Final Project Proposal

Kyle D. Patterson

May 2018

Overview I propose that I will create a 2D particle-based simulation to model heat transfer in a computer case. Specifically, I am interested in testing the effect on the CPU temperature over time of different case volumes, case shapes, numbers of fans, fan directions, air molecule density, and rate of heat generation.

The System Assume that I will model an area of a case with only a rectangular region, of uniformly distributed temperature, for the heat spreader of the CPU. I will assume that the CPU draws a constant wattage and therefore generates heat at a constant rate. I will also model an area outside the case. In the open areas (i.e. areas without a wall of the computer case, the CPU heat spreader, or other particles), I will randomly generate a specific quantity of particles which represent the average molecule of air. The particles will initially have an ambient amount of kinetic energy and be travelling in random directions.

Heat Transfer As the CPU runs, it will generate heat, thereby raising its temperature by producing heat. Heat is a measure of energy transfer with units of joules. When heat is transferred from a solid metal to molecules of air (e.g. H2O, N2), the energy transfer can be in potential or kinetic energy. I will assume that these molecules do not change phase or react and that they only exchange kinetic energy. This is a reasonable assumption since with the exception of water vapor, which is not a large part of air, most gases do not change phase around room temperature. Thus, the molecules will transfer kinetic energy (according to Newton's Third Law of equal and opposite reactions) when they bump into one another. The molecules will additionally change speed accordingly (kinetic energy

$$\text{KE} = \frac{1}{2}mv^2$$

with mass m and speed v). Since I am only modeling change of kinetic energy, these collisions must be elastic, meaning kinetic energy is conserved. Although some collisions in a gas are not elastic, others are super elastic meaning that they gain energy, so on average the collisions are elastic. So, elasticity is a reasonable assumption. After bumping into one another, the particles in my system will change velocity but conserve both kinetic energy and momentum. Thus,

$$m_1u_1 + m_2u_2 = m_1v_1 + m_2v_2 \qquad \text{(Conservation of momentum)}$$

$$\frac{m_1u_1^2}{2} + \frac{m_2u_2^2}{2} = \frac{m_1v_1^2}{2} + \frac{m_2v_2^2}{2} \qquad \text{(Conservation of energy)}$$

$$\Rightarrow v_1 = \frac{u_1(m_1 - m_2) + 2m_2u_2}{m_1 + m_2} \text{ and } v_2 = \frac{u_2(m_2 - m_1) + 2m_1u_1}{m_1 + m_2}$$

The Particle Different kinds of molecules have different specific heats and are different sizes, but air is locally (on the scale of centimeters) very similar in composition. So, I will make the simplifying assumption that a composite molecule will effectively mirror the heat-transfer phenomena of a more complex system with multiple kinds of molecules.

The CPU The final goal of this simulation will be to do thermodynamics experiments, tracking the temperature of the CPU, with different case volumes, case shapes, numbers of fans, fan directions, air molecule density, and rate of heat generation. The CPU will generate heat at a constant rate and the change in energy of the molecules of the heat spreader will be equal to this amount. When air molecules pass through the rectangular region in the case where the heat spreader is, I

will calculate a weighted average of kinetic energies between the heat spreader and the molecule (based on an experimental parameter for how much more mass the heat spreader should count for) and then redistribute heat. For example,

$$\mathrm{KE}_{1_{\mathrm{new}}} = \frac{m_1\mathrm{KE}_1 + m_2\mathrm{KE}_2}{m_1 + m_2}$$

Furthermore, since I am simulating with the assumption that energy will only be transferred in the form of kinetic energy, I can relate the heat generation, not removed by air molecules by

$$KE_{\mathrm{avg}} = \frac{3}{2}kT$$

where k is Boltzmann's constant ($1.38 \cdot 10^{-23}$ joules per kelvin).

Code The simulation will consist of a step function; a two dimensional array to store the positions of case edges, the CPU, and the particles; and a class to create particle objects with the properties of position, velocity, acceleration, and kinetic energy. I will express output using a graph of CPU temperature and a GIF of particle movements. If I have time, I will also produce a graph of pressure.

Exercises

1. Collision of two particles of equal mass and velocity heading toward each other (to test collision system)
2. Collision of two particles of unequal velocity but equal mass (to test collision system)
3. Box with higher density of particles on one side (to test whether particles reach in gradient distribute more uniformly over time)
4. Different numbers of particles with CPU heat spreader without a case to see how purely air density effects cooling (more particles should be better...)
5. Test the above exercise with a case that is not thermally conductive
6. Make the case thermally conductive
7. Test 3 different case volumes
8. Add a couple fans in the case. The fans will not actively pull the particles toward them but will function like leakage channels: the fans will allow molecules to enter only from one side. The fans will also increase the velocity/KE of the molecules that have passed through, thereby creating a vacuum inside the case, which should have a "pulling" effect and a high pressure zone outside the case
9. Test number of fans facing inward versus outward and position of fans. Possibly create a function to measure pressure by counting number of particles per volume inside/outside case.

Validation For code validation, I will investigate edge cases in addition to testing functions like collision and heat redistribution in the above exercises.

1. Is it possible for particles to spawn on top of each other? If so, do they get stuck or what happens?
2. What happens when a particle reaches the edge of the simulation?
3. How does changing the time step affect collisions and other parts of the simulation? Does the time step need to be smaller than a certain amount for the simulation to work correctly?
4. Do particles ever get stuck or have glitchy behavior?