CS 190C
Science Education in Computational Thinking

Case Studies in Computational Physics

Last time we introduced some concepts of and ways of thinking about thermal physics to provide a context in which we could illustrate the productive role that computational thinking can play in physics.

Today, we continue our exploration of the way in which computational thinking helps us to construct a model that predicts properties of an ideal gas in thermal equilibrium.

We also discuss the construction of a computational model that predicts properties of a magnetic film in thermal equilibrium. This system is interesting because its equilibrium properties are dramatically different when its total energy is above or below a critical energy. It undergoes a phase transition.

Mark Haugan, 2/27/2008

Q1: A bag contains a large number of balls that are the same size and weight but which come in 5 different colors.

You do the following 1000 times: Choose a ball from the bag, make a note of its color, put the ball back in the bag and shake the bag.

In the end, you find that 70 of the balls you chose were green and that 350 of them were red.

What is the relative probability of getting a green rather than a red ball on your next choice?

1. 0.07
2. 0.35
3. 0.20
4. 0.42
5. 5.00

Modeling a 1-d Ideal Gas of N Particles and Total Energy E

We noted last time that particle energy in an ideal gas does not depend on particle position. Consequently, we can represent the microstate of an N-particle ideal gas system as an N-element list or 1-d array of numbers, each one representing the velocity component of a particle.

$$[V_1, V_2, \ldots, V_N] = [523.4, -245, \ldots] \text{ m/sec}$$

The total energy of an ideal gas, which defines its macrostate, is simply the sum of the kinetic energies of its particles. The mass of each particle is m.

$$E = \frac{m}{2} \sum_{i=1}^{N} V_i^2$$

Since we use the demon algorithm to generate new microstates of the gas, we must define an initial microstate of the combined gas-demon system that has the energy that we want our model gas to have. For example, we can assign each gas particle an Nth of the desired total energy and the demon zero energy.

Generating New Microstates Using the Demon Algorithm

Select a particle at random and consider making a random change in its velocity component, ΔV. The corresponding change in the energy of the particle is

$$\Delta E = \frac{m}{2}(V + \Delta V)^2 - \frac{m}{2}V^2$$

- If this is negative, accept the change (the new microstate) and increase the demon's energy by a corresponding amount so that the energy of the combined gas-demon system does not change.

- If it is positive and less than or equal to the demon energy, accept the change (the new microstate) and decrease the demon's energy by a corresponding amount so that the energy of the combined gas-demon system does not change.

- If it is positive and greater than the demon energy, do not accept the change (the new microstate) and try again.

A "step" in a Monte Carlo simulation of an N-particle gas consists of N of these trials.

After each "step" you store or display information about the microstate of the gas needed to predict the gas properties you are interested in. For example, as we discussed last time, you can use the distribution of demon energy that accumulates during a simulation to estimate the temperature of the gas.

After a specified number of "steps" you stop the simulation.

Magnetic Systems and the Ising Model

Magnetic fields are generated by electric currents.

Atoms can generate such fields because of the currents circulating within them and the particles they consist of.
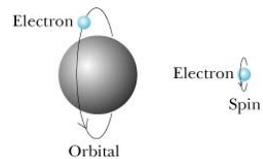

The magnetic fields generated by single atoms are minute, but if many of them are aligned so that currents circulate in the same direction the magnetic fields can add up to become significant. Remember how big Avogadro's number is!

Ferromagnetic materials like iron and nickel have interactions between their atoms that can keep the atoms aligned in this way at ordinary temperatures. However, if you heat a magnet, you discover that for every ferromagnetic material there is a critical temperature above which the atoms no longer remain aligned. The material is not magnetized about this temperature, its Curie temperature.

The Ising model of a 2-d film of magnetic atoms assumes very simple interactions between the atoms but does predict a phase transition, a complex collective phenomenon.

The atoms of an ideal solid film are arranged in a regular lattice configuration (it is as if each atom is located in the cell of a grid like the ones you used in your percolation project).
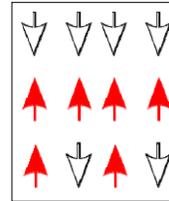
$$M = \mu \sum_{i=1}^{N} s_i$$

The figure shows one microstate of a 3 by 4 lattice of spins. Arrows depict the magnetization of atoms. The Ising model is quantum mechanical, so, the arrows may only point in one direction or in the opposite direction. We generally say that they point "up" or "down". The magnetization of the system is the (vector) sum of the magnetization of its atoms.

The Total Energy of an Ising System

As we saw in the case of the ideal gas, the thermal properties of a system reflect the way in which energy is shared among the system's constituents in different microstates.

To predict properties of the Ising model we must, therefore, understand how the total energy (macrostate) of the magnetic system is related to its microscopic structure. The Ising expression for the energy is determined by the interaction of atoms with their nearest neighbors in the lattice,

$$E = -\frac{J}{2} \sum_{i=1}^{N} \sum_{j=nn(i)} s_i s_j$$

The "coupling constant" J is positive for a ferromagnetic system. The energy of spins that are aligned is then lower than when they point in opposite directions.
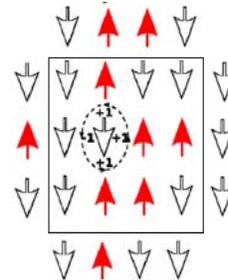
Notice that atoms at an edge of the lattice have different environments. To remove "edge effects" and, so, model a larger magnetic system, the grid's edges are identified (magnetic film on the surface of a torus [donut]!).

Modeling a 2-d Ising System and Total Energy E

We can represent the microstate of an *n* by *n* lattice of magnetic atoms using an *n* by *n* array whose elements have the values plus or minus one, representing the "up" or "down" orientation of each spin. The symbols s_i and s_j in the expressions for the energy and magnetization of the Ising system represent the values of the elements of the array representing the orientation of the ith and j[th] atoms.

$$E = -\frac{J}{2} \sum_{i=1}^{n^2} \sum_{j=nn(i)} s_i s_j \qquad\qquad M = \mu \sum_{i=1}^{n^2} s_i$$

Part of the cleverness of using the demon algorithm to predict the thermal behavior of this system is that it avoids the relatively complicated computation of the system's energy during the simulation. We set the energy (macrostate) when we define an an initial microstate for the system, and the demon algorithm automatically maintains this energy throughout the simulation.

Although it is usually an improbable microstate of the system, we are free to define an initial microstate in which all of the atoms are aligned (all s_i values the same so that $E_{init} = -2\,J\,n^2$) and the demon has energy $E_d = E_{sys} - E_{init}$, where E_{sys} is the energy of the macrostate we want to simulate.

Generating New Microstates Using the Demon Algorithm

Select an atom at random and consider making a random change in its orientation.

Compute the corresponding change in the energy of the atom. For example,

Before   After
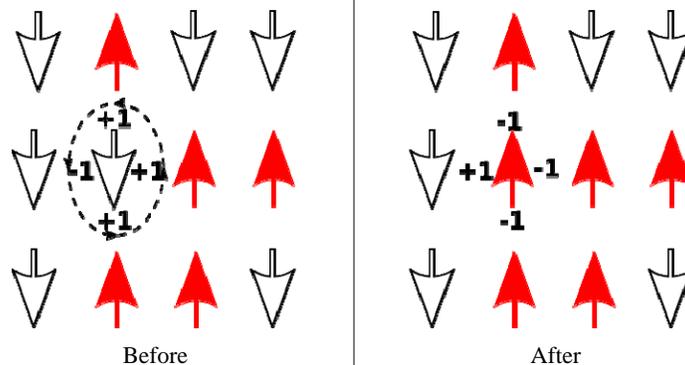
$$\Delta E = -2J$$

- If this energy change is negative, accept the change (the new microstate) and increase the demon's energy by a corresponding amount so that the energy of the combined gas-demon system does not change.

- If it is positive and less than or equal to the demon energy, accept the change (the new microstate) and decrease the demon's energy by a corresponding amount so that the energy of the combined gas-demon system does not change.

- If it is positive and greater than the demon energy, do not accept the change (the new microstate) and try again.

_____

A "step" in a Monte Carlo simulation of an n^2 atom system consists of n^2 of these trials.

After each step, we can compute and display the system's magnetization to see how it behaves when the system is in equilibrium.

Ising2dDemon2.py

We terminate simulation after a specified number of steps.