

Abstract

This supplementary paper will focus on the development of my particle simulations. The purpose of these simulations is to illustrate a couple of phenomena in particle physics. In the respected field, a key challenge is the scale at which these phenomena occur. We cannot see particles with the human eye, so it defaults to a troublesome learning experience for others. Computer simulations in this field help provide insight into what happens in these phenomena. For this project, I made a simulation based on particle collisions and another simulation based on heat. Keep in mind that they may not be 100% accurate since they are supposed to be simplifications of the actual phenomenon, yet still mostly accurately depict it.

Simulation Methodology

A methodology is a system of methods used in a particular area of study. When it comes to simulations, there exist concepts and equations involved with the development process. When looking at my particle collision simulation, it may look quite simple, but there are a lot of underlying concepts and details involved. A particle collision simulation utilizes the conservation of momentum and the conservation of energy. When a particle collides with another, momentum and energy are conserved. The conservation laws shown on the macroscopic level are also apparent in the particle realm. The sum of the initial kinetic energies (KE) and momentums (p) of the particles is equal to the sum of the final (prime ') energies and momentums of the particles.

$$\text{KE}_1 + \text{KE}_2 = \text{KE}_1' + \text{KE}_2'$$
$$\mathbf{p}_1 + \mathbf{p}_2 = \mathbf{p}_1' + \mathbf{p}_2'$$

When making my heat simulation, however, there was a divergence in the concept of conservation. I had to use an equation that relates kinetic energy with temperature:

$$K = \frac{3}{2} \frac{R}{N_A} T$$

K = average kinetic energy (measured in Joules)

R = gas constant (8.314 J/mol · K)

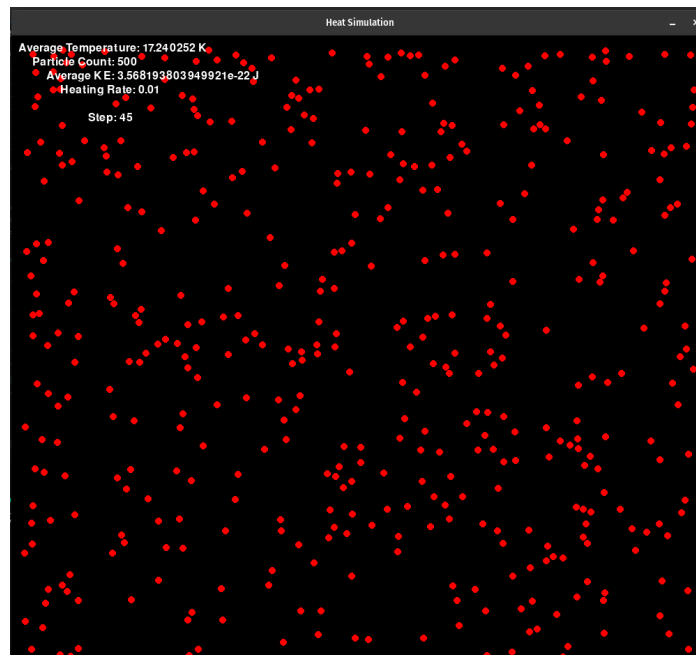
N_A = Avogadro's number (6.022x10²³ atoms/mol)

T = temperature (measured in Kelvin)

When a particle heats up, it gains kinetic energy and moves faster. The temperature of the system increases and results in an increase in the average kinetic energy, which concludes that there is no longer conservation within the system because the averages are no longer constant. A heating rate was implemented in the simulation to produce the effect of the particles heating up. As time goes on, the particles visually start moving faster and faster, ultimately producing the desired effect of what happens when heat is applied to a material over some time.

Challenges and Problem Solving

I was somewhat overconfident, thinking it would be as easy as putting a shirt on. What I should have taken into account was that I barely used Python. I was more accustomed to JavaScript and Java, and also lower-level languages. Python is much easier to use than those, but the fact that I barely had experience writing in the language made it both a challenge and a fun experience. I needed to research libraries that I could use and how to use the language in this context, which is making computer simulations. Some pesky bugs also needed fixing. When making the particle collisions simulation, my code worked, except I forgot to implement code to clear each new frame, which resulted in the particles drawing tails using the particles from the previous frames. Annoyed, I spent more than 10 minutes writing that section multiple times over, getting the same result. When I talked with my mentor, he automatically realized I forgot to add a piece of code that clears the previous frame. When it came to my heat simulation, it was a bit more complex. I had to figure out how to model what heat does. I started by trying to simulate heat transfer, but then changed my mind, then changed my mind again and came up with the perfect idea. My mentor liked my final simulation, and I was also pretty proud of it.



A Screenshot of my Heat Simulation

Lessons Learned

Going through the process of making these simulations was an excellent learning experience. The simulations weren't too overbearingly complex, but it's just the thought process required for putting everything together took a little more than just writing some code. I had to dig deeper into what I wanted to get out of making these simulations and focus on the main goal. I wanted to contribute to the idea of gaining insight using computational methods. Another valuable thing I learned was to avoid underestimating something because of my experience in related areas. You can get caught up and confused about something that ended up having an easy fix.