Vishal Puppala

Using Monte Carlo Methods to Study the Ising Model

1. Background and Motivation:

The Ising Model is an important mathematical model in statistical mechanics for describing the spin states of a lattice of atoms. Modeling spin configurations of a system allows physicists to ascertain important quantities like the configuration's magnetization or potential for exhibiting a phase transition. This formulation models how a system consisting of a LxL square lattice where each site in the lattice has two possible (+1 or -1) spin states behaves in response to perturbations caused by applied magnetic fields or temperature changes.[2] One such important quantity that can be calculated from the Ising model simulations is magnetization which is directly proportional to the energy of the system and depends upon the degree to which the spin positions in the system are aligned. One such way to calculate quantities like the average magnetization of any given state would be via a brute force method where the magnetization of every state would be calculated and then a weighted average would be obtained where each state is weighted by its probability. However, this is computationally prohibitive since there are 2^n possible states, where n is the total number of lattice sites.[2]

2. Monte-Carlo Methods:

The nature of a high energy lattice of discrete magnetic spins can be modeled using Markov chains. The system can approach a more energetically stable configuration through a series of small-scale local changes that accumulate to produce a large-scale change over the course of time. Thus, by assigning each change a transition probability, we can use MCMC algorithms to approximate a target density to represent the final energetically stable configuration that is achieved after the aforementioned series of local changes. Thus, by allowing p for the exploration of as many states as possible, MCMC algorithms in theory, are capable of exploring complete phase space.[3]

MCMC algorithms are especially powerful because they leverage Monte-Carlo methods which allow for the approximation of high-dimensional probability distributions that are difficult to computationally simulate. The Ising model is especially computationally burdensome to simulate as the equations governing expected values of lattice configurations require knowing the partition function $Z = \sum_{s} e^{\frac{-\epsilon_s}{KT}}$. The partition function is the sum of the energy states of each possible state. Since there are LxL occupied sites with each site having two spins, the number of energy calculations would be in the order of $O(2^{n})$. Furthermore, ϵ can be calculated using the equation for the hamiltonian, which is just the total energy of the system, for a spin lattice which is $H =- J \sum_{<i,j>} \sigma_i \sigma_j$, where σ_i and σ_j comprise a pair of adjacent spin sites and J is the coupling constant. Z is a constant of proportionality needed to completely simulate the Boltzmann energy distribution. An alternative to computationally simulating the Boltzmann energy distribution is to approximate it as a target distribution using Monte-Carlo methods.[2] Using a MCMC algorithm, as the number of iterations n approaches infinity, the sample Monte-Carlo estimator should approach the population metric. Thus, after a certain period (burn-in length), the distribution simulated and displayed on a trace plot by a MCMC should converge upon the stationary distribution which is treated as the true distribution for the quantity we are trying to estimate.[5]

In order for MCMC algorithms to generate random variables, the Markov chains must also be aperiodic and irreducible. [1] By sampling sites at random in the Ising lattice, we can ensure that the chain remains aperiodic. Irreducibility refers to the potential to reach all other possible states from any given state. This can be maintained since the MCMC algorithms we are exploring are stochastic and not deterministic, meaning that one state does not determine with certainty the next state that will be reached.

3. Metropolis-Hastings:

We can use the Metropolis-Hastings algorithm to simulate the Ising model, if we are able to calculate the probability of the configuration being in any given state. In fact, this is called Boltzmann Distribution which is often expressed as $p_i \propto e^{-\epsilon_i/(KT)}$, where p_i is the probability of the configuration being in a given state i, $\epsilon_i$is the change in energy of a given conformation, K is Boltzmann's constant, and T is the temperature. ϵ_i will be expressed as the change in energy caused by perturbing the physical system.

Suppose we have a series of successive, sequential states starting with x_i=0 and proceeding with i=1,2,3,etc. Suppose we have a square lattice where each row has L occupied sites and each column has L occupied sites. Each occupied site has a spin of either -1 or +1. We can then uniformly sample a random site x_i on the lattice. Let p(x_i) be the probability of the current spin configuration at x_i. Let p(x'_i) be the probability of the current spin configuration at x_i being inverted ((i.e. +1$\rightarrow -$ 1)). We can then formulate the acceptance ratio as $\alpha = \frac{p(x_i')}{p(x_i)}$,

which simplifies to $\alpha = \frac{e^{-\epsilon_{x'_i}/(KT)}}{e^{-\epsilon_{x_i}/(KT)}} = e^{\frac{\epsilon_{x_i} - \epsilon_{x'_i}}{KT}}$. [2] Note that if ϵ_{x_i}>$\epsilon_{x'_i}$, then ϵ_{x_i}-$\epsilon_{x'_i}$ would be positive which is suboptimal, as we want the acceptance probability to be less than 1 in order for the candidate spin to not be rejected. Thus, candidate positions are more likely to be accepted if the new configuration is energetically higher.

Pseudocode:

```
Initialize spins of LxL lattice
for(i in 1:n):
    Randomly sample a site $x_i$ on a lattice where sites are evenly distributed.
    Invert $x_i$ to get candidate $x_i$'.
    Calculate acceptance probability of $\frac{p(x_i')}{p(x_i)}$.
    if U~(0,1)<$\frac{p(x_i')}{p(x_i)}$, accept candidate $x_i$'
    else, reject $x_i$'and keep $x_i$
```

Note that $p(x_i)$, as referenced in the pseudocode is estimated through the following equation: $e^{\frac{\epsilon_{x_i}-\epsilon_{x'_i}}{KT}}$, where $\epsilon_x =- J \sum_{<i,j>} \sigma_i\sigma_j$ where σ_i *and* σ_j can take values of either +1 or -1.

4. Gibbs Sampling:

Another Monte-Carlo Markov Chain method we can use is Gibbs sampling. For the Gibbs method, during each iteration, after x_i is randomly selected, the proposal density should be g(x)=p($x_i|x_0, x_1, x_2,, x_{i-1}, x_{i+1}, x_{i+2},, x_n$) where i is some value in the range of (0, n) where n is L x L or L^2. We can simplify this proposal density

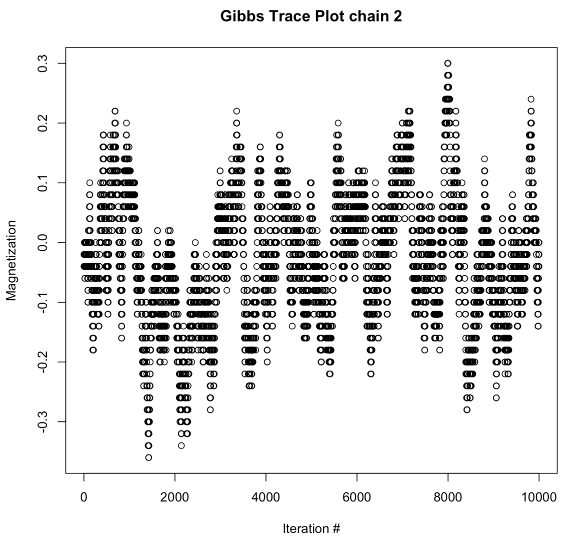
$$p(x_i|x_{-i}) = \frac{p(x_i)p(x_{-i}|x_i)}{p(x_{-i})}, \textit{ where } x_{-i} \textit{ is a vector of all states excluding } x_i$$

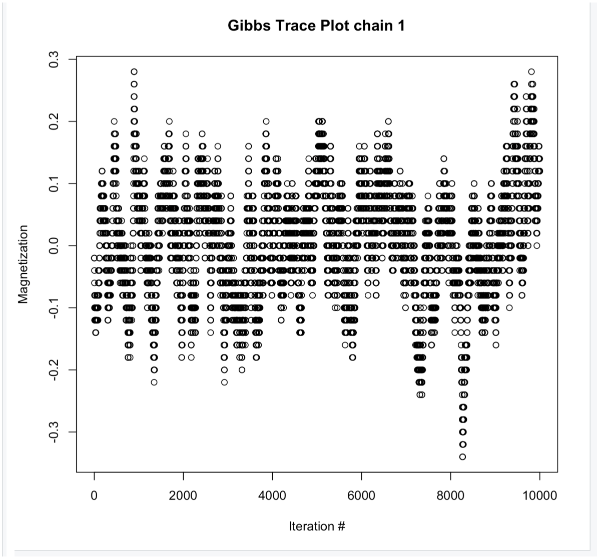
Since $p(x_i|x_{-i}) = \prod_{i=1}^{i=n} \mathrm{p}(x_1|x_{-1})\mathrm{p}(x_2|x_{-2})\ldots\mathrm{p}(x_{i-1}|x_{-(i-1)})\mathrm{p}(x_{i+1}|x_{-(i+1)})\ldots p(x_n|x_{-n})$

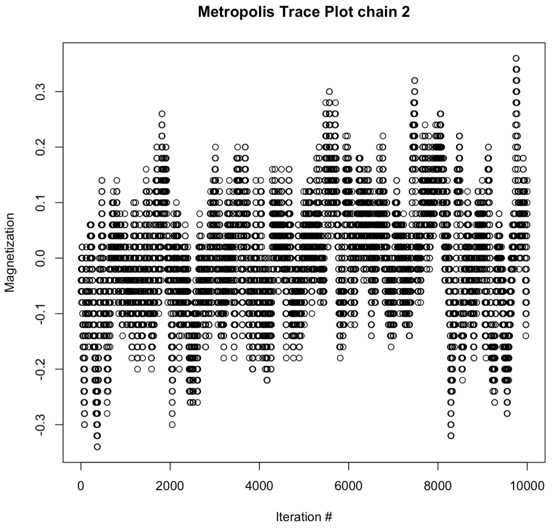
Thus sampling from $p(x_i|x_{-i})$ can be conceptualized as sampling from the joint distribution. Thus, since $p(x)_{x_i}$ is equivalent to p(x_i) which is why the Gibbs process is equivalent to using the Metropolis Hastings algorithm with acceptance probability always being 1.

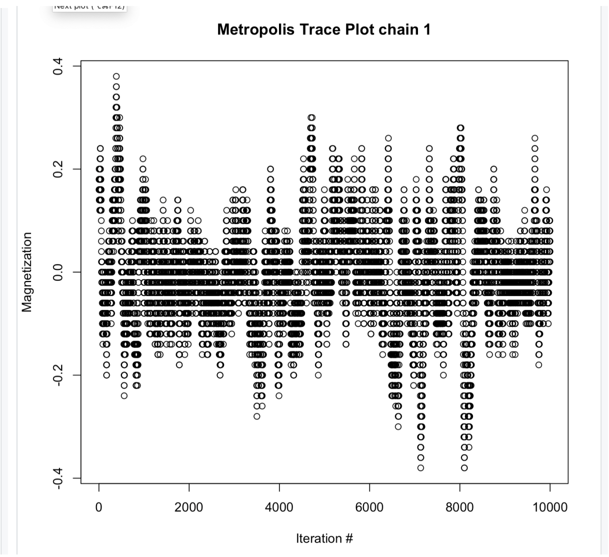
5. Results/Comparison of Two Approaches:

The algorithm (both general case of Metropolis-Hastings and Gibbs Sampling) was run on R studio and produced the following trace plots as shown below with hyperparameters temperature(T) and coupling(J) both arbitrarily chosen to as T=1, J=1 with the number of iterations chosen to be 10,000. Note that each algorithm was run twice to produce two Markov chains per algorithm, as this is a good practice to hedge against algorithms getting stuck in one area of a target distribution and consequently not being able to fully explore sample space. This works, as running an algorithm twice means that the starting value will differ, which can potentially lead to different Markov trajectories as each Markov state is fully only dependent on the state immediately reached prior to the current state and independent of all other states.





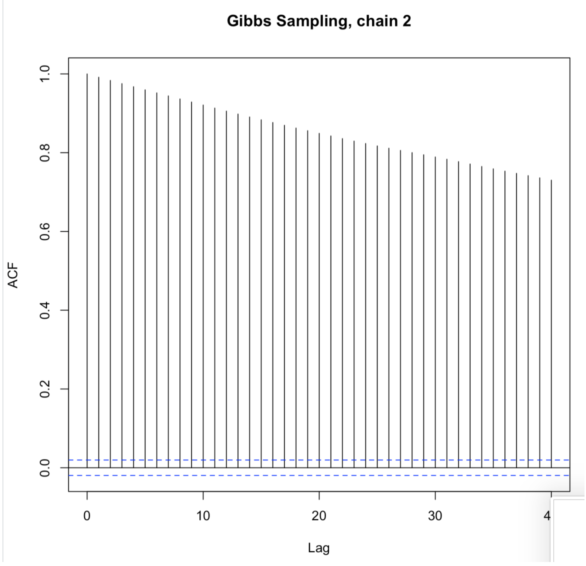




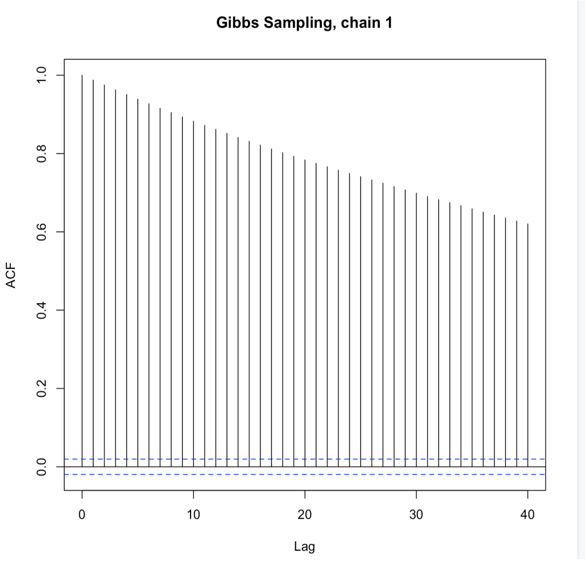
Visually inspecting the trace plots, it appears that there is good mixing and the algorithms seem to explore the phase space in an ergodic fashion, as evidenced by the path not converging upon straight horizontal lines but instead oscillating over various different values of the magnetization.

In terms of comparing the trace plots between Gibbs sampling and Metropolis-Hastings, I was not able to gauge any significant differences in the sample paths that would warrant any claims made in regards to which algorithm is better. This is surprising because I initially decided to experiment with Gibbs sampling because Thus, I decided to also examine autocorrelation as another metric to evaluate which methodology has the better performance for Ising Model simulation.

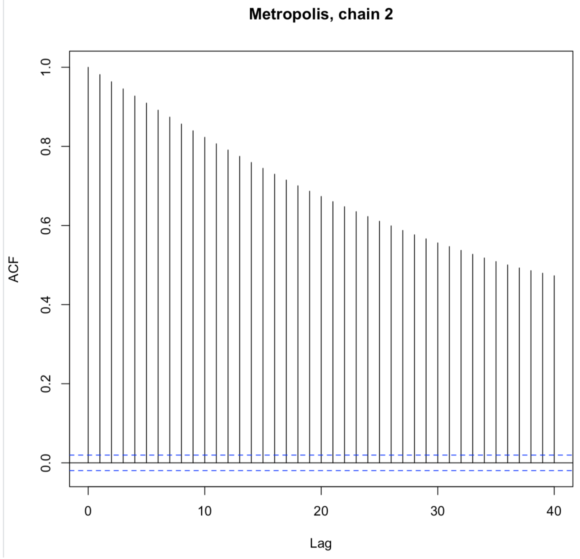
Below is an autocorrelogram where a lag of i on the x-axis is a measurement of the correlation between two iterations that are i iterations apart. An autocorrelelogram which shows rapid decay with increasing lag is indicative of “good mixing.”[2]



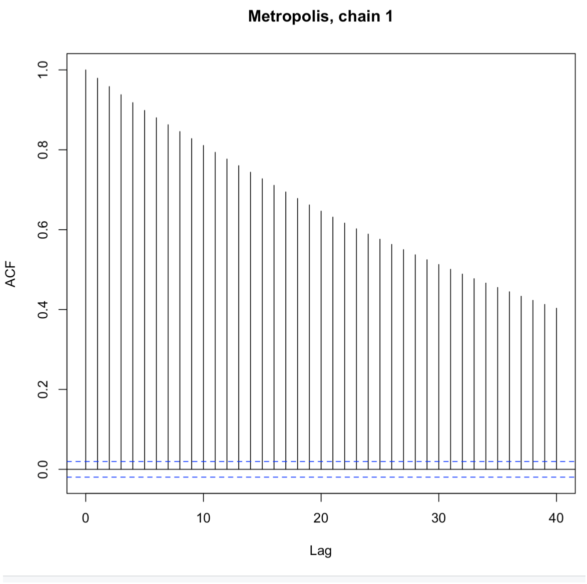
Gibbs Sampling, chain 2



Gibbs Sampling, chain 1

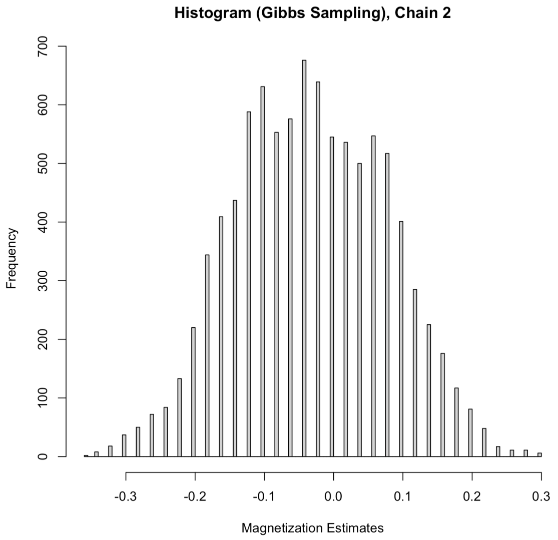


Metropolis, chain 2

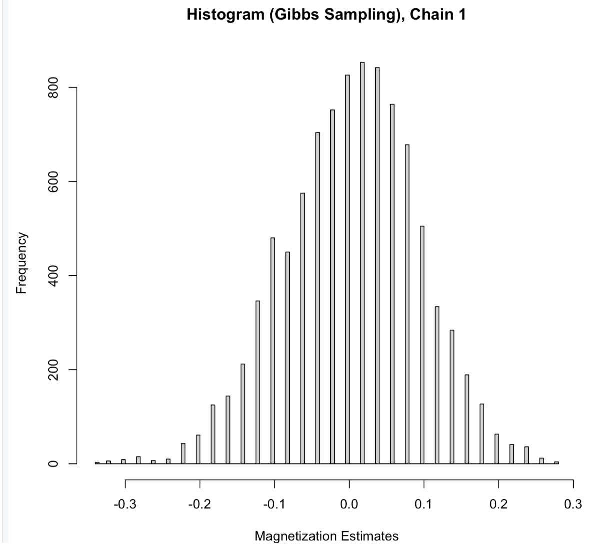


Metropolis, chain 1

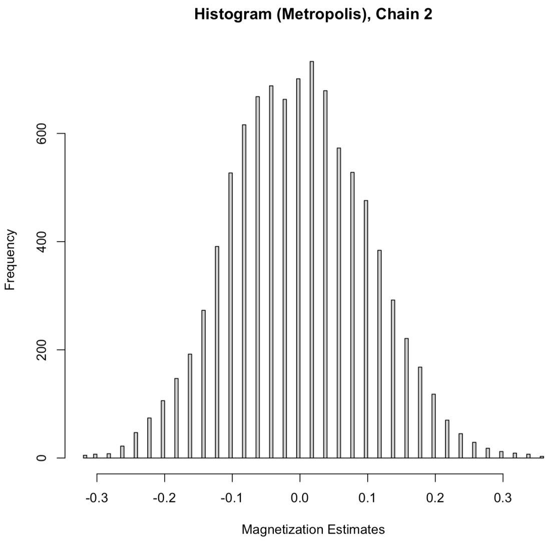
Interestingly, there is more autocorrelation in Gibbs Sampling, which was unexpected given the acceptance probability is always 1 so the new configuration is always accepted. Moreover, not only is the Metropolis autocorrelation lower for a given lag value i, the Metropolis autocorrelation also decays faster as lag increases. There is also a between-chains variance in autocorrelation between the two chains using the Gibbs sampling approach.

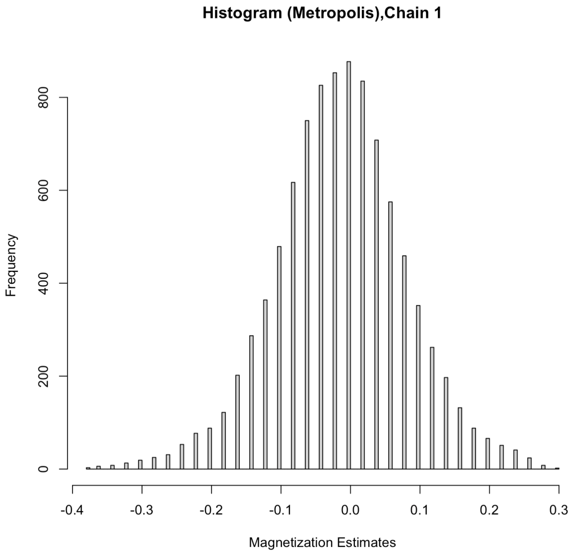


Histogram (Gibbs Sampling), Chain 2



Histogram (Gibbs Sampling), Chain 1





Similarly, even when comparing the histogram showing the frequency of the MCMC estimates, there does not appear to be a significant difference in the shape of the distribution. As the shapes of the histogram appear to approximate that of a normal distribution, we can assume the sample distribution of the estimates is normally distributed and thus we can assume the sample mean of the magnetization estimates approximates the true expected value of the magnetization. Thus, by taking the mean of all estimates, we obtain a mean magnetization of -0.008282 for the first Metropolis chain and -0.00419 for the second Metropolis chain. We can do the same to obtain mean magnetizations of the Gibbs Sampling process: 0.005750974 for the first Markov chain and -0.029988 for the second Markov chain. Negative values raised suspicion and caused me to worry that taking the mean of all estimates starting from the first to last iteration might not be statistically robust. This is because I am including estimates outputted before the MCMC algorithm converges on the target density. Thus, I decided to invoke the method of burn-in, where D iterations are discarded and not included in computation, such that the final computed values are more representative of the final approximation of the target density and less influenced by initial values.

Examining the trace plots that are posted above, it appears that after ~500 iterations, the Markov chain distribution begins to approximate the final distribution that appears present near the end of the algorithm. Thus, I decided to set D=500. However, before going ahead with the burn-out, I decided to assess the validity of this decision using the Gelman-Rubin statistic.

An acceptable value of D is indicated when the square root of the Gelman-Rubin statistic (R) is less than 1.2.[1]. Note that the Gelman-Rubin statistics requires more than one chain to be run in order to be a valid criterion which is one reason why I decided to run two chains per algorithm. Calculating the Gelman-Rubin statistics, I got a Metropolis R value of 1.009563 ($\sqrt{R}$~1.005<1.2) and a Gibbs R value of 0.9998947 ($\sqrt{R}$~.999<1.2). These are both acceptable R values, suggesting that the approximated target distributions after 500 iterations are indeed stationary.

Recalculating the expected values after discarding the first D or 500 iterations, I get a mean magnetization of 0.0005894116 for the first Metropolis chain and 0.01460478 for the second Metropolis chain. For the Gibbs sampling process, I got an expected magnetization of -0.01992001 for the first chain and -0.02473003 for the second chain. While, I still get magnetization values that are negative, these values are still different from the original values obtained without the application of the burn-out strategy.

The Lewis-Rafferty diagnostic is another statistic that diagnoses the sample size or number of iterations required for a 95% accuracy rate in prediction. [6] However, when I tried running the Lewis-Rafferty diagnosis on the Markov chains for both algorithms, I got a suggested sample size from 60,000-120,000 for each of the Markov chains. This is computationally strenuous on my computer so I decided to sacrifice on accuracy and not increase my sample size (or number of iterations) from 10,000 to ~100,000.

6. Conclusion:

This paper showcases how reified approximation techniques from statistical computation can be leveraged to solve computationally expensive problems in physics. An entire subfield of physics called statistical mechanics exists which seeks to not postulate laws governing trends and phenomena but to utilize probability theory and statistical techniques to model systems of microscopic particle interactions and behaviors.

I hope that I successfully accomplished my purpose of exploring physics through the use of statistical models. My methodology relied on the assumption that a demonstrative simulation could be achieved with fixed values of the temperature (T) and coupling (J) constants, without a loss of generality. I assumed this was acceptable because the Boltzmann distribution is a function of energy and not T or J. Moreover, T and J are proportionality constants that only affect the shape of the Boltzman energy distribution. However, upon further research, it appears that at temperatures above 570°C (also called the Curie temperature), magnetization is supposed to become less ordered as spins become less aligned. It would have thus been interesting and insightful to also try the simulations with different temperature values (including at least one above the Curie temperature).

Another issue with this investigation is that I could not compare my magnetization values obtained from the MCMC process with actual ground truth magnetization values for the same configuration and same state variables (T and J). Ideally, I would be able to compare these MCMC approximations with a closed-form analytic solution, which while possible to calculate, was too cumbersome and difficult for me to accomplish with limited computing power.

Furthermore, I feel like because the general variant (acceptance probability<1) of the Metropolis-Hastings is so similar to the Gibbs sampling algorithmically as they both sample from the same distribution, a fuller investigation would have considered sampling from other distributions. One possibility would be to try a different algorithm that does not rely on rejection sampling such as the Hamiltonian Markov Chain Monte Carlo algorithm.

Link to Code: Note that the script is divided into two parts (first half is Metropolis-Hastings, second half is Gibbs sampling)

Please visit https://github.com/v-puppala/Ising-Model-Simulation/blob/main/full_ising.R

References:

[1]. Givens, Geof H., and Jennifer A. Hoeting. *Computational Statistics*, Wiley, Hoboken, NJ, 2013.

[2] Molkaraie, Mehdi. "An Importance Sampling Algorithm for the Ising Model with Strong Couplings." *arXiv: Computation* (2014): n. pag.

[3]. Robert, Christian P.. "The Metropolis–Hastings Algorithm." *arXiv: Computation* (2004): 267-320.

[4]. Shekaari, Ashkan and Mahmoud Jafari. "Theory and Simulation of the Ising Model." (2021).

[5]. Roy, Vivekananda. “Convergence Diagnostics for Markov Chain Monte Carlo.” *arXiv: Computation* (2019): n. Pag.

[6] Raftery, A.E. and Lewis, S.M. (1995). The number of iterations, convergence diagnostics and generic Metropolis algorithms. *In* Practical Markov Chain Monte Carlo (W.R. Gilks, D.J. Spiegelhalter and S. Richardson, eds.). London, U.K.: Chapman and Hall.