Stability of Integrators, Temperature-Pressure Relationship, and Deviation From Ideal Gas in Molecular Dynamics Engine Using Python

Allen Chen, 2024

Abstract:

The purpose of this project was to write a simple molecular dynamics (MD) engine for simulating ideal and Lennard-Jones gas in a fixed box. This MD engine was used to assess Euler and velocity Verlet integrators and the effects of various simulation parameters upon long-term stability of the system, as well as the pressure-temperature relationship and Lennard-Jones gas deviation from ideal gas under various temperatures. This engine was written in Python and used NumPy arrays to store gas particle information. Particles were generated on a uniform grid within a box of specified size and assigned initial velocity using a Maxwell-Boltzmann distribution at specified temperatures. Lennard-Jones (LJ) interactions were modeled using Lennard-Jones 12-6 potential, and particles were integrated using either Euler or velocity Verlet algorithms. The system could optionally be coupled to a Berendsen thermostat for temperature control. Data (time, net energy, net LJ potential, net kinetic energy, temperature, pressure) were periodically outputted to a file while particle positions were stored in Protein Data Bank (PDB) format. To test stability of integration algorithms, 15 picosecond long simulations were run for both integrators using 1, 2, 4, 8, and 16 femtosecond integration timesteps without the effects of a thermostat, and conservation of energy was assessed using standard deviation and drift rate of the total energy of the system. For analysis of pressure-temperature relationship, simulations were run with thermostat using Euler integration at temperatures from 200K to 400K with a 25K increment, and pressure was analyzed in relation to ideal gas. For small timesteps (<4 femtoseconds), drift and standard deviation of net energy in Euler and velocity Verlet algorithms were generally on the same order of magnitude. Larger timesteps (>4 femtoseconds) induced enough error where the integration algorithm no longer simulated accurately regardless of what was used. Pressure and temperature were found to have a linear relationship where the average pressure deviation from ideal gas decreased (0.84% deviation to 0.17% deviation) as simulation temperature increased, which is the expected behavior. This MD engine worked well to run simple simulations of Lennard-Jones or ideal gas under small timesteps, regardless of integration algorithm used.

Methods:

Written in Python, the molecular dynamics engine was built step-by-step by adding additional functionality to the existing code. First, data structures were written to store gas particle coordinates, store gas particle velocities, assign random coordinates given the box size, and output the coordinates in PDB format for visualization using Visual Molecular Dynamics (although this was not utilized for data analysis). For this, NumPy arrays were used instead of Python lists for more efficient handling of data, where coordinates and velocities were stored in three dimensions for each particle.

Next, a basic Euler integrator was introduced to update particle coordinates each step based upon velocity and confine particles to the box, and a Maxwell-Boltzmann distribution was used to initialize particles with an appropriate initial velocity. In this state, the script already serves as an accurate simulation of ideal gas.

The Euler integrator updates particle positions with the following algorithm:

$$x(t + \Delta t) = x(t) + v(t) \cdot \Delta t$$

Where x represents particle position and v represents particle velocity at time t. This integration was applied separately to each degree of freedom.

Maxwell-Boltzmann velocity distributions were generated using the following algorithm:

$$v = \sqrt{\frac{R \cdot T}{m}} \cdot N(0,1)$$

Where v represents particle velocity in one degree of freedom, R represents the ideal gas constant, T represents system temperature, m represents the mass of the particles, and $N(0,1)$ is a normal distribution. These velocities were generated for each degree of freedom for each particle separately.

To simulate Lennard-Jones gas, additional arrays were added to store particle force. By taking the partial derivative of Lennard-Jones 12-6 potential for each degree of freedom for each particle, the force array could be appropriately updated for each particle. A cutoff was also implemented to only calculate LJ potential for particles within a set distance of each other. Each step, this force array was used to update particle velocities before particle velocities were used to update particle positions. Additional global variables were created to calculate and store the instantaneous net energy, instantaneous net Lennard-Jones potential, and instantaneous net kinetic energy for the purpose of assessing total energy conservation. Also calculated were instantaneous temperature and pressure. All of these values were output to a text file.

Lennard-Jones 12-6 potential was modeled using the following formula:

$$V_{LJ}(r) = 4\varepsilon \cdot [(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6}]$$

Where V_{LJ} represents Lennard-Jones potential, r represents intermolecular distance, ε represents the depth of the potential well, and σ is the distance where Lennard-Jones potential is zero.

Particle velocity were updated using the following algorithm:

$$v(t + \Delta t) = v(t) + \frac{f(t)}{m} \cdot \Delta t$$

Where m represents particle mass, v represents particle velocity, and f represents particle force at time t. Velocities were updated separately for each degree of freedom.

Net kinetic energy was calculated with the following algorithm:

$$\Sigma E_k = \sum_{i=1}^{n} \frac{1}{2} m v_n^{\;2}$$

Where ΣE_k represents net kinetic energy, n represents number of particles, m represents the mass of the particles, and v_n represents the net velocity of the nth particle.

Total energy of system was calculated using the following formula:

$$\Sigma E = E_{LJ} + E_k$$

Where ΣE represents total energy, E_{LJ} represents net Lennard-Jones potential, and E_k represents net kinetic energy. Total energy was updated every integration step.

Instantaneous temperature was calculated using the following formula:

$$T = \frac{2}{3NR} E_k$$

Where T represents instantaneous temperature, N represents number of gas particles, R represents the ideal gas constant, and E_k represents net kinetic energy.

Instantaneous pressure was calculated using the following formula:

$$P = \frac{NkT}{V} + \frac{1}{3V} \sum_i r_i \bullet f_i$$

P is pressure, N is the number of atoms, k is the Boltzmann constant, T is instantaneous temperature of the system, V is the volume of the box, r_i is the position of atom i, and f_i is the force acting upon atom i.

An alternative integration algorithm, velocity Verlet, was also implemented as an alternative to basic Euler integration. Finally, a Berendsen thermostat was implemented to rescale velocities every step towards a target temperature, as particle collisions and internal fluctuations cause changes in temperature during simulation.

Velocity Verlet integration was done according to the following algorithm:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2$$
$$v(t + \Delta t) = v(t) + \frac{a(t)+a(t+\Delta t)}{2}\Delta t$$

Where t is time, x is particle position, v is particle velocity, a is particle acceleration. This integration was done separately in each degree of freedom for each particle.

Berendsen thermostat velocity rescaling was done using the following algorithm:

$$\lambda = \sqrt{1 + \frac{\Delta t}{\tau}\left[\frac{T_{bath}}{T(t)} - 1\right]}$$

Where λ is the velocity rescaling factor, τ is the coupling constant, Δt is the timestep, T_{bath} is the temperature of the heat bath, and $T(t)$ is the instantaneous temperature of the system.

To assess stability of integration algorithms, simulations were run without the thermostat. A system of 100 atoms in a 100 by 100 by 100 angstrom box was simulated over 15 picoseconds with initial temperatures generated at 300K. For Lennard-Jones 12-6 potential, 4 angstrom was used as the distance where potential energy is zero and the depth of the potential well was 1000 J/mol. This simulation was run with 1, 2, 4, 8, and 16 femtosecond timesteps once using Euler integration and once using velocity Verlet integration. With the output data, linear regression was used to calculate drift rate of net energy in J/mol per picosecond for each timestep and integration algorithm. Additionally, the standard deviation of the total energy was used to assess the size of energy oscillations for each timestep and integration algorithm, scaled according to the average calculated instantaneous temperature for each simulated system.

To assess the pressure-temperature relationship, simulations were run with the thermostat implemented. A system of 300 atoms in a 120 by 120 by 120 angstrom box was simulated over 10 picoseconds with temperatures from 200K to 400K with a 25K increment using Euler integration. For Lennard-Jones 12-6 potential, 4 angstrom was used as the distance where potential energy is zero and the depth of the potential well was 1000 J/mol. With output data, average temperature, average pressure, and standard deviation of the pressure was calculated for each quarter of the simulation and overall to assess how the pressure changed as the system approached the thermostat target temperature. Pressure was also analyzed as a function of temperature and compared to the relationship indicated by the ideal gas law.

Results:

Stability of Integrators:

To analyze stability of integrators, simulations were run at various timesteps using either Euler or velocity Verlet integration algorithms and total energy conservation and stability was assessed.

In a theoretically perfect simulation, energy conservation would be perfect and total energy would not drift. Linear regression was used to calculate the rate at which total energy drifted for the integration algorithms at different timesteps (Table 2, Figure 5). Generally, greater integration timesteps led to larger drift and therefore more order. At the smallest timestep of one femtosecond, Euler integration drifted at a rate of 1.60E-01 kJ/ps*mol and velocity Verlet integration drifted at a rate of 4.90E-01 kJ/ps*mol. At the greatest timestep, sixteen femtoseconds, Euler integration drift rate was 9.65E+00 kJ/ps*mol and velocity Verlet integration drift rate was 1.21E+00 kJ/ps*mol , an order of magnitude larger. Generally, both integration algorithms tended to drift at a rate on the same order of magnitude and neither was clearly more accurate than another. While the graph (Figure 5) visually seems to imply that velocity Verlet might have smaller drift and better long-term stability at larger timesteps, drift rate for both integration algorithms increased by an order of magnitude for timesteps larger than two femtoseconds and so at large timesteps the error introduced by the timestep is enough that integration algorithm is immaterial.

To measure the strength and frequency of oscillations of total energy (Figure 6), the standard deviation of total energy was calculated for the integration algorithms at different timesteps (Table 2, Figure 4). Greater integration timesteps led to larger standard deviation and larger fluctuations in total energy. Velocity Verlet integration had consistent and noticeably greater standard deviation than Euler integration, with velocity Verlet standard deviation being an order of magnitude greater (1.40E+02 kJ/mol) than Euler standard deviation (5.29E+01 kJ/mol) at a four femtosecond timestep. However, as earlier established, timsteps greater than two femtoseconds started leading to drift rates and errors at greater orders of magnitude; at timesteps equal to or less than two femtoseconds, standard deviation of total energy for velocity Verlet and Euler integration were on the same order of magnitude.

It should also be noted that for this relatively short simulation on a relatively small system, the average drift rates were also impacted by the collision frequency and temperature of each system; a system that generated with more intermolecular interaction would have greater drift and standard deviation than a different, more subdued system would. Generally, the timestep seemed to have a much greater effect on total energy drift and fluctuations than integration algorithms did; at small timesteps, both integration algorithms seemed to function similarly.

Timestep (fs)	Euler Drift Rate (kJ/mol per ps)	Verlet Drift Rate (kJ/mol per ps)	Euler SD (kJ/mol)	Verlet SD (kJ/mol)
1	1.60E-01	4.90E-01	1.21E+01	2.01E+01
2	2.90E-01	2.80E-01	2.46E+01	4.38E+01
4	6.49E+00	4.19E+00	5.29E+01	1.40E+02
8	7.00E-02	1.07E+00	1.17E+02	1.99E+02
16	9.65E+00	1.21E+00	2.07E+02	7.78E+02

Table 2: Simulations of 100 atoms in a 100 by 100 by 100 angstrom box, run for 15 ps using indicated timestep and initial temperatures generated by Maxwell-Boltzmann distribution at 300K. “Verlet” represents “velocity Verlet” for brevity.

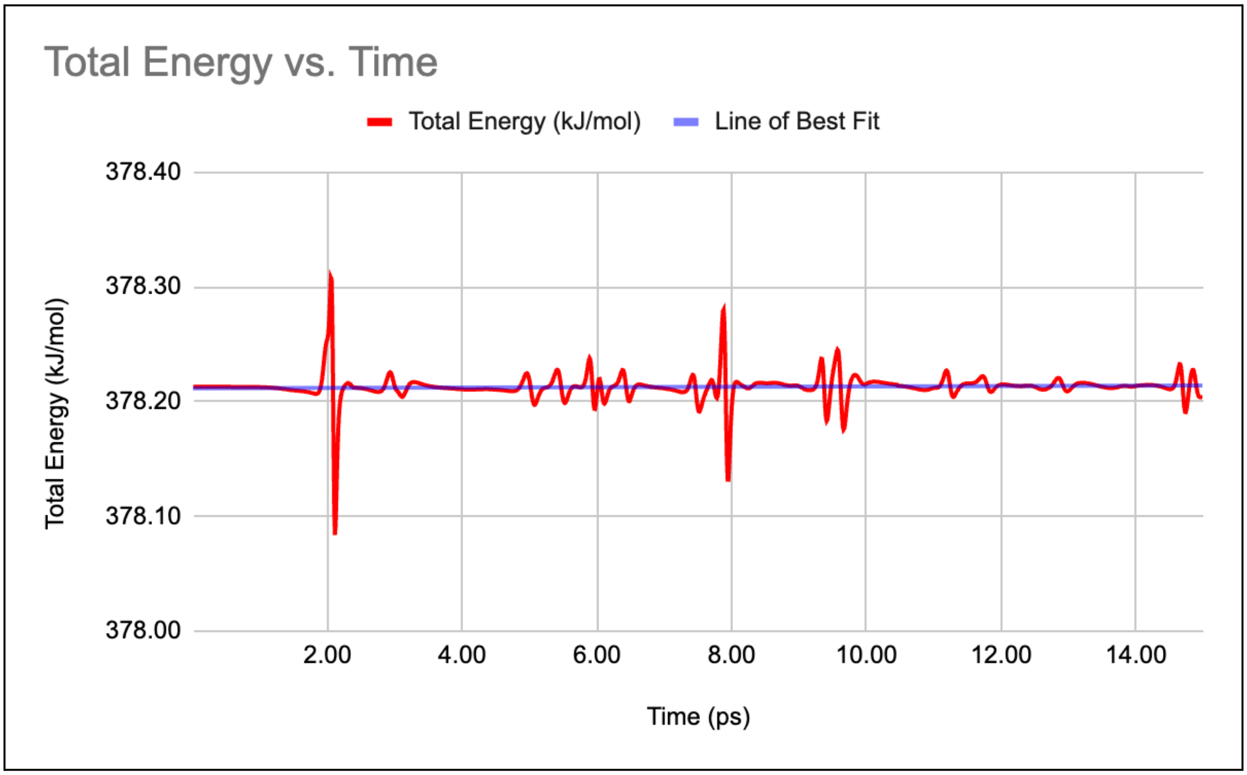

Figure 6: Graph of total energy against time for simulation run using Euler integration algorithm at 1fs timestep to illustrate visually how total energy fluctuates and drifts over time.

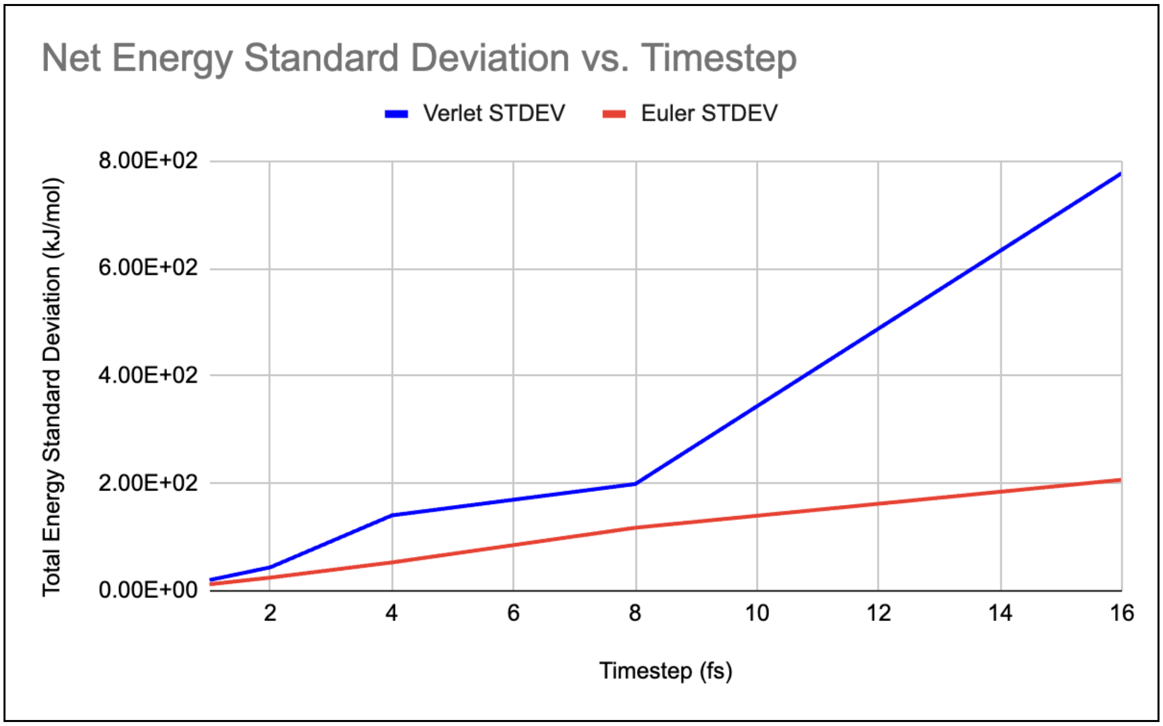

Figure 4: Standard deviation of total energy in Euler and velocity Verlet integration algorithms against the timestep simulation was run at. Total energy has much greater standard deviation at larger timeseps.

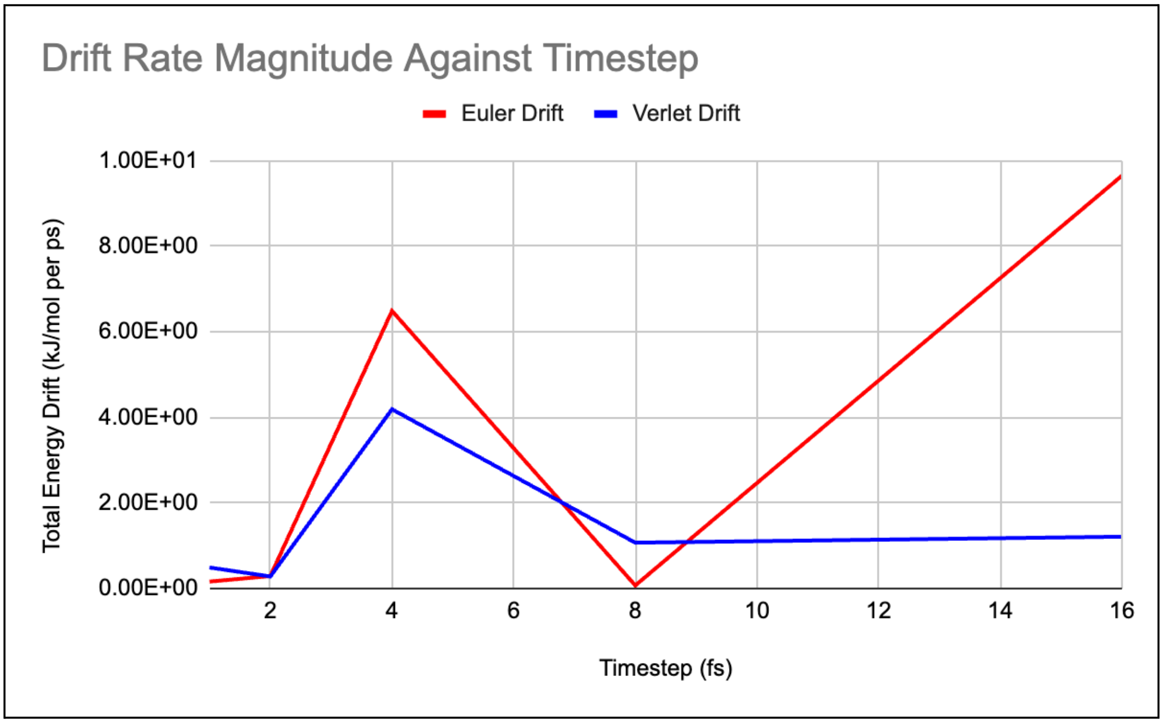

Figure 5: Drift rate of total energy for Euler and velocity Verlet integration algorithms graphed against various timesteps.

Pressure-Temperature Relationship, Thermostat, and Deviation from Ideal Gas

To analyze pressure-temperature relationship, simulations were run at various temperatures with a thermostat and pressure was analyzed against temperature and compared to ideal gas.

Maxwell-Boltzmann distributions generated an initial net temperature not exactly equivalent to the desired temperature for each simulation (due to random assignment via normal distribution), though this temperature stabilized towards the desired temperature as the simulation went on.

The average difference between calculated temperature and desired temperature during the first quarter of the simulation trajectory among all nine simulated systems was 3.46K and average standard deviation of average temperatures among the nine systems during the first quarter was 2.45K. By the fourth quarter of the simulation trajectory, average difference between calculated and desired temperature had decreased to 0.19K and average standard deviation of average temperatures had decreased to 0.81K (Table 1). This change reflected the Berendsen thermostat bringing the system to the desired temperature and the system stabilizing.

This was not reflected in the standard deviation from average pressure in each quarter. The four quarters of the simulation trajectory had an average pressure of 7.18E+5 Pa, with a SD of 3.38E+03 Pa. The four quarters had an average pressure SD of 1.72E+04 Pa, with an extremely minimal SD 3.33E+02 Pa (Table 1). In other words, there was relatively little variance in average pressure or fluctuations of pressure (SD of average pressure) throughout the entire simulation trajectory. This is because the zero order of magnitude kelvin fluctuations in temperature over the course of the simulation (which represents <1% of the temperature), while noticeable in regards to the Berendsen thermostat, would induce very minimal amounts of change in overall pressure; instead, the fluctuations in pressure would receive much greater impact from particle collisions and the instantaneous state of the system.

The relationship between temperature and pressure was relatively linear, which is the expected behavior. Using the average pressures and average temperatures for each system trajectory, the temperature-pressure relationship was fit using linear regression. The average of the residuals for this line of best fit was equal to 0.0 Pa and the average of the absolute values of the residuals for this line was 1.68E+3 Pa, which is 0.23% of the average pressure (Figure 1). The simulated temperature and pressure relationship had extremely minimal deviation from a linear relationship.

The pressure-temperature relationship of the simulated Lennard-Jones gas systems were graphed and compared to the pressure-temperature relationship of ideal gas (Figure 2). The average deviation between simulated LJ gas pressure and ideal gas pressure over all temperatures was 3.21E+3 Pa, which is less than 1% of total pressure. The deviation between Lennard-Jones gas pressure and ideal gas was relatively small. This deviation was also graphed against the average temperature of the system (Figure 3). As temperature of the system increased, the deviation between LJ gas pressure and ideal gas pressure seemed to decrease, which is the expected behavior. It is worth noting that the parameters chosen for Lennard-Jones 12-6 potential $(\sigma = 4A, \varepsilon = 1000J/mol)$ are close to the parameters for the noble gas Argon, which should theoretically behave similarly to ideal gas.

Simulation Quarter	System ID	System Target Temp. (K)	Avg. Temp. (K)	Temp. SD (K)	Avg. Pressure (Pa)	Pressure SD (Pa)
First	1	200	205.52	2.93	493,488.60	10,894.31
First	5	300	302.98	0.90	728,654.79	12,648.96
First	9	400	404.43	1.54	972,047.16	14,920.35
Second	1	200	201.53	1.05	477,913.02	14,771.12
Second	5	300	301.14	0.78	708,363.23	18,767.48
Second	9	400	400.85	1.38	963,876.58	22,160.93
Third	1	200	200.42	0.83	475,958.14	12,364.69
Third	5	300	300.11	0.96	717,519.32	19,985.17
Third	9	400	400.39	1.11	958,873.79	25,173.04
Fourth	1	200	199.98	0.89	471,635.08	19,459.62
Fourth	5	300	300.02	0.76	716,645.27	19,747.44
Fourth	9	400	399.87	1.02	960,557.73	23,485.89

Table 1: Simulations of systems of 300 atoms in a 1728000 cubic angstrom cube box, run for 10 picoseconds with 2 femtosecond timesteps; table shows three representative systems for each quarter of the simulation trajectory to demonstrate temperature and pressure over time.

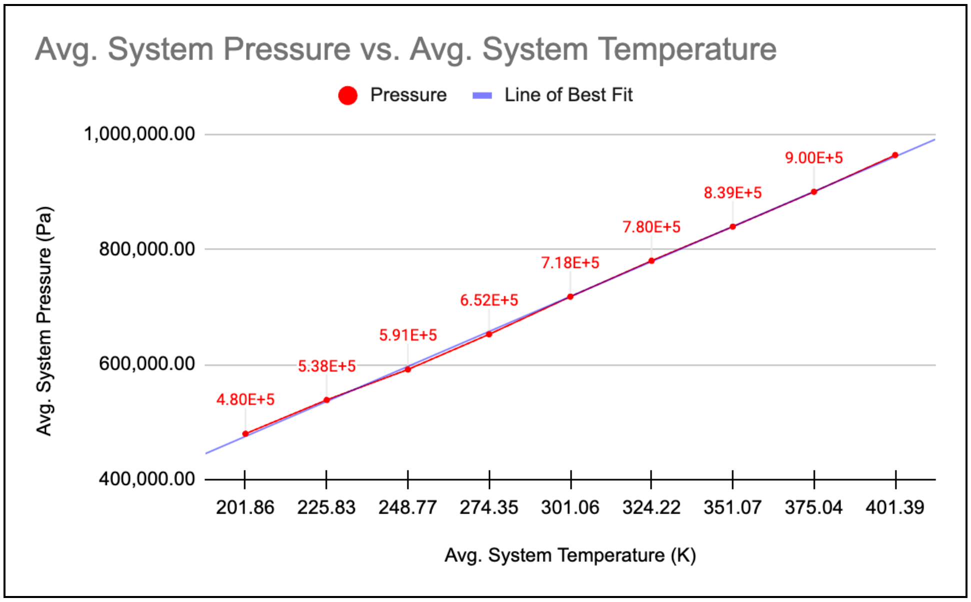

Figure 1: Average pressure graphed against average temperature for each system, with line of best fit to illustrate the linear relationship between pressure and temperature.

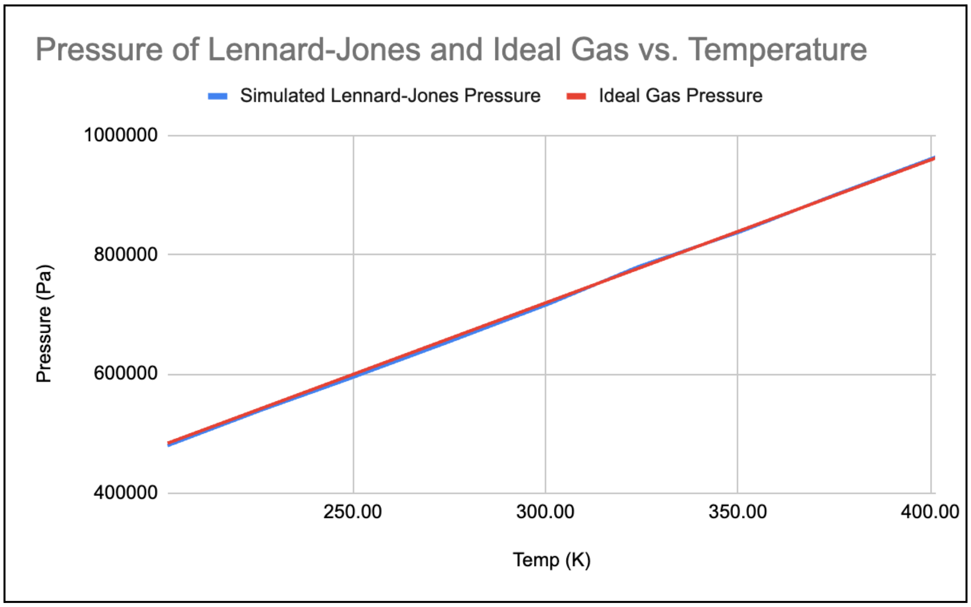

Figure 2: Simulated system average pressures and ideal gas pressures graphed against temperature.

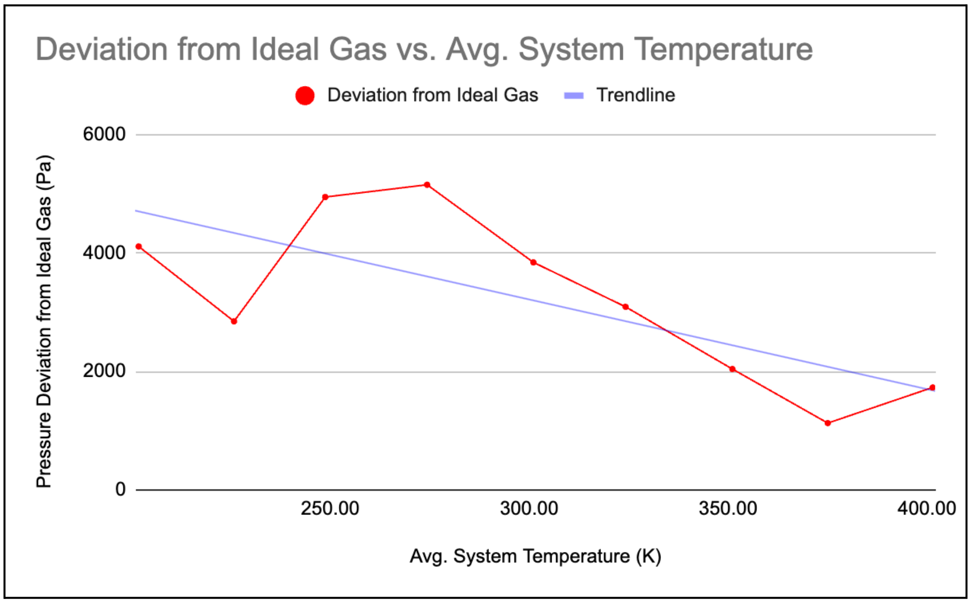

Figure 3: Magnitude of the deviation of simulated Lennard-Jones gas pressures from ideal gas pressures graphed against temperature with line of best fit.