A source file is given in c++. You can start from that or write your own code in your favorite language.

HOMEWORK 11

from Cornell University - course by prof Zabaras

Problem 1: Condensation and Pattern Formation in a Binary Lenard Jones System

In this homework we will perform Monte Carlo simulations for a two-dimensional system containing argon and krypton. In order to do this, we will use a Monte Carlo code provided by Veera Sundararaghavan. The source code in C++ is included in this ~~homework.~~ The code uses periodic boundary conditions and performs the simulation using reduced LJ units and normalized atomic coordinates. Two moves are coded: The jump move, in which an atom of one type jumps to a new location, and the exchange move, where an atom switches types. This type of MC dynamics is known as Glauber Dynamics. Please look through the source code to see how an MC program works.

MC-alloys-software.rar

In main.cpp you will find the general method for this program is this:

1. Read in input from the user. Input is fraction of the system which is argon, the initial number of atoms, the number of Monte Carlo steps to perform, and the temperature of the system. Note that the initial density is hardcoded into the program at 0.025 atoms/squared angstroms. This means that is the size of the system is defined by the number of atoms you input.
2. Define the "system". In this program, system is defined as a data type which holds a collection of "atoms" (another defined data type). In order to define the system, all parameters are converted in terms of the Lenard Jones parameters (reduced LJ units). The size of the edge, as well as the potential cutoff and verlet cutoff are defined. The jump size is also defined.
3. MC moves are performed. Look at the function void MC( ) for details. You will see that moves are accepted or rejected based on energy calculations and parameters like jump size (or in the case of exchange, calculated energies and type of atom).
4. At every MC step, xyz coordinates and energies (in joules) are printed to two files: positions.xyz and energies.txt. The xyz file can be visualized using a program like VMD (you will need to register to download this software). or see JMOL jmol.sourceforge.net/ or whatever you want
5. The program stops when the time steps are completed.

A make file is provided to create the executable. To run it use the following format.

```
./main.exe (fraction of krypton atoms) (initial number of atoms) (total number of MC steps) (Temperature in Kelvin)
```

For example, for a run with 50% krypton, 1000 atoms initially, 100 MC steps, and a temperature of 200 K, type:

```
./main.exe 0.50 1000 100 200
```

You can also use the batch file provided by simply typing

```
./a.bat
```

In order to explore our system, perform the following two tasks.

a) Use a system of 1000 atoms with a composition of 100% Krypton atoms. In this case, the exchange move will not be activated. Plot any one sample of the system after convergence at room temperature (300 K) and below the condensation point (17 K). Comment on the general behavior of the system.

b) Repeat (a) with 80% Krypton and 20% Argon. Comment on the differences.

NOTE: When plotting in VMD, use "points" drawing method and increase the size of points (using the slider) for clarity.

Problem 2: The Potts Model

In this problem, we will explore the Potts Model. In the simple Ising model, particles are able to take one of two states. The Potts Model generalizes the Ising Model to handle particles with q number of states. In a q-state Potts model in two dimensions on a regular square lattice, there are spins s_i with q different states, s_i = 1, ..., q, and its action and partition functions are:

$$S = \beta \sum_{<i,j>} 1 - \delta(s_i\,, s_j)\,, \qquad Z = \beta \sum_{<s_i>} e^{-S}$$

where the sum goes over the nearest neighbor pairs on a square lattice. $\delta(s_i, s_j)$ is the delta function =1 for the same states s_i and s_j, otherwise zero. We assume here periodic boundary conditions using heat bath or Metropolis algorithm, for simulations on rectangular V=L_x x L_y lattices.

Absolute magnetization is given as:

$$|M| = \frac{3}{2}\left[\max(M_1, M_2, M_3) - \frac{1}{3}\right], \qquad where\ M_s = 1/V \sum_i \delta(s\,, s_i)$$

where normalization is such that the absolute magnetism lies between 0 and 1.

Perform a series of simulations on a 64 x 64 lattice in the β range of 0.5 to 1.5 at 10 different values of β and at least 30000 update sweeps for each value of β. Calculate $\langle |M| \rangle(\beta)$. Show that the results indicate a phase transition at $\beta = \ln(1 + sqrt(3))$. Do one simulation at this value of β. Plot $\langle |M| \rangle(\beta)$ and autocorrelations of $|M|$ as functions of β.

Included in this homework is a program for a 2D Ising model and a 2D Potts model, written for the course Monte Carlo Simulation at the University of Oulu in Finland. Three subdirectories with correct makefiles are provided one for each of the Ising, Potts and error calculations. In the directory Errors, you can compute the autocorrelation using the output files generated from the Potts or Ising programs (cut and paste the generated output in the directory Errors). The files associated with mersenne are for random number generation. When you run the programs, input is asked for values of β, output file, and random number seed. For all β cases, run both the Ising and Potts programs to identify similarities and differences regarding the location of the transition and the observed magnetization and autocorrelations of $|M|$ as functions of β.