ASEN 6061 Project 2: Direct Simulation Monte Carlo

Kalvin Y. Monroe[*]
University of Colorado, Boulder, CO, 80303

An investigation of thermal nonequilibrium conditions are explored from a molecular standpoint. A Direct Simulation Monte Carlo (DSMC) code was developed to explore this. In this paper, DSMC methodology and results are presented for a shock tube and a confined volume with Argon (Ar) as the working fluid. For the shock tube, multiple freestream mach numbers and densities are chosen and the resulting shock width is computed. For the confined volume, various Knudsen numbers and wall conditions are chosen, and gas quantities are tracked in time. Variable wall conditions include wall temperature and thermal accommodation coefficient. Results show that collisions drive a system towards equilibrium by transferring energy between particles.

I. Nomenclature

DSMC	=	Direct simulation monte carlo
VHS	=	Variable hard-sphere
NTC	=	No-time counter
Kn	=	Knudsen number
λ	=	Mean free path [m]
ρ	=	Density [kg/m^3]
γ	=	Ratio of specific heats
R	=	Gas constant [J/kgK]
T_w	=	Wall temperature [K]
□$_1$	=	State 1 (freestream) temperature [K]
□$_2$	=	State 2 (post-shock) temperature [K]
T	=	Reservoir temperature [K]
d_{ref}	=	Reference diameter for Ar [m]
$\mathbf{g}$	=	Relative velocity [m/s]
g_{ref}	=	Reference relative velocity magnitude for Ar [m/s]
ω	=	Viscosity law exponent
σ_T	=	Particle collision cross section [m^2]
L_x	=	Domain length [m]
L_y	=	Domain height [m]
d_x	=	Cell length [m]
d_y	=	Cell height [m]
V_{cell}	=	Cell volume [m^3]
a	=	Thermal accommodation coefficient
n	=	Number density [m^{-3}]
N_p	=	Number of simulated particles
W_p	=	Particle weighting
$\bar{\mathbf{u}}$	=	Bulk velocity [m/s]
$\mathbf{u}$	=	Thermodynamic fluctuating velocity [m/s]
$\mathbf{W}$	=	Center of mass velocity [m/s]
m	=	mass of Argon atom [kg]
k_b	=	Boltzmann constant [J/K]
χ	=	Particle collision scattering angle
θ	=	Particle collision scattering angle

[*]PhD Student, Ann and H.J. Smead Aerospace Engineering Sciences

II. Introduction

Thermal nonequilibrium conditions have long been studied over a range of topics, including fusion reactors, re-entry vehicles, and plasma physics. A thermal nonequilibrium condition within a gas typically occurs when a gas is rarefied, and it's Knudsen number (Kn), defined as the ratio of the molecular mean free path λ to a characteristic length, is large.

$$Kn = \frac{\lambda}{L} \tag{1}$$

When $Kn << 1$, continuum dynamics governed by the Navier-Stokes equations can be assumed. A continuum suggests that local thermodynamic fluctuations are small compared to flow gradients, and thus, local thermodynamic equilibrium us assumed. Continuum dynamics also assume the no-slip condition, where the fluid velocity goes to zero at a solid wall. These assumptions, however, break down for rarefied flows where $Kn > 1$. In this regime, particle collisions and energy exchanges are not frequent, resulting in non-negligible local gradients in thermodynamic properties. Additionally, due to large intermolecular spacing, the no-slip condition is no longer true.

For this reason, the Navier-Stokes equations are no longer valid, and we must turn to the Boltzmann Equation, seen in Eqn 2, as a governing equation. The Boltzmann equation describes the particle dynamics of a thermodynamic system and can resolve nonequilibrium states. The dimensions of the Boltzmann equation include three spatial dimensions, three velocity dimensions, and one time dimension. The unknown variables of this equation are the velocity distribution function ($f(C)$), and a number density (n), and thus, can be thought of as a conservation law for particles in particle and velocity space. The left had side of the equation governs changes to n and $f(C)$ due to, from left to right, temporally unsteady processes, advection through physical space, and acceleration through velocity space. The right-hand side is known as the collision integral, and captures changes to n and $f(C)$ due to particle-particle collisions.

$$\frac{\partial(nf)}{\partial t} + C \cdot \frac{\partial(nf)}{\partial r} + a \cdot \frac{\partial(nf)}{\partial C} = \int_{-\infty}^{\infty} \int_{0}^{4\pi} n^2 \left[f(C')f(Z') - f(C)f(Z) \right] g\sigma d\Omega dZ \tag{2}$$

A popular method to solve the Boltzmann Equation though the Direct Simulation Monte-Carlo (DSMC) method. In this method a probabilistic Monte Carlo simulation of virtual fluid particles, each with a position and velocity, is performed. Particle motion and collisions are decoupled over a time step Δt that is smaller than the local mean free time. There are multiple models to determine particle-particle collision pairs and frequency, with a popular approach being Bird's No-Time Counter (NTC) method [1]. Collision models, such as the Variable-Hard-Sphere (VHS) or Variable-Soft-Sphere model, then determine the resulting post-collision velocity using conservation of momentum and energy. The collision results in a transfer of energy between particles and allows for local gradients in the fluid to decay and for the fluid to reach equilibrium.

In this study, a DSMC code is created and applied to several test cases involving thermal nonequilibrium. First the methodology behind the DSMC code is explained in detail. Finally, the DSMC code is applied over test problems and the results are discussed. One is a shock tube experiment, with Argon (Ar) as it's working fluid. The second is a confined volume of Ar with a heated wall. Gas temperatures and computed as a function of time, and the effects of thermal accommodation coefficients and particle collision parameters are discussed.

III. DSMC Methodology

Within a DSMC framework, a domain is first discretized into cells. The cell size is expected to be smaller than the mean free path, λ, of the gas. Fluid particles are introduced into the domain and are assigned position and velocity. Within a simulation timestep, particles travel in a straight line according to their velocity vector. If a particle encounters an object or wall boundary condition, the particle is reflected from the surface and continues its propagation until the end of the timestep. After particle movement has concluded, an algorithm is run to determine which particles within a cell are to be flagged for collision. In this code, the No-Time-Counter (NTC) method as described by Bird [1] is implemented to accomplish this. This method functions by randomly choosing particle pairs within a cell and computing the probablity that each pair collides as a function of their relative velocity magnitude, g, and their collision cross-section, σ. Further details on this method can be found in Section 6.2.3 of Boyd [2]. A verification of the NTC method is discussed in section III.A. Once a particle pair is selected to collide using the NTC method, a collision model is used to compute the post-collision velocities. In this code, the Variable-Hard-Sphere (VHS) is used. This model functions by varying the total collision cross-section, σ, of the particle pair flagged for collision as a function of their relative velocity g. This variability in σ results the VHS model having characteristics closer to experimental measurements compared to the simplistic hard-sphere model. Further details on this method can be found in Section 6.3.1 of Boyd [2]. It should be noted that the VHS model approximates elastic collisions. Internal energy modes are not considered.

A. No-Time-Counter Method

In order to determine which particles within a cell are to collide, the NTC method was implemented in this DSMC formulation. To verify the implementation of the NTC method, as test problem was run to compute the simulated collision rate per particle, ν_{sim}. The expected value of ν_{sim}, pulled from example 6.1 in Boyd [2] can be seen below in Eqn. 3. Under the simulation parameters described in [2], a Collision Rate Fraction of near 0.0250 results in $\nu_{sim} = 1.759 \times 10^5$collisions/particle/sec.

$$\nu_{sim} = \frac{\langle \text{Collision Rate Fraction} \rangle}{\Delta t_{\text{DSMC}}} = 1.759 \times 10^5 \text{collisions/particle/sec} \tag{3}$$

An identical simulation case to example 6.1 was created using the DSMC code. A cell with $N_p = 20000$ simulated particles was initialized, and the NTC algorithm was implemented to determine the number of particles that were to collide. At each iteration, the total number of expected collisions computed by the NTC algorithm was stored and both ν_{sim} and the Collision Rate Fraction were computed. The Collision Rate Fraction was then plotted as a function of iteration count, in order to reproduce Figure 6.6b from [2]. Both the plot from [2] and the resulting plot from this study can be found in Figs. 1 and 2. It is observed that the correct value of the Collision Rate Fraction near 0.025 was recovered. This suggests that the NTC algorithm in the current DSMC formulation was implemented correctly.

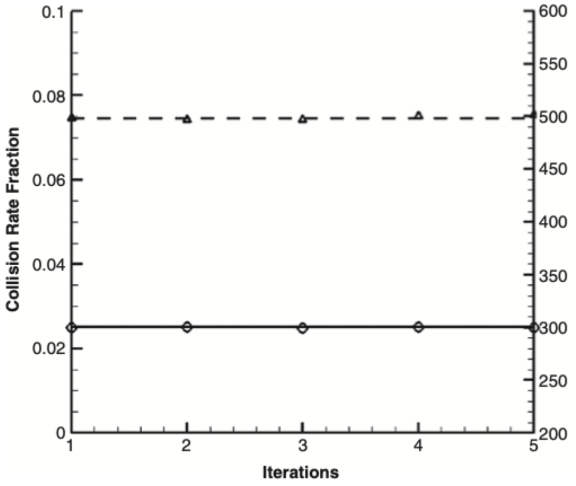

Fig. 1 Instantaneous collision rate fraction for Argon with $N_p = 20000$ pulled from Figure 6.6 in [2]

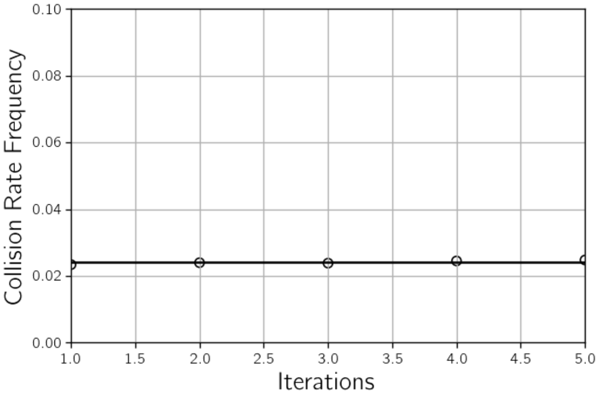

Fig. 2 Instantaneous collision rate fraction for Argon with $N_p = 20000$ from the created DSMC code.

B. DSMC Procedure

The general workflow of the DSMC code is described below. The DSMC code was written in python.

1) User specifies the domain size, number of computational cells, boundary conditions, computational timestep, simulation time, and initial condition of the gas.
2) A computation grid is created and populated with particles according to the user-specified initial conditions.
    1) A global particle weighting (W_p) and value of $< \sigma_T g >_{max}$ is assigned to each cell.
    2) Each particle is randomly placed within the domain and is assigned a position and velocity.
3) The DSMC algorithm is looped for the total specified simulation time.
    1) Particles enter the domain from any relevant boundaries
    2) Particles are propagated along their velocity vector for the specified timestep.
        1) If the particle hits a boundary, the particle reflects from the wall according to the user-specified thermal accommodation coefficient and continues along its new velocity vector

for the remainder of the timestep.

3) Particles positions are updated at the end of their propagation and assigned to the cell they lie within.
4) A NTC method is applied to each cell. This determines which particles will collide within a cell.
5) A VHS model is used to update the velocities of particles that have collided.
6) Cells are sampled to compute quantities such as temperature and number density.
7) If there is total simulation time remaining, loop back to step 1 of the DSMC algorithm.

This methodology was implemented in python. A particle class was created to store velocity and position information in an organized manner. Similarly, a cell class was created to store cell-based information (e.g., W_p, x and y coordinates, $< \sigma_T g >_{max}$, etc).

C. Simulation Setup - Shock Tube

The first test problem to apply the DSMC code towards is a shock tube, following example 6.2 in Boyd [2]. In this example, a shockwave forms within a computational domain that is one cell in height with specular walls on the top and bottom boundaries. This results in two states left and right of the shock, referred to as states one and two. As particles cross the shock into a different state, particle-particle collisions will rapidly cause the particle to come into equilibrium with the new state. For this example Argon (Ar) is used as the working fluid. The final grid had 400 cells in the x-direction and 1 cell in the y-direction. Additional relevant parameters of the simulation can be found below.

m	$6.6335209 \times 10^{-26}$ kg
L_x	0.08 m
L_y	0.08 m
dx	0.0002 m
dy	0.0002 m
λ_1	0.0008 m
M_1	9
M_2	0.456
T_1	300 K
T_2	7856 K
ρ_1	1.069×10^{-4} kg/m^3
ρ_2	4.123×10^{-4} kg/m^3
N_{cells}	400

Table 1 DSMC Problem 1 Simulation Parameters

The computational domain was initialized with both states and run until the total simulation time has reached 25 times the average collision time of state one. This ensures that the shock has had sufficient time to reach a steady-state profile. As specified in the example, the viscosity law exponent, ω, was chosen to be equal to 0.7, and each cell was initialized with $N_p = 1000$ simulated particles. The expected steady state profile of the solution from Boyd [2] can be seen below in Fig. 3.

Fig. 4 shows the steady state output of the created DSMC code. Quantities of gas density and temperature were normalized between states and plotted. Expressions for cell temperature can be found in Section VI. We note far downstream and upstream of the shock, states one and two recover the expected initial conditions from the simulation parameters. The shock width is estimated to be around $10\lambda_1$, where λ_1 is the mean free path corresponding to state one. Looking back to the expected solution for the $\omega = 7$ case, the shock width is also approximately $10\lambda_1$. This suggests that the created DSMC code is in agreement with example 6.2.

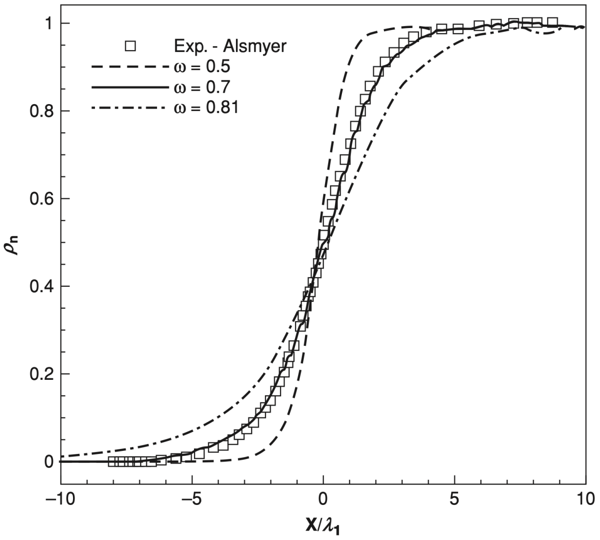

Fig. 3 Normalized density from example 6.2 [2]

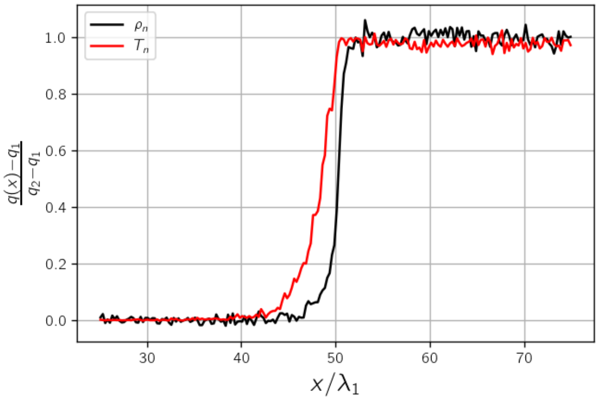

Fig. 4 Normalized density and temperature from the created DSMC code for $\omega = 0.7$.

D. Simulation Setup - Confined Volume

The second test problem to apply the DSMC code towards is a confined volume. In this example, gas is kept in a confined volume with a bulk temperature T that does not equal the wall temperature T_w. It is expected that through particle collisions with the wall, energy will be transferred between the wall and the particles, and through particle-particle collisions, energy will be transferred throughout the domain. Analytically, we expect the velocity distribution of the particles in this domain to approach a Maxwell-Boltzmann distribution at the temperature of the wall T_w. For this example Argon (Ar) is used as the working fluid. The final grid had 20 cells in both the y- and x-directions. Additional relevant parameters of the simulation can be found below.

m	$6.6335209 \times 10^{-26}$ kg
L_x	0.004 m
L_y	0.004 m
dx	0.0002 m
dy	0.0002 m
a	1
λ	0.0008 m
T	300 K
T_w	800 K
ρ	1.069×10^{-4} kg/m^3
N_{cells}	400

Table 2 DSMC Problem 2 Simulation Parameters

The computational domain was initialized with particles sampled from a Maxwell-Boltzmann distribution at $T = 300$ K and run until the velocity distribution function (VDF) reaches a steady state (approximately 40 timesteps). The viscosity law exponent, ω was chosen to be equal to 0.7, and each cell was initialized with $N_p = 1000$ simulated particles. The thermal accommodation coefficient a was selected to be equal to one, resulting in fully diffuse collisions. More discussion on particle-wall reflections can be found in section V.B. The VDF and temperature of the domain at various timesteps can be seen below in Figs. 5, 6, 7, and 8.

We note that at the first timestep, the particle VDF clearly does not match the expected VDF and the mean temperature in the domain is around 300 K. However, looking at the edges of the domain, we see that after one timestep, a few

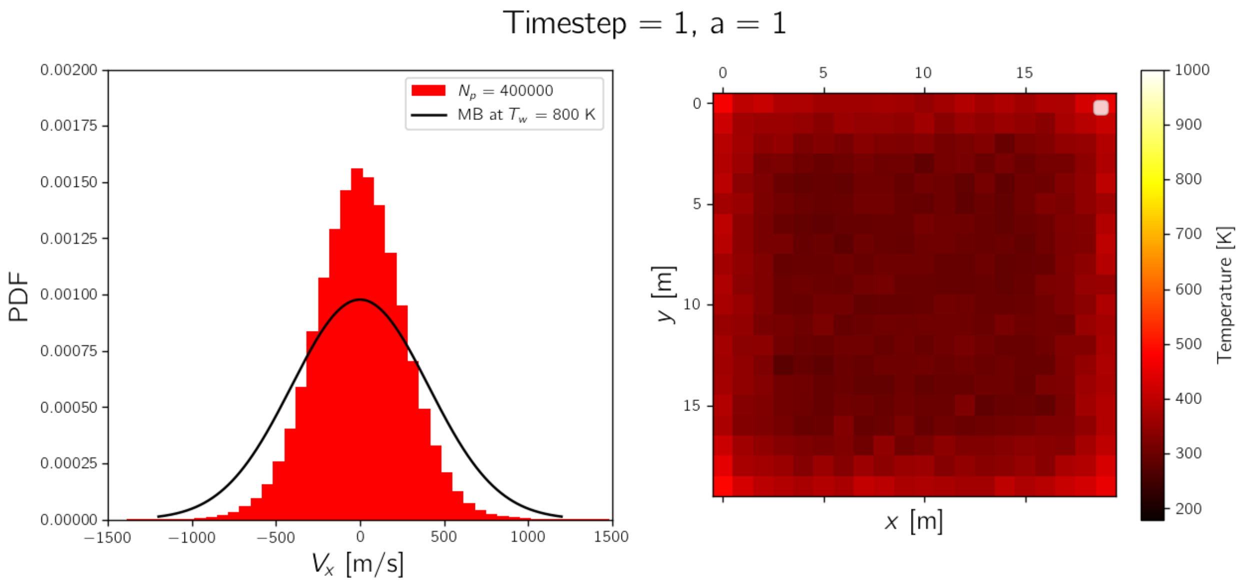

Fig. 5 Temperature and velocity histogram of the confined volume at $t = 1 \times \Delta t$

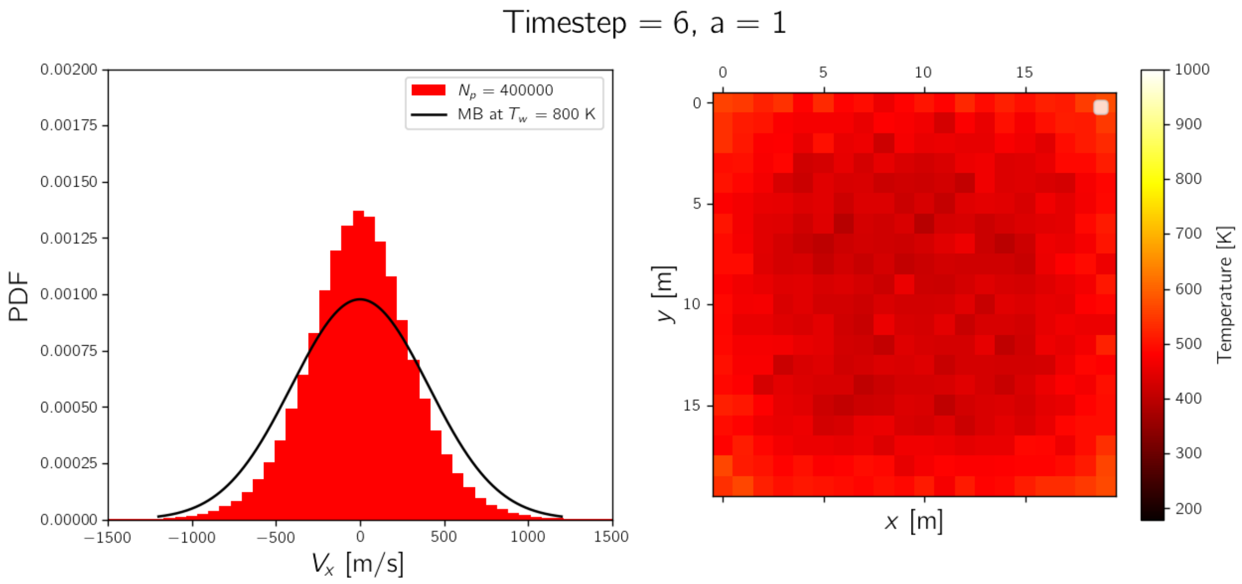

Fig. 6 Temperature and velocity histogram of the confined volume at $t = 6 \times \Delta t$

particles have already collided with the wall and have increased in temperature. At timesteps 6 and 11, we see again a gradient in temperature within the domain, with the hottest portion of the domain being near the wall. Additionally, the VDF has drastically relaxed from its initial state and is approaching the analytical distribution. Finally, at 40 timesteps, the domain has had sufficient collisions such that it has reached equilibrium with the wall temperature. The VDF is well-matched with the analytical distribution at T_w = 800 K. This suggests that the DSMC code has been correctly implemented to capture the effect of wall collisions.

IV. Parameter Space Exploration - Shock Tube

The parameter space for the shock tube was explored to better understand the dynamics of the shockwave and it's sensitivity to the state of the gas. For the first study, the Knudsen number of the gas was modified. For the second study

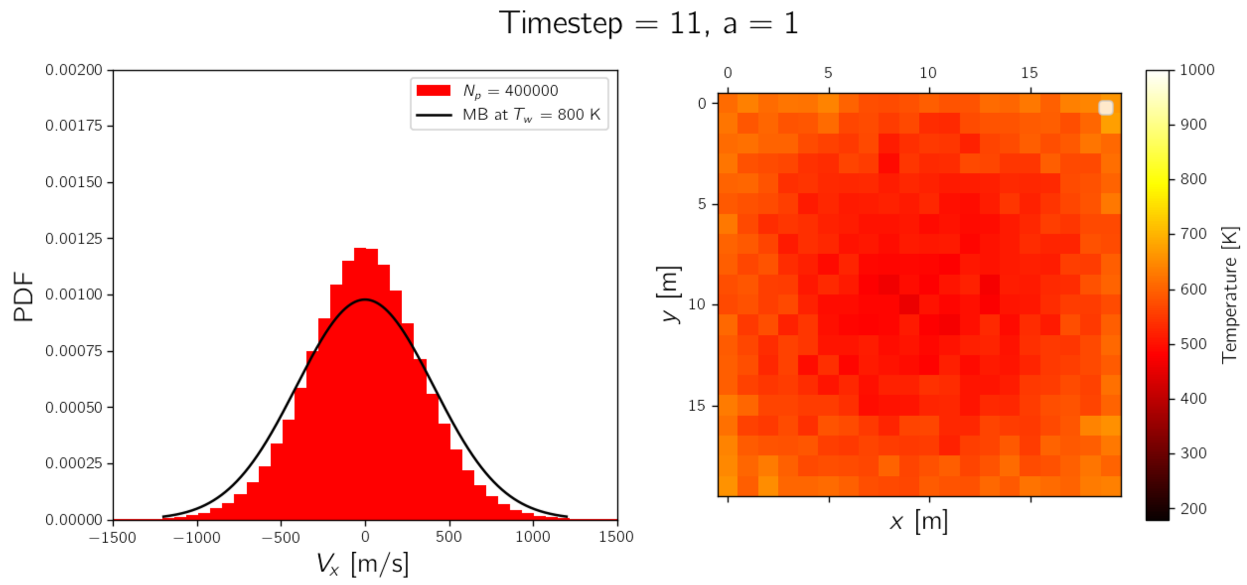

Fig. 7 Temperature and velocity histogram of the confined volume at $t = 11 \times \Delta t$

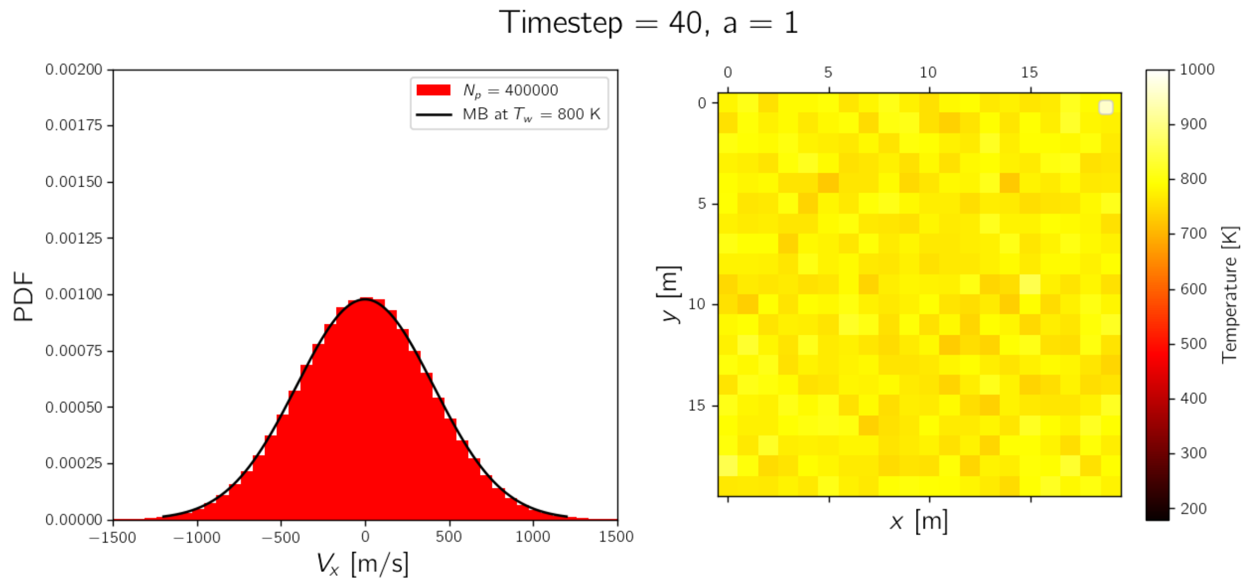

Fig. 8 Temperature and velocity histogram of the confined volume at $t = 40 \times \Delta t$

the incoming Mach number, and resultantly the downstream mach number, were modified.

A. Variable Knudsen Number

From equation 1, the Knudsen number is a function of the mean free path, λ, of the gas. Note that λ is proportional to the density ρ of the gas. As a result, scale factors on the values of ρ will result in various Knudsen numbers. The example case from section III.C had a Knudsen number approximately equal to $Kn \approx Lx/(100dx) = 4$. Two additional cases were run with identical simulation parameters beyond a Knudsen number of $Kn = 2$, and $Kn = 8$ attained by modifying the pre- and post-shock densities ρ_1 and ρ_2, seen in Fig. 9.

It is observed that the Knudsen number is directly proportional to the thickness of the shock. From example 6.2, it is observed that the thickness of the shock, defined as the length required for a gas quantity to change from state 1 to 2, is

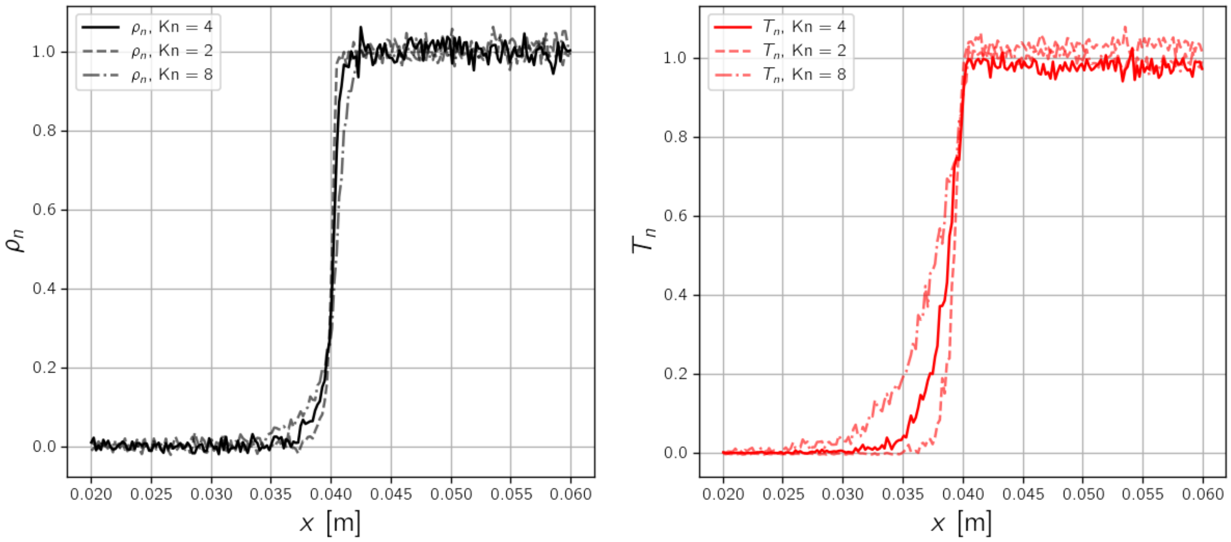

Fig. 9 Plots of normalized density, ρ_n, and temperature, T_n, through the shock at varied Knudsen numbers.

0.010 m. In doubling the value of Kn, seen as the dot-dashed lines in Fig. 9, we observe that the shock thickness is significantly larger, estimated to be near 0.015 m in length. This is best seen on the temperature plot. In halving the value of Kn, seen as the dashed lines, we observe that the shock thickness is smaller, estimated to be 0.005 m in length.

These results make sense as we know that the Knudsen number is effectively a measure of how frequent particle collisions are. When the Knudsen number is doubled, particles in the shock tube share information and collide with other particles less frequently. Particles crossing the shock need to travel a farther distance to collide with particles in the opposite state, and reach equilibrium, thus, increasing the thickness of the shock. Conversely, when the Knudsen number is reduced, particles are able to collide and share information more frequently in the shock tube. An individual particle traveling from state one to state two needs to travel a shorter distance before enough collisions have occurred to bring it in equilibrium with state 2. Thus, the shock thickness is reduced. One can imagine that as the Knudsen was increased, the shock thickness would continue to increase in length, until upon reaching the free molecular limit, the shockwave would not form at all. If the Knudsen number was reduced, one would see the shock thickness decrease until, at the inviscid limit, the shockwave would be infinitely thin.

B. Variable Mach Number

The compressibility of the gas can be quantified through the Mach Number, M, defined as the ratio of the bulk velocity to the speed of sound of the fluid. Notably, through analytical normal-shock relations, the downstream conditions of a gas through a normal shock can be completely characterized given the upstream conditions, and the gas properties. By modifying the bulk velocity of state 1, the mach number of state one, and thus, the mach number of the state downstream of the shock, can be predicted.

The example case from section III.C had an incoming Mach number of $M_1 = 9$. Summarized in table 1 this resulted in a downstream $M_2 = 0.456$. For this study, incoming mach numbers of $M_1 = [4, 13]$ were explored, and the shock width was calculated by plotting quantities of gas temperature and density across the shock. Using normal-shock relations, the downstream Mach number was computed to be $M_2 = [0.589, 0.450]$ respectively. Results of this study can be seen below in Fig. 10.

It is observed that changes to the freestream Mach number had little effect on the shock width for this problem As observed in Fig. 10, comparing Mach numbers of $M = [4, 9, 13]$, the width of the shock in viewing density is mostly constant. In viewing temperature, a small change in shock width is observed of around 0.001 m, but it is close to being negligible. From these results, one can conclude that the strength of the normal shock, quantified by the freestream Mach number M_1, has little affect on the thickness of the shock. While the downstream conditions are very different between the cases, the physical actions and timescales governing the transfer of information between the states is unchanged.

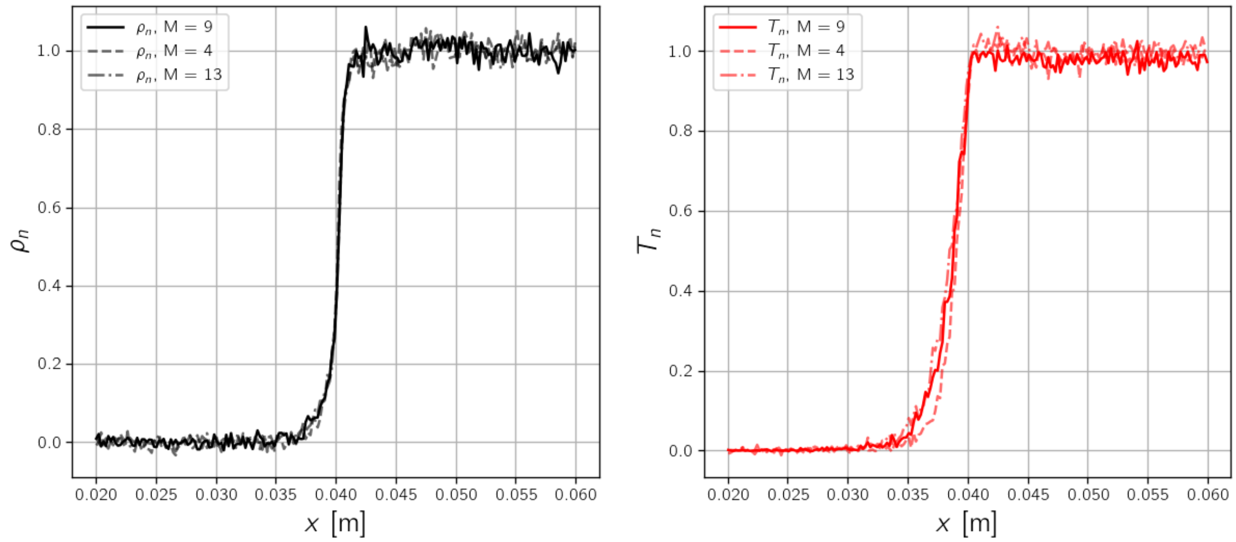

Fig. 10 Plots of normalized density, ρ_n, and temperature, T_n, through the shock at varied Mach numbers.

V. Parameter Space Exploration - Confined Volume

The parameter space for the confined volume was also explored to better understand the sensitivity of the case to particle-particle collisions. For the first study, the Knudsen number of the gas was varied. For the second study the thermal accommodation coefficient was varied. Finally, the effect of particle-particle collisions is investigated. Temperature of each cell was computed using expressions from section VI.

A. Variable Knudsen numbers

Similarly to section IV.A, we note that the Knudsen number will largely govern the rate at which a fluid can return to equilibrium with its surroundings. Density of the fluid in the confined volume was modified to change the Knudsen number, and the experiment was re-run for a set length of time of $50 \times \Delta t$, where $\Delta t = 1.6 \times 10^{-7}$. This length of time was chosen as it was approximately the time required for the case presented in III.D to reach the wall temperature. The average temperature of the volume was plotted as a function of time and the temperature convergence was compared between cases, seen in Fig. 11. For all cases, the wall temperature was taken to be $T_w = 800$ K, and the initial temperature of the fluid was taken to be $T = 300$ K.

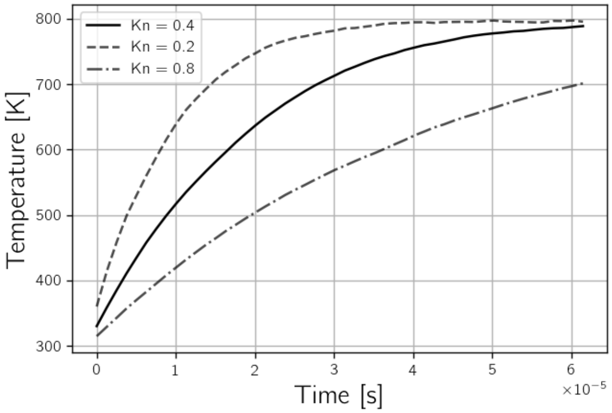

Fig. 11 Plots of average temperature in the confined volume for various Knudsen numbers.

Viewing Fig. 11, we immediately see that a higher Knudsen number case has not converged to the expected temperature. This makes sense as we know that the higher Knudsen number means less frequent collisions, and thus, at the same total time, the fluid has had fewer collisions compared to the example case. Similar behavior is seen with the lower Knudsen number case. The lower Knudsen number means collisions are more frequent, and local gradients in the solution are smoothed out faster. Visually, this is shown with the temperature of lower Knudsen number case converging faster to T_w.

B. Variable Thermal Accommodation Coefficient

Under this DSMC formulation, when a particle hits a wall, reflections can either be *specular* or *diffuse*. A specular reflection results in only the normal velocity component being reversed. For a diffuse reflection, the particle's velocity is re-sampled from a Maxwell-Boltzmann distribution taken at the wall temperature T_w. Only through diffuse reflections can information about the wall temperature be transferred to the fluid. The type of reflection is determined by the thermal accommodation coefficient a. For each particle-wall collision, a random number $R_1 \in [0, 1]$ is generated. If $R_1 > a$, then the reflection is diffuse. Otherwise, the reflection is diffuse. For this study, three separate values, $a = 1, 0.7, 0.5$ were considered, and the average temperature of the system was plotted as a function of time in Fig. 12.

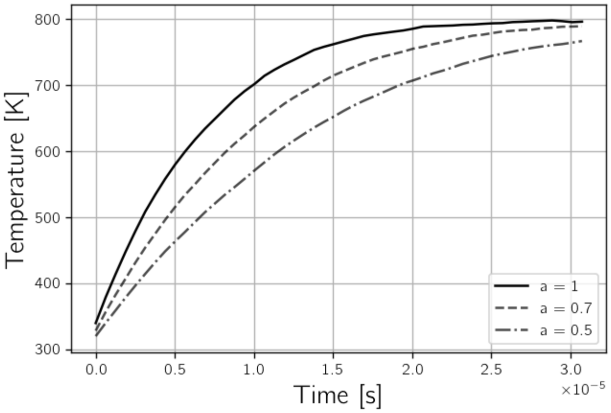

Fig. 12 Plots of average temperature in the confined volume for various thermal accommodation coefficients.

We observe that as a is lowered, the domain requires longer times to reach thermal equilibrium with the wall. At the end of the simulation, for $a = 0.7$, the average temperature is near 787 K and is just below the expected temperature of 800 K. However, for $a = 0.5$, the system is significantly cooler then the expected temperature, at 762 K. This can be explained as with a lower value of a, large percentages of particle-wall collisions are specular. Specular reflections do not change the velocity magnitude of the particle, and therefore, the temperature of the cell, as seen from our temperature calculation in Section VI. Thus, more particle-wall collisions are necessary to bring the particles to equilibrium.

C. Particle-Particle Collisions

Finally, we consider the isolated effect of particle-particle collisions. An case similar to section III.D is run twice: once with particle-particle collisions enabled, and again with particle-particle collisions deactivated. For both cases, $a = 1$. Domain average temperature is plotted as function of time and compared between cases. Results can be seen in Fig. 13.

Clearly, it is observed that the case with no collisions was unable to reach equilibrium with the wall in the allotted time. However, the final temperature is close to the expected temperature, indicating many particles did have a reflection with the wall. A hypothesis for the observed difference would be that, as only particle-wall collisions transfer information about the wall temperature, particles with a low velocity magnitude and an initial position near the center of the domain are unable to collide with the wall in the allotted time, and pollute the average temperature of the domain. When collisions are active, these slow particles have the additional opportunity to collide with particles that have already interacted with the wall. For this reason, we can expect that with collisions the domain temperature will reach equilibrium with the wall at a faster rate. Viewing Fig. 13, this is in fact, observed.

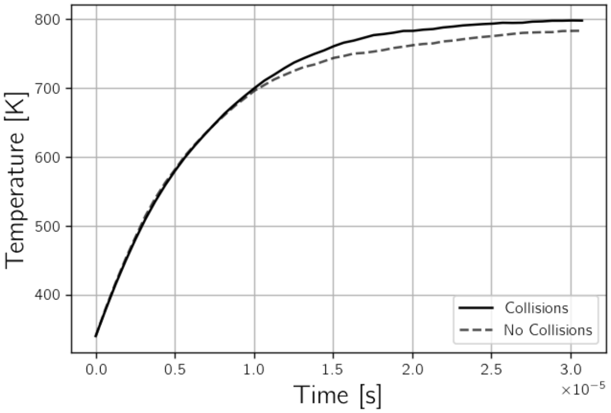

Fig. 13 Plots of average temperature in the confined volume with and without particle-particle collisions.

VI. Conclusion

A DSMC code was created in python for the purposes of simulating thermal nonequilibrium conditions in a shock tube and a confined volume. The code discretizes the domain into computational cells has the ability to vary the wall temperature, bulk inflow velocity, reservoir conditions, thermal accommodation coefficient, and number of simulated particles. A No-Time-Counter algorithm was verified and implemented in the DSMC code to compute a simulated particle collision frequency within each cell. A Variable-Hard-Sphere model was also implemented as a collision model to govern particle-particle collisions. The implementation of the NTC algorithm was then validated against a previous study, supporting its correct implementation. Finally, a shock tube and confined volume case were computed. Modifications were then made to the simulation parameters to better study thermal nonequilibrum. For all cases, it was observed that particle-particle collisions push the system to equilibrium.

In the case of the shock tube, it was observed that the Knudsen number was inversely proportional to the width of the shock, with a higher Knudsen number increasing the thickness of the shock. This expected, as at high values of Kn, particles travel a longer distance between collisions and require more time to equilibrate with particles around them. In the case of the confined volume, it was observed that the Knudsen number and thermal accommodation coefficient was inversely proportional to the time required for the system reach equilibrium. This is understandable, as again, higher values of Kn decreases and less diffuse collisions reduce the rate at which information about the wall can be spread through the domain. An additional case was run with particle-particle collisions deactivated. With this case, it was seen that particle-particle collisions act to increase the rate at which the domain reaches equilibrium with the wall. A possible explanation for this behavior was that particle-particle collisions act upon slow moving particles near the center of the domain that would struggle to collide with the wall during the course of the simulation.

Appendix A

For cell volume properties, we denote p as a particle that crosses a cell j. Number density of a cell n_j is calculated simply as the total number of particles that cross the cell divided by the cell volume, seen below as

$$n_j = \frac{\sum p}{\Delta V} \tag{4}$$

We also denote $u'_i = u_i - < u_i >$ as the thermal component of a velocity in the i direction of a particle p, where $< u_i >$ is the expected velocity component in the same direction. The temperature contribution of this particle towards a cell can be calculated as shown in Eqn. 5. Summing this relation for every particle contribution results in the final temperature of the cell.

$$T_j = \frac{m}{3k_b}({u'_1}^2 + {u'_2}^2 + {u'_3}^2) \tag{5}$$

References

[1] Bird, G. A., "Molecular Gas Dynamics and the Direct Simulation of Gas Flows," , 1994.

[2] Boyd, I. D., and Schwartzentruber, T. E., *Nonequilibrium Gas Dynamics and Molecular Simulation*, Cambridge Aerospace Series, Cambridge University Press, 2017. https://doi.org/10.1017/9781139683494.