Simulating Deuterium-Tritium Fusion Using Python

Noah Brear, Adviser: Dr. Heddle

1 Abstract

This project is a program created in Python that simulates deuterium-tritium (D-T) fusion in a small container. Nuclear fusion is a nuclear reaction where the nuclei of low-mass elements fuse together to form higher-mass elements. The physical models that this simulation relies on are contained within four sections, each utilizing a main algorithm. The first models the particles using kinematics and the kinetic theory of gases. It gives particles a random position in the simulation and then assigns them a velocity based on a Maxwell-Boltzmann distribution of thermal velocities. The phase then finds the momentum and detects if any collisions occurred. If so, it calculates the force acting on the walls of the container. It then uses these forces to find the pressure on the walls.

The second phase finds the changes in the particles' velocities from the Coulomb repulsion force. To do this, the simulation first takes a reference particle. It then finds the separation distances between the reference particle and each other particle in the simulation. Using this separation distance and the charges of the two particles, it calculates the forces acting on the reference particle. Using these forces, it then finds the change in velocity of the reference particle. FInally, it passes this value back to the first phase, and the separation distance to the third phase.

The third phase models the fusions based on quantum tunneling. It first decides if two particles can fuse (that is, one is deuterium, the other is tritium). It then uses the separation distance between the two particles to calculate a probability of fusion. It then generates a random decimal between zero and one. If the random probability is lower, the particles fuse. The fourth phase finds the

vectors of the resulting particles and the energy released. This phase takes two fusing particles and replaces their masses with the resulting particles' masses (a neutron and a helium-4 nucleus). It then finds their new velocities based on the conservation of energy and momentum. The resulting particles masses are less than the total initial mass and this change in mass is directly converted to energy, via Einstein's energy-mass equivalence. The methods for these phases are then run in a simulation created by using the Python library Simpy.

2 Introduction

This simulation was designed to further an understanding of how nuclear fusion energy functions, and explore how independent variables such as the temperature and the number of particles affect the fusion rate. The expected results from the simulation were that the rate at which fusions occur will increase as either the temperature, the number of particles, or both increase. An increase in temperature was expected to cause the particles to have more kinetic energy which would allow them to approach closer to other particles. Likewise, an increase in the number of particles was expected to cause them to become closer due to the decrease in the amount of empty space.

The technique used for the first and second phases is to model the particles individually as opposed to a gas as a whole. This is preformed in this manner since the goal of the program is to show how nuclear fusion functions on the scale of the particles themselves. This is also a needed technique for the second phase as the simplest and most effective way to model the Coulomb repulsion force for each particle on one another is as forces between point charges. Quantum mechanics dictates that subatomic particles have a wave-particle duality. However, in this simulation, the particles are modeled solely in their particle forms. This is done in order to simplify the process needed in the third phase.

In the third phase itself, the effects of the nuclear force are modeled by a function that is constructed to allow for few fusions below 100 million kelvin. Since the nuclear force is intricate and difficult to model, the probability of fusion is modeled by a piecewise function that represents the nuclear force. Finally, the main technique used in the fourth phase is to calculate the energy produced as a constant value. Since the only type of fusion reaction simulated is deuterium-tritium fusion, the energy produced is constant.

3 Theory

3.1 Phase One

The theory behind the first phase is that the particles within the container will behave as an ideal "gas" and obey the laws of kinematics. Though the particles can be treated as such, the particles become ionized at high temperatures and become a plasma. Particles within the container will have velocities that are distributed along a Maxwell-Boltzmann distribution. The most probable speed is given by

$$v_{\text{prob}} = \sqrt{\frac{kT}{m}}, \tag{1}$$

where k is the Boltzmann constant, T is the temperature, and m is the mass of the particle The differential of the probability density function for a Maxwell-Boltzmann distribution in a three dimensional system is

$$dP(v) = (\frac{1}{2\pi {v_{\text{prob}}}^2})^{\frac{3}{2}} e^{-\frac{v^2}{2v_{\text{prob}}}} dv, \tag{2}$$

where $P(v)$ is normalized. A particle with a velocity v and mass m will then have a momentum p given by

$$p = mv. \tag{3}$$

When a particle collides with a wall, the momentum in the direction that the wall impedes will be reversed. The change in momentum of the particle produces a force that acts on the wall. The force is then given by

$$F_{\text{i}} = \frac{\Delta p_{\text{i}}}{\Delta t}, \tag{4}$$

where Δp is the change momentum of a particle, Δt is the duration of the collision, and i is the index of the particle. The pressure created by an individual particle can then be found by

$$P_{\text{i}} = \frac{F_{\text{i}}}{A}. \tag{5}$$

The average pressure over the course of the simulation can then be found as a rolling average given by

$$P_{\text{avg}} = \sum_{i=1}^{N} \frac{P_{\text{i}}(1t)}{t_{\text{tot}}}, \tag{6}$$

where $1t$ is one unit of time (to allow units to be correct), and t_{tot} is the total time that the simulation has been running.

3.2 Phase Two

The theory behind the second phase is that a the nuclei of the deuterium and tritium particles will repel each other by the Coulomb repulsion force. The repulsion force is given by

$$F = \frac{kq_{\text{a}}q_{\text{b}}}{{r_{\text{a,b}}}^2}, \tag{7}$$

where k is the Coulomb constant, q_{a} is the net charge on particle a, q_{b} is the net charge on a particle b, and $r_{\mathrm{a,b}}$ is the separation distance between the two particles. The change in velocity that this force causes can be found by combining equations (3) and (4) and solving for Δv_{a} to get

$$\Delta v_{\mathrm{a}} = \frac{F_{\mathrm{a}} \Delta t}{m}. \tag{8}$$

3.3 Phase Three

The theory behind the third phase is that at close enough ranges, the nuclear force, which binds protons and neutrons together in nuclei, will allow particles to overcome the Coulomb repulsion force and join together to form higher mass element nuclei. The probability of whether or not two nuclei will fuse depends on their type and separation distance. The probability is found by the piecewise function

$$P(r) = \begin{cases} 1 - r & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases}, \tag{9}$$

where r is the separation distance in fermi (10^{-15} meters).

3.4 Phase Four

The theory that is used in the fourth phase is that with enough energy put into a system, deuterium and tritium nuclei fuse together in the chemical reaction

$${}^{2}_{1}\mathrm{H}^{+} + {}^{3}_{1}\mathrm{H}^{+} \longrightarrow {}^{4}_{2}\mathrm{He}^{+} + \mathrm{n}^{0}, \tag{10}$$

where ${}^{2}_{1}\mathrm{H}^{+}$ is a deuterium nucleus, ${}^{3}_{1}\mathrm{H}^{+}$ is a tritium nucleus, ${}^{4}_{2}\mathrm{He}^{+}$ is a helium-4 nucleus, and n^{0} is a neutron. The velocities of the resulting particles must be found in accordance with the conservation of momentum

$$p_{\text{tot}} = p'_{\text{tot}} = m_{\text{a}}v_{\text{a}} + m_{\text{b}}v_{\text{b}} = m_{\text{c}}v'_{\text{a}} + m_{\text{d}}v'_{\text{b}} \tag{11}$$

and the conservation of energy

$$KE_{\text{tot}} = KE'_{\text{tot}} = m_{\text{a}}{v_{\text{a}}}^2 + m_{\text{b}}{v_{\text{b}}}^2 = m_{\text{c}}{v'_{\text{a}}}^2 + m_{\text{d}}{v'_{\text{b}}}^2, \tag{12}$$

where p_{tot} is the total momentum before the reaction, p'_{tot} is the total momentum after the reaction, KE_{tot} is the total kinetic energy before the reaction, KE'_{tot} is the total kinetic energy after the reaction, m_{a} is the mass of the deuterium nucleus, m_{b} is the mass of the tritium nucleus, m_{c} is the mass of the helium-4 nucleus, m_{d} is the mass of the neutron, v_{a} is the velocity of the deuterium nucleus, v_{b} is the velocity of the tritium nucleus, v'_{c} is the velocity of the helium-4 nucleus, and v'_{d} is the velocity of the neutron. The total mass of the resulting particles is less than the total mass of the initial particles. This difference in mass is converted directly into energy via Einstein's mass-energy equivalence

$$E = \Delta mc^2, \tag{13}$$

where E is the energy released, Δm is the change in mass, and c is the speed of light in a vacuum.

4 Methods

In the first phase, its main algorithm simulates kinematics for the particles and finds the average pressure on the walls of the container. First, it gives the particles a random position within the container. It then gives the particles a velocity that is distributed along a Maxwell-Boltzmann distribution. Using this velocity, it then finds the momentum of each particle. After time has passed, the method that finds the momentum then detects if particles collide with any walls. If so, it then finds the forces these collisions create. The algorithm then finds the pressures on the walls that each force create. It then sums all of these pressures together and finds the average pressure of the simulation.

The main algorithm used in the second phase finds the changes in velocities for each particle due to the Coulomb repulsion forces between them. It first finds all the separation distances between every particle. Using each of these distances, the algorithm then calculates the changes in velocities for each particle due to all the forces acting on them. The simulation then passes the changes in velocities to the first phase for implementation.

In the third phase, its algorithm uses the identity of each particle to determine if any two particles can fuse (one is deuterium, one is tritium). If so, it takes the separation distance between the two particles and finds the probability that the two particles will fuse. If the particles are within 1 fermi of one another, the probability is found with the piecewise function in equation (9). If the separation distance between the two particles is greater than 1 fermi, the probability of fusion is zero. The algorithm then generates a random decimal between zero and one. If the random decimal is less than or equal to the probability of fusion, the two particles fuse. Otherwise, they do not. The algorithm then creates a boolean that contains the information of whether the particles fused or not, and passes it to the simulation.

The algorithm that is used in the fourth phase finds the directions of the resulting particles and finds the energy released in the fusion reactions. If the fusion boolean for two particles is true, the algorithm replaces the masses of the deuterium and tritium particles with the masses of a helium-4 nucleus and a neutron. It then find velocities for the new particles such that the conservation of energy and momentum are upheld by using a random direction for the velocity of the neutron. The algorithm then finds the energy released by using equation (12). It then returns the new values for the resulting particles and the energy released back over to the simulation.

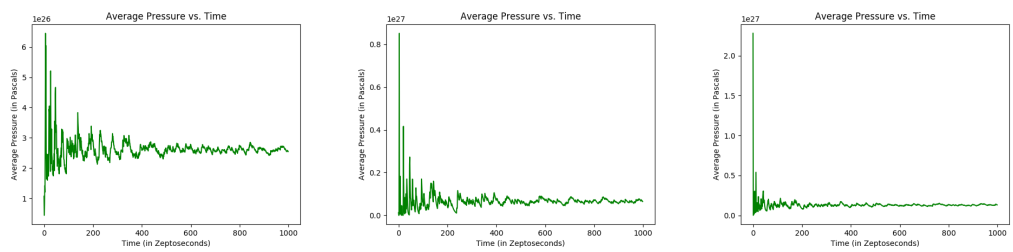
5 Data

The data collected from this program are stored in text files. For each text file created, a user-friendly text file is created to allow the user to read and understand the data if they wish to see the details of each simulation. Within the program, the data collected is then used to create three graphs: a plot of the average pressure vs. time, a histogram of the closest of approaches, and a plot of the energy produced across the simulation. The temperature data for this report was taken at temperatures 80, 100, and 120 million kelvin, for three trials each. For this portion of data taken, the time simulated for each was 1000 zeptoseconds (10^{-21} seconds), and the number of particles was 20. Keeping the number of particles low allowed for better analysis of how the temperature affects the simulation on its own.

Data was also taken for this report to explore how changing the number of particles affected the fusion rate. The numbers of particles used were 50, 100, and 150 particles. For this portion of the data, the time simulated for each run was 100 zeptoseconds. The temperature used was 150 million kelvin. Since a higher number of particles means more cycles through the methods of the second phase, the simulation took much more data points for each increase in the number of particles. The duration of the simulation was made shorter in an attempt to take the same amount of data for each type of trial. While testing the effects of changing the number of particles, the temperature was set to above ignition temperature so that there would be no concern that the particles could fuse.

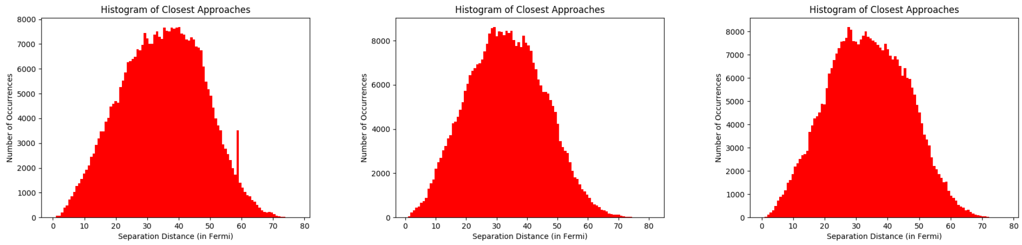
NOTE: The time labels displayed for the energy vs. time plots read "seconds", and should instead read "zeptoseconds". Changing the labels on the plots would mean needing to retake all of the data.

5.1 Temperature (80 Million Kelvin)



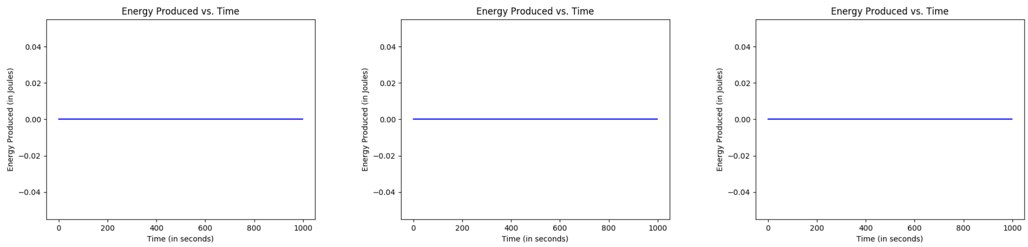
(a) Trial 1 (b) Trial 2 (c) Trial 3

Figure 1: Average Pressure vs. Time, 80 Million Kelvin



(a) Trial 1 (b) Trial 2 (c) Trial 3

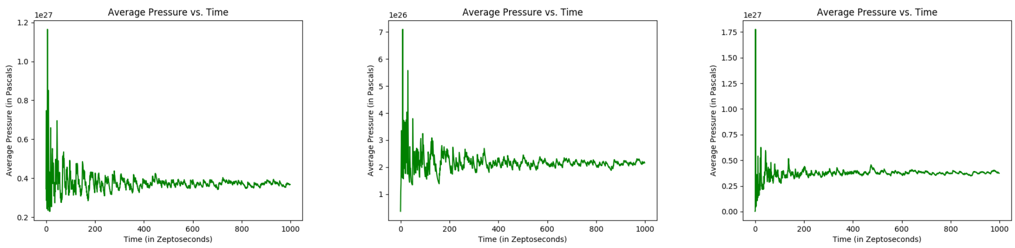
Figure 2: Closest Approaches, 80 Million Kelvin



(a) Trial 1 (b) Trial 2 (c) Trial 3

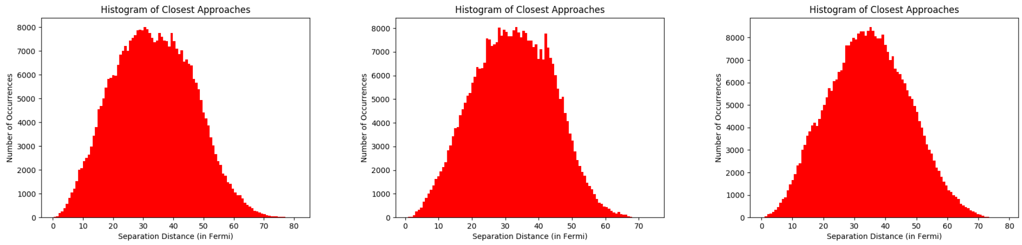
Figure 3: Energy Produced vs. Time, 80 Million Kelvin

5.2 Temperature (100 Million Kelvin)



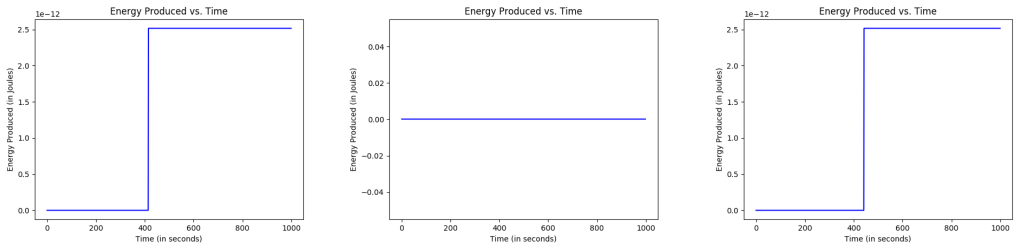
(a) Trial 1 (b) Trial 2 (c) Trial 3

Figure 4: Average Pressure vs. Time, 100 Million Kelvin



(a) Trial 1 (b) Trial 2 (c) Trial 3

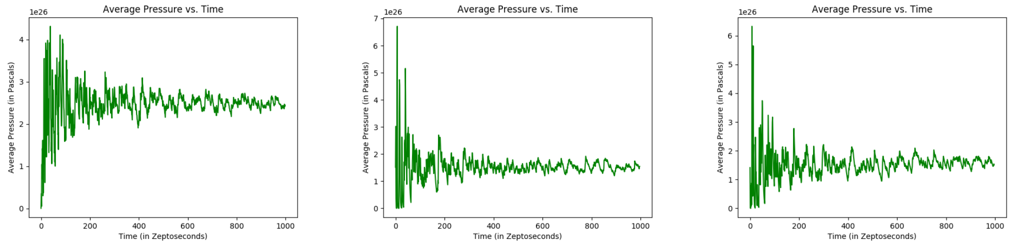
Figure 5: Closest Approaches, 100 Million Kelvin



(a) Trial 1 (b) Trial 2 (c) Trial 3

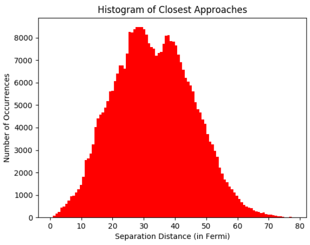
Figure 6: Energy Produced vs. Time, 100 Million Kelvin

5.3 Temperature (120 Million Kelvin)

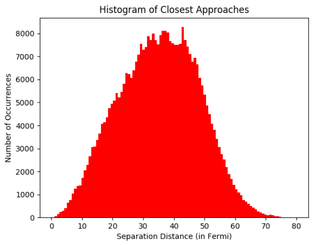


(a) Trial 1 (b) Trial 2 (c) Trial 3

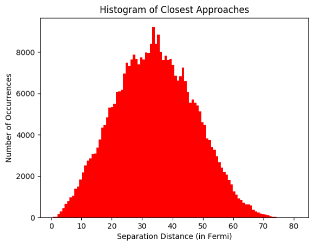
Figure 7: Average Pressure vs. Time, 120 Million Kelvin



(a) Trial 1

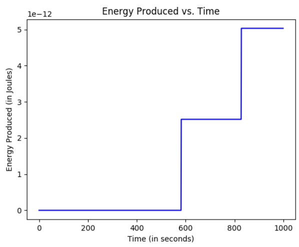


(b) Trial 2

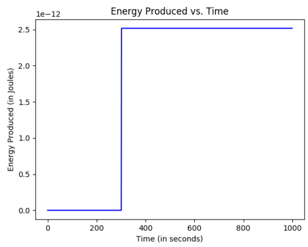


(c) Trial 3

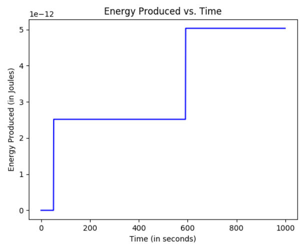
Figure 8: Closest Approaches, 120 Million Kelvin



(a) Trial 1



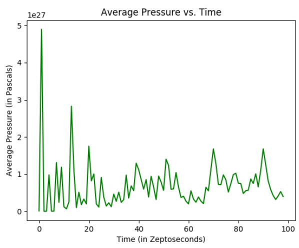
(b) Trial 2



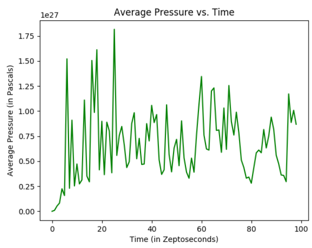
(c) Trial 3

Figure 9: Energy Produced vs. Time, 120 Million Kelvin

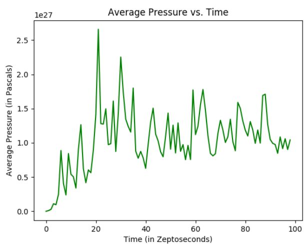
5.4 Number of Particles (50 Particles)



(a) Trial 1

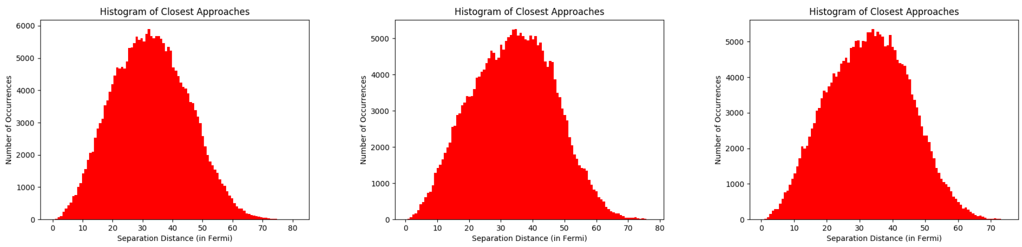


(b) Trial 2



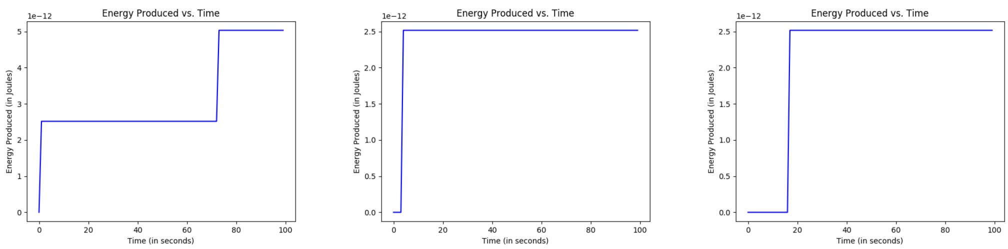
(c) Trial 3

Figure 10: Average Pressure vs. Time, 50 Particles



(a) Trial 1 (b) Trial 2 (c) Trial 3

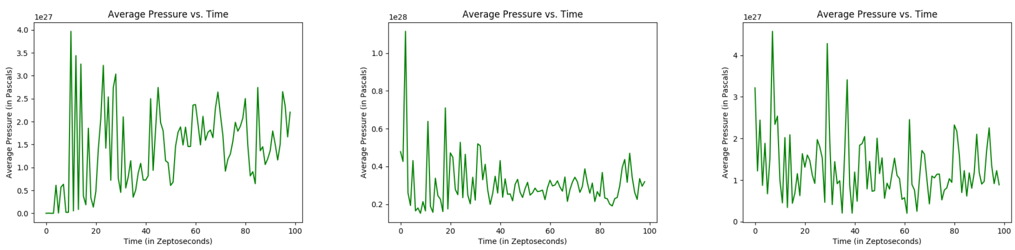
Figure 11: Closest Approaches, 50 Particles



(a) Trial 1 (b) Trial 2 (c) Trial 3

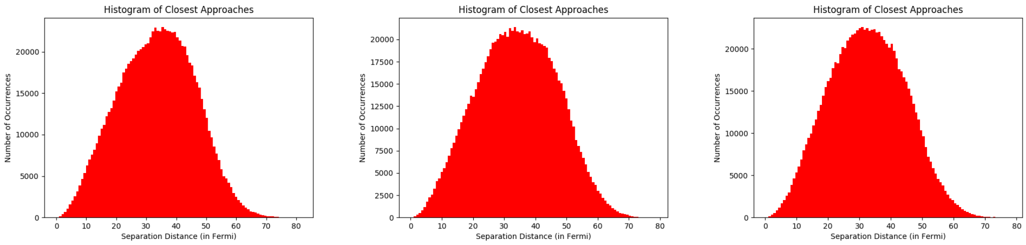
Figure 12: Energy Produced vs. Time, 50 Particles

5.5 Number of Particles (100 Particles)



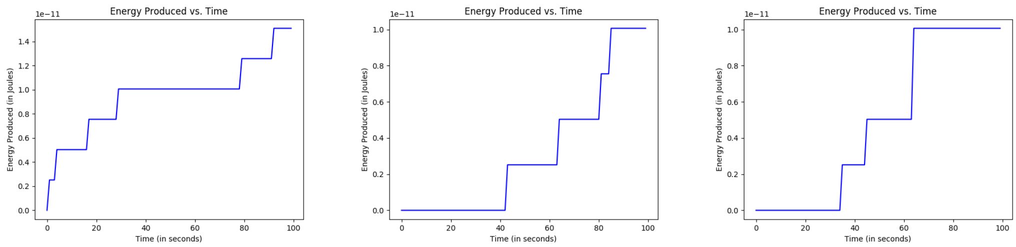
(a) Trial 1 (b) Trial 2 (c) Trial 3

Figure 13: Average Pressure vs. Time, 100 Particles



(a) Trial 1 (b) Trial 2 (c) Trial 3

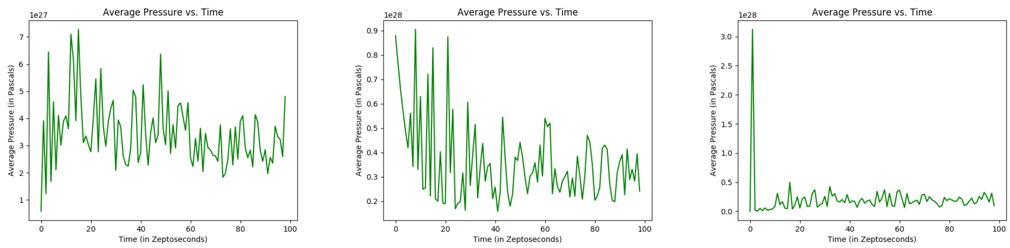
Figure 14: Closest Approaches, 100 Particles



(a) Trial 1 (b) Trial 2 (c) Trial 3

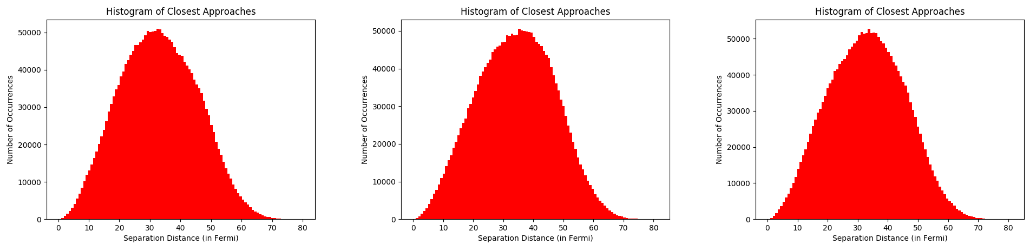
Figure 15: Energy Produced vs. Time, 100 Particles

5.6 Number of Particles (150 Particles)



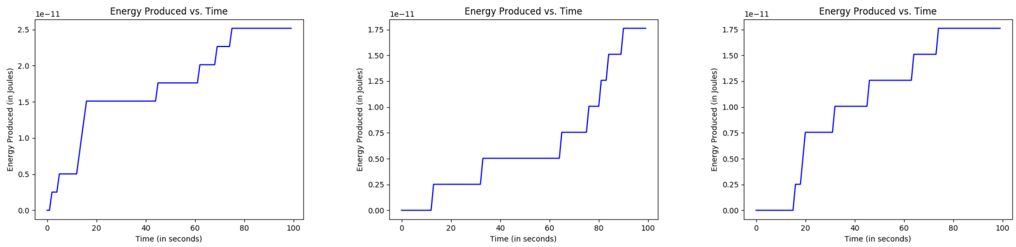
(a) Trial 1 (b) Trial 2 (c) Trial 3

Figure 16: Average Pressure vs. Time, 150 Particles



(a) Trial 1 (b) Trial 2 (c) Trial 3

Figure 17: Closest Approaches, 150 Particles



(a) Trial 1 (b) Trial 2 (c) Trial 3

Figure 18: Energy Produced vs. Time, 150 Particles

6 Discussion and Conclusions

The data collected for the pressure vs. time and the histogram of closest approaches for each trial show that the methods of the first two phases function properly. The pressure simulated nearly matches the expected result values and the histograms of closest approaches show that the particles are rarely coming within one fermi of one another. Any differences between these values and the expect values most likely comes from the discrete nature of this simulation. The particles do not collide with the wall at the same points in time or with the same forces across all simulations. With this, the values for the pressure will fluctuate and the number of approaches that are less than one fermi will vary. Due to the distributions involved, these values may increase as the values of the parameters decrease. For instance, the pressure calculated at 100 million kelvin has a probability of being less than the pressure calculated for a temperature of 120 million kelvin. However, the probabilities of these happening are low. Generally, as the temperature and/or number of particles increase, the pressure will increase, and the number of approaches that are less than one will also increase.

The energy produced vs. time plots hold the results for the third and fourth phases. The accuracy of the piecewise function used in the third phase can be obtained from the data collected as the temperature increased. At 80 million kelvin, there were no fusions recorded across all three trials. For the plots at a temperature of 100 million kelvin, fusions begin occurring. However, as can be seen from the second trial, fusions are not guaranteed at this temperature. At 120 million kelvin, fusions occur with almost every run. These values show that the piecewise function used in the third phase is functioning properly, and places ignition temperature at about 100 million kelvin. The term "ignition temperature" is the minimum temperature that is generally accepted for when fusion reactions begin. However, this is a soft limit for the minimum temperature, as the fusion rate also depends on the density of the plasma, that is, the number of particles contained within the simulation.

The energy produced vs. time plots for a varying number of particles compared to the same plots for a varying temperature reveal which parameter affected the rate of fusion the greatest. The varying temperature simulations ran for ten times longer than the varying particle simulations. However, the maximum temperature tested still produced less energy in 1000 zeptoseconds than the maximum number of particles tested produced in 100 zeptoseconds. This shows that the parameter that affects the rate of fusion the most is the number of particles. The reasoning for this is that as temperature increases, the fusion rate increases as well because the particles have a greater velocity so they may penetrate deeper into the potential barriers of other particles. As the number of particles increases, the rate of fusion also increases because the particles have less space to move around. The number of particles has a greater impact on the rates of fusion because it permanently limits the mean free path of the particles in the plasma, while a greater temperature simply permits the particles to possibly approach closer.

7 References

- W. S. C. Williams, *Nuclear and Particle Physics*, Oxford University Press Inc., New York, NY (1991).
- John R. Taylor, *Classical Mechanics*, University Science Books, Sausalito, CA (2005).
- David J. Griffiths *Introduction to Electrodynamics: Fourth edition*, Oxford University Press Inc., New York, NY (2017).
- Stephen J. Blundell, and Katherine M. Blundell, *Concepts in Thermal Physics: Second edition*, Oxford University Press Inc., New York, NY (2010).