This homework is due on Monday Nov 9.

The Ising model is written as,

$$E = J\sum_{\langle ij\rangle} S_i S_j = J\sum_{\langle ij\rangle}(2\sigma_i - 1)(2\sigma_j - 1) \tag{1}$$

where $S_i = \pm 1$ and we have written $S_i = 2\sigma_i - 1$. Note that $\sigma_i = 0, 1$ is the way we will store the spin values in our arrays. We now set $J = -1$ without any loss of generality. If we attempt to change the i^{th} spin, the ratio of Boltzmann weights is,

$$R = \frac{e^{-\beta E_{\text{new}}}}{e^{-\beta E_{\text{old}}}} = e^{2\beta(\sigma_i^{\text{new}} - \sigma_i^{\text{old}})(2(\sum_j \sigma_j) - 4)}, \tag{2}$$

in the last equation we have specialized to the square lattice with PBC where each site has 4 neighbors, the sum on j is over the nearest neighbors of i. Note since R only takes on a few values we store it in a vector of vector of vectors called "wght_tbl" that takes the possible values of σ^{new}, σ^{old} and $\sum_j \sigma_j$ as indices.

I. METROPOLIS ALGORITHM

Read the unfinished code "ising_met_hw.cpp" in the "code" folder. Finish all the areas where you find the comment "// ———COMPLETE CODE HERE———!"

Complete the code to simulate the Ising model using the single-spin flip Metropolis algorithm. For full credit on this part, make sure you show clearly in your write-up that you have debugged each subroutine, step by step. You should always assume your code has bugs in it, and you should make a convincing case that you have looked and found them all out! The program should print binned averages of e, e^2, m, m^2 (where the lower case indicates intensive quantities). *(120 points)*

To help you check if your code is working properly, please fill in this empty table with the results of your code (mc for monte carlo) and enum (full enumeration from last week) (Quote all values in units in which $J = -1$, when calculating χ use $\langle m\rangle = 0$ allowing you to use the standard error formula to estimate the error):

L	β	$Neql$	$Nmcs \times Nbin$	χ(mc)	χ(enum)	$\langle e\rangle(mc)$	$\langle e\rangle$ (enum)
4	0.5	10^3	10^7				
4	1	10^3	10^7				
4	2	10^3	10^7				
3	0.5	10^3	10^6				
3	1.0	10^3	10^6				

II. SIMULATION RESULTS

A. $C_V(T)$ and $\chi(T)$

In last week's homework we made a plot of these two functions using the exact enumeration method and were able to reach system sizes of $L = 4$. With the new MC algorithm we can simulate much larger lattices. Compute these functions for $L = 4$ and then $L = 8$ (which would be impossible with exact enumeration!). Make a graph showing the three curves together, the MC for $L = 4, 8$ and the exact enumeration for $L = 4$. What do you observe to $C_V(T)$ as L is increased? *(30 points)*

PS 1: Remember, you do not want to loop over the temperatures in the C++ code. In fact now that we have tested the C++ code for a single temperature we do not want to mess with it at all. Write a simple python script that will automate the collection of data at different temperatures.

PS 2: Computing errorbars for the specific heat is subtle. You do not need to do it this week. We will discuss how to do this properly in class shortly.