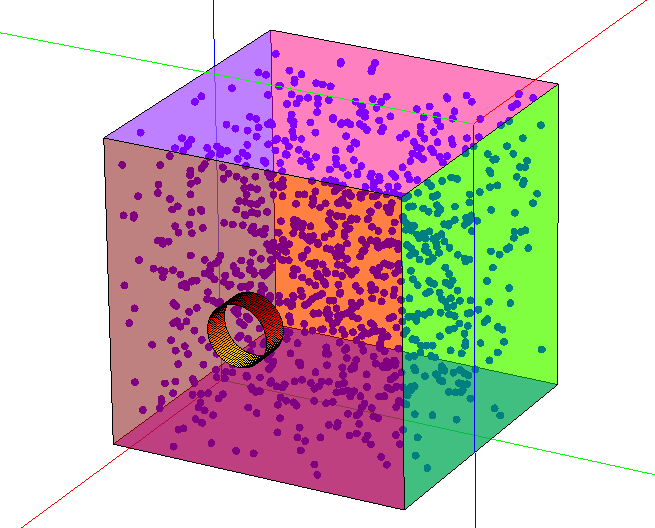


Worcester Polytechnic Institute

Physics Minor Capstone Project

Simulation of Gas Kinetics

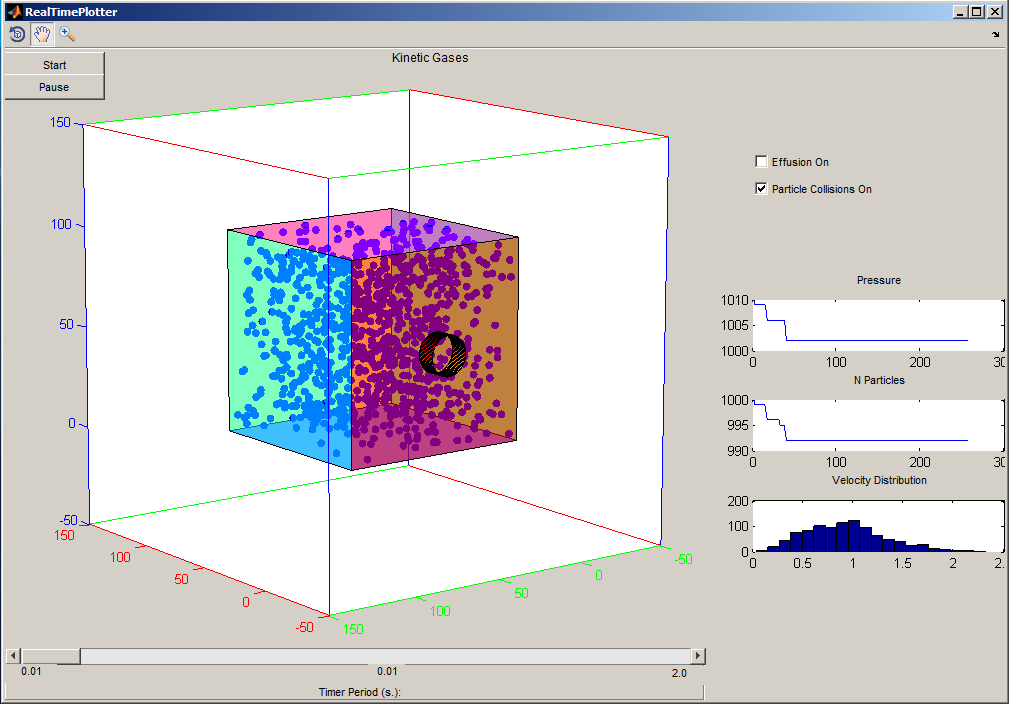
Daniel Fitzgerald



# Abstract

The Kinetic Theory of Gasses developed by Maxwell and Boltzmann relates the emergent macroscopic properties of gasses, such as temperature and pressure, to the fundamental kinetic particle interactions as microscopic levels. One useful way to explore and verify this phenomena is with the aid of computer simulations that calculate individual particle kinetics but can simultaneously visualize the overall behavior of the gas. Such simulators may also be used to verify or predict models numerically.

This paper details one such simulator, written in MatLab, which is capable of calculating the kinetics and interactions of pure idealized gas particles constrained within a box container and visualizing them in real time. The simulator’s design and performance are evaluated and its results are compared to those predicted by the kinetic theory model.



Kinetic Gas Simulator

# Introduction

The Maxwell-Boltzmann distribution for particle velocity derived from kinetic gas theory is the probability density function that, for a gas with certain macroscopic properties, gives the probability that any given gas particle will have a specific velocity. It assumes the components of the three dimensional velocity vectors of particles are normally distributed with a mean of zero, such that the gas has zero net momentum in any direction. Related distributions exist for the kinetic energies, momentum, the components of velocity and momentum in a given dimension, and for the absolute magnitudes of the velocity and momentum. This distribution underlies more common physical relations, such as the ideal gas law, and provides a bridge between the almost ideal Newtonian interactions between individual particles and the statistical relations that emerge from large numbers of those interactions in a gas.

A simulator was developed in Matlab to calculate particle inactions and states over time. The particles are contained in a cuboid box and may escape through a circular hole, shown as a tube/hole coming out of the box. The particles experience elastic collisions with the sides of the container and with each other. The momentums and kinetic energies of the particles are recorded and displayed in real-time plots, as is the pressure of the gas, calculated using kinetic gas theory. These results are compared to the expected behavior from the Maxwell-Boltzmann distribution and the ideal gas law. In addition, the gas particles are allowed to effuse through a circular opening in one side of the container, and their effusion rate is likewise calculated and displayed. This rate is then compared to that expected by the gas effusion equation.

# Computing Environment

There is a variety of programming languages and computational toolboxes available for science and engineering. These include Matlab, Python, Wolfram Mathematica, Processing, and others. For this project, a balance of efficiency, 2D and 3D visualization capability, readability/ease-of-use, and rapid development time was prioritized. As such, Matlab was chosen as the platform for this simulator for its highly efficient matrix pipeline and build-in plotting functionality.

# Kinetic Theory

The kinetic theory of gasses makes several assumptions: particles obey Newton’s laws of motion, they undergo ideal elastic collisions with other particles and any container, no other forces (electric force between particles, the effect of magnetic fields, gravity, etc.) are considered, and the number of particles is large with sufficiently large space between them.

## Particle-Particle Collisions

Completely elastic collision of ideal gas particles in one dimension are governed by two primary laws: the conservation of momentum and the conservation of kinetic energy. For the two particle case, where particle of mass and initial velocity collides with particle of mass and initial velocity , the conservation of momentum states that the total initial momentum after the collision must equal that before the collision.

Similarly, the conservation of kinetic energy states that the sum of the kinetic energies must be constant before and after the collision.

When the masses and initial velocities are known, this system of equations can be solved for the final velocities for each particle (ignoring the trivial case where there is no collision and the final velocities are the same as the initial velocities.)

|  |  |
| --- | --- |
|  |  |
|  |  |

Before

Collision

After

Collision

For collisions between particles in two or three dimensions, this one dimensional relation holds for the components of the velocities along the line through the centers of mass of the particles (common normal, assuming spherical particles with uniform density.) The other components of each particle’s velocity remain unchanged.

The line is derived from the three dimensional positions of the centers of the particles. The vector connecting the centers is

And the unit in the direction of is then

The component of a particle ’s velocity on to this line is given by vector projection

Or in matrix form for three dimensions

The remaining component of the velocity that is perpendicular to is the vector rejection , the difference between the total velocity and the component along .

The final velocity vectors of each particle after a collision are the sum of the unaffected rejection velocity and the new velocity along the line of collision.

## Particle-Container collisions

Idealized collisions between particles and the walls of the container also obey the conservation of momentum and the conservation of kinetic energy. When the container is fixed, such that its velocity momentum and kinetic energy are always zero, both laws will the same solution for the resulting velocity of the particle: the component of velocity normal to the wall will be inverted after the collision. For a wall with normal along , the component of a colliding particle’s velocity will be

# Pressure

For particles bouncing between two walls, the impulse on a wall from a particle “bounce” collision is twice the particle’s momentum in the direction normal to the wall (assuming elastic collisions and immobile walls.)

Assuming no other interactions, every particle will collide with the wall at regular intervals as it bounces back and forth. This “round trip time” is given by the round trip distance, which is twice the separation between the walls, divided by the particle’s speed in the direction normal to the walls.

The average force on each wall due to a single particle is then the impulse per particle normalized by the time between applications of that impulse (time between collisions with the wall.)

Scaling by particles between the walls the total average force on a container wall is

Where the average square of the speeds of the particles is the sum of the square of their speeds in the normal direction of the wall normalized by the number of particles.

The pressure of a gas in terms of the kinetics of it’s constituent particles is derived from its definition as Force per unit area. For any wall of the container, the pressure at any given time is equal to the average force distributed over its surface area.

For a rectellinear container, the product of the area of a wall and the distance to the opposite wall is the volume of the container.

The pressure then simplifies

Assuming the particles have velocities in random directions, all components of their velocity have the same average, so all of their squares are equal.

The square of a particles speed is then, on average, three times the square of its speed along given any dimension.

The average pressure in the container can then be expressed in terms of the sum of the squares of particle speeds scaled by the mass of each particle normalized by the container volume.

This is equivalent to the standard formula for gas pressure derived from kinetic theory but is more efficient to compute for a discrete set of particles with known velocity components in each direction.

# Effusion

Effusion of a gas through a single hole in its container in to a vacuum is considered.

The rate at which particles escape is the average particle flux through the area of the hole.

Where is the number of particles escaping per unit time, is the average particle flux in particles per unit area per unit time and is the area of the hold in unit areas.

The rate of change of the number density of the remaining particles is the inverse of this escape rate.

The average particle flux is the product of the average number density, , and average particle speed in the direction normal to surface, . (similar to the calculation of force on a container wall for the derivation of pressure above)

The rate of change of the number of particles remaining in the container is then

Using the Maxwell-Boltzmann distribution for particle velocity components

The rate of change of the number of particles is then

For this First order ODE, the number of particles remaining at time is of the form

Where is the initial number of particles. This is assuming constant container volume, effusion hole area, and temperature of the remaining particles. The energies of the particles remaining does not change, so their average velocity does not change. The rate of effusion is the inversely proportional to the number of remaining particles, so the number of remaining particles exhibits classic exponential decay over time. Conversely, the time to reach a given pressure from an initial pressure is given by

# Implementation

The simulation is run over discrete timesteps. For computational simplicity, all physical quantities that are constant over the duration of the simulation are computed with a value of 1 unless otherwise noted. Thus the simulation is not numerically accurate for any particular unit system, but relations between quantities will still hold. At each time step, the following is calculated:

1. Movement update: Particle positions are updated by adding the product of the timestep (1 unit time) and their velocity (unit distance per unit time) to their previous position.
2. Effusion: If effusion is turned on, any particles colliding with the hole are removed
3. Particle-Particle Collisions: Every combination of remaining particle pairs (N choose 2) is checked for the collision condition.

Where is the Euclidian distance between the centers of the particles and is the sum of their radii.

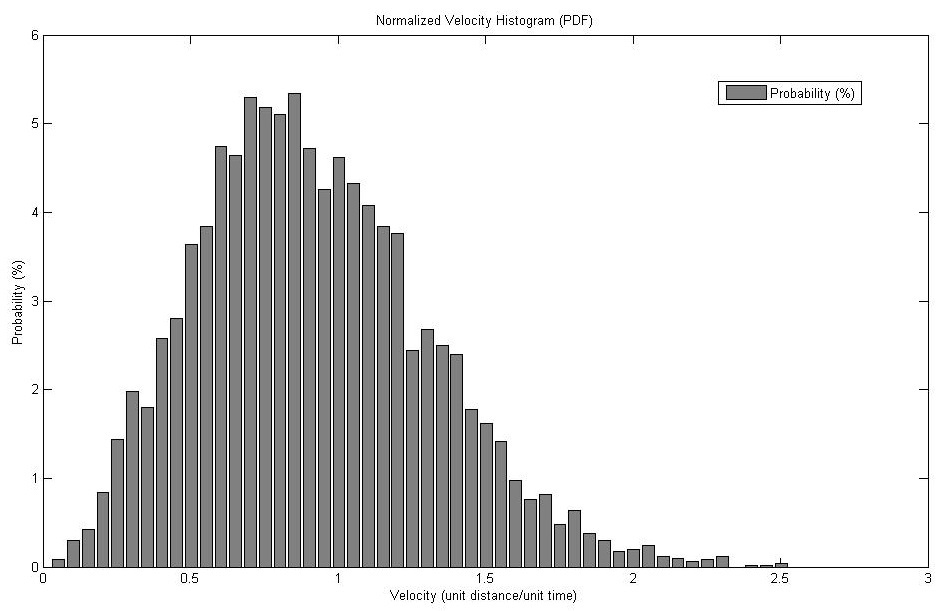
For all pairs of particle, and , that are in collision:

* 1. Calculate the line between the center positions of A and B as
  2. Divide this line vector by it’s normal to get the unit vector from in the direction of .
  3. Project the velocities of each particle on this unit vector to find the components along the line. This is the normalized line vector times the dot product of itself and each velocity.
  4. Find the remaining (rejected) components as the differences of the original velocities and the projected components.
  5. Find the norm of each projected velocity. Because this method of finding scalar projected vectors only produces magnitudes, the signs are adjusted relative to the line vector.
  6. Calculates the corresponding components of the momentums of each particle. These are the particle masses times the signed magnitude of their projected velocities.
  7. Calculate the new projected velocities after collision by the equation derived above.
  8. Calculate the final post-collision velocities as the sum of these new projected components and the old rejected components.

1. Particle-Wall Collisions: Any remaining particles that are in collision with the container have their appropriate velocity components inverted.
2. The particles are rendered as a 3D Scatter plot.
3. The plots of the number of particles and pressure vs. time are updated.
4. The histogram of particle velocity magnitudes is updated.

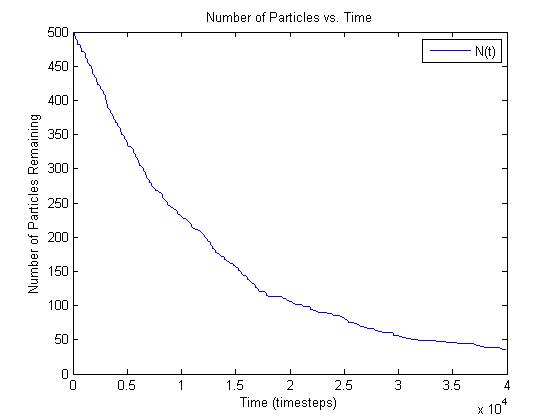
# Results

After running the simulation for a few hundred time steps with particle collisions enabled, the histogram of particle velocities settles is shown.



Normalized Velocity Histogram for 5000 particles after 5 hours of simulation

For a different run with 500 particles in which particle effusion was also enabled, the number of particles over time is shown.



Number of remaining particles over time

# Analysis

The distribution of particle speeds fits an approximate Maxwell-Boltzmann distribution.

The number of particles remaining experiences an exponential decay over time.

# Discussion

One observation of interest is that the distribution of particle speeds when effusion is enabled does not as closely match the Maxwell Distribution. This is because those particles with higher velocities, which are represented by the “tail” of the distribution, are the most likely to escape through the hole because they have a greater rate of collisions with the walls. This tends to eliminate the tale of the distribution, making it more normally distributed.

A potential inaccuracy was observed as an unexpected rise in the velocity distribution for very high velocities, making the last few highest velocity bins higher than those in the preceding tail. One explanation is that interactions for higher velocity particles are more likely to be “missed” between time steps. Only those collisions that occur at each discrete time step are detected, so particles effectively pass through each other in the time between updates. If one particle would have intersected another during this time, but their positions at the next time step are not intersecting, then this collision will be missed. This is more likely to happen for higher velocity particles, because they travel a farther distance per step over which there are potentially more particles. These particles thus experience less collisions than they should. Particle interactions produce the Maxwell Distribution by effectively slowing down fast particles and speeding up slow ones until an equilibrium distribution of velocities is reached. These high velocity particles interact less and thus have less opportunities to transfer their momentum to slower particles, so they end up keeping more of their momentum than they should and do not follow the expected distribution.

# Conclusion

A program was implemented in Matlab to simulate the kinetics of ideal gas particles in a container. The particles and container are visualized and simulated in real time. The particles experience elastic collisions with the walls of the container and each other, and their velocities reach the expected Maxwell-Boltzmann distribution over time. A representative histogram of this distribution is shown. In addition, the gas particles may escape through a hole in the container, and the number of particles and pressure of the gas are plotted over time, both of which exhibit expected exponential decay. The simulator demonstrates the emergence of these statistical gas behaviors from individual Newtonian particle interactions.

# Future Work

Real-time visual simulator are valuable for providing immediate and understandable indications on the effects certain parameters have on a system and its properties. The more flexible the simulator is in the parameters that are settable and properties that are visualized, the more useful it becomes. As such, adding more controls to this simulator would be the first step for future improvement. This might include:

* A slider bar to control the number of particles in the simulator. New particles would be added with random velocities based on the current average temperature.
* A slider bar to control the temperature of the gas. The velocities of the particles could be scaled to maintain the appropriate distribution by an algorithm similar to histogram equalization.
* Visualization of kinetic temperature in addition to pressure (although they are both proportional to the kinetic energy, which, for a fixed number of pure gas particles, is proportional to the sum of the squares of their velocities.)
* Overlays of the expected temperature, pressure, and particle number density over time based on ideal gas formulas.
* Visualization of effusing particles (They are currently deleted as soon as they escape the container through the hole.)
* Indication of the speed of particles by their color.
* Support multiple types of gas with different masses and particle sizes. This feature would allow for calculation of partial pressures and would serve to demonstrate the relation between particle mass and velocity in a gas mixture of homogeneous temperature. (All particles in a gas have the same average kinetic energy, so lighter particles must have much greater velocities.)
* Particle visualization as true spheres rather than pixel markers. Although this is much more computationally intensive (and was attempted), rendering the particles as sphered gives a much better representation of their position and arrangement in 3D space.
* The overall accuracy of any such simulator increases with the number of particles simulated and the discretization of time and space used (smaller timesteps.)

There are of course numerous computational optimizations to be made to the code and aesthetic/functional improvements to the graphical user interface (GUI.)

Finally, further comparison to empirical and analytic results is needed to further verify the accuracy of the program.

# References

"Kinetic Temperature." *Hyperphysics*. Georgia State University, n.d. Web. 04 May 2014.

Pisacane, Vincent L. *The Space Environment and Its Effects on Space Systems*. Reston, VA: American Institute of Aeronautics and Astronautics, 2008. Print.

Weisstein, Eric W. "Maxwell Distribution." *Wolfram MathWorld*. Wolfram Research, n.d. Web. 04 May 2014.