Problem Set 5 **R08244002 Shin-Rong, Tsai**

- **Single Cluster Algorithm**

Show that the single cluster algorithm satisfies detailed balance, which means this equation holds:

$$\delta\rho = \rho_C \cdot P(C \to D) - \rho_D \cdot P(D \to C) = 0$$

where

C, D: a set of spin lattice, where D is the result of C after flipping the spin inside the cluster.

P: propability from one set to another.

ρ, ρ_C, ρ_D: distribution density.

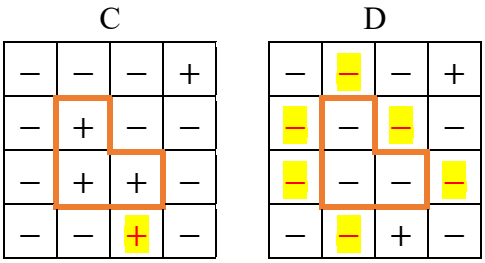
$\langle proof \rangle$:

$\delta\rho = \rho_C \cdot P(C \to D) - \rho_D \cdot P(D \to C) = 0$

$\Rightarrow \frac{\rho_C}{\rho_D} = \frac{P(D \to C)}{P(C \to D)} = e^{-\frac{E(C)-E(D)}{kT}}$ ∵ distribution follows partition function Z

For probability, it can be written as the probability of sequence of trial steps to reach some selection of a cluster (T), times the probability of accepting (A):

$$P(C \to D) = T(C) \cdot A(C \to D)$$
$$P(D \to C) = T(D) \cdot A(D \to C)$$



The probability of forming bound is P_b. And inside the two cluster in C and D respectively, their spins are opposite. For the orange cluster reaches its limit like this, which means not forming bound with neighboring spin at all, it has probability:

$P_b = 1 - e^{-2J/kT}$

$T(C) = (1 - P_b)^n$

$T(D) = (1 - P_b)^m$

n, m: # of neighboring spin same with cluster in C, D

For energy difference between two lattice C and D, only the neighboring spins matter, which is:

$$\Delta E = E(C) - E(D) = -J[(n - m) - (m - n)] = -2J(n - m)$$

So

$$\frac{\rho_C}{\rho_D} = \frac{P(D \to C)}{P(C \to D)} = e^{-\frac{E(C)-E(D)}{kT}}$$

$$\Rightarrow \frac{T(D)\cdot A(D\to C)}{T(C)\cdot A(C\to D)} = \frac{(1-P_b)^m \cdot A(D\to C)}{(1-P_b)^n \cdot A(C\to D)} = e^{\frac{2J(n-m)}{kT}}$$

$$\Rightarrow \frac{A(D\to C)}{A(C\to D)} = (1-P_b)^{n-m}\cdot e^{\frac{2J(n-m)}{kT}} = e^{-\frac{2J(n-m)}{kT}}\cdot e^{\frac{2J(n-m)}{kT}} = 1$$

$$\Rightarrow A(D\to C) = A(C\to D)$$

The final step after the cluster reaches the limit, we flip all spins no matter what, so that it satisfies detailed balance.

- **Two-Dimensional Ising Model**

1. Result

Run code *ising_model_2D.c*, output result as .txt file, and then analysis with python code (*exact_Cv.py*, *exact_X.py*, *cal_Cv_X.py*). Compare to exact solution of $\langle e\rangle$ and $\langle m\rangle$ by running the code *ising2.c* (mentioned in class), which gives:
At conditions : $\mathrm{N} = 100, \mathrm{J} = 1.0, \mathrm{B} = 0.0, \mathrm{T} = 2.26, k_B = 1.0$

	$\langle e\rangle$	$\langle m\rangle$	C_v	χ
Exact Solution	-1.440966	0.613371	1.30615×10^4	3.10507×10^5
Metropolis	-1.44531 $\pm 1.823\times10^{-3}$	0.654918 $\pm 8.572\times10^{-3}$	1.77467×10^4 $\pm 2.062\times10^5$	2.26623×10^5 $\pm 3.748\times10^4$
Heat Bath	-1.45093 $\pm 3.509\times10^{-3}$	0.674597 $\pm 8.992\times10^{-3}$	3.18657×10^4 $\pm 1.586\times10^6$	2.855191×10^6 $\pm 7.245\times10^6$
Single Cluster	-1.44091 $\pm 1.063\times10^{-3}$	0.621728 $\pm 3.877\times10^{-3}$	2.14204×10^4 $\pm 1.399\times10^5$	6.85926×10^5 $\pm 1.446\times10^5$

2. Discussion

A. Finding C_v and χ

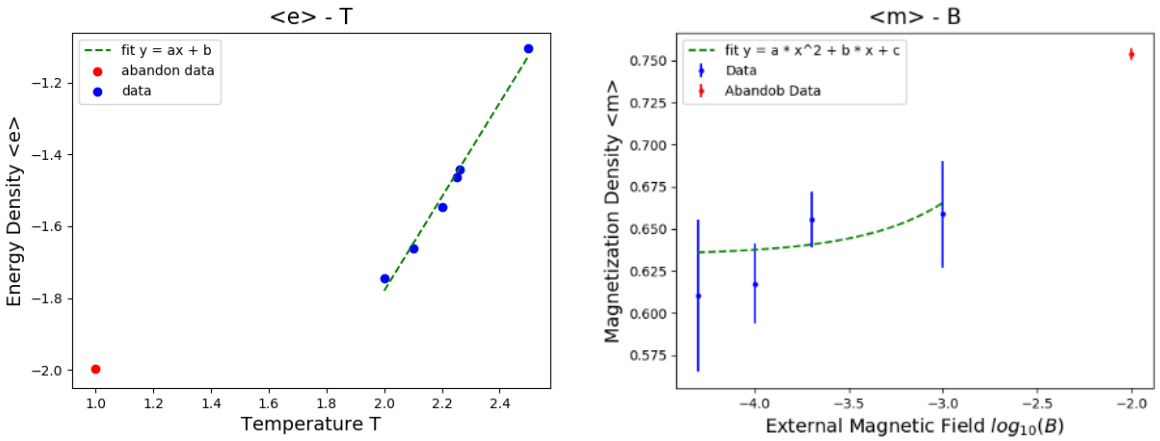
The exact solution of C_v and χ are calculated by finding the slope of $\langle e\rangle - T$ graph and $\langle m\rangle - B$ graph times total sites of the lattice.

$$C_v = \frac{\partial\langle E\rangle}{\partial T} = N_{site}\cdot\frac{\partial\langle e\rangle}{\partial T} \text{ and } \chi = \lim_{B\to0}\frac{\partial\langle M\rangle}{\partial B} = N_{site}\cdot\lim_{B\to0}\frac{\partial\langle m\rangle}{\partial B}$$

Since there are no exact solution for $\langle m\rangle$ for $\mathrm{B}\neq 0$, $\langle m\rangle - B$ graph's data is determined by altering the external magnetic field in Metropolis simulation (because both $\langle e\rangle$ and $\langle m\rangle$ are more closer to the exact solution, and the criteria has B dependency). And $\langle e\rangle - T$ graph is calculated by the exact solution of $\langle e\rangle$.

For $\langle e\rangle - T$ graph, I fit the data with $\mathrm{y} = \mathrm{a}\cdot\mathrm{x} + \mathrm{b}$, where the temperature (x axis) is close to 2.26, which means to abandon temperature at 1.0.

For $\langle m \rangle - B$ graph, I fit the data with $y = a \cdot x^2 + b \cdot x + c$, where B (x axis) is close to zero, then take the derivative at $B = 0$.



For the other simulation methods, I calculated by using these formulas:

$$C_v = \frac{1}{kT^2}(\langle E^2 \rangle - \langle E \rangle^2) \text{ and } \chi = \frac{1}{kT}(\langle M^2 \rangle - \langle M \rangle^2)\Big|_{B=0}$$

B. Accuracy

For $\langle e \rangle$ and $\langle m \rangle$, it seems that the accuracy of the algorithm is:

$$\text{Single Cluster} > \text{Metropolis} > \text{Heat Bath}$$

For C_v and χ, we can see that both approach are in the same order, though with great divergence. The solution of C_v is a lot promising than χ, since we have exact solution of $\langle e \rangle$. It would be nonsense if we compare the other results to the exact solution of χ, since it's estimated and calculated by metropolis' output.

Another fun fact is that when I try to calculate the error of the simulation through this,

$\langle E^2 \rangle = a \pm b$, where error is estimated by binning with Jackknife.

$\langle E \rangle = c \pm d, \langle E \rangle^2 \approx c^2 \pm 2 \cdot cd$, where error is estimated by binning with Jackknife.

$\langle E^2 \rangle - \langle E \rangle^2 \approx (a - c^2) \pm (b - 2 \cdot cd)$

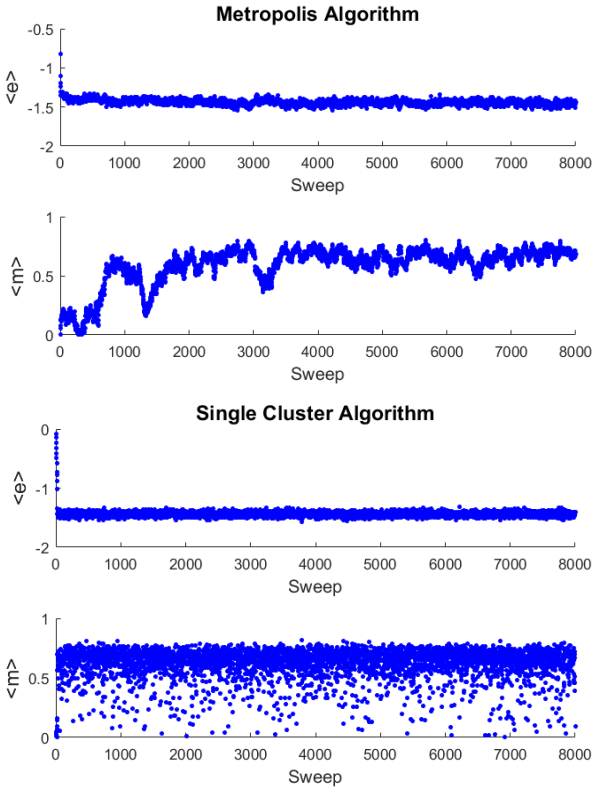
Most of them give really bad result, but I typed it in the result anyway.

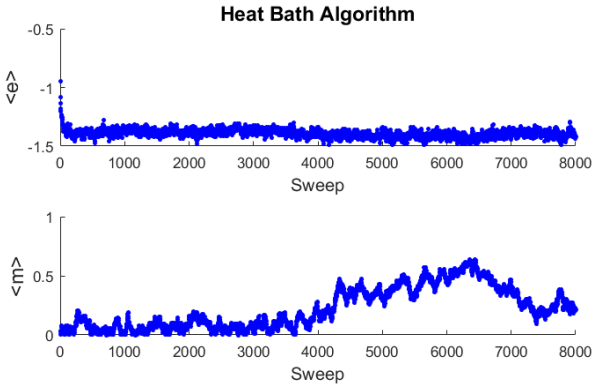
C. Sweeps needed to reach thermalization

Energy density is more stable than magnetization density.

And I think the stability of the $\langle m \rangle - Sweep$ matches the accuracy of the result. For Single Cluster, the magnetization density looks more neat. But for heat bath, it varies much. So the range and interval we measure affects the result widely, which might cause the inaccuracy.

Apart from this, it is obvious that the bigger the site of the lattice is, the more steps it needs to reach thermalization.





- **Integrated Autocorrelation Time in Monte Carlo Simulation**

1. Result

First, find out the critical temperature T_c for different box length of the square lattice, and generate their data as .txt file (in *findL-Tc.c*). Then use shell script (in *getCorLoop.sh*) to help analyze the data, since I write the auto-correlation function (in *ising_model_2D_auto.c*) in a form of reading command line argument. Finally, plot and fit the result using python (in *auto_fit.py*).

We get:

	Metropolis	Heat Bath	Single Cluster
Dynamical Critical Exponent	2.0519	1.5358	-1.2385

2. Discussion

For Metropolis $z \approx 2$ was mentioned in class, which matched the result here.

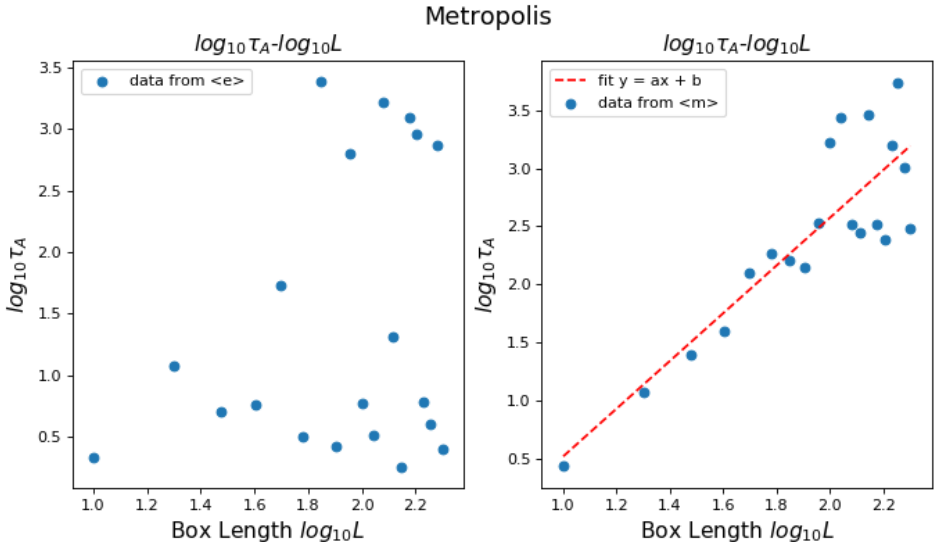
Dynamical Critical Exponent z:

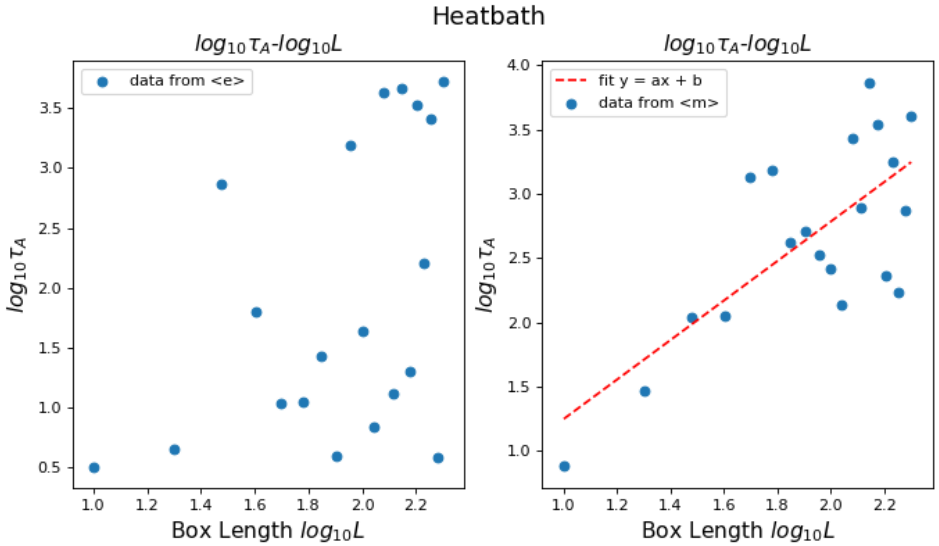
$$\text{Metropolis} > \text{Heat Bath} > \text{Single Cluster}$$

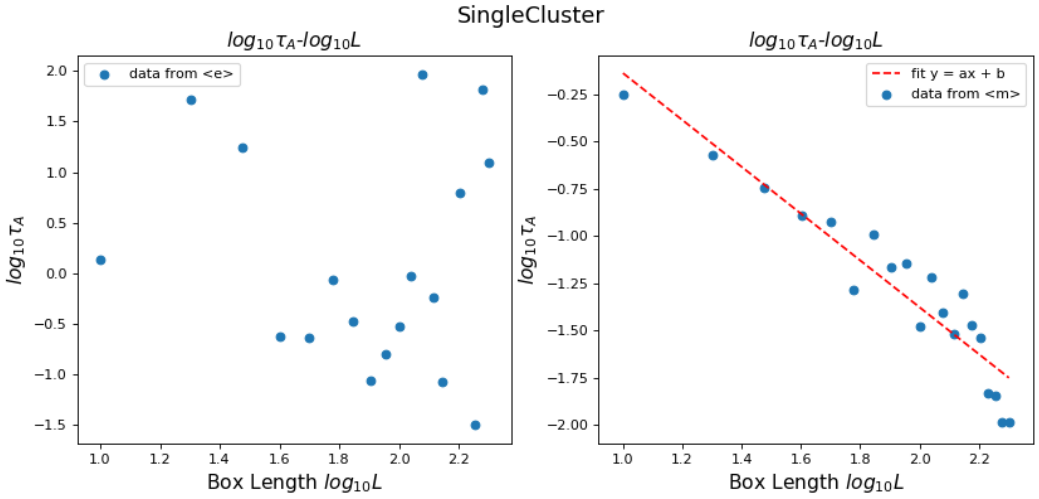
The higher z the algorithms have, the more likely they will suffer from critical slow down $\tau_A \to \infty$, where $\tau_A \sim \xi^z$.

It is not surprising that Single Cluster Algorithm has the lowest z, since there are no criteria to flip the spin or not, it doesn't depend on other spin.

All of their dynamical critical exponent are calculate from the observable $\langle m \rangle$, since the result from $\langle e \rangle$ scattered everywhere. I still don't know what is the problem here…







Through finding T_C for respective L for each algorithm, we can plot the result. For larger correlation length, the critical temperature is smaller. Since the higher the temperature is, the messier the spin configuration will be, which results in smaller correlation length and vice versa.

It looks like L and T_C has an inverse proportional relation.