PHYSICS 115/242

Molecular Dynamics Project

Due in class, Friday, May 9.

Give yourself plenty of time for this assignment. Don't leave it to the last minute.

In this mini-project you will show that an interacting, classical system comes to equilibrium in which the probability of having a particular configuration in "phase-space" (i.e. the positions and momenta of the particles) is given by the Boltzmann distribution

$$P(\{x,p\}) \propto \exp(-E(\{x,p\})/k_B T) \tag{1}$$

where $\{x,p\}$ refers to the set of all positions and momenta, E, is the energy, k_B is Boltzmann's constant, and T is the temperature in Kelvin.
Here we will consider particles of unit mass moving in one dimension. Hence the Hamiltonian (energy) is given by

$$E = \sum_{i=1}^{N} \tfrac{1}{2} v_i^2 + \sum_{i<j} V(x_i - x_j)\,. \tag{2}$$

The probability that a particle has velocity v in equilibrium is therefore given by

$$P_{\text{equil}}(v) = \frac{1}{\sqrt{2\pi T}} \exp\left(-\frac{v^2}{2T}\right)\,, \tag{3}$$

where, from now on, we use units where $k_B = 1$. The coefficient in front of the exponential is to ensure that the probability is normalized. You should be able to show that the mean square velocity is given by

$$\langle v^2 \rangle_{\text{equil}} = T\,. \tag{4}$$

Consider a system of particles coupled by "anharmonic springs" such that they are in equilibrium (i.e. the potential is a minimum) if the spacing between them is 1, and the potential energy between an adjacent pair is given by

$$V(x_i - x_{i+1}) = \tfrac{1}{2}(x_i - x_{i+1} - 1)^2 + \tfrac{1}{4}(x_i - x_{i+1} - 1)^4 = \tfrac{1}{2}(y_i - y_{i+1})^2 + \tfrac{1}{4}(y_i - y_{i+1})^4\,, \tag{5}$$

where

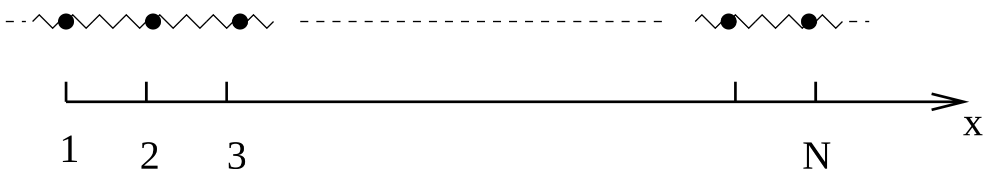
$$y_i = x_i - i \tag{6}$$

is the *deviation* of the particle away from its equilibrium position $x_i = i$. (It will be most convenient to use y_i as the basic variable in your calculations.) Assume that a particle only interacts with its two neighbors; see the figure below. It follows that the force on particle i is given by

$$F_i = -\left[\frac{\partial V(y_i - y_{i+1})}{\partial y_i} - \frac{\partial V(y_{i-1} - y_i)}{\partial y_i}\right] = f(y_i - y_{i+1}) - f(y_{i-1} - y_i) \tag{7}$$

(notice the signs) where

$$f(y) = -y - y^3\,. \tag{8}$$



You are recommend to use "periodic boundary conditions" in which the right hand neighbor of the last particle ($i = N$) is the first particle ($x = 1$) and vice versa. (This means that the system has no edges.)

Start the system off with the positions of the particles at the minimum of the potential energy, $x_i = i$, and let the velocities, v_i, take values ± 1 with equal probability. In fact it is best to choose *exactly* half the particles to have $v_i = +1$ and half -1. The reason is that the net total velocity ($\sum_i v_i$) that occurs with a completely random choice for the sign of v_i is of order $\sqrt{N}$ and is not negligible for a small number of particles. Furthermore it is *conserved* (momentum conservation). A C routine which will generate exactly half the velocities to be $+1$ and the other half to be -1 at random is given at the end. The initial distribution of velocities is therefore

$$P_{\text{init}}(v) = \frac{1}{2}\left[\delta(v-1) + \delta(v+1)\right], \tag{9}$$

which is far from the (equilibrium) Boltzmann distribution in Eq. (3).

The energy is given by

$$E = \frac{N}{2}. \tag{10}$$

The energy will stay constant but, it is a central assumption of statistical mechanics, that, after an "equilibration" time, it will be divided between the potential and kinetic energy according to the Boltzmann distribution. In particular the probability of a particle having velocity v will be given by Eq. (3).

For this project you are to show that this is true and determine the temperature T the system settles down to. More precisely, you need to do the following:

1. Decide on a value of N. Don't make it too large. N in the range 20–30 should do.

2. Decide on an algorithm for integrating the equations of motion. I strongly suggest the position/velocity Verlet (leapfrog) method since this is simple and, being symplectic, the energy will stay close to its initial value.

3. After some time t_{equil} (which you may need to estimate by trial and error) compute $\langle v^2 \rangle$ by averaging both over the N particles and times greater than t_{equil}. Hence estimate the temperature according to Eq. (4).

4. Produce a histogram of the velocity distribution and show that it fits the expected Gaussian distribution in Eq. (3).

5. *Optional:* Consider a pair of neighboring particles and show that their relative separation, $y \equiv y_{i+1} - y_i$, has a Boltzmann distribution

$$P(y) \propto \exp\left[-\frac{(y^2/2 + y^4/4)}{T}\right] \tag{11}$$

with the *same* T as found from the velocity distribution.

Note: Please ask me, well in advance of the deadline, if you are not clear what is expected. A crucial part is to get the force on a particle correct. Remember each particle interacts *both* with the particle to its left and the one on its right.

```
/**************************************************************************
This program creates a randomized array of initial velocities of either -1
or 1, with N/2 of each.  I choose N/2 sites randomly. Sites in this list
are given velocity +1, the others -1.

Original version by Deva O'Neil.  Modified by Peter Young.
**************************************************************************/
#include <time.h>
random_vs (int N, double v[])
{
    int k, num;
    double sum;
    int taken[N] ;   // shows whether a given number is already taken

    srand(time(NULL));  // Seed the random number generator

    for(k=0; k<N; k++)
    {
        v[k] = -1.0;    // Initially set all the velocities to be -1
        taken[k] = 0;   // Indicates that all sites are initially not taken
    }

    for(k=0; k<N/2; k++) // Main loop gives k/2 random sites with +1 velocities
    {
       while(1)          // Continue until get a number that isn't taken
       {
            num = rand()%N;
            if (taken[num] == 0) // We have a number that isn't taken
            {
                taken[num] = 1;  // num is now taken
                break;           // Exit from the loop
            }
       }
       v[num] = 1.0;            // Set velocity of site "num" to be +1
    }                           // End of main loop

                         // Omit the rest from here once the routine works
    sum = 0              // Test if it worked. Print out v[k]'s and the sum.
    for(k=0; k<N; k++)
    {
        printf("v[%2d]= %10.4f\n",k,v[k]); // Print out the velocities
```

```
        sum += v[k];
    }
    printf("sum  = %10.4f\n", sum); // Print out the sum; should be zero.
}
```