Simulating the Relationship Between Pressure and Other Variables in a 3D Environment

Luke Carlson
Trevor Day School
March 11, 2013

Introduction

The Ideal Gas Law describes the characteristics of ideal gas in a container. Often written as $PV = nRT$, this law displays the relationship between Pressure, Volume, Temperature, moles of the particle, and the universal gas constant in a system. The Ideal Gas Law can be derived from combining three other gas laws: Boyle's Law, Charles's Law, and Avogadro's Law.

Boyle's Law postulates that in a system with uniform temperature, the pressure of an ideal gas is inversely proportional with volume of the gas. Thus, the pressure times the volume is equal to a constant value in the system, often shown as $PV = k$ (where k is the constant). Since the constant is the same no matter the circumstances in the system, the law can be used to relate changes in pressure or volume as $P_1V_1 = P_2V_2$ (where 1 indicates the initial and 2 is the final state).

Charles's Law states that in a system with uniform pressure, the temperature is inversely proportional to the volume of the container holding the ideal gas ($V \propto T$). Since this law applies to any variation in volume or temperature, it can be written as $\frac{V_1}{T_1} = \frac{V_2}{T_2}$

Avogadro's Law declares that in a system with a constant temperature and pressure, equivalent volumes of the same ideal gas will contain an equal number of particles. Mathematically, the relationship can be shown using $\frac{V}{n} = k$ (where k is the constant in the system).

These three laws can be combined mathematically to create $\frac{PV}{Tn} = R$ (R is a constant in the system). When rearranged, this creates $PV = nRT$ or the Ideal Gas Law.

Purpose

In this lab, I set out to create a 3D simulation of ideal gas particles in a cubic container in order to experimentally determine the pressure of the gas based on given circumstances. From there, I planned to explore the relationship between pressure and volume as well as pressure and number of particles.To produce a simlation, a replication of a real world circumstance using programming of a gas particle, it is first necessary to understand exactly how particles affect the pressure of a system. Pressure is the amount of force over a specific area, also written as $Pressure = F/A$. Force can also be described as change in momentum over change in time: $F = \frac{\Delta p}{\Delta t}$. The change momentum of a single particle equals its mass multiplied by its change in velocity: $\Delta p = m\Delta v$. Since there is more than one particle in a system, the entire change in momentum is the combined change in velocities of each particle that hits the specified area. Thus, the following formula can be used to determine total force:

$$F = \frac{2m * \sum_{i=1}^{n} v}{\Delta t}$$

Where n is the number of collisions and v is the velocity of the particle hitting the wall. Since the change in velocity is double the initial velocity, the 2 can be placed outside the summation along with the mass.

Once the force has been computed using the momentum of the particles, the pressure can then be determined with the initial formula $P = F/A$.

Hypothesis

Increasing the number of particles in the simulation will yield a greater pressure. Furthermore, increasing the volume of a container will decrease the pressure in the system.

Method

To create my simulation, I used Processing, which is an open source program designed for programming visual projects. Processing uses a form of the Java programming language

Computing Pressure

I began constructing my simulation with '3D Balls Bouncing' (a project from Open Processing) as a base. Starting off with a system that could already handle 3D collisions of small objects inside a container was necessary [see Failed Methods, p.3]. From there, I created small spheres with the properties of an ideal gas. Their speed was roughly determined on a Maxwell-Boltzman distribution and assuming that the most probable speed (V_p) would occur the most often (calculated using the formula $V_p = \sqrt{\frac{2kT}{m}}$. Next, I focused on one wall in the container and tracked each time a particle collided with the area of the wall. I could then compute the change in momentum over the area since I had the masses and velocities of the particles. The simulation did not keep track of total time since initiation, which is equivalent to time, but I knew that aspects of the code were executed every frame. The project ran at a fixed number of frames per second (60) so I designed this formula to figured out the change in time: $total\ time(seconds) = \frac{total\ frames}{frames\ per\ second}$.

I inserted that data into this formula $F = \frac{2m * \sum_{i=0}^{n} v}{\Delta t}$ (see Introduction/Purpose, p.1) to find the total force on the wall. To obtain the pressure, I just divided the answer by the area of the wall.

Testing Pressure vs. Number of Particles

Now that I was able to compute the pressure, I could test my hypothesis by increasing the number of particles in the system and analyzing the pressure readings. I started off with 5 particles and then tried 10, 15, and 20, 50 and 100 particles. I calculated the total pressure of each system once every 5 seconds for 20 seconds.

Assessing the Relationship Between Pressure and Volume

The next step of my experiment involved manipulating the volume of the container while keeping the number of particles constant. In order to do so, I altered one line of code in the box Class:

```
int boxsize = 300;
```

This variable alters the dimensions of the box (currently 300x300x300 pixels). I changed the boxsize to 400, 500, and 600, and calculated the pressure every 5 seconds for 20 seconds.

Failed Methods

I originally designed my own 3D collision system but it was less efficient so a computer could not render as many particles in the simulation. I knew that my approximation for pressure would be further off and I also realized that it would be too time consuming to focus on designing the base of the system when I could use open source alternatives.

Results

My data shows that increasing the number of the particles will increase the pressure of the system. Furthermore, expanding the volume of the container leads to a lower pressure.

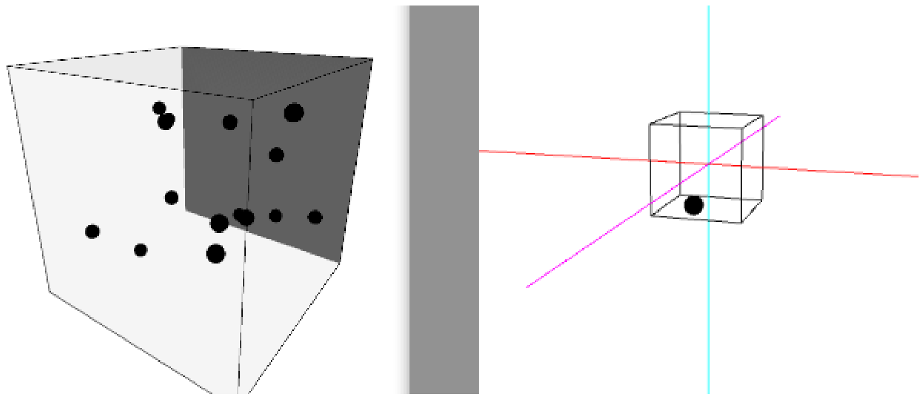

Figure 1: A side-by-side comparison of the final simulations (right) and my initial simulation (left)

Data

Number of Particles	5	10	15	20	50
First Pressure Reading	8.48E-07	2.04E-06	2.92E-06	4.21E-06	1.44E-05
Second	8.70E-07	1.45E-06	3.06E-06	4.30E-06	1.28E-05
Third	8.79E-07	1.31E-06	2.85E-06	4.26E-06	1.15E-05
Fourth	8.90E-07	1.30E-06	2.52E-06	3.32E-06	1.08E-05
Average	8.72E-07	1.52E-06	2.84E-06	4.02E-06	1.24E-05

Dimensions of Box	300	400	500	600	700
First	6.10E-06	4.85E-06	3.29E-06	2.65E-06	2.72E-06
Second	6.41E-06	4.48E-06	3.23E-06	3.13E-06	2.13E-06
Third	6.70E-06	4.37E-06	3.21E-06	3.08E-06	2.87E-06
Fourth	6.46E-06	4.31E-06	3.25E-06	3.01	2.78e-06
Average	6.42E-06	4.50E-06	3.25E-06	2.97E-06	2.63E-06

Analysis

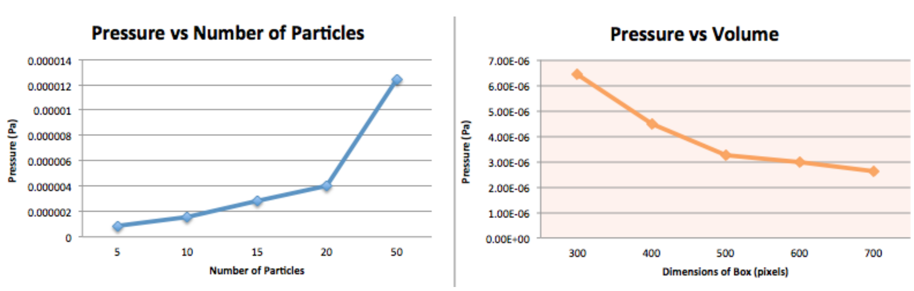

Figure 2: Graphs of Pressure vs # of Particles and Pressure vs Volume

Discussion

In this lab, I created a 3D simulation to explore the relationship between pressure and number of particles in a system. In order to do so, I derived a formula for pressure based on how the particles interact with a specific section of the container. Next, I tested how increasing the volume affects the pressure by changing the dimensions of the square container. The results support my hypothesis: increasing the number of particles will increase the total pressure and increasing the volume decreases the pressure.

The results of my simulation can be used to mathematically display the relationship between these variables (when all other variables remain constant), as $P \propto V$ and $P \propto 1/N$ ($N =$ number of particles). The trend lines on my graphs in the Analysis section show that the relationships were not perfectly proportional (as is postulated by the gas laws). If they were, the trend line would have been Ce^{-1} (where C is a constant) but the difference could be attributed to the fact that my simulation does not perfectly replicate a real world situation.

Improvements

1. Since the simulation could not handle a large number of particles, I had to resort to using an unrealistically low number of particles (e.g. 5 particles vs $5x10^{23}$).
    (a) This also meant that my Maxwell–Boltzmann distribution would not be as precise because if a particle obtained a truly outlier speed, it would greatly affect the results.
2. Increasing the number of trials would create a larger picture of the correlation between the variables. This would not change the results of my trial but the added data would create a larger foundation of evidence.
3. Allowing the variables to change as the simulation runs would be an interesting addition to the project and could provide helpful visuals.

Going Further

My simulation can be altered or tweaked to explore many characteristics of ideal gases. Charles's Law, which argues that increasing the temperature will increase the volume, could be tested using my simulation as well. I would change the temperature of the system, keep the pressure constant, and track the change in volume. Similarly, I could experimentally determine the relationship between temperature and pressure when volume remains constant. If I were to involve another gas in the simulation, I could try to experimentally prove Dalton's law of Partial Pressures, which states that the total pressure of a mixture of different ideal gases in a container is equal to the combined pressures of the individual gases. Dalton's Law can also be displayed as: $P_{total} = P_1 + P_2 + P_3 \ldots$ (where the numbers represent different gases)

Besides using the simulation to provide evidence for these two ideal gas laws, it can be used to understand other ones. If the user accepts that pressure and volume are directly proportional, they can test the Combined Gas Law. This law argues that the constant created within Boyle's Law ($PV = k$) is inversely proportional to the temperature of the system. This can be written as $\frac{PV}{T} = K$ (K is a system constant).

Bibliography

- "Maxwell Speed Distribution Directly from Boltzmann Distribution." Development of Maxwell Distribution. N.p., n.d. Web. 07 Mar. 2013.<http://hyperphysics.phy-astr.gsu.edu/%E2%80%8Chbase/kinetic/maxspe.html>.
- "Language Reference (API) Processing 2." Language Reference (API) Processing 2. N.p., n.d. Web. 07 Mar. 2013. <http://processing.org/reference>.
- "3D Balls Bouncing- OpenProcessing." 3D Balls Bouncing- OpenProcessing. N.p., n.d. Web. 07 Mar. 2013. <http://www.openprocessing.org/sketch/20136>.
- "Kinetic Theory." Wikipedia. Wikimedia Foundation, 03 Apr. 2013. Web. 07 Mar. 2013.
- "The Distribution of Molecular Speeds." ChemEd. University of Wisconsin, n.d. Web. 10 Mar. 2013. <http://chemed.chem.wisc.edu/chempaths/GenChem-Textbook/Kinetic-Theory-of-Gases-\The-Distribution-of-Molecular-Speeds-941.html>.
- "Including Graphics in a LaTeX Document." Including Graphics in a LaTeX Document. University Of Colorado Boulder, n.d. Web. 10 Mar. 2013. <http://amath.colorado.edu/documentation/LaTeX/reference/figures.html>.