Monte Carlo Methods
The 2D Ising-Lenz model

Tinoco, D.[1]
California State University San Bernardino

(*Electronic mail: 006186618@coyote.csusb.edu)

(Dated: 31 May 2022 GMT)

This paper presents the Metropolis-Hastings and Wolff algorithm programmed using the Rust language, using rayon threading library for scaleable multithreading. A basic analysis of the two algorithms is presented as well as a introduction into Monte-Carlo methods, with expectation values for the Ising-Lenz model presented in output figures. The code used can be found at https://github.com/ramenspazz/Ising_model_rust.

I. THE ISING MODEL

The Ising-Lenz model was first proposed by the German physicist Whilhelm Lenz as a problem for his student Ernest Ising in 1920[1] as a mathematical model for describing ferromagnetism in martensitic materials, but also can extend to paramagnetism and diamagnetism given the correct hamiltonian and parameters.

Before the ubiquity of the computer circa the 1960's, pure analytical methods were used to analyse this model using the transfer matrix method, with the first analytical solution being given by Lars Onsager[2] in 1944. The problem with this method is that the specifics of the method change depending on the lattice symmetry and number of dimensions used. The model can be arbitrarily expanded, but the difficulty in solving the system analytically becomes harder with each dimension added.

The systems of interest in this paper are 2D lattices with c4v and c3v crystal symmetry, where the lattice is characterized by its size a and b, where a and b define the number of basis elements used in a linear combination to give the total range of the system in the span of the basis vectors.

An important symmetry of the system is reflection symmetry[3], where Ulli Wolff generalizes this symmetry into an operator capable of acting on specific spins. In 2D, the operator acts as such:

$$\sigma_i := \sigma \cdot r$$

$$R(r)\sigma = \sigma - 2(\sigma \cdot r)r$$

$$\therefore R(r)\sigma \cdot r = -\sigma \cdot r$$

The action of this operator is to flip a spin pointed to by r in σ. Under applying this operator to all spins, figure 1(a) and 1(b) show the reflectional symmetry of the system, with the only difference in the two plots being a negation of the magnetization at all points.

Using a singular value for J and B for all nearest neighbor exchange values and the magnetic field, we can write the hamiltonian of the Ising model as:

$$H = -\Sigma_{i=0}^{n-1}\Sigma_{j=0}^{n-1}J(\sigma \cdot r_i)(\sigma \cdot r_j) - \mu\Sigma_{i=0}^{n-1}B(\sigma \cdot r_i)$$

where μ is set to one Ampere meter squared.

II. MONTY CARLO ALGORITHMS

In 1953, Metropolis revolutionized the fields of chemistry and physics, with statistics following in the 90's with a new scheme of random sampling techniques named a metropolis algorithm[4] by simulating a liquid in equilibrium with a gas. Later in 1970 Hastings[5] generalized the new Metropolis algorithm method into what became known as a Metropolis-Hastings algorithm, hallmarked by the use of what became known as gibbs sampling as discussed by Geman and Geman[6].

The name "Monte Carlo" comes from the original use of the method to mostly jokingly simulate gambling in the popular Monte Carlo casino, thereafter it was realized to be a powerful tool for extracting expectation values out of random systems – or in the case of a real casino prove that they are biased.

The expectation value of interest can be estimated from a function acting on the state space of the system, here called g acting on ξ. Let the expectation value be defined as such:

$$\hat{\mu} = \frac{1}{n}\Sigma_{i=1}^{n} g(\xi_i)$$

Then we can express the variance of the system as:

$$\sigma^2 = var(g(\xi))$$

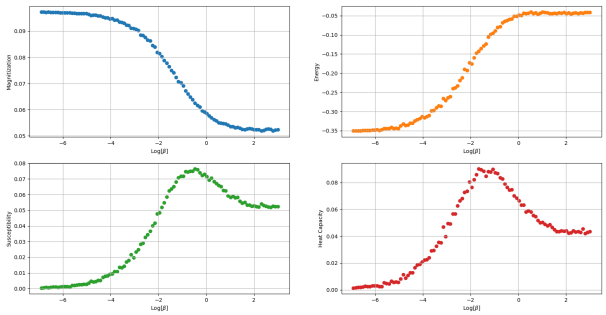
Applying the central limit theorem, we can skip the use of the transfer matrix method and avoid the computationally expensive task of solving N degree polynomial equations. The central limit theorem then for a large enough set of trials approximates $\hat{\mu}$ as the standard mean of the number of trials. This then allows the variance to be estimated as such:

$$\hat{\sigma}_n^2 = \frac{1}{n}\Sigma_{i=1}^{n}(g(\xi) - \hat{\mu})^2$$

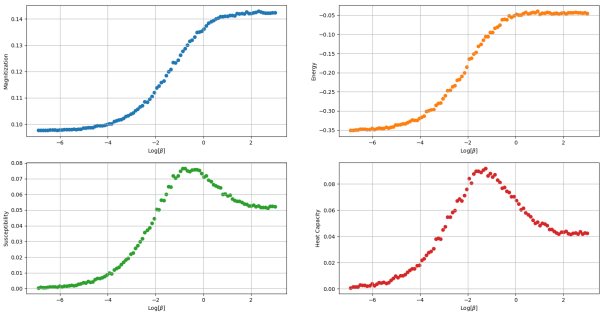
As can be seen so far, there is not much a difference if any from a pedantic standpoint from standard statistics on sets of real valued numbers.

We can define as the instantaneous magnetization of the system as:

$$M = \frac{1}{N}\Sigma_{i=0}^{N-1}(\sigma \cdot r_i)$$



(a) Fig 1. Wolff plot in a c4v system with 20% of spins up.



(b) Fig 1. Wolff plot in a c4v system with 80% of spins up.

where N is the total number of spins in the system. Then for k iterations in a single simulation trial, we can write the expectation as:

$$\hat{M} = \frac{n}{k}\Sigma_{i=0}^{k-1} M_i$$

and the the susceptibility of the system becomes:

$$\hat{\chi} = \frac{1}{k}\Sigma_{i=0}^{k-1}(M_i - \hat{M})^2$$

The heat capacity of the system for a given β is defined as the standard deviation of the energy of the system as given by the hamiltonian:

$$\hat{C_v} = \frac{1}{k}\Sigma_{i=0}^{k-1}(H_i - \hat{H})^2$$

III. MARKOV CHAIN MONTY CARLO SIMULATIONS

An algorithm is called a Markov chain if given some microstate, the transition probability from a state ξ to some new state ξ' has an equal probability independent of all previous and concurrent states of the system based on its probability distribution – called the transition probability distribution.

An Ordinary Monte Carlo simulation is a special case of the Markov Chain Monte Carlo scheme where the system is time independent and reversible[7] such that a transition from state ξ to some new state ξ' is equally as probable as moving from ξ' to ξ.

The two algorithms presented in this paper are both Markov Chains[8] where each generates a chain of states – called a Markov chain – where the n'th state is derived from the $n-1$'th state with a probability of appearing proportional to its Boltzmann weight based on the Boltzmann distribution.

IV. THE METROPOLIS ALGORITHM

The Metropolis algorithm is a Monte Carlo method who's transition probability distribution is parameterized by the change in energy ΔE of moving from a state $\xi(E)$ with macroenergy E to some new state $\xi'(E')$ with macroenergy E'.

The transition probability distribution of interest is the Boltzmann distribution, giving the probability that a system will be in a state given the macroenergy of that states collective microstates, and is given by:

$$p(\xi) \propto e^{-E_i\beta}$$

Using the transition probability distribution and the concept of reversibility, a balance equation can be derived as such:

$$P(\xi \to \xi') = P(\xi' \to \xi)$$

$$p(\xi) = p(\xi')$$

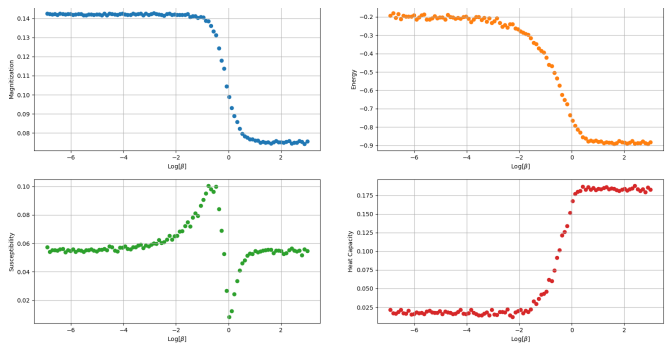
$$\implies e^{-E_i\beta} = e^{-E_f\beta}$$

Taking the ratio of this, we arrive at our transition probability statement for a MCMC update to accept or reject a proposed change in the system:

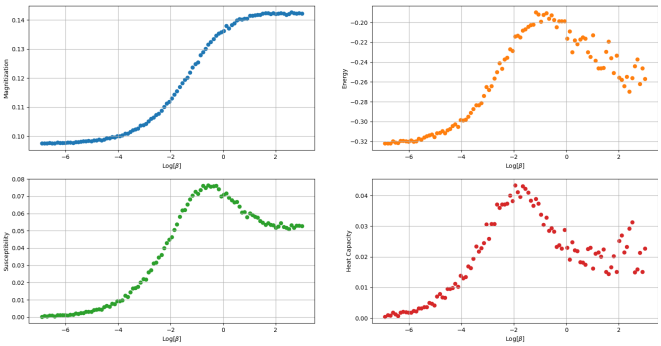
$$\frac{p(\xi)}{p(\xi')} = e^{-\beta(\Delta E)}$$

Therefore for a given choice of βJ, the only deciding factor for accepting or rejecting an evolution is the change in energy of the system from ξ to ξ'. We then choose a random number inclusively between 0 and 1 and compare it to the balance expression each iteration. The parameterization of the balance equation by the change in energy of the system from ξ to ξ' causes the Metropolis-Hastings algorithm as applied to the Ising model to seek a local minima of the energy of the system.

The problem with this scheme is that the system only flips single spins at a time, making the traversal of phase space limited in its total span. By starting the system at high temperatures where the probability of a flip is higher and gradually cooling the system, we can lower the system into a minimal energy state in a more reliable way. This process is called annealing the system and can be used in conjunction with other



(c) Fig 2. Metropolis-Hastings c4v plot using $J = 1$ and $B = 0.1$.



(d) Fig 2. Wolff c4v plot using $J = 1$ and $B = 0.1$.

methods as a starting point for a minimal energy state for a given set of simulations across a range of beta values.

Annealing the system using a 128x128 spin c4v system, the Metropolis-Hasting scheme applied for 100,000,000 iterations with an initial unit-less configuration energy of -3540, and an initial magnetization is 3779 produces a final energy of -13993 and magnetization of -7169 at a β of 1, while annealing the system from the same initial configuration across β values of results in a final unit-less energy of 82 and the initial magnetization of 81.

A shortcoming of the Metropolis-Hastings algorithm is related to the ergodicity of the system. Ergodicity is the ability of the system to visit all states in the configuration space of the problem regardless of the number of iterations it takes to do so[8], but with a single flip algorithm like the Metropolis-Hastings algorithm, the system can get stuck in local minima and miss global minima, and overall preforms slower as the system evolves. This problem is called the critical slowing down, and occurs when the system approaches a critical temperature where spins are flipped with much greater frequency than surrounding temperatures[7].

This is due to the treatment of the action in the Metropolis-Hastings algorithm. The algorithm only models short wavelength interactions by modeling the spin interaction as limited to nearest neighbors. In systems near the critical temperature, spins begin to become correlated at longer distances across the lattice[8], and the number of iterations required increase as $O(e^n)$.

V. THE WOLFF ALGORITHM

The Wolff algorithm as introduced by the German physicist Ulli Wolff preforms much better for modeling long wavelength correlation of spins across the lattice around the critical temperature of the system due to its treatment of spin flips. The mechanism for spin flips in the Wolff algorithm is as follows[3]:

1. Select a random spin σ_i in the lattice.

2. Mark σ_i, then visit all nearest neighbors of σ_i with the same spin value if they are not already marked.

3. Generate a random number between 0 and 1 inclusive, and if it is less than the balance condition flip the node, repeat the process from step 2 from this new spin.

The reason for flipping a cluster of spins at in a single iteration of the simulation comes from the realization of Swendsen and Wang[9] that cluster flipping causes much shorter relaxation times providing to more efficient simulations of larger systems near the critical temperature.

First consider that even in a minimal energy ground state, domain walls are already present in the system at any temperature, and for the system to have action on small variations of the magnetic field, some domains have to reverse their spin orientation[8], ergo flipping clusters of spins instead of single spins as in the Metropolis-Hastings algorithm.

The balance condition used relies on the definition of a general spin flip operation[3], where σ_y is equal to $-\sigma_x$, and can be reduced to:

$$P(\sigma_x \to \sigma_y) = 1 - e^{-2\beta|\sigma|^2}$$

Because of the cluster flipping nature of the Wolff algorithm, it is easy to calculate the susceptibility of the system by keeping track of the average size of clusters flipped[3], which works out to the equivalent standard deviation of the magnetization.

VI. DIFFERENCES IN RESULTS BETWEEN ALGORITHMS

The Metropolis-Hastings algorithm produces different results when measuring the heat capacity and susceptibility of the system when near the critical temperature of the system, as opposed to the Wolff algorithm due to the lack of long wavelength interactions in the Metropolis-Hastings algorithm.

As can be seen in figure 2, the two algorithms get markedly different results for the same set of input parameters and the same starting state.

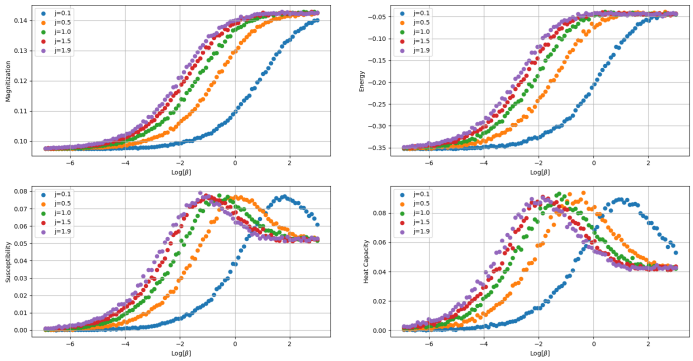


Fig 3. Wolff c4v plot using $J \in [0.1, 1.9]$ and $B = 0$ for a 128x128 spin system.

VII. SPIN EXCHANGE AND THE HEAT CAPACITY

The reason for this difference is in the balance equation of each algorithm. The Metropolis-Hastings algorithm is attempting to quench the system in a heat bath by assuming that the internal temperature of the system at all spin sites is homogeneous, and moves through phase space such that the energy is minimized for often than not in any given iteration.

The Wolff algorithm simulates a more realistic spin flipping model, where clusters up to their domain wall can be flipped each iteration. In this scheme, the long range correlation of the system is preserved and near the critical temperature, phase space is traversed more quickly, providing a lower relaxation time.

System size is the main factor in the heat capacity of the system, as a larger system will predictably be able to contain more heat for a given size. The temperature of the maximum heat capacity the system obtains changes with the spin exchange value as can be seen in figure 3, however increasing the system size allows for the system to retain more heat by having a higher and more stable energy configuration before the critical temperature, thereafter the system settles into a lower energy ground-state. Table 1 lists the values obtained.

Size	Maximum C_v	*Pre-Critical temperature limit*
32×32	0.21±0.009	0.12±0.008
64×64	0.23±0.004	0.16±0.007
128×128	0.24±0.002	0.13±0.03
256×256	0.20±0.001	0.16±0.002
512×512	0.21±0.07	0.16±0.002
1024×1024	0.24±0.001	0.23±0.002

An applied magnetic field changes the energy of the object greatly, and stabilizes the maximum heat capacity of the object. This is the system being magnetized, and then losing the majority of its internal magnetic field due to random spin flips after the curie point.

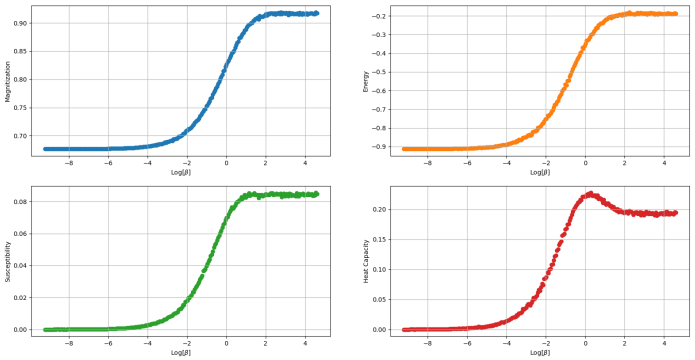


Fig 4. Wolff c4v plot using $J = 1$ and $B = 0.00$ for a 256×256 spin system with B=0.00.

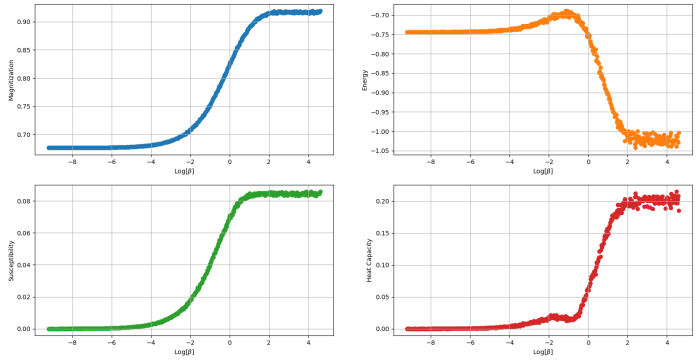


Fig 5. Wolff c4v plot using $J = 1$ and $B = 0.25$ for a 256×256 spin system with B=0.25.

[1] W. Lenz, "Beiträge zum verständnis der magnetischen eigenschaften in festen körpern," Physikalische Zeitschrift **21**, 613–615 (1920).

[2] L. Onsager, "Crystal statistics. i. a two-dimensional model with an order-disorder transition," Phys. Rev. **65**, 117–149 (1944).

[3] U. Wolff, "Collective monte carlo updating for spin systems," Phys. Rev. Lett. **62**, 361–364 (1989).

[4] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," The Journal of Chemical Physics **21**, 1087–1092 (1953).

[5] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," Biometrika **57**, 97–109 (1970).

[6] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-6**, 721–741 (1984).

[7] S. Brooks, *Handbook of markov chain Monte Carlo* (Chapman amp; Hall, 2011).

[8] N. M. E. J. and G. T. Barkema, *Monte Carlo Methods in statistical physics* (Clarendon Press, 2011).

[9] R. H. Swendsen and J.-S. Wang, "Nonuniversal critical dynamics in monte carlo simulations," Physical Review Letters **58**, 86–88 (1987).