Exercise 1, TFY4235 Computational physics

Martin K. Johnsrud

Introduction

The goal of this exercise is to simulate particles as flat, hard disks in a square, 2D container. This is done with a event-driven simulation, as described in the exercise [1]. This is implemented in Python, using the built-in library heapq. The simulation is used to first tested with scenarios we know the outcome of, then used to demonstrate the Maxwell-Boltzmann distribution and to investigate the effect of a large, heavy disk hitting a large number of small, inert particles.

Implementation

The main engine of the code is the function `run_loop()` in `utillities.py`. It follows the algorithm, as laid out in [1], using the objects:

- `particles`, a numpy array with the position and velocity of all the particles.
- `collisions`, a priority queue containing the time of the collision, the index of the particle(s) involved, and the type of collision it is.
- `last_collided`, a list of when each particle was involved in a collision.

To be expanded ...

Tests

Several functions were developed to test the accuracy of the simulation. First, one particle, starting at in the middle of the box, all the way to the left, and with a velocity with at $45°$ to the x-axis should move in a titled rectangle. With $\xi = 1$ it should also conserve energy. Figure 1 shows that this is still the case after $10,000$ events. To test the validity of the particle collision, one small, light particle is sent towards a single, large and heavy particle, with varying impact parameters. The relationship between the impact parameter is is (ref til goldstein, begunn hvorfor det blir cos) $\frac{\mathrm{d}s}{\mathrm{d}\theta} = a/2\sin(\theta/2)$. The result is shown i Figure 2, and is in good agreement with the theory. Lastly, the energy from a simulation of N particles over T events is shown in

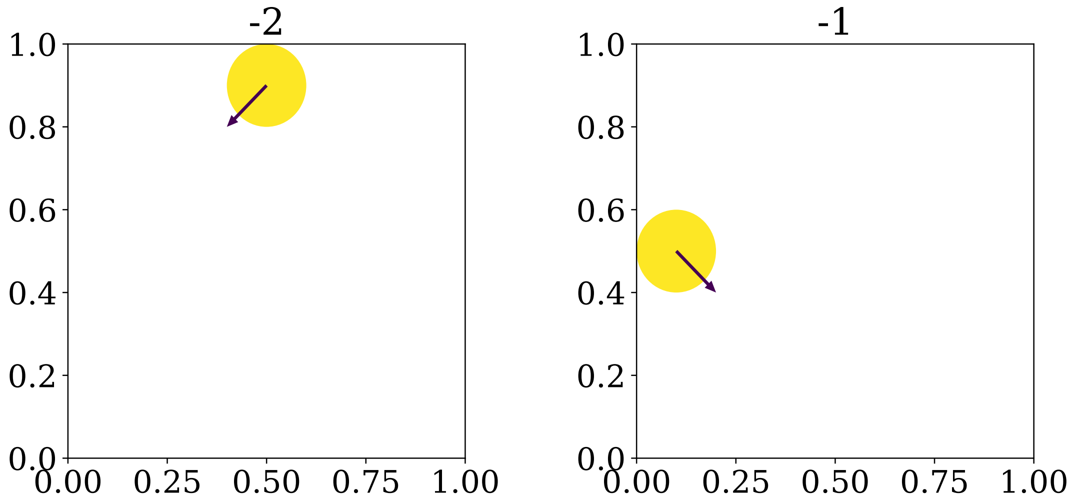


Figure 1: One ball being simulated. after 10000 steps, it still follows a regular pattern.

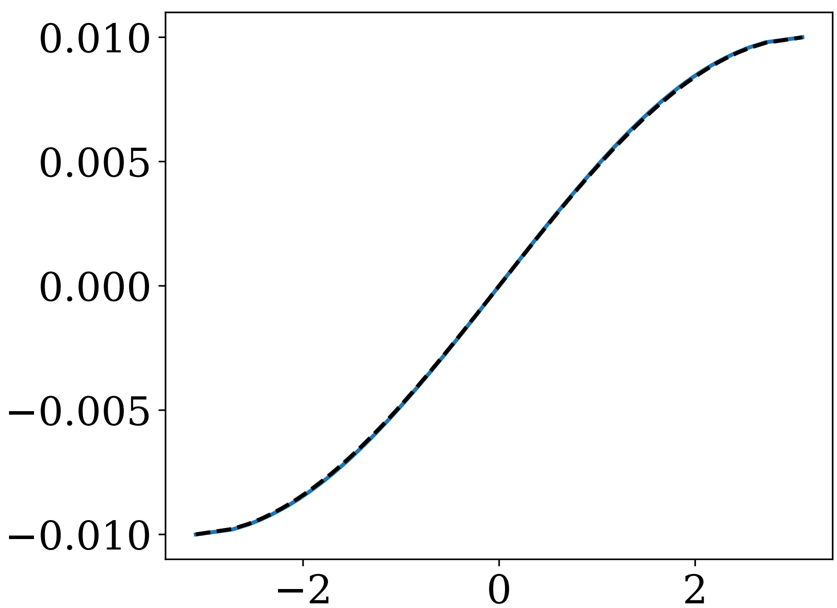
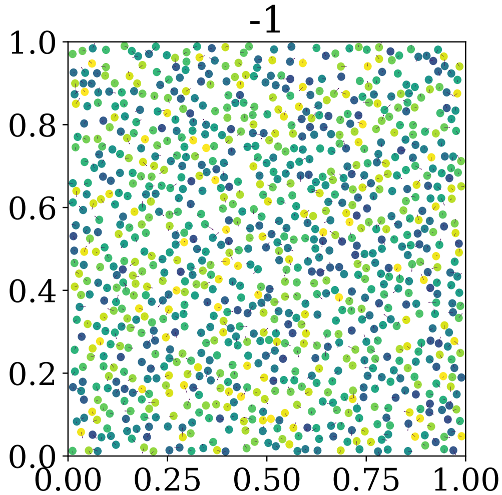


Figure 2: The



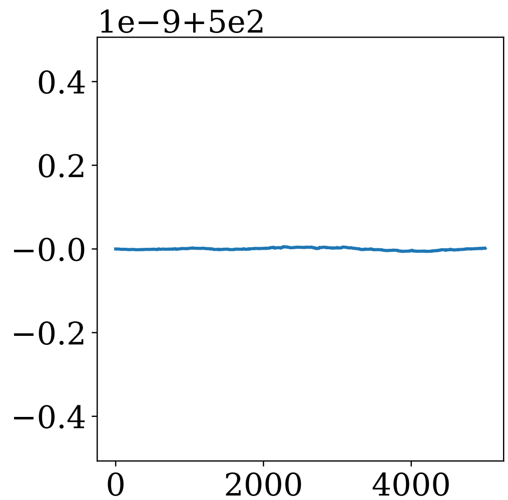


Figure 3: On the left is a snapshot of the particles. The arrows represent the velocities. On the right, the energy is plotted as a function of events. The energy loss can be seen to be negligible.

Statistical distribution

As the system equilibrates, it should reach the Maxwell-Boltzmann distribution, which in 2D is

$$f(v) = \frac{mv}{T} \exp\left(-\frac{mv^2}{2T}\right),$$

when using units in which $k_b = 1$. The equipartition theorem gives the temperature $T = E$ in 2D. Figure 5 was used to find a good starting point for when the simulation has reach equilibrium. After that, the simulation is sampled every N event, where N is the number of particles. This ensures somewhat independent samples. Figure 4 shows the velocity distribution, compared to the Maxwell Boltzmann distribution

References

[1] TFY4235 Computational Physics Exercise 1, 2021. Institutt for fysikk.

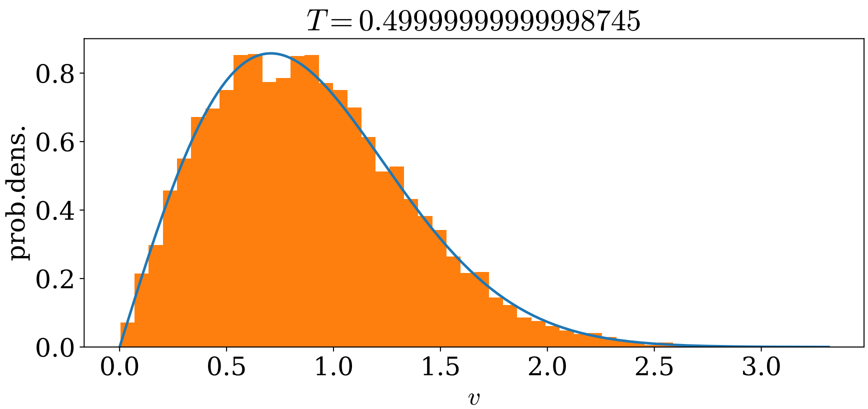


Figure 4: The velocity distribution is a good fit with the Maxwell-Boltzmann distribution, shown as a blue line.

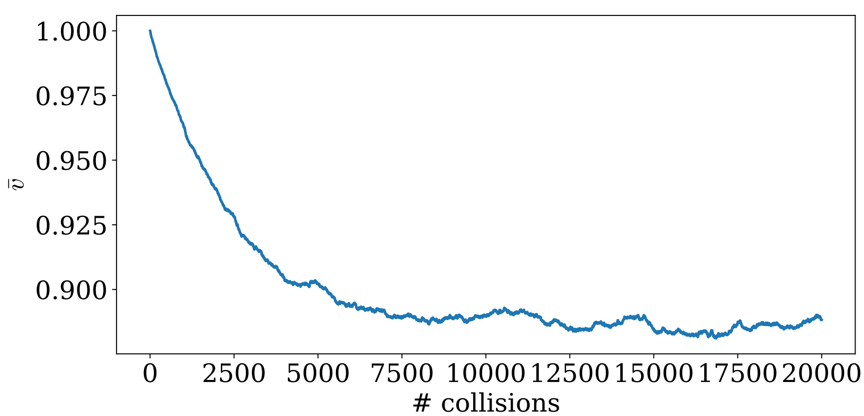


Figure 5: Average velocity, as a function of collisions. The distribution reaches equilibrium around 6000 collisions, or $3N$