Physics G4080/G6080 – Problem Set 4
Spring 2015
Due by 5:00 PM Wednesday April 15, 2015

1 Monte Carlo Simulation of Liquid Argon

The Monte Carlo method can be used to simulate argon liquid, at a given temperature, through the canonical ensemble. The results of a canonical simulation should agree very closely with the results of a microcanonical simulation (molecular dynamics), differing only due to the fact that the number of particles, while large, is not infinite.

Your molecular dynamics code can be easily modified to do a Monte Carlo simulation by replacing the Verlet algorithm part of the code with a Metropolis update for each particle and velocity. Of course, you do not need to do the velocity via Monte Carlo, since it is a Maxwell distribution, but it is a negligible computational overhead to include it and it keeps the changes in your code small.

Modify your argon molecular dynamics code to do a Monte Carlo simulation at a temperature of $T = 1.069$ and density $\rho = 0.75$. Measure the same variables as in problem set 2 and check that your answers agree. Include statistical errors for your results.

In the molecular dynamics simulations, the autocorrelation times for observables are related to physical quantities, since the evolution represents real dynamics of the system. For the Monte Carlo, the autocorrelation times reflect the algorithm used for the update. Quote measured integrated autocorrelation times for the measured temperature.

2 Cluster Algorithm for the 2d Ising Model

We discussed the the cluster algorithm for the 2d Ising model in class. We saw that the partition function could be written as

$$Z = \sum_{\sigma_i} \left\{ \sum_{n_{ij}} \prod_{<ij>} \left[(1-p)\delta_{n_{ij},0} + p\delta_{\sigma_i,\sigma_j}\delta_{n_{ij},1} \right] \right\} \exp(-B\sum \sigma_i) \qquad (1)$$

where $p \equiv 1 - e^{-J}$ and B is an external magnetic field. In class, we discussed a single cluster, recursive program to simulate the Ising model for $B = 0$ and a copy of this program is posted on the course web page.

In this problem, you are asked to modify this code for the case of a non-zero magnetic field. The sample program implicitly changes spins in the main array `spin` as it produces the cluster. With an external magnetic field, the spin of the cluster can only be determined after the cluster has been found, when the factor $\exp(-B\sum \sigma_i)$

can be evaluated. This means you can use the single cluster code, but must keep a copy of the original configuration in order to be able to return to it if the spin flip is not accepted. You will also need to keep track of the size of the cluster to evaluate the Metropolis step for the contribution of the external magnetic field.

Use this code to determine the magnetization as a function of temperature and compare it to the known result

$$M = \pm N_s \left\{1 - \sinh^{-4}(2J)\right\}^{1/8} \tag{2}$$

for $J > J_c$, where M is the magnetization and N_s is the number of sites. To extrapolate to the $B = 0$ limit, you will need to run simulations with $B > 0$ and extrapolate. You will also want to check that your results, for a non-zero value of B, are in the large volume limit.

The random.C code on the web page uses a random number generator called rand3(). You must compile this with your program and link it. This can be done using

gcc -o cluster cluster.C random.C

on a Linux/Mac machine.

3 Correlations in the 2d Ising Model (G6080 only)

As the Ising model approaches its critical point, the size of spatial clusters grows. It is this growth in the average size of clusters which is responsible for the eventual phase transition and the slow evolution through phase space of the simple, local site Metropolis algorithm. In this problem, you should measure the spatial correlator

$$\langle \sigma(\vec{x})\, \sigma(\vec{y}) \rangle \tag{3}$$

as you approach T_c from below. Note that since this quantity is unchanged by flipping all spins, you don't need an external magnetic field, as in problem 2, to keep it in a single magnetization state.

To get good statistics, you need to average this over all $(\vec{x}, \vec{y})$ pairs with a fixed $\vec{x} - \vec{y}$. As a simplification, you can concentrate on locations where $\vec{x} - \vec{y} = m\hat{x}$ or $n\hat{y}$. Note that rotational invariance allows you to average measurements with the same values for m and n.

3.1 Simple Observable

Start by measuring the correlation in Eq. (3) somewhat below T_c. This correlation function will generally fall exponentially, as $\exp(-|\vec{x} - \vec{y}|/a(T))$, so start where the rate of exponential fall-off is about 1/2 in lattice units, *i.e.* $a(T) \sim 1/2$. Increase T towards T_c measuring a few values for $a(T)$ between 1/2 and 1/8. For larger values of $a(T)$, you may need to work on a large volume. Plot $a(T)$ versus T.

3.2 An Improved Estimator

Consider the correlation in Eq. (3) and the cluster algorithm. Comparing the values that enter this measurement before and after a call to the flip() function, we readily see that if $\vec{x}$ is in the cluster and $\vec{y}$ is not, the value for those two points in the average of the correlator is zero, since the spin at the point $\vec{x}$ is of either sign with equal probability. Implement this improved method for measuring the observable and compare with the simple implementation from the first part. Is the error on the measurement reduced? Can you measure $a(T)$ values smaller than 1/8?