MM553/837 - Computational Physics Assignment 1 (Molecular Dynamics) - Due Oct. 21st, 2019

This assignment concerns applying the techniques we have developed for integrating the equations of motion to a particular system. Any of the algorithms we have discussed in class (and any of the corresponding C++ code) may be used. NB: this assignment is somewhat difficult and involves a non-trivial amount of coding. Please do not leave it until the last minute! Also, if you get stuck or have any question about what is required please ask me or Michael for help.

The system in question consists of N particles moving in one dimension. These particles all have unit mass and interact with their immediate neighbors via a spring-like potential.i Assume that the system has fixed boundary conditions so that particle N and particle 1 are always fixed to thier equilibrium positions $x_1(t) = x_N(t) = 0$. The Hamiltionian (total energy) of the system is thus

$$H = E = \sum_{i=1}^{N} \frac{1}{2} p_i^2 + \sum_{\langle i,j \rangle} V(x_i - x_j),$$

$$V(x_i - x_{i+1}) = \frac{1}{2}(x_i - x_{i+1})^2 + \frac{1}{3}(x_i - x_{i+1})^3 + \frac{1}{4}(x_i - x_{i+1})^4.$$

where x_i is the displacement of the ith particle from its equilibrium position and the sum in the potential term is over all adjacent pairs of particles. Note that each pair of particles is counted only once.

- Find Hamilton's equations for this system (analytically by taking derivatives) and provide an expression for the force on a single particle.
- Decide on a value of N (larger than 100 but smaller than 1000 should be fine, but more is always better!). Start the system in the configuration where all the $x_i = 0$ so that the potential energy is zero and

$$p_i(t = 0) = \sin\left\{\frac{2\pi(i + \delta)}{N}\right\},$$

where i on the RHS is the particle index (not the imaginary unit!) and δ can take any value in the interval $[0, 1]$ except 0, 1 or 1/2. To understand the significance of these initial conditions, do the following analysis. Consider $p_i(0) = f(s)$ to be a function of a continuous variable s instead of the discrete index i, but note that because of the boundary conditions $f(0) = f(N) = 0$. Next examine the Fourier transform of $f(s)$ defined as

$$f(s) = \int dk \, \mathrm{e}^{iks} \, \tilde{f}(k)$$

and find restrictions on the allowed values of k based on the boundary conditions. If $\delta = 0$, what is $\tilde{f}(k)$?

- Decide on an algorithm for integrating the equations of motion. Choose a small enough step size to ensure that the integration is accurate. Monitor the total energy violations and ensure that they are small. You should make a plot showing the energy as time evolves, to illustrate that it is approximately constant.

- Run the system for some time to allow it to equilibrate, and then take measurements of the average velocity $\langle v^2 \rangle$ over all particles at a fixed time. Plot $\langle v^2 \rangle$ as a function of time. Throughout this integration over time, make sure that the violations of the total energy are small. From this plot, does it appear that the system stays in a single Fourier mode, i.e. is v^2 roughly constant?

- Finally, take all of these individual measurements of the velocity at a single time and put them in a histogram[1] to get an idea of how they are distributed. Statistical mechanics states that in equilibrium at a temperature T, the probability of finding a particular particle with velocity (magnitude) v is given by the Maxwell-Boltzmann distribution

  $$P_{\mathrm{eq}}(v) = \sqrt{\frac{2}{a^6 \pi}} v^2 \exp\left\{ -\frac{v^2}{2a^2} \right\}.$$

  Does your histogram look like this distribution?

- EXTRA CREDIT/MM837: Repeat the above process with the (anharmonic) cubic and quartic terms in the potential energy removed. How does the plot of average velocity as a function of time and the histogram change?

- MORE EXTRA CREDIT/MM837: How should the computational time scale with the number of particles N? How should the step size ϵ change so that the energy violations are roughly constant as N is increased? Take some timings of runs with different N and make a scaling plot of the execution time vs. N. NB: you may find the C++ STL library `<chrono>` helpful.

[1] An example of how to make a histogram with `matplotlib` can be found at `http://matplotlib.org/1.2.1/examples/pylab_examples/histogram_demo.html`.