstream speed. The vertical scale, however, is different for each of three graphs in order not to sacrifice detail. Although the intermediate points are interpolated between data points as described previously, this plot is useful in grasping and explaining the process of shock development. The contour plot does not involve shifting of profiles in time in order to get a composite profile or having to re-define the origin of the x axis.

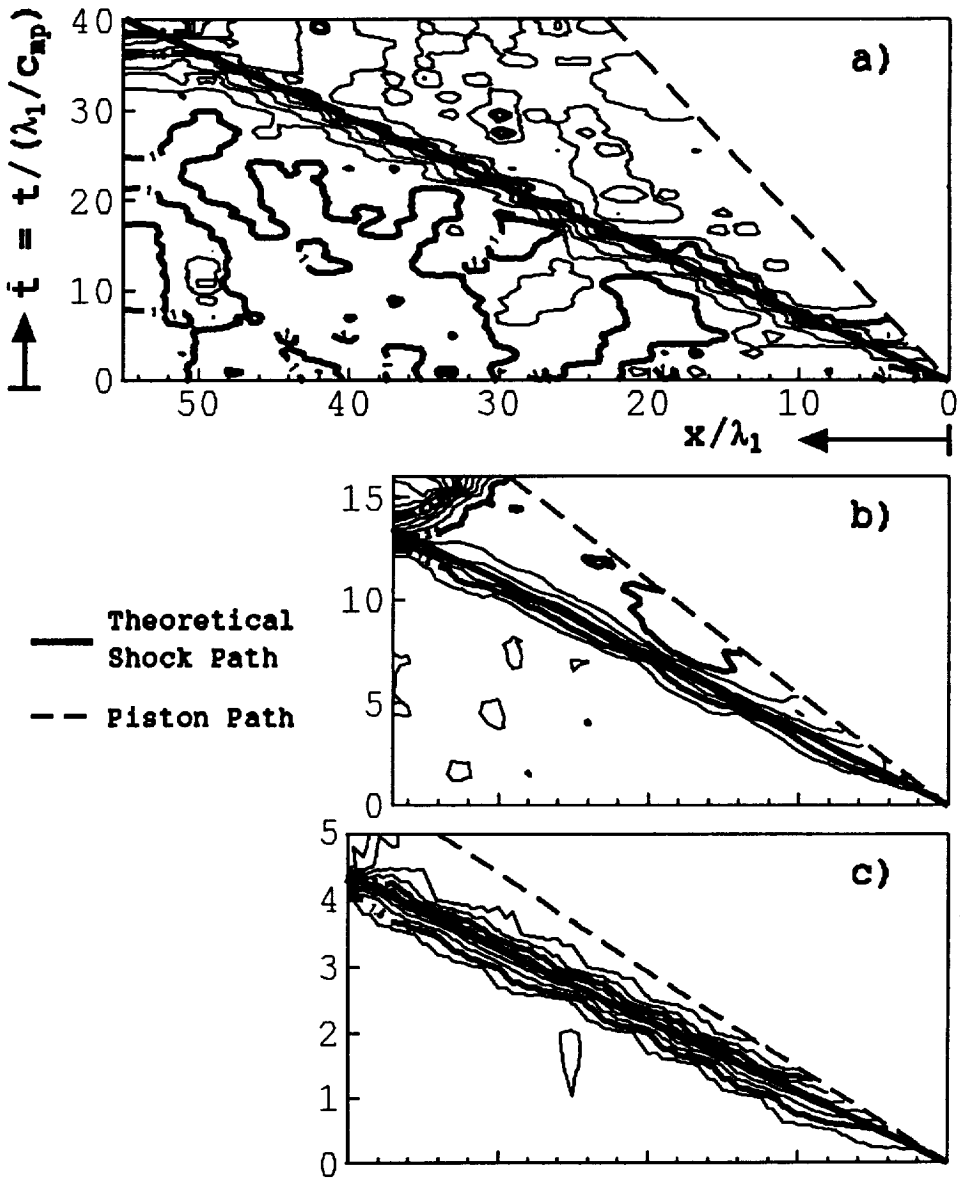

Figure 7 Equi-density contour plot;
a)M=1.5, N=3,000, L=54,
b)M=3, N=3,000, L=37,
c)M=10, N=4,000, L=41

A sketch describing the coordinate system with remarks is shown in Figure 8.

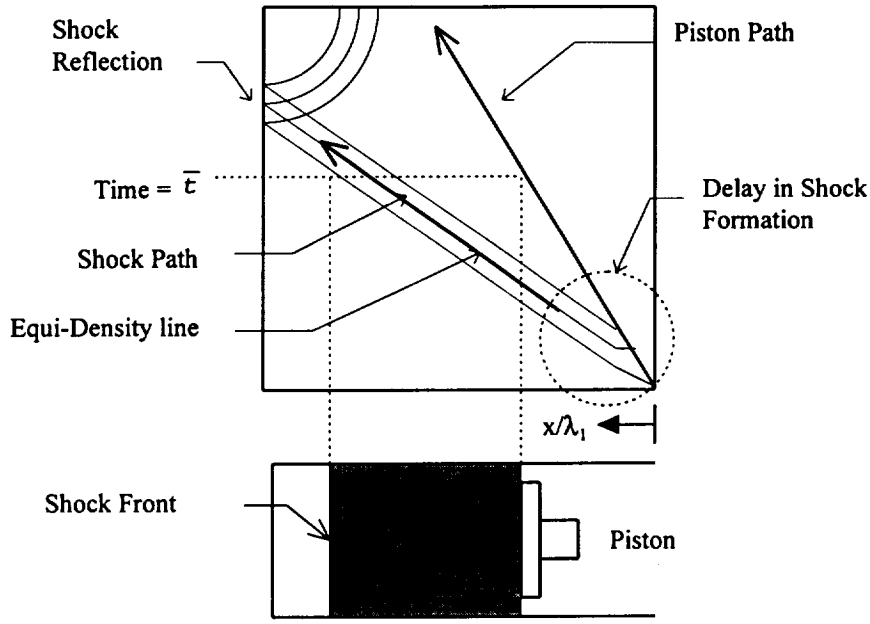

Figure 8 Results from equi-density contour plot

Examining Figure 7, there seem to be delays in the formation of shock in all Mach numbers simulated as illustrated in Figure 8. The shock profiles builds up with time, the downstream density gradually reaches an asymptotic value. It appears that it takes longer to generate a fully developed shock profile at low Mach numbers.

Another interesting observation is the shock reflection as shown in Figure 7 b). The shock reflection is shown clearly. The shock reflection is seen momentarily because the distance between the end wall and the piston was not large when the reflection did occur. For more thorough study of the event, a longer computational

region is needed in order to give more time to sustain the reflected shock.

The slope of the shock path gives the wave speed. The theoretical wave speed is drawn directly on Figure 7 so as to make direct comparisons. Since fairly straight paths, which are parallel to the theoretical shock path, are formed in each cases, the results of simulation not only show that the speed of shock wave remains constant but also show that it agrees with the theoretical speeds for all cases. Another measure of the constant wave speed is how parallel the equi-density lines are to each other. Figure 7 a), b), and c) show that equi-density lines are approximately parallel to each other, especially in the high Mach number simulations.

Since the equi-density lines are results of interpolation and the resulting graph can be different for different techniques of interpolation, the contour graph is not an exact representation of more complete data set. Nevertheless either comparing the slope (thus the velocity) with the theoretical speed or looking at the equi-density lines and examining how parallel they are, gives a good idea on the speed and the constancy of the speed. Another measure of the wave speed and the constancy of it can be made by looking at the

wave speed and the constancy of it can be made by looking at the scatter of actual data points. Since the profiles of the density ratio obtained at different time during the simulation can be shifted as described earlier, to be superimposed, assuming that the shock is moving at the theoretical wave speed, all profiles should collapse into one forming a narrow band. If the assumption either on the speed of the shock wave or the constancy of the shock speed is not correct, then the results will show that a narrow band does not form. The composite profile forming a narrow band is shown as Figure 9. This plot can be thought of as one looking the equi-density plot standing at the origin in the direction of the shock path. This plot emphasizes the constant shock speed.

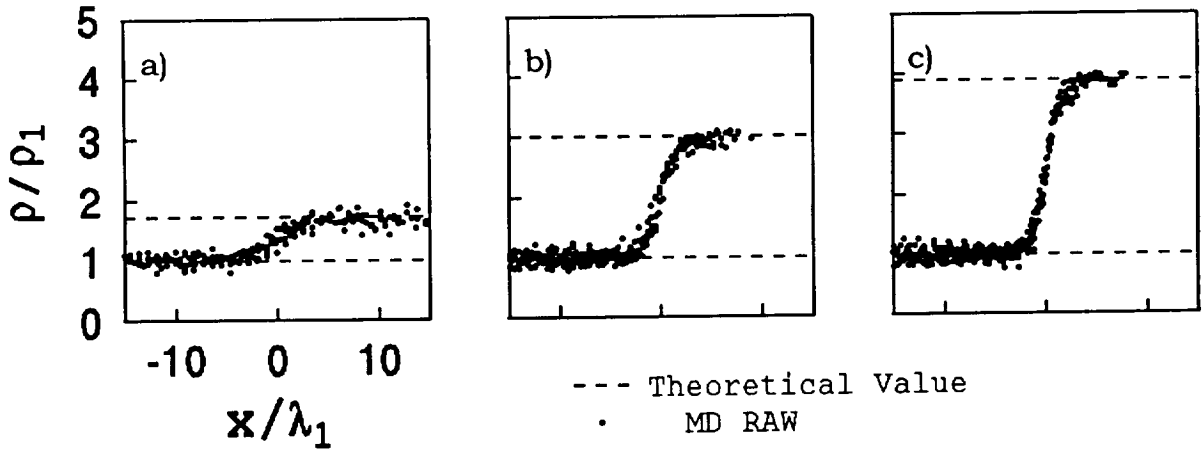

Figure 9 Wave speed and data scatter in density profiles of MD; a)M=1.5, b)M=3, c)M=10

In each case, data points form a narrow band showing that the speed of the shock wave is constant. The amount of data scatter increases as the Mach number decreases. This trend is most evident when the scatter in the transitional region is compared between Mach 10 and Mach 1.5. The large scatter is at the lower Mach numbers because the mean velocity and the most probable speed become closer to each other as Mach number decreases and also because the lateral velocity is relatively larger.

4.2 Profiles of properties

4.2.1 Density and velocity profiles

To show how the density changes across the shock, profiles of density ratio are plotted. The density ratio is the ratio of the local density to the upstream density far upstream of the shock. Figure 10 shows the absolute density ratio profile for each Mach number. Both MD and DSMC are plotted using different markers to show that both methods give discrete results, not continuous lines. As shown the results of MD and DSMC agree closely.

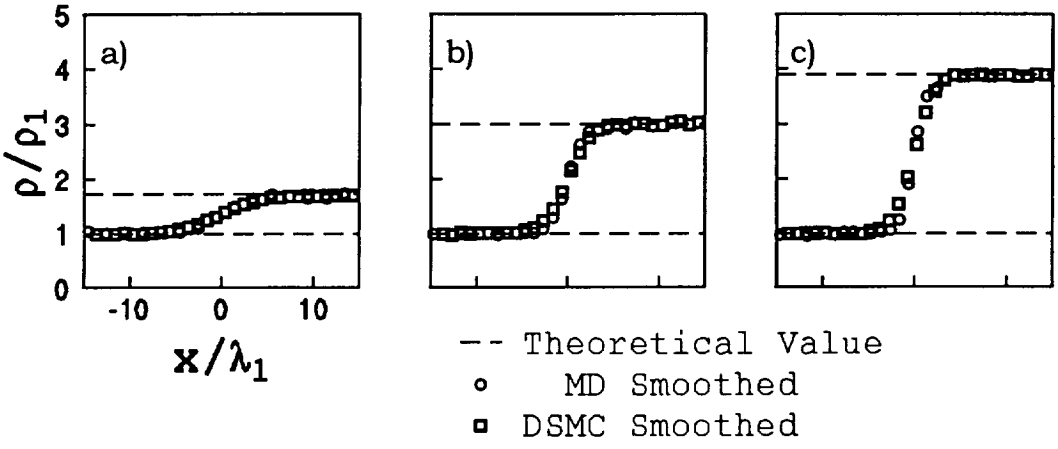

Figure 10 Density profiles; a)M=1.5, b)M=3, c)M=10

The origin for all profiles is determined from the density profile as described earlier. Since the origin of the coordinate system does not exist any more due to the shifting of the coordinate system in order to superimpose set of profiles, a new origin is located where the value of the density ratio is half way between the upstream and downstream values.

There are no surprises in profiles of the streamwise velocity. For the sake of completeness, velocity profiles are presented as Figure 11. The DSMC results are presented as curves from now on to show MD results clearer, rather than as marker as done in the profile of density ratio. The local velocities are obtained in the laboratory

frame of reference which is then transformed to one which the observer is riding on the wave front. The streamwise component of velocity is plotted as the ratio of the local streamwise velocity to the shock wave velocity. The velocity profiles are not forced in any way and obtained directly from the simulation, not from density profiles.

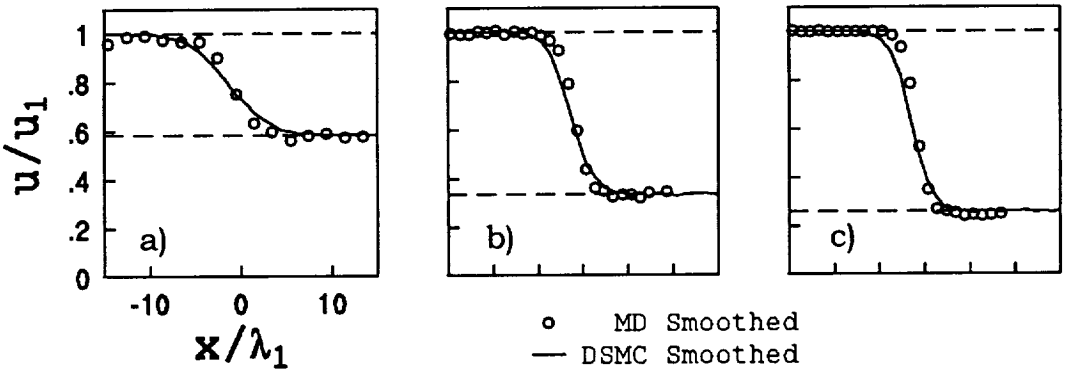

Figure 11 Streamwise velocity profiles; a)M=1.5, b)M=3, c)M=10

Since the mass flux, $\dot{m}$, must be conserved for our shock problem, we can multiply the absolute density ratio by absolute velocity ratio and should obtain unity, 1, everywhere within the computational region. The degree of preservation of the mass flux also tests the accuracy of the method used to compute the average velocity. Following table shows the statistical description of the mass flux obtained.

Case	Mach number	L/λ_1	N	n/λ_1^3	$\dot{m}$	Standard deviation
a	1.5	54.	3,000	24.	0.99	0.031
b	1.5	31.	500	24.	0.96	0.066
c	3	49.	4,000	75.	1.00	0.035
d	3	19.	4,000	75.	0.99	0.036
e	10	41.	4,000	125.	0.99	0.034

Table 3 Conservation of mass flux

The mean mass flux is closer to 1 and the standard deviation is less when many simulation particles and a longer computational region are used. The effect of having a larger number of simulation particles is stronger than having a longer computational region, as the cases **d** and **b** shows. The case **d** has the shortest computational region thus about 50 data points only in overall, yet the computed mass flux and standard deviation is as good as in other cases.

4.2.2 Temperature profiles

As discussed, it has been shown by DSMC and in a number of investigations,[32, 60] based on Mott-Smith's work, that the component of temperature in the direction of the streamwise velocity has a peak,

of temperature in the direction of the streamwise velocity has a peak, a maximum value higher than the downstream value, somewhere within the profile. Experimentally,[26] the existence of a peak has also been shown. Results from simulation also show the existence of the peak in the axial temperature profile as shown in Figure .

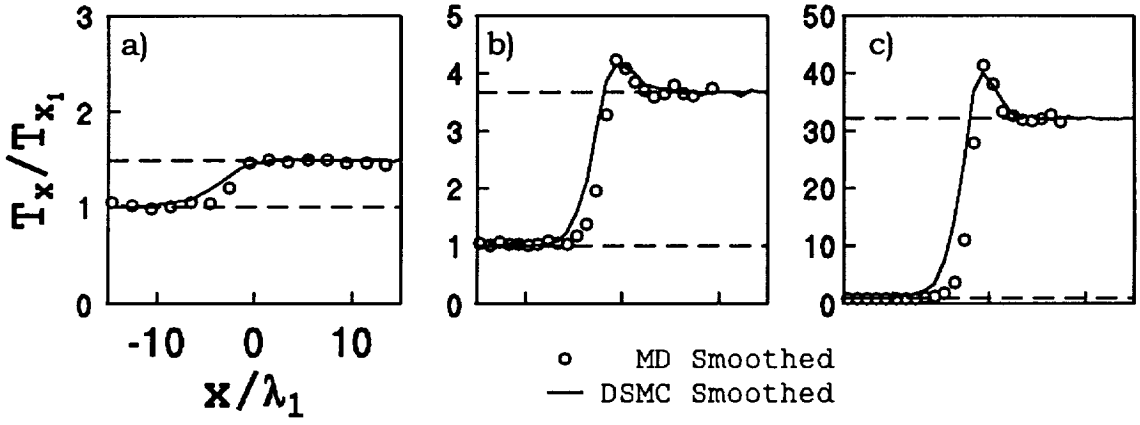

Figure 12 Axial temperature profiles and their peaks;
a)M=1.5, b)M=3, c)M=10

Numerical values of the axial temperature peaks, measured in units of the upstream temperature, are displayed in Table 4. All three methods give values close to each other; while the molecular dynamics results are somewhat closer to Yen's theoretical results, than to DSMC.

M_1	MD	DSMC	Yen (Mott-Smith)
1.5	1.51	1.50	1.51
3	4.24	4.13	4.27
10	41.40	40.00	42.2

Table 4 Peak values in axial temperature

The slope of the transition region is sharper therefore the change is more sudden in the MD than the DSMC results. The peaks occur at the same locations where DSMC has the peaks, even though the beginning of the transition is downstream of DSMC. The horizontal coordinate is the same as the one obtained from the density profile and no shifting is performed.

The two lateral components of temperature show no peaking and are virtually identical to each other. For the sake of completeness, their profiles are shown in Figure 13.

The slopes of the transition region are steeper for MD therefore the change is more sudden than DSMC. The beginning of the transition region for all lateral temperature profiles is downstream of DSMC as shown in Figure 13. This tendency is especially visible for Mach 3 and 10.

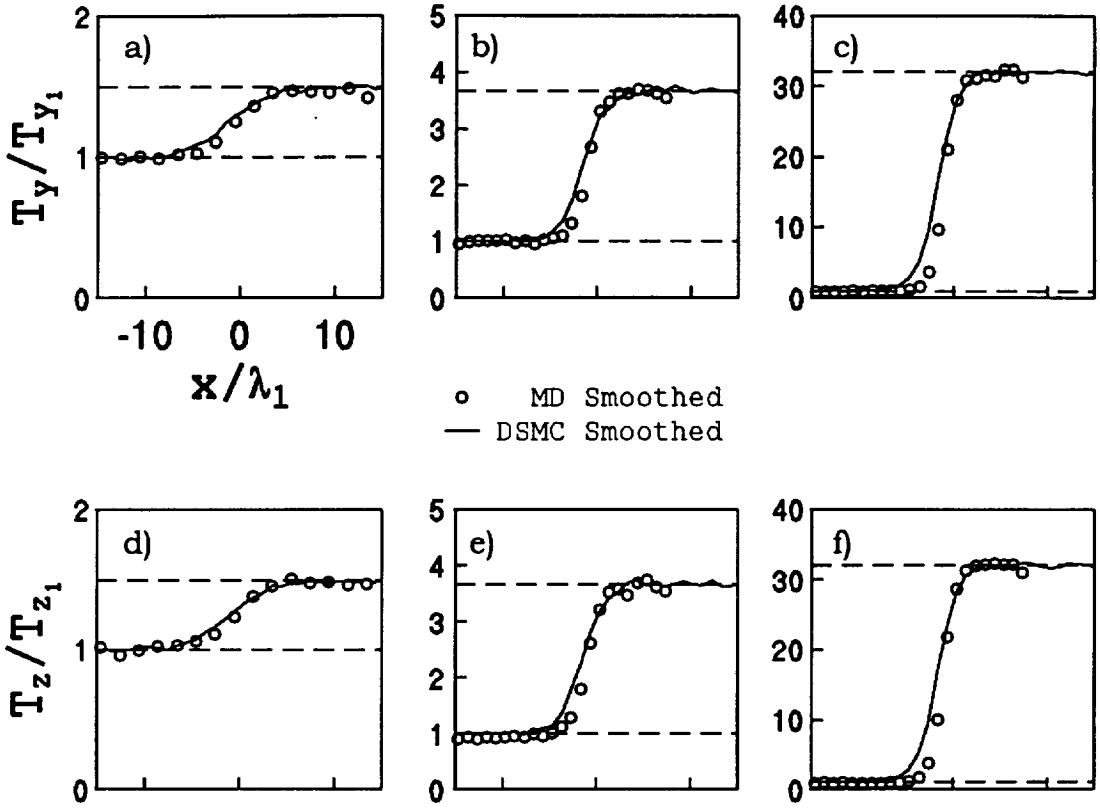

Figure 13 Lateral Components of Temperature;
y Temperature - a)M=1.5, b)M=3, c)M=10;
z Temperature - d)M=1.5, e)M=3, f)M=10

Summation of all three components of the temperature gives the overall temperature. They are shown in Figure 14. The overall profile confirms the overall trend that the slopes are sharper in MD profiles than DSMC profiles.

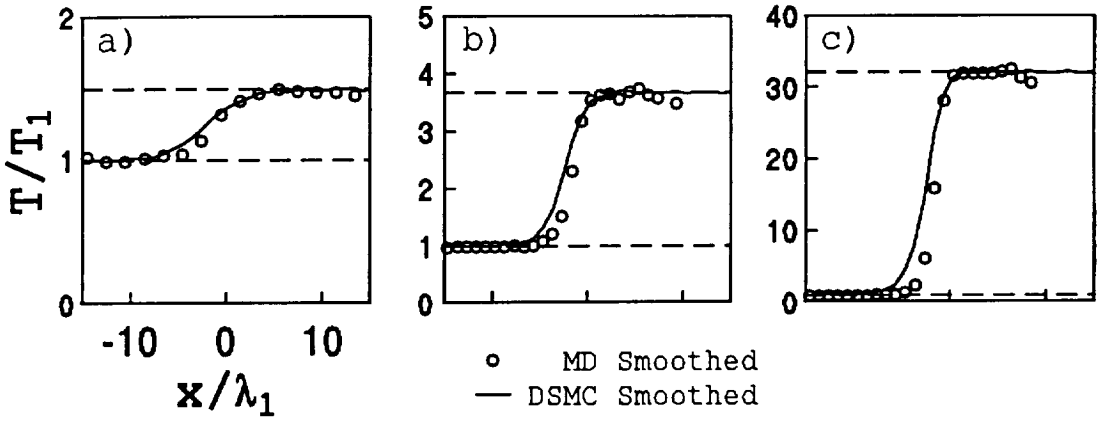

Figure 14 Overall temperatures; a)M=1.5, b)M=3, c)M=10

Shock thickness

The shock thickness is obtained most easily by plotting the reduced density versus the reduced distance as sketched in Figure 15. In this graph, the inverse of the maximum slope is equal to the shock thickness. Since this normalized density ratio goes from 0 to 1 for all Mach numbers, this form of plot is especially useful to observe the shock thickness variation with Mach number.

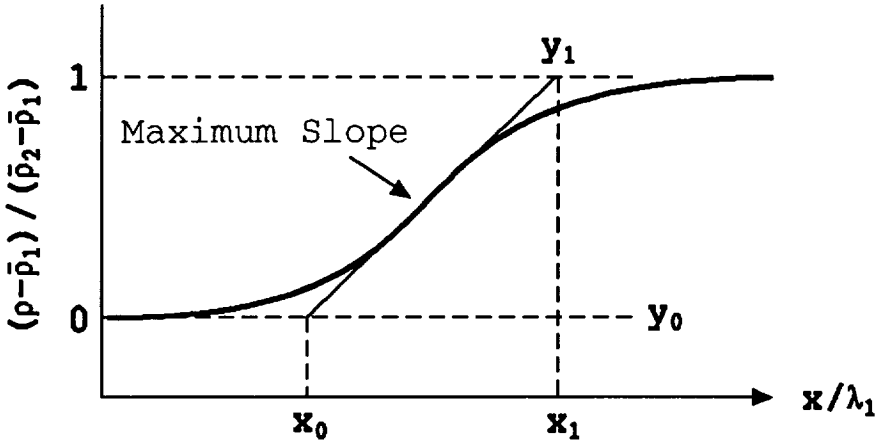

Figure 15 Normalized density ratio

The reduced density is a ratio between the difference of the local density to the upstream density and the maximum density difference. Similar definition can be made for velocity and temperature as well.

$$\text{reduced density} = (\rho - \bar{\rho}_1) / (\bar{\rho}_2 - \bar{\rho}_1)$$

Shock thickness obtained from the density profiles are compared with those of DSMC, Navier-Stokes and Mott-Smith in Figure and numerical values are given the Table 5.

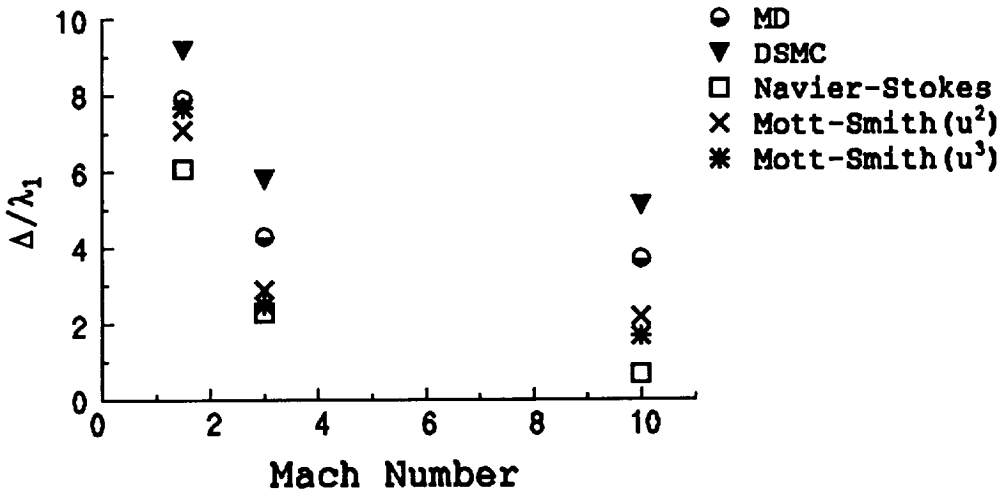

Figure 16 Shock thickness comparison

Mach number	MD	DSMC	N-S	M-S(u^2)	M-S(u^3)
1.5	7.9	9.2	6.1	7.1	7.7
3	4.3	5.8	2.3	2.9	2.5
10	3.7	5.1	> 1.	2.2	1.7

Table 5 Shock thickness comparison

Using the Mott-Smith's bimodal model, Muckenfuss showed that the shock thickness is not a function of Mach number for strong shocks for the hard sphere molecules.[12] The shock thickness reduces greatly for all methods in Table 5 when the Mach number changes from 1.5 to 3. The shock thickness changes little when the Mach number changes from 3 to 10, compared with changes seen for

Mach number going to 3 from 1.5. The results of MD simulation also shows that the shock thickness is not a strong function of Mach number for strong shocks for the hard sphere molecules. The numbers given for the shock thickness from the maximum slope definition can be misleading. Thickness differences found by the maximum slope definition give an exaggerated idea of how much the shock thickness changes with Mach number for strong shocks. One sees in Figure 17 b) that the qualitative characteristic thickness is not much different at Mach number 3 and 10.

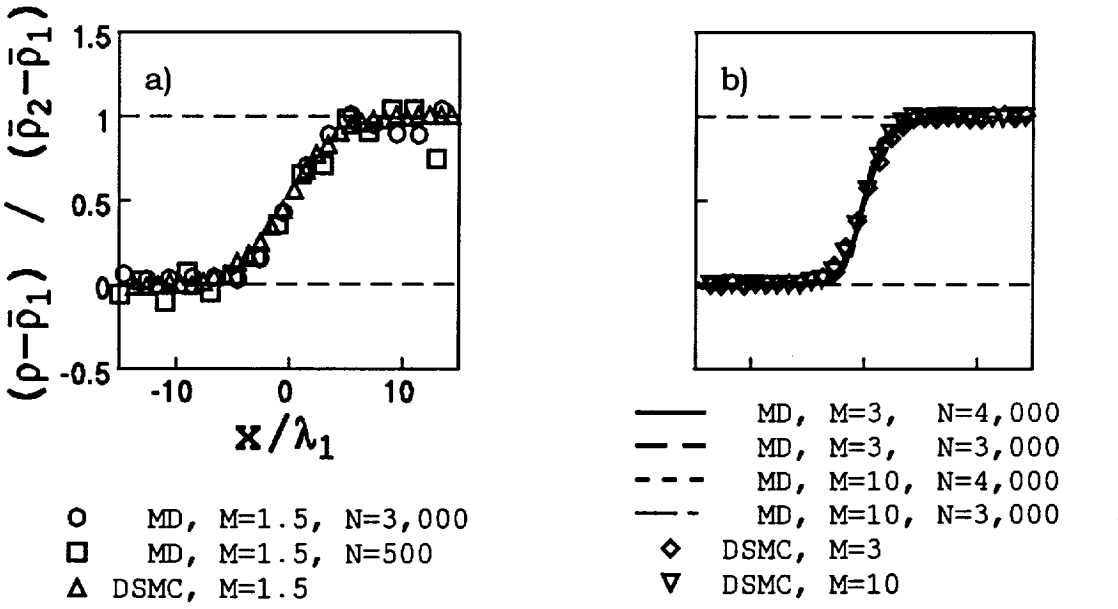

Figure 17 Shock thickness comparison between MD and DSMC in reduced scale graphs; a)M=1.5, b)M=3 & 10

The difference in the shock thickness, as obtained from the maximum slope definition, between DSMC and MD is consistently about 1.5 times the upstream mean free path for all three cases. Mott-Smith's results using u^3 moments at Mach 1.5 is closer to the MD result. Incidentally, MD results for Mach 3 and 10 are larger by about 2 mean free paths than the corresponding Mott-Smith's results using u^3 moments. It is interesting to note that all of the kinetic theory results predict broader shocks than those deduced from the Navier-Stokes equation.

The effect of the computational Knudsen number on the shock profiles and thickness are further examined by changing the computational Knudsen number for the Mach 1.5 and 3 simulation. As stated, the lateral dimension should become increasingly important as Mach numbers decrease, since the lateral velocity is relative larger at low Mach numbers. If the lateral dimension, the diameter of the computational region in our case, does affect the slope of the shock profile, the effect should be most visible at low Mach numbers. For this reason, computational Kn is tested for Mach 1.5 and 3. The following table shows computational conditions.

Mach number	Kn_1 (= λ_1/D)	N	n/λ_1^3	L/λ_1	d/λ_1
1.5	0.56	3,000	24.	54.	.096
1.5, Test run	0.98	500	24.	31.	.096
3	0.81	4,000	74.	49.	.055
3, Test run	0.52	4,000	74.	19.	.055

Table 6 Parameters used in test runs

The total number of particles needed for Mach 3 test run is estimated to be over 15,000 particles which is not a practical run for currently available computing resources. Therefore a rather short computational region is used for Mach 3 to address the problem. It appears that the shock wave profile is in a translation motion not only when it is fully developed but also during the formation. This phenomena is observed from Figure 7, 9, and 19. Figure 7 shows that the equi-density lines are straight and parallel to each other supporting the observation. Figure 9 show that data points form a narrow band, also supporting the observation. Figure 19 show that the profiles in the transitional region remains the same, regardless of the length of the computational region. Since the shock thickness is determined using the maximum slope, the results from a short

computational region and from a long computational region should show that the slope in the transitional region remain the same regardless of whether a full profile of shock wave has been obtained or not. Results of test runs are shown in Figure 18.

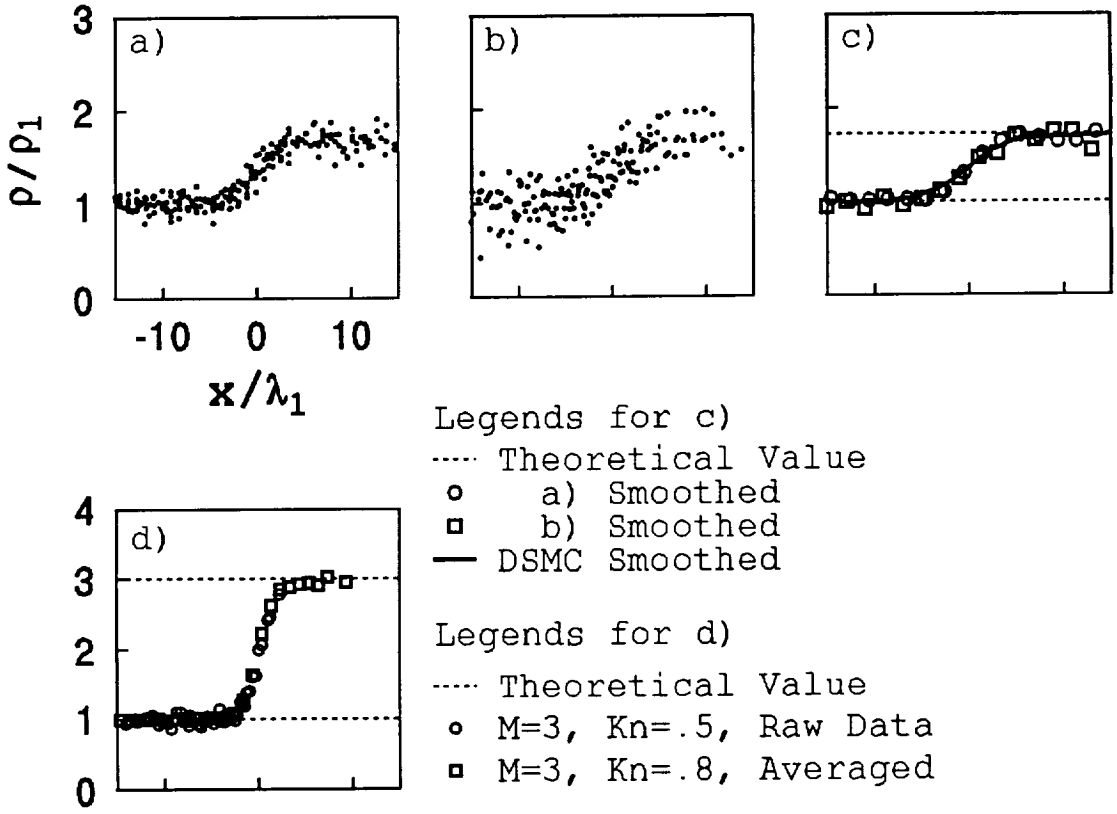

Figure 18 Effect Kn on shock thickness; M=1.5 - a)Kn=0.5, b)Kn=1, c)comparisons between averaged results of a) and b); M=3 - d)comparison between the averaged data of Kn=0.8 and the raw data of Kn=0.5

A visual inspection shows that the scatter has increased with increasing *Kn* at Mach 1.5, probably due to the small number of particles used. But when the same averaging technique is applied to both cases as shown in Figure 18 c), results are close to each other. The shock profile also remained the same at Mach 3 even when the computational region length was such that the full profile of a shock wave was not obtained before the shock reflection. Therefore the shock thickness is not a strong function of *Kn* for the tested range of *Kn* and Mach numbers.

A longer computational region is used for Mach 3 and 10 because of a negative slope in the profiles of the density and the temperature in the neighborhood of the piston. The longer computational region gives extra time for the high density area to build up. Therefore the longer length is used to get sufficiently uniform behavior in the high density area.

As Figure 19 shows, profiles shows uniform behavior showing the correct values at the high density area. The shock thickness is not affected by the change in the computational region length.

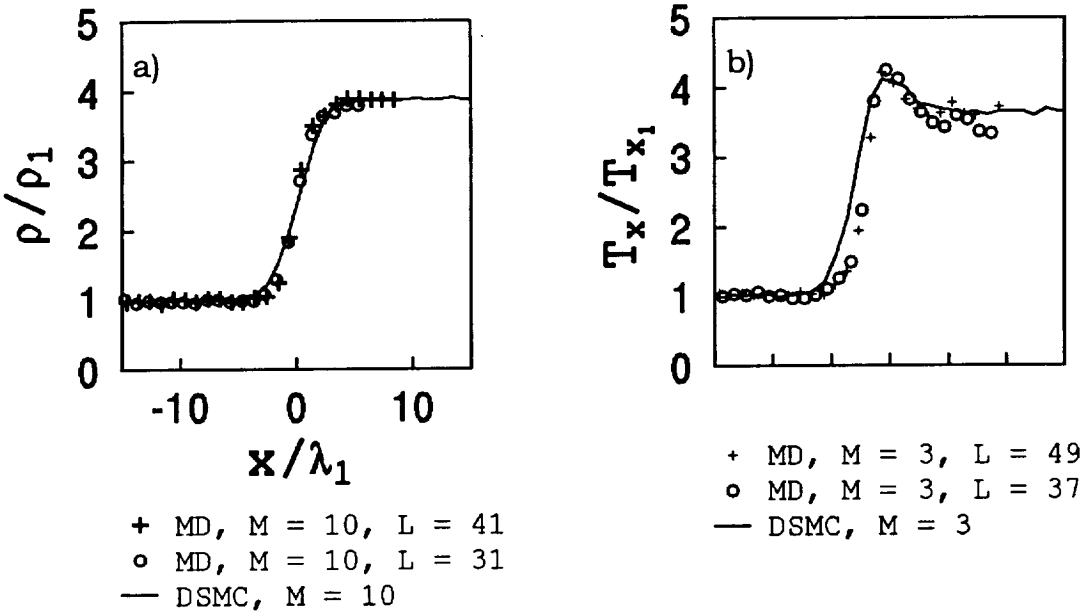

Figure 19 Improvement of asymptotic behaviors at high density regions when longer computational region is used; a)density improvement for M=10, b)axial temperature Improvement for M=3.

5. Conclusion

The comparisons of MD profiles at three Mach numbers with DSMC profiles at the same Mach numbers show that the profiles of the shocks are sharper. Though measured differences from the maximum slope definition are of the order of 1 mean free path for all three Mach numbers, one expects that MD is more likely to produce a dispersed shock than DSMC because the particles are free to travel within the whole computational region. In DSMC, randomly selected particles in a sub-region go through a chosen number of collisions with a randomly selected partner within that sub-region and are not allowed to leave the sub-region until the chosen number of collision is completed. Then the particles are released from the region and allowed to travel according to their individual velocity such that particles may cross the sub-region boundaries. A disturbance is likely to travel faster in MD technique, yet MD produced sharper profiles. Bird showed that the shock thickness obtained using DSMC is sharpest for the hard sphere molecule.[33] MD produced sharper shocks at a given Mach number than DSMC of any intermolecular force law model.

The effect of *Kn* is examined for possible explanations of the sharper profiles. The *Kn* was increased to 1.0 from 0.5 at Mach 1.5 and decreased to 0.5 from 0.8 at Mach 3. At Mach 1.5, increasing *Kn* resulted decreasing the total number of particles. Fewer particles gave more scatter but the averaged profiles showed changes neither in the sharpness of the shock nor the downstream values of density, velocity and temperature. At Mach 3, decreasing *Kn* resulted increasing the total number of particles to an extent that a definitive run could take too long to compute for currently available computing resources. But a test run at Mach 3 with a shorter overall length of the computational region also showed that the sharpness of the profiles did not change.

When the profiles at the high density region showed a negative slope near the piston, the length of the computational regions increased. The results of the longer computational region showed the correct asymptotic behavior and the drops near the piston disappeared in the high density area. Since the rest of the shock profiles remains the same, the length of the computational region appears to influence the high density regions, not the shock thickness.

Density profiles, velocity profiles and temperature profiles are in good agreement with DSMC profiles albeit differences in the thickness of shocks. Asymptotic values obtained for the density ratio, the velocity ratio and the temperature ratio for the downstream region were virtually identical to the Rankine-Hugoniot. The axial temperature is observed to have a peak before it settles to the downstream value. The values of the peaks are somewhat closer to Yen's theoretical values than to the results of DSMC are to Yen's, for strong shocks.

It should be re-emphasized that the asymptotic values in the high density area are results of our simulation, and are not forced in anyway. There is no built in statistics in the simulation since all boundary conditions are specular.

To conclude, the piston driven shock tube simulations by MD have proven that MD is capable of simulating dilute gases in three different cases of the Mach numbers ranging from a weak shock to a strong shock; especially the piston driven shock tube simulations using MD have been able to generate and verify shock profiles. Considering the difference in simulation techniques, the degree of

agreement in the resulting profiles between MD and DSMC is visually striking for all three Mach numbers.

Since the hard sphere model for a monatomic gas is examined in detail, the natural order of events is to study the shock structure using force law type monatomic molecules. Currently, gas-surface interaction is being examined by colleagues for the force law type monatomic molecules. The dynamic portion of the simulation for the gas-surface interaction can be substituted for the dynamics of the hard sphere molecules, decreasing source code development time. Since the shock thickness is dependent even for the strong shocks on the Mach number if a force law other than the hard sphere model is used, comparison with published results of other methods should be interesting. A group at CWRU is currently examining molecular dynamics computations of a diatomic gas using a dumbbell mechanical model of molecules. Since air consists mostly of diatomic gases, a study of shock profiles using the mechanical dumbbell model of diatomic molecule is another interesting problem.

6. Literature cited

[1] W. J. M. Rankine, "On the thermodynamic theory of waves of finite longitudinal disturbance," Trans. Roy. Soc. (London) 160, pg. 277 (1870)

[2] Lord J. Rayleigh, "Aerial plane waves of finite amplitude," Proc. Roy. Soc. (London) A84, pg.247 (1910)

[3] G. I. Taylor, "The conditions necessary for discontinuous motion in gases," Proc. Roy. Soc. (London) A84, pg. 371 (1910)

[4] L. H. Thomas, "Note on the Becker's theory of the shock front," Journal of Chemical Physics, Vol. 12, no. 11, pg. 449 (1944)

[5] D. Gilbarg, "The existence and limiting behavior of the one-dimensional shock layer," American Journal of Mathematics, vol. 73, no. 2 (1951)

[6] R. von Mises, "On the thickness of a steady shock wave," Journal of the Aeronautical Science, pg. 551 (1950)

[7] S. H. Radin and D. Mintzer, "Orthogonal polynomial solution of the Boltzmann equation for a strong shock," The Physics of Fluid, vol. 9, no. 9, pg. 1621 (1966)

[8] S. Chapman and T. G. Cowling, "The mathematical theory of non-uniform gases," Cambridge University Press (1939)

[9] H. Grad, "The profiles of a steady plane shock wave," Communications on Pure and Applied Mathematics, vol. v, pg. 257 (1952)

[10] H. Grad, "On the kinetic theory of rarefied gases," communications on Pure and Applied Mathematics, vol. 2, no. 4 (1949)

[11] H. M. Mott-Smith, "The solution of the Boltzmann equation for a shock wave," Physical Review, vol.82, no. 6, pg. 885 (1951)

[12] C. Muckenfuss, "Some aspects of shock structure according to the bimodal model," The Physics of Fluid, vol. 5, no. 11, pg. 1325 (1962)

[13] H. Salwen, C. E. Grosch and S. Ziering, "Extension of the Mott-Smith method for a one-dimensional shock wave," The Physics of Fluid, vol. 7, no. 2, pg. 180 (1964)

[14] S. H. Radin and D. Mintzer, "Orthogonal polynomial solution of the Boltzmann equation for a strong shock," The Physics of Fluid, vol. 9, no. 9, pg. 1621 (1966)

[15] D. L. Rode and B. S. Tanenbaum, "Mott-Smith shock thickness calculations using the v_x^p method," The Physics of Fluid, Vol. 10, pg. 1352 (1967)

[16] D. Gilbarg and D. Paolucci, "The structure of shock waves in the continuum theory of fluids," Journal of Rational Mechanics and Analysis 2, pg. 617 (1953)

[17] M. Greenspan, "Attenuation of sound in rarefied helium," Physical Review 75, pg. 197 (1949)

[18] M. Greenspan, "Propagation of sound in rarefied helium," Journal of Acoustical Society of America 22, pg. 568 (1950)

[19] C. Truesdell, "Precise theory of the absorption and dispersion of forced plane infinitesimal waves according to the Navier-Stokes equation," Journal of Rational Mechanics and Analysis 3 (1954)

[20] G. R. Cowan and D. F. Hornig, "The experimental determination of the thickness of a shock front in a gas," The Journal of Chemical Physics vol. 18, no. 8, pg. 1008 (1950)

[21] E. F. Greene, G. R. Cowan, and D. F. Hornig, "The thickness of shock fronts in argon and nitrogen and rotational heat capacity lags," The Journal of Chemical Physics, vol. 19, no. 4, pg. 427 (1951)

[22] M. Linzer and D. F. Hornig, "Structure of shock front in argon and nitrogen," The Physics of Fluids, vol. 6, no. 12, pg. 1661 (1963)

[23] D. F. Hornig, "A method of measuring the thickness of a shock front," Physical Review, vol. 72, American Physical Society, Contributed Paper, pg. 179 (1947)

[24] F. Robben and L. Talbot, "Measurement of shock wave thickness by the electron beam fluorescence method," The Physics of Fluids, vol. 9, no. 4, pg. 633 (1966)

[25] B. Schmidt, "Electron beam density measurements in shock waves in argon," Journal of Fluid Mechanics, vol. 39, part 2, pg. 361 (1969)

[26] E. P. Muntz and L. N. Harnett, "Molecular velocity distribution function measurements in a normal shock wave," The Physics of Fluids, vol. 12, no. 10, pg. 2027 (1969)

[27] T. Holtz, E. P. Muntz and S.-M. Yen, "Comparisons of measured and predicted velocity distribution functions in a shock wave," The Physics of Fluids, vol. 14, no. 3, pg. 545 (1971)

[28] G.A. Bird, "Molecular gas dynamics," Clarendon Press, Oxford. © Oxford University Press (1976)

[29] B. J. Alder and T. E. Wainwright, "Phase transition for a hard sphere system," The Journal of Chemical Physics, vol. 27, pg. 1208 (1957)

[30] G. A. Bird, "Shock-wave structure in a rigid sphere gas," Rarefied Gas Dynamics (Ed. de Leeuw), pg. 216, New York: Academic Press (1965)

[31] G. A. Bird, "Velocity distribution function within a shock wave," J. Fluid Mech., vol. 30, part 3, pg. 479 (1967)

[32] S.-M. Yen, "Temperature overshoot in shock waves," The Physics of Fluids, vol. 9, no. 7, pg. 1417 (1966)

[33] G. A Bird, "Aspect of the structure of strong shock waves," The Physics of Fluids, vol. 13, no. 5, pg. 1172 (1970)

[34] G. A. Bird, Proceedings of the Sixth International Symposium on Rarefied Gas Dynamics (1968)

[35] D. H. Tsai and S. F. Trevino, "Thermal relaxation in a dense liquid under shock compression," Physical Review A, vol. 24, no. 5, pg. 2743 (1981)

[36] W. G. Hoover, "Structure of shock-wave front in a liquid," Physical Review Letters, vol. 42, no. 23, pg. 1531 (1979)

[37] B. L. Holian, W. G. Hoover, B. Moran and G. K. Straub, "Shock-wave structure via nonequilibrium molecular dynamics and Navier-Stokes continuum mechanics," Physical Review A, vol. 22, no. 6, pg. 2798 (1980)

[38] J. A. Barker, R. A. Fisher and R. O. Watts, "Liquid argon: Monte Carlo and molecular dynamics calculations," Molecular Physics, vol. 21, no. 4, pg. 657 (1971)

[39] E. A. Guggenheim, F.R.S. and M. L. McGlashan, "Interaction between argon atoms," Proc. R. Soc. A, vol. 255, pg. 456 (1960)

[40] M. L. McGlashan, "Effective pair interaction energy in crystalline argon," Discuss. Faraday Soc., vol. 40, pg. 59 (1965)

[41] K. A. Fiscko and D. R. Chapman, "Comparison of shock structure solutions using independent continuum and kinetic approaches," SPIE Symposium on Innovative Sciences and Technology, Los Angeles, California, 10-15 January (1988)

[42] K. Niki and S. Ono, "Molecular dynamics study on the structure of shock wave front in a hard-core fluid," Physics letters, vol. 62A, no. 6, pg. 427 (1977)

[43] S.-M. Yen and W. Ng, "Shock-wave structure and intermolecular

collision laws," J. Fluid Mech., vol. 65, part 1, pg. 127 (1974)

[44] B. L. Hicks, S.-M. Yen and B. J. Reilly, "The internal structure of shock waves," J. Fluid Mech., vol. 53, part 1, pg. 85 (1972)

[45] P. L. Bhatnagar, E. P. Gross, and M. Krook, "A Model for collision process in gases. I. small amplitude processes in charged and neutral one-component systems," Physical Review, vol. 94, no. 3, pg. 511 (1954)

[46] G. A. Bird, "Direct simulation of high-vorticity gas flows," Phys. Fluids 30 (2), pg. 364 (1987)

[47] G. A. Bird, "Monte Carlo simulation of gas flows," Ann. Rev. Fluid. Mech., vol. 10, pg. 11 (1978)

[48] See the list of surveyed literature in the back of [56]

[49] I. Greber and H. Wachman, "Scaling rules and time averaging in molecular dynamics computations of transport properties," Rarefied Gas Dynamics: Theoretical and Computational Techniques, Edited by E. P. Muntz, D. P. Weaver, and D. H. Campbell, Vol. 118 of Progress in Astronautics and Aeronautics, AIAA, Washington, DC, ISBN 0-930403-55-X (1989)

[50] M. J. Woo, I. Greber and H. Y. Wachman, "Molecular dynamics computations of two-dimensional supersonic rarefied gas flow past blunt bodies," 29th Aerospace Science Meeting, AIAA-91-0322, 7-10 January (1991)

[51] M. J. Woo, "Molecular dynamics test computations of two dimensional hypersonic flows," Master thesis, Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, Ohio (1990)

[52] Kelvin, Lord, "Nineteenth century clouds over the dynamical theory of heat and light," Philos. Mag. 2:1-40 (1901)

[53] G. A. Bird, "Approach to transitional equilibrium in a rigid sphere gas," Physics of Fluids, vol. 6, pg. 1518 (1963)

[54] G. A. Bird, "Direct molecular simulation of a dissociating diatomic gas," J. Of Comput. Phys. (1977)

[55] W. G. Hoover, "Nonequilibrium molecular dynamics," Ann. Rev. Phys. Chem., vol. 34, pg. 103 (1983)

[56] D. J. Evans and W. G. Hoover, "Flows far from equilibrium via molecular dynamics," Ann. Rev. Fluid Mech. vol. 18, pg. 243 (1986)

[57] See page 366 of G. A. Bird, "Direct Simulation of high-vorticity gas flows," Phys. Fluids, vol. 30, no. 2, pg. 364 (1987)

[58] Paul Barnhardt, Personal communications, Supervisor, Aerospace Analysis Section, LeRC Group, Sverdrup Technology, Inc. 16530 Commerce Court, P.O.Box 30650, Midpark Branch, Middleburg Heights, Ohio 44130, NASA-LeRC, 216-433-1182.

[59] Cohort Software, "Coplot Manual Revision 2.02," Cohort Software, ©Cohort Software, Berkeley, CA., pg. 88 (1990)

[60] Lowell H. Holway, Jr., "Temperature overshoot in shock waves," The Physics of Fluids, Research Note, Vol. 8, no. 10, pg. 1905 (1965)

7. Appendix

7.1 Source Code

The main program is divided into small sections according to their function. Source code is not grammatically correct as there are strange capitalization and mis-spelled words. All lower case "L" in the source code are deliberately capitalized regardless of their position in a word in order to distinguish them with the number 1. Some of the subroutine name are mis-spelled to accomodate the restriction on the number of letters in a variable name.

7.1.1 Initialization routine

The following source code computes the diameter of particles and selects position and velocities for particles, given total number of particles, length and the diameter of computational region as well as the smallest spacing ratio, which occurs in the high density region.

```
Program Start
      parameter(n = number of particLes, nw = number of boundaries,
     + tL = Length of computationaL region, nx = number of data
     +      coLLecting ceLLs)
      Dimension x(n), y(n), z(n), u(n), v(n), w(n)
      pi = acos(-1.)
      xma = simuLation mach number
      tsd = the worst spacing ratio
```

```
      errr = 1e-10  'error toLerated'
      R = .5  'radius of the computationaL region'
      yc = .5  'y Location for center of comput. region'
      zc = .5  'z Location for center of comput. region'
      xw = tL  'initiaL Location of piston'
      gam = 5. / 3.  'Specific heat ratio for hard sphere'
      cmp1 = 1.  'Most probabLe speed is set at 1'
      s1 = sqrt(gam / 2.) * xma  'upstream speed ratio'
      u11 = s1 / cmp1  'upstream veLocity'
      cmp2 = cmp1 * sqrt((2.*gam*xma**2-(gam-1.))*
     +       ((gam-1.)*xma**2+2.)/((gam+1.)**2*xma**2))
c 'most probabLe speed at downstream region'
      xma2=sqrt(((gam-1.)*xma**2+2.)/(2.*gam*xma**2-(gam-1.)))
c 'downstream mach number'
      s2 = sqrt(gam / 2.) * xma2
c 'downstream speed ratio'
      u21 = s2 * cmp2
c 'downstream veLocity in terms of most probabLe speed'
      uw = -u11 + u21 'the piston speed'
      rhoR = ((gam+1.)*xma**2)/((gam-1.)*xma**2+2.)
c 'density ratio'
c knowing rhoR and target s/d, we can find nominaL s/d
      sd = tsd * rhoR**(1./3.)
c given n and target s/d, radius is determined by triaL and error
      step = 1.
      fac = 100.
      dia = 1. / (sd * ((reaL(n) / reaL(nx)) /
     +      ((tL / reaL(nx)) * pi * R**2))**(1. / 3.))
11    tr = dia / 2.
      rhs = tsd * (reaL(n) * rhoR / reaL(nx))**(1./3.) * dia *
     +      ((tL-dia) / reaL(nx) * pi * (R - tr)**2)**(-1./3.)
      write(6,*)rhs - 1.
      if (abs(rhs - 1.) .Le. errr) goto 21
      if ((rhs - 1.) .Lt. 0.) then
         dia = dia + step / fac - step / (fac * 10.)
         fac = fac * 10
         goto 11
         end if
      dia = dia - step / fac
      goto 11
21    effL = tL - dia
      barn = (reaL(n) / reaL(nx)) /
     +       ((tL - dia) / reaL(nx) * pi * (R-tr)**2)
      soverd = 1. / (barn**(1./3.) * dia)
      rknud = 1. / (sqrt(2.) * barn * dia**2 * pi)
      write(6,*)soverd, rknud
c particLes are randomLy distributed and given maxweLLian veLocity.
      do 10 i = 1, n
         xi = effL * ranf() + tr
30       theta = ranf() * 2. * pi
```

```
          tm = ranf() * (R - tr)
          zi = tm * cos(theta) + zc
          yi = tm * sin(theta) + yc
          do 20 j = 1, i - 1
             xj = x(j)
             yj = y(j)
             zj = z(j)
             xs = (xj - xi)**2
             ys = (yj - yi)**2
             zs = (zj - yi)**2
             diff = sqrt(xs + ys + zs)
             if (diff .Lt. dia) goto 30
20           continue
          x(i) = xi
          y(i) = yi
          z(i) = zi
c InitiaLize the veLocities for particLes.
100       cn = (ranf() - .5) * 6.
          vp = exp(-cn * cn)
          if (ranf() .gt. vp) goto 100
110       c1 = (ranf() - .5) * 6.
          vp = exp(-c1 * c1)
          if (ranf() .gt. vp) goto 110
120       c2 = (ranf() - .5) * 6.
          vp = exp(-c2 * c2)
          if (ranf() .gt. vp) goto 120
          u(i) = cn
          v(i) = c1
          w(i) = c2
          k1 = mod(i, 100)
          if (k1 .eq. 0) write(6,*) i
10        continue
       nstart = 1
       nunit = 12
       tt = 0.
C particLe position and veLocity is recorded in a fiLe.
       open(unit = 12, fiLe='init.vpi')
       write(12, *)nstart, tt, tr, xw, uw
       do 121 i = 1, n
          write(12, *)x(i),y(i),z(i)
          write(12, *)u(i),v(i),w(i)
121       continue
       cLose(12)
       stop
       end
```

7.1.2 main routine

The following routine finds the next time to collide. The previous time to collide is subtract from all elements as the result of computing the time to collide only for particles which have different velocities as a result of interaction. For these newly computed time to collide, the previous time to collide is added at the time of computation.

```
      st1 = big 'shortest time is initiaLized to be a big number'
      do 100 i = 1, n 'n is number of particLes'
         do 110 j = i + 1, npnw 'npnw is n pLus number of waLLs'
            stime = t(i, j) - stb
            t(i, j) = stime
            if (stime .Lt. st1) then
               st1 = stime
               ni = i
               nj = j
               end if
110           continue
100        continue
c checks the found time for possibLe bLunder. The shortest time to
c coLLide can not be a negative number.
      if (st1 .Lt. 0.) then
          write(6, *) 'st1 is negative...'
          write(6, *) 'st1 : ',st1
          write(6, *) nco, ni, nj
          stop
          end if
c simuLation time is updated and the shortest time is saved
c for use in finding the shortest time for the next time.
      tt = tt + st1
      stb = st1
c position of the piston is updated and checks if it has run into
c the end waLL for possibLe programming bLunder.
      xw = xw + uw * st1
      if (xw .Le. 0.) then
         write(6, *)'piston is at waLL... error'
         stop
```

```
      end if
```

7.1.2.1 time to collide between two particles

Following subroutine compute the time to collide between particle i and j. The i^{th} particle is represented by xi, yi, and zi for the position and ui, vi, and wi for the velocity. Likewise the j^{th} particle is represented by xj, yj, zj, uj, vj, and wj. The algorithm is given in Section 3.2.1.1.

```
      subroutine moLecuLe(xj,xi,yj,yi,zj,zi,uj,ui,vj,vi,
     +                    wj,wi,dia2,t1,big,errr)
      c = (xj - xi)**2 + (yj - yi)**2 + (zj - zi)**2 - dia2
c Instead checking to see if a vaLue is zero, the absoLute vaLue of
c the variabLe is compared whether this number is Less than error
c toLerated.
      if (sqrt(abs(c)) .Le. errr) then
         t1 = 0.
         return
         end if
      b = ((xj - xi) * (uj - ui) +
     +     (yj - yi) * (vj - vi) +
     +     (zj - zi) * (wj - wi)) * 2.
      if (b .gt. 0.) then
         t1 = big
         return
         end if
      a = (uj - ui) * (uj - ui) +
     +    (vj - vi) * (vj - vi) +
     +    (wj - wi) * (wj - wi)
      if (abs(a) .Le. errr) then
         t1 = big
         return
         end if
      determ = b * b - 4. * a * c
      if (determ .Lt. 0.) then
         t1 = big
         return
         end if
      drt = sqrt(determ)
```

```
      if ((-b) .Le. drt) then
         t1 = big
         return
         end if
      neo = nint(b / abs(b))
      q = -0.5 * ( b + reaL(neo) * drt)
      t1 = c / q
      return
      end
```

7.1.2.2 time to collide between a particle and boundaries

For a particle with its position represented as xi, yi, and zi with velocities ui, vi, and wi, following subroutine gives the time to collide with boundary 1, the upstream boundary, is given as t1, with boundary 2, the piston, is given as t2 and the time to collide with the lateral boundary is t3.

```
      subroutine bounday(uw,xw,ui,vi,wi,tr,xi,yi,zi,
     +     big,rc,yc,zc,t1,t2,t3)
      errr = 1.e-15
      if (ui .gt. 0.) then
         t1 = big
         t2 = (xi + tr - xw) / (uw - ui)
         end if
      if (ui .Lt. 0.) then
         t1 = (tr - xi) / ui
         t2 = (xi + tr - xw) / (uw - ui)
         if (t2 .Lt. 0.) t2 = big
         end if
      if (ui .eq. 0.) then
         t1 = big
         t2 = (xi + tr - xw) / uw
         end if
c foLLowing routine are for cyLindericaL computationaL region.
c checking the distance between the center and the particLe...
      c = (zi - zc)**2 + (yi - yc)**2 - (rc - tr)**2
```

```
c If c is 0, then the particLe is in coLLision position.
      if (abs(c) .Le. errr) then
         t3 = 0.
         return
         end if
c At this stage, c must be Less than zero.
      if (c .gt. 0.) then
         write(6,*)'particLe outside....'
         write(6,*)'error has occured....'
         stop
         end if
      b = 2. * (vi * (yi - yc) + wi * (zi - zc))
      a = vi**2 + wi**2
      if (a .eq. 0.) then
         t3 = big
         return
         end if
c c must be negative at aLL times
c if the particLes are within the computationaL region.
c a is aLways positive.
c b can be positive or negative or even zero.
c Both roots must be reaL, one positive and one negative.
c It is the positive root that we are interested in.
      drt = b**2 - 4. * a * c
      if (drt .Lt. 0.) then
         write(6,*)'imaginary root ..  det. is negative..'
         stop
         end if
      q = -0.5 * (b + b / abs(b) * sqrt(drt))
      x1 = q / a
      x2 = c / q
      if (x1 .Lt. 0.) then
         t3 = x2
         return
         end if
      t3 = x1
      return
      end
```

7.1.2.3 Particle to particle collision

For two colliding particles, their velocities are computed using the source code below. ni represents the first particle number and nj is the second particle number.

```
      ui = u(ni)
      vi = v(ni)
      wi = w(ni)
      uj = u(nj)
      vj = v(nj)
      wj = w(nj)
      xr = x(nj) - x(ni)
      yr = y(nj) - y(ni)
      zr = z(nj) - z(ni)
      ci = (ui * xr + vi * yr + wi * zr) / dia2
      cj = (uj * xr + vj * yr + wj * zr) / dia2
      cij = ci - cj
      cji = cj - ci
      u(ni) = ui + xr * cji
      v(ni) = vi + yr * cji
      w(ni) = wi + zr * cji
      u(nj) = uj + xr * cij
      v(nj) = vj + yr * cij
      w(nj) = wj + zr * cij
c write the pertinent info. for data coLLection
      write(10,*) ni, nj, st1
      write(10,*) u(ni), v(ni), w(ni)
      write(10,*) u(nj), v(nj), w(nj)
```

7.1.2.4 particle to boundary collision

Following source code show how the interaction between a particle and a boundary is treated. Line starting with number 1, 2 and 3 corresponds to the interaction of the particle with the corresponding boundary.

```
1     ui = -ui
      xi = tr
      t(ni, n1) = big
      t(ni, n2) = (xi + tr - xw) / (uw - ui) + st1
      u(ni) = ui
      x(ni) = xi
      goto 455
2     ui = uw - (ui - uw)
      xi = xw - tr
      u(ni) = ui
      x(ni) = xi
      t(ni, n1) = (tr - xi) / ui + st1
      t(ni, n2) = big
      c = (zi - zc)**2 + (yi - yc)**2 - (rc - tr)**2
      if (abs(c) .Le. errr) then
         t3 = 0.
         goto 454
         end if
      if (c .gt. 0.) then
         write(6,*)'particLe outside....'
         write(6,*)'error has occured....'
         stop
         end if
      b = 2. * (vi * (yi - yc) + wi * (zi - zc))
      a = vi**2 + wi**2
      if (a .eq. 0.) then
         t3 = big
         goto 454
         end if
      drt = b**2 - 4. * a * c
      if (drt .Lt. 0.) then
         write(6,*)'imaginary root ..  det. is negative..'
         stop
         end if
      q = -0.5 * (b + b / abs(b) * sqrt(drt))
      x1 = q / a
      x2 = c / q
      if (x1 .Lt. 0.) then
         t3 = x2
         goto 454
         end if
      t3 = x1
454   t(ni, n3) = t3 + st1
455   write(10,*)ni, nj, st1
      write(10,*)xi, ui
      goto 460
```

```
c  LateraL waLLs

3      caLL LtrL(pi,yi,zi,vi,wi,yc,zc)
       v(ni) = vi
       w(ni) = wi
       write(10,*)ni, nj, st1
       write(10,*)vi, wi
       b = 2. * (vi * (yi - yc) + wi * (zi - zc))
       a = vi**2 + wi**2
       t(ni, n3) = -b / a + st1
```

Since the interaction of a particle with the computational lateral bounary requires a coordinate transformation from a Cartesian to the normal and tangential coordinate system and back to the Cartesian coordinate system, following subroutine is used.

```
subroutine LtrL(pi,yi,zi,vi,wi,yc,zc)
c
c Inverting the veLocity to normaL and tangentiaL component
c Get the normaL directionaL vector from the center to the particLe.
c ReaLize that the particLe position is aLready at the
c coLLision position.
c First, find the axis rotation angLe, rw
c
       zdi = zi - zc
       ydi = yi - yc
       dinom = sqrt(zdi * zdi + ydi * ydi)
       argum = ydi / dinom
       if (yi .ge. yc) then
          if (zi .gt. zc) angLe = asin(argum)
          if (zi .Le. zc) angLe = pi - asin(argum)
             eLse
                if (zi .gt. zc) angLe = 2. * pi + asin(argum)
                if (zi .Le. zc) angLe = pi - asin(argum)
          end if
       sinw = sin(angLe)
       cosw = cos(angLe)
c
c FoLLowings are the incident veLocities in normaL and tangentiaL
c coordinate
c
       vit = vi * cosw - wi * sinw
       vin = vi * sinw + wi * cosw
```

```
c For specuLar condition, onLy normaL component reverses the sign.

      vrt = vit
      vrn = -vin
c Now we must convert them back to Cartesian coordinate system.
c We go through the same procedure with angLe = -angLe.
c ResuLting vi and wi are the resuLting veLocity component
c of specuLar refLection in Cartesian coordinate system.
      sinw = sin(-angLe)
      cosw = cos(-angLe)
      vi = vrt * cosw - vrn * sinw
      wi = vrt * sinw + vrn * cosw
      return
      end
```

7.1.3 data extraction routine

Following source code works with the data files generated from the main program and write out the information needed to extract density, velocity and temperature at a constant interval. The interval is equal to the time it takes the shock wave to move one upstream mean free path moving at theoretical velocity.

```
      program Look
c
c coLLects data in time intervaL which is the time for the shock
c to move 1 upstream mean free path.
c
      parameter (n = number of particLe, nw = number of boundaries,
     *       nx = number of data coLLecting ceLL, tL = totaL Length)
c extra dimension is added to soLve a memory confLict probLem and to
c improve the execution speed on CRAY C90 under the advise of the
c CRAY system operator.
      dimension  xpw(nx+2),pw(nw+1),totaL(nx+1),
     *           x(n+1),y(n+1),z(n+1),u(n+1),v(n+1),w(n+1),
     *           tau(nx+1),usum(nx+1),vsum(nx+1),wsum(nx+1),
     *           nsum(nx+1),usm2(nx+1),vsm2(nx+1),wsm2(nx+1)
c if isave is set to 1, program save the positon of the veLocity and
c position regardLess whether the program has anaLyzed aLL the input
c or not.  This provides a safety net in case something goes worng.
```

```
      isave = 0
      ncp = 0  'number of particLe to particLe interaction'
      ncw = 0  'number of particLe to waLL interaction'
      ncw1 = 0  'number of particLe to waLL 1 interaction'
      ncw2 = 0  'number of particLe to waLL 2 interaction'
      ncw3 = 0  'number fo particLe to waLL 3 interaction'
c read the intiaL position and veLocity of particLes
      open(unit = 10, fiLe='/scr2/mwoo/3i.vpi')
      read(10, *)nstart, tt, tr, xw, uw
      tb = tt
      do 20 i = 1, n
         read(10, *)x(i),y(i),z(i)
         read(10, *)u(i),v(i),w(i)
c if starting from other than the initiaL, read in the time to
c coLLide as weLL.
         if (nstart .ne. 1) then
            do 24 j = i + 1, n + nw
               read(10, *) dum
24             continue
            end if
20       continue
         cLose(10)
      tr2 = tr * tr
      dia = tr + tr
      dia2 = dia * dia
      barn = reaL(n) / reaL(nx) * 4. / pi
      rknud = 1. / (sqrt(2.) * pi * barn * dia**2)
      soverd = 1. /(barn**(1./3.) * dia)
c time it takes for the shock wave moving at the theoreticaL speed a
c upstream mean free path.
      deLt = rknud / abs(uw)
      deLtn = tt + deLt
c width for the data coLLecting ceLLs
      cL = (tL - dia) / reaL(nx)
      do 10 i = 1, nx+1
         xpw(i) = reaL(i-1) * cL + tr
10       continue
      pw(1) = 0.
      pw(2) = tL
c if staring from intermediate resuLts, variabLes coLLected from
c previous run are read in in order to continue the data gathering.
      if (nstart .ne. 1) then
         open(unit = 10, fiLe ='/scr2/mwoo/31.avg')
         read(10,*)tt, deLtn
         do 23 i = 1, nx
            read(10,*) tau(i),nsum(i),usum(i),vsum(i)
            read(10,*)wsum(i),usm2(i),vsm2(i),wsm2(i)
            read(10,*)totaL(i)
23          continue
         cLose(10)
```

```
        end if
c opening the coLLision info. fiLe.
50    open(unit = 12, fiLe='/scr2/mwoo/31.out')
      read(12, *)nstrt1, nend
c if the coLLision dynamics fiLe does not have the same starting and
c ending coLLsion counter, probabLy it is the wrong fiLe.
       if (nstart .ne. nstrt1) then
         write(6,*)'wrong data set'
         stop
         end if
c write the header for the resuLt fiLe.
      open(unit = 10, fiLe='/scr2/mwoo/31.prn')
      write(10,*)'u11 ', u11
      write(10,*)'u21 ', u21
      write(10,*)' '
      write(10,*)'piston speed =      ', uw
      write(10,*)'soverd =            ', soverd
      write(10,*)'Knudsen number =    ', rknud
      write(10,*)'Cmp2                ', cmp2
      write(10,*)'density ratio       ', rhoR
      write(10,*)'radii               ', tr
      write(10,*)'Mach number is      ', xma
      write(10,*)'deLta t = ', deLt
      write(10,*)'totaL time = ',tt
      write(10,*)' '
      write(10,*)'starting coLL. counter : ', nstart
      write(10,*)'FinaL coLL. counter    : ', nend
      write(10,*)' '
c
c beginning main Loop
c
      do 90 nco = nstart, nend
c read the index for the coLLision pair which shows what type of
c coLLision.
         read(12,*)ni, nj, st1
         tt = tt + st1
         stb = st1
c the piston advances to new position
         xw = xw + uw * st1
         ip = int((xw - tr) / cL) + 1
c see which data coLLecting ceLL the piston is at.  If the ceLL
c index is Larger than the Last index for the data coLLecting ceLL,
c it is at starting position and causes error in the rest of the
c computation.  Therefore eLiminate this possibLe error by assigning
that the piston is in the Last ceLL.
         if (ip .gt. nx) ip = nx
c
c spatiaL data coLLection in 1-d
c
         do 120 i = 1, n
```

```
c use dummy variabLes as their vaLues are often requested.
            ui = u(i)
            vi = v(i)
            wi = w(i)
            xi = x(i)
            yi = y(i)
            zi = z(i)
            ui2 = ui**2
            vi2 = vi**2
            wi2 = wi**2
c position of the particLe at the end of the shortest time to
c coLLide.
            xf = xi + ui * st1
            yf = yi + vi * st1
            zf = zi + wi * st1
c L is the ceLL index when the particLe is at xi, and m is the ceLL
c index when the particLe is at xf.
            L = int((xi - tr) / cL) + 1
            if (L .gt. ip) L = ip
            m = int((xf - tr) / cL) + 1
            if (m .gt. ip) m = ip
c
c if movement of particLe is restricted within a ceLL, then..
c Note that zero u veLocity is aLso taken care of by this routine.
c
            if (m .eq. L) then
               x(i) = xf
               y(i) = yf
               z(i) = zf
               tau(m) = tau(m) + st1
               nsum(m) = nsum(m) + 1
               usum(m) = usum(m) + ui
               vsum(m) = vsum(m) + vi
               wsum(m) = wsum(m) + wi
               usm2(m) = usm2(m) + ui2
               vsm2(m) = vsm2(m) + vi2
               wsm2(m) = wsm2(m) + wi2
               totaL(m) = tt
               go to 120
               end if
c
c if the movement is over severaL ceLLs....
c
c compute fraction of ceLL occupation
c
            if (ui .gt. 0.) then
               bfci = (xpw(L+1) - xi) / cL
               bfcf = (xf - xpw(m)) / cL
               eLse if (ui .Lt. 0.) then
                  bfci = (xi - xpw(L)) / cL
```

```
                bfcf = (xpw(m+1) - xf) / cL
                eLse
                   pause
             end if
          neo = nint(ui / abs(ui))
          factor = reaL(neo) * cL / ui
          tau(L) = tau(L) + bfci * factor
          totaL(L) = tt
          tau(m) = tau(m) + bfcf * factor
          totaL(m) = tt
          do 129 j = L, m, m - L
             nsum(j) = nsum(j) + 1
             usum(j) = usum(j) + ui
             vsum(j) = vsum(j) + vi
             wsum(j) = wsum(j) + wi
             usm2(j) = usm2(j) + ui2
             vsm2(j) = vsm2(j) + vi2
             wsm2(j) = wsm2(j) + wi2
             totaL(j) = tt
129          continue
          do 130 j = L+neo, m-neo, neo
             tau(j) = tau(j) + factor
             nsum(j) = nsum(j) + 1
             usum(j) = usum(j) + ui
             vsum(j) = vsum(j) + vi
             wsum(j) = wsum(j) + wi
             usm2(j) = usm2(j) + ui2
             vsm2(j) = vsm2(j) + vi2
             wsm2(j) = wsm2(j) + wi2
             totaL(j) = tt
130          continue
          x(i) = xf
          y(i) = yf
          z(i) = zf
120       continue
c write the info for avg.
C if current simuLation time is greater than the deLtn, the shock
c wave moved a upstream mean free path and it is the time to output
c the vaLue of variabLes gathered so far.
       if (tt .ge. deLtn) then
          deLtn = deLtn + deLt
          write(10,*)' '
          write(10,*)nco, tb, tt,' piston is at ',xw
          dm = 0.
          do 760 i = 1, nx
             where = (xpw(i) + xpw(i+1)) / 2.
             write(10,*)where,tau(i),nsum(i),usum(i),vsum(i),
     +          wsum(i),usm2(i),vsm2(i),wsm2(i),totaL(i)
             dm = dm + tau(i)
760          continue
```

```
            write(10,*)' '
            write(10,*)'sum of part. / ceLL = ', dm / tt
            write(10,*)' '
            write(10,*)' '
            write(10,*)'# of coLLisions with waLLs'
            write(10,*)'    1 : ',ncw1,'  2 : ',ncw2,'  3 : ',ncw3
            write(10,*)' '
            write(10,*)'Average incoming veLocity  '
            if (ncw1 .ne. 0) write(10,*)'up ',sumup / reaL(ncw1)
            if (ncw2 .ne. 0) write(10,*)'dn ',sumdn / reaL(ncw2)
            write(10,*)'sum of coLLisions with      waLL  (pw) ',ncw
            write(10,*)'sum of coLLisions with particLes  (pp) ',ncp
            if (ncp.ne.0)write(10,*)'pw/pp is',reaL(ncw)/reaL(ncp)
            tb = tt
            end if
c if the coLLision counter has reached the end of the coLLsion
c dynamics information fiLe, then the data gathering is over.  To
c continue, vaLues of the varibLes must be saved in order to
c continue the data gathering Later.
         if (nco .eq. nend) then
            open(unit=14, fiLe='/scr2/mwoo/31.avg')
            write(14,*)tt, deLtn
            do 761 i = 1, nx
               write(14,*) tau(i),nsum(i),usum(i),vsum(i)
               write(14,*)wsum(i),usm2(i),vsm2(i),wsm2(i)
               write(14,*)totaL(i)
761            continue
            cLose(14)
c save veLocity and position of particLes at the end of run.
            if (isave .eq. 1) then
               open(unit=14, fiLe='/scr2/mwoo/31.end')
               write(14,*)nend + 1, tt, tr, xw, uw
               do 762 i = 1, n
                  write(14,*)x(i),y(i),z(i)
                  write(14,*)u(i),v(i),w(i)
762               continue
               cLose(14)
               end if
            end if
c
c coLLision dynamics
c
c if the index nj is greater than the number of particLes, then it
is the interactin between the ni^th particLe with the (nj-n)^th waLL.
         if (nj .gt. n) then
            go to 270
            eLse
               go to 280
            end if
270      nuw =  nj - n
```

```
          go to (1, 2, 3) nuw
c onLy thing that changed in the interaction with waLL 1 or 2 is the
c u veLocity of particLe.  The x position is reset to a correct
c vaLue just in case the position has accumuLated enough machine
c accuracy.
1         read(12, *)x(ni), u(ni)
          ncw1 = ncw1 + 1
          goto 460
2         read(12, *)x(ni), u(ni)
          ncw2 = ncw2 + 1
          goto 460
c for waLL 3, onLy v and w veLocity change.
3         read(12, *)v(ni), w(ni)
          ncw3 = ncw3 + 1
460       ncw = ncw + 1
          go to 1000
c
c particLe to particLe coLLision
c
c particLe to particLe interaction changes aLL components of
c veLocity for both particLes
280       read(12, *)u(ni),v(ni),w(ni)
          read(12, *)u(nj),v(nj),w(nj)
          ncp = ncp + 1
c
1000      continue
90        continue
       cLose(12)
       cLose(10)
       stop
       end
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1995	Final Contractor Report

4. TITLE AND SUBTITLE

Molecular Dynamics Simulation of a Piston Driven Shock Wave in a Hard Sphere Gas

5. FUNDING NUMBERS

WU-233-01-0E
G-NAG3-795

6. AUTHOR(S)

Myeung-Jouh Woo and Isaac Greber

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Case Western Reserve University
Cleveland, Ohio 44106

8. PERFORMING ORGANIZATION REPORT NUMBER

E-9603

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR-195463

11. SUPPLEMENTARY NOTES

This report was submitted as a dissertation in partial fulfilment of the requirements for the degree Doctor of Philosophy to Case Western Reserve University, Cleveland, Ohio. Project Manager, Dale C. Ferguson, Power Technology Division, NASA Lewis Research Center, organization code 5470, (216) 433-2298.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 72

This publication is available from the NASA Center for Aerospace Information, (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Molecular dynamics simulation is used to study the piston driven shock wave at Mach 1.5, 3, and 10. A shock tube, whose shape is a circular cylinder, is filled with hard sphere molecules having a Maxwellian thermal velocity distribution and zero mean velocity. The piston moves and a shock wave is generated. All collisions are specular, including those between the molecules and the computational boundaries, so that the shock development is entirely causal, with no imposed statistics. The structure of the generated shock is examined in detail, and the wave speed, profiles of density, velocity, and temperature, and shock thickness are determined. The results are compared with published results of other methods, especially the direct simulation Monte-Carlo method. Property profiles are similar to those generated by direct simulation Monte-Carlo method. The shock wave thicknesses are smaller than the direct simulation Monte-Carlo results, but larger than those of the other methods. Simulation of a shock wave, which is 1-dimensional, is a severe test of the molecular dynamics method, which is always 3-dimensional. A major challenge of the thesis is to examine the capability of the molecular dynamics methods by choosing a difficult task. This work is essentially the doctoral dissertation of Myeung-Jouh Wood, performed with Isaac Greber as Faculty Advisor.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Shock structure; Molecular dynamics; Kinetic theory	123
	16. PRICE CODE: A06

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	