The Wolff Cluster Algorithm*

Ram Surya Sri Shourie Mummadavarapu and Rudra Kalra
(Dated: March 4, 2022)

The Ising model is perhaps the most famous and widely used models to probe critical behaviour of ferromagnetic systems. Given it's simplicity, many algorithms have been created to make computations as efficient as possible. Monte Carlo simulations are the way to go. The well known Metropolis algorithm, straightforward as it may be, is far from perfect. Here we demonstrate an algorithm from a class of so called cluster algorithms: the Wolff Cluster Algorithm. It reduces computation times near the phase transition significantly and hence succeeds in effectively eradicating the elusive phenomenon of critical slowing down.

Usage: The code and relevant information can be found in our GitHub repository `https://github.com/Radium2000/p760-cpbonn-wolff`

I. INTRODUCTION

The phenomenon of phase transitions shows up in many areas of life beyond Physics, in places like the influence of peers in a corporate scenario [1], financial markets [2] and biological systems [3] to name a few. Hence, it becomes essential to study the properties of such systems more closely. Any Statistical Mechanics course taught around the world includes an analysis of the famous Ising model. At it's heart, the Ising model is one of simplest and most intuitive models for ferromagnetic (or anti-ferromagnetic) systems. It's utility, however, transcends it's simplicity as it has proven to be an invaluable tool to understand critical behaviour and phase transitions.

The model is described by a lattice of spins in a suitable dimension, where the spins can be either pointing up (+1) or down (-1). For simplicity, periodic boundary conditions are invoked. The Hamiltonian is given by:

$$H = -h\sum_i \sigma_i - J\sum_{<ij>} \sigma_i\sigma_j \tag{1}$$

where h is the external magnetic field & J is the spin coupling constant. σ_i and σ_j are spin states, and can assume values ± 1; $< ij >$ indicates the sum over nearest neighbour spin states. The sign of J decides whether the system is ferromagnetic ($J > 0$) or anti-ferromagnetic ($J < 0$). For this project, we assume a ferromagnetic system devoid of any external field ($h = 0$). This ferromagnetic Hamiltonian favors aligned spins.

The system exhibits a phase transition at the critical (or Curie) temperature, T_C. Below T_C, the spins start to align with their neighbours and at very low temperatures, barring a few outliers, all spins are aligned. Above T_C, this long range order vanishes. This fact is demonstrated by the mean magnetization, which is simply the magnetic moment per lattice site, defined as follows:

$$\langle m \rangle = \frac{1}{N}\sum_i \sigma_i \tag{2}$$

Above T_C, due to the random orientations of the spins, m goes to 0 and below T_C, there is spontaneous symmetry breaking as a result of which m becomes either +1 or -1. This value is determined on the basis of initial conditions and general stochastic processes. Onsager, in a landmark paper [4] was able to exactly calculate the free energy of the 2 dimensional Ising model. Since then, there have been relatively easier methods devised to solve the same. The derivation using the transfer matrix approach can be found in [5].

Direct computational methods for the model using the partition function can be performed, but only for small systems. As the system size increases, listing all spin configurations becomes too computationally expensive. Exact solutions in most cases fortunately, can be achieved by Monte Carlo models given a few conditions are satisfied (detailed balance and ergodicity). The trade off is efficiency for inevitable statistical errors, since the configurations are sampled rather than enumerated. Importance sampling Monte Carlo methods which focus on local sampling like the Metropolis algorithm (which flips one spin per iteration) become very slow near the critical point because of reduced acceptance rates. This is the phenomenon of *critical slowing down* which ruins the party. To address this problem, so called cluster algorithms were developed which flip whole clusters of similar spins simultaneously rather than individuals. The Wolff cluster algorithm first described by Uli Wolff [8] is what we are demonstrating in this project. It has been shown and will be demonstrated that this algorithm reduces critical slowing down.

Our aim in this project will be to compare the effects of critical slowing down in the Metropolis algorithm to the Wolff algorithm, probe more closely into the phase transition at T_C, determine critical exponents and extend the ideas from the 2D Ising model to a more general 3D case. But first, we will start with the basic theoretical foundations required for a coherent understanding.

* Final project for the p760:Computational Physics course offered by University of Bonn

II. THEORETICAL BASIS

Since we are taking $h = 0$, the Hamiltonian reduces to the following,

$$H = -J \sum_{<ij>} \sigma_i \sigma_j \tag{3}$$

The probability of any configuration is then given by:

$$P(\sigma) \propto e^{-\beta H\{\sigma\}} \tag{4}$$

where $\beta = \frac{1}{k_B T}$ is the commonly used form of inverse temperature. In the context of the computations, the following parameters are important:

- Coupling constant, $J = 1$
- Grid dimension, L
- Total number of spins, N = L^2 or L^3
- Total magnetization, $M = \sum_i \sigma_i$
- Mean magnetization, m (as defined by equation 2)
- Magnetic susceptibility, χ. χ is related to the variance of M
- Critical exponents, α, $\tilde{\beta}$, γ defined by the following equations:
  $C_V \propto (T - T_C)^{\alpha}$; $m \propto (T - T_C)^{\beta}$; $\chi \propto (T - T_C)^{-\gamma}$

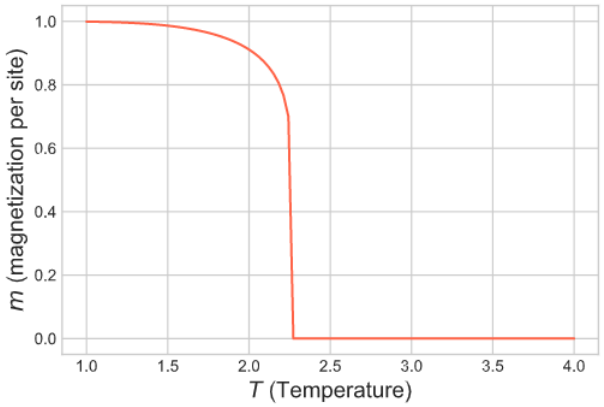

FIG. 1: Theoretical form of m. Above T_C, m goes to zero. X-axis is normalised to T_C.

Theoretical expressions. Following [6], the exact expression for m (Fig 1) is given by (in dimensionless form):

$$m = \left(1 - \frac{1}{\sinh^4 \frac{2}{T}}\right)^{\frac{1}{8}}$$

The slope of the graph changes discontinuously at $T = T_C$, indicating a second order phase transition. It can be shown that χ_m diverges at T_C,

$$\chi_m = \begin{cases} \frac{1}{k_B(T - T_C)} & T > T_C \\ \frac{1}{2k_B(T - T_C)} & T < T_C \end{cases} \tag{5}$$

We will use this divergence (which will show as a peak due to finite size scaling) and the peak of C_V vs T graph for computing the value of T_C, since the mean magnetization plot introduces ambiguity. The exact value of T_C, again in dimensionless form is,

$$T_C = \frac{2}{\ln(1 + \sqrt{2})} \approx 2.269$$

Detailed balance. Any Monte Carlo Markov Chain which hopes to achieve statistically independent configurations needs to satisfy the condition of detailed balance. In it's simplest form, the condition is,

$$\pi(a) P(a \to b) = \pi(b) P(b \to a) \tag{6}$$

where $\pi(x)$ is the probability of the system to exist in state x and $P(x \to y)$ is transition probability from state x to state y. The Metropolis algorithm for Ising model satisfies this by considering,

$$\tilde{P}(a \to b) = \min\left[1, e^{-\beta \Delta H}\right] \tag{7}$$

During evolution, near T_C, there are bubbles of similar spins. The probability that you will chose a spin which is in the neighbourhood of similar spins is therefore very high. This leads to a lot of rejections in the energy checking stage of the Metropolis algorithm (see Subsection III A). This algorithm is inefficient near the critical point because of this low acceptance rate. Fortunately, there is more than one way to satisfy 6. The Wolff cluster algorithm makes it so that $\tilde{P}$ is always 1. Following [9] and it's notations, now,

$$\tilde{P}(a \to b) = \min\left[1, \left(\frac{e^{-2\beta}}{1 - p}\right)^{n_2} \left(\frac{1 - p}{e^{-2\beta}}\right)^{n_1}\right]$$

with a special $p = 1 - e^{-2\beta}$. Therefore, instead of flipping one spin at a time, we form flip clusters of similar spins in all iterations since $\tilde{P}$ is always 1 due to our choice of p. Thus, computation complexity is transferred from accepting/rejecting moves to the creation of clusters.

Binders Cumulant[13], it is the fourth order cumulant which is defined as the kurtosis of the order parameter. It is often used to find the critical points in models describing phase transition. In this particular project we used this method to determine the critical temperature.

$$U_4 = 1 - \frac{< m^4 >}{3 < m^2 >^2} \tag{8}$$

Application to 3D. After 33 pages of intense work [4], one can derive the free energy for the 2D Ising model. This can't be done for the 3D case however. Therefore, we will check the correctness of our code for 2D and simply extend it for a more general 3 dimensional system.

III. METHODS

A. Algorithms

Metropolis Algorithm Flip single spins stochastically,

1. Start with some initial configuration of spins
2. Randomly chose a site
3. Calculate energy change if this spin is flipped, ΔH
4. Accept the move with probability $\tilde{P}$ from equation 7. If rejected, stay on the same configuration
5. Repeat step 2
6. Measure the observable only after some thermalisation steps and in some predefined interval

Wolff Cluster Algorithm Form clusters stochastically and flip every time,

1. Start with some initial configuration of spins
2. Randomly chose a site
3. Start forming a cluster of similar spins using the selected spin as the starting point. Add first spin to `cluster` and `frontier` list:
   a. Select a random spin from `frontier`
   b. Loop over all neighbours (which are not in `cluster`) of that particular spin in `frontier` and add them to `cluster` and `frontier` with probability p (see Section II)
   c. Remove that spin from `frontier`
   d. Repeat steps (a) and (b) till `frontier` is empty
4. Flip the whole cluster of spins
5. Repeat from step 2
6. Measure observable only after some thermalisation steps

B. Simulation Strategies

Given the ease of the algorithm, there wasn't much room for innovation. However, there were a few things which make life a bit easier,

- Instead of going for the intuitive 2-dimensional array for storing the values of the given spins, we do some mental arithmetic before and opt for a 1-dimensional array. This makes the calculation and use of neighbours easier.
- We need not have to recalculate the magnetization after each update of the lattice by summing over all spins. For both algorithms, each change of spin just contributes `2*spins[i]` where `spins[i]` is the new value of that spin.
- For the determination of the neighbours of a particular spin, we pre-determine all the neighbours for each spin. When required, we simply read them from the stored list

C. Error Analysis

Most of the error analysis done is by repeating the same simulation multiple times and taking the standard deviation of the data to be the error. While fitting the data to compute characteristic time the error analysis employed is the standard root mean squared error.

IV. RESULTS

All figures have dimensionless axes. First, we present the results for the 2 dimensional Ising model.

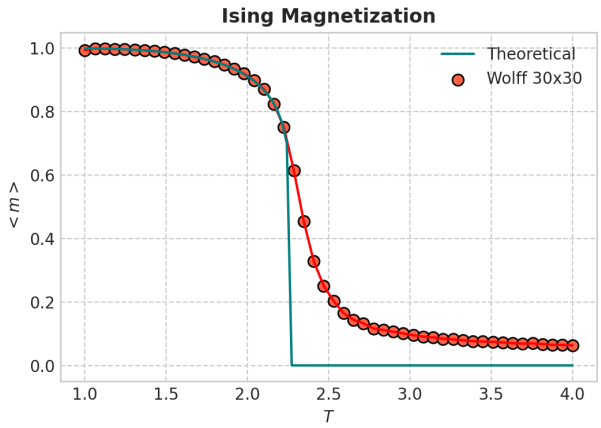

FIG. 2: Mean magnetization per site for a range of temperatures with a 2 dimensional 30x30 lattice.

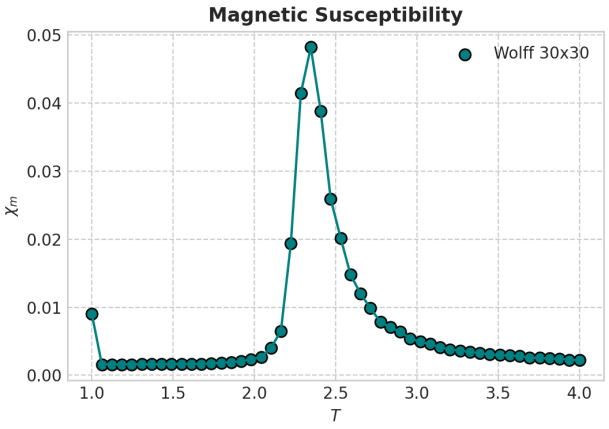

FIG. 3: Magnetic susceptibility for a range of temperatures with a 2 dimensional 30x30 lattice.

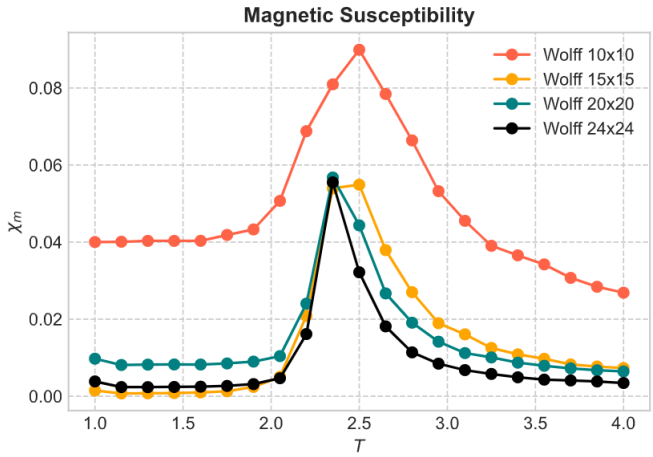

FIG. 5: A comparison between the Wolff cluster magnetic susceptibility for a range of lattice sizes.

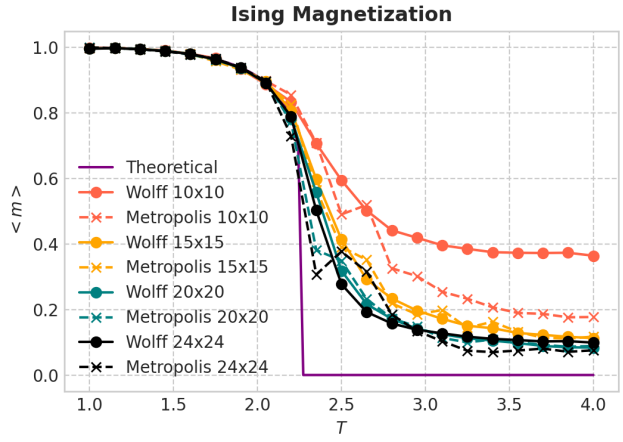

FIG. 4: A comparison between the Wolff cluster and Metropolis magnetization for a range of lattice sizes.

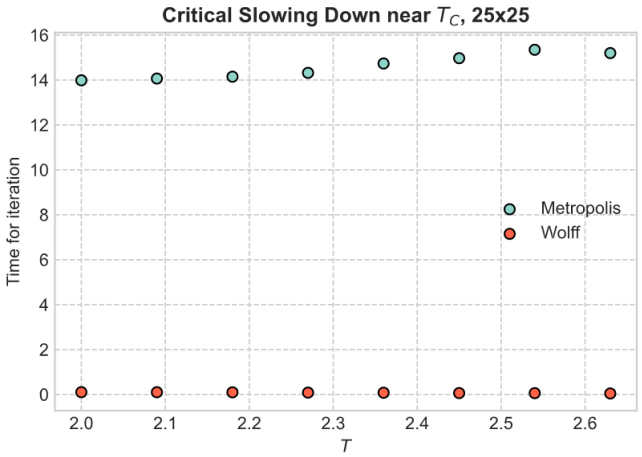

FIG. 6: A comparison of effects of critical slowing down for a 25x25 lattice for the 2 algorithms.

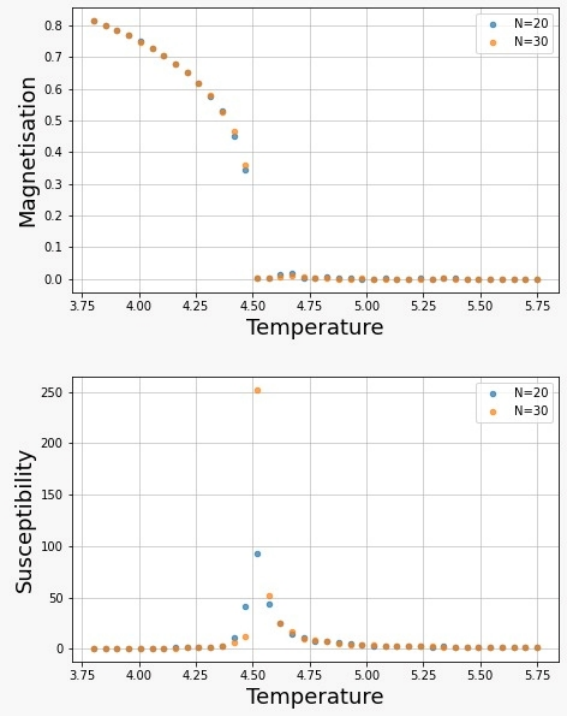

FIG. 7: Mean magnetization and magnetic susceptibility for the 3 dimensional lattice using Wolff Cluster Algorithm for 2 lattice sizes

The critical temperature for 3D ising model found using Binding Cumulant is 4.5 ± 0.020 and for 2D Ising model it is 2.25 ± 0.056.

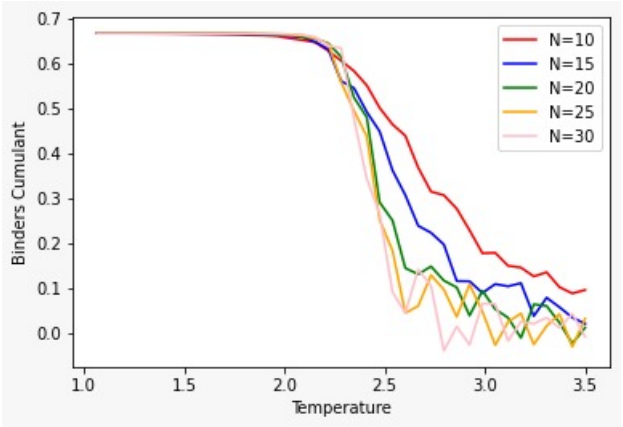

FIG. 8: The binders cumulant for 2D case; useful for determining T_C.

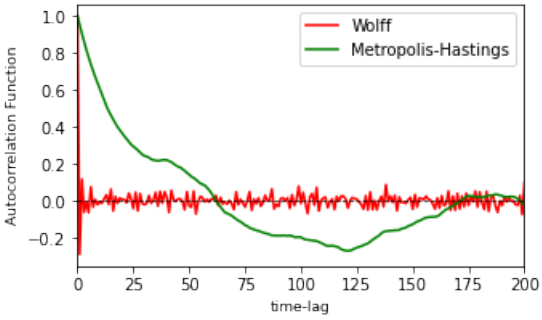

FIG. 9: Autocorrelation function for 2D lattice

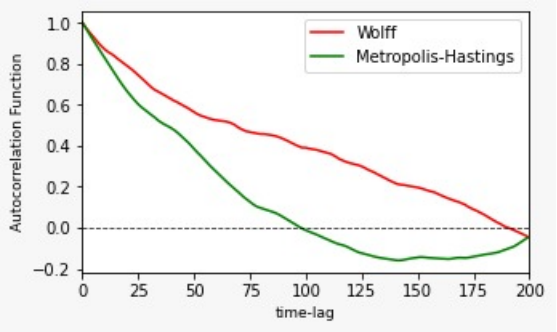

FIG. 10: Autocorrelation function for 3D lattice

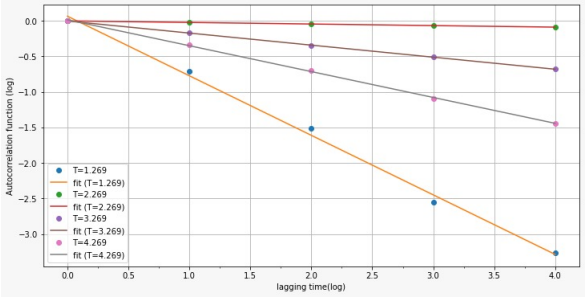

FIG. 11: Autocorrelation(log) vs lagging time(log). Used for determining characteristic time

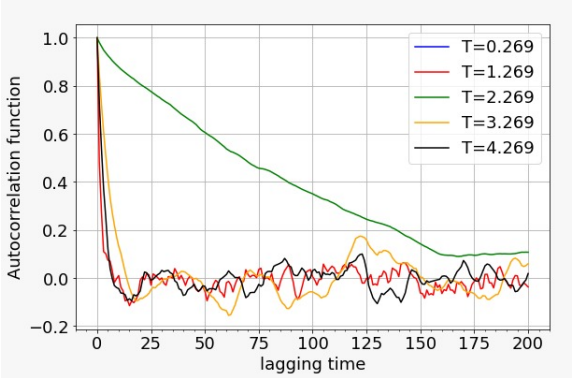

FIG. 12: Autocorrelation vs lag time for Metropolis Hastings algorithm in the case of 2D Ising model

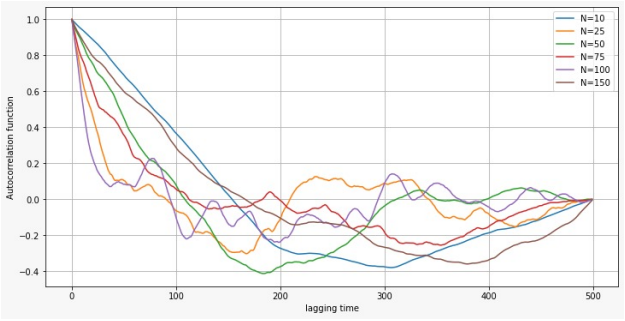

FIG. 13: Autocorrelation for 2D metropolis-hastings algorithm for various lattice sizes. Used for showing that the performance of the algorithm gets worse with increase in lattice size.

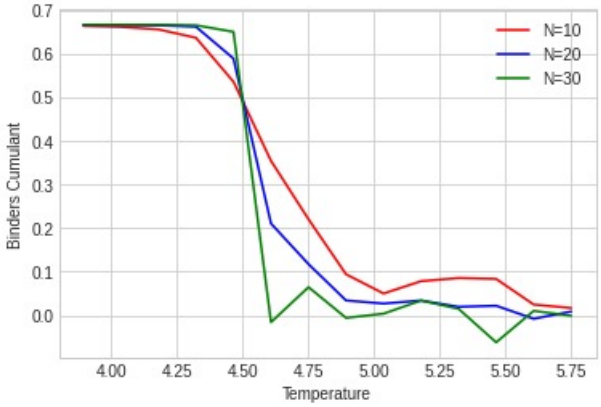

FIG. 14: Binders cumulant for the 3D ising model calculated using wolff-Cluster algorithm.

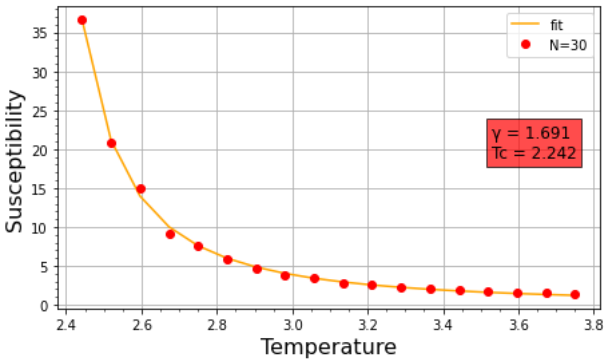

FIG. 15: Determination of critical exponent, γ from magnetic susceptibility

Temperature	Characteristic Time	Error
0.269	∼0	NA
1.269	1.194	1.72E-04
2.269 (T_C)	45.48	-5.74E-05
3.269	5.89	2.87E-05
4.269	2.74	-5.74E-05

TABLE I: Table describing the correspondence between temperature and characteristic time for 2D Metropolis Hastings algorithm.

Lattice Size	Characteristic Time at $T = T_C$	Error
10	173.31	1.63E-02
25	26.01	-2.18E-04
50	70.28	-9.18E-05
75	42.81	3.32e-04
100	12.97	5.31E-04
150	88.12	-2.92E-04

TABLE II: Table describing the correspondence between lattice size and characteristic time for 2D Metropolis Hastings algorithm at critical temperature.

V. DISCUSSION

Figure 2 shows the result of the mean magnetization for a 30x30 lattice. This is not quite what we should expect theoretically [14]. The reason for the magnetization not going to 0 with increasing T is due to the stochastic nature of the algorithm which prefers like spins forming clusters. At higher T's, there still exists some residual magnetization as a result. However, we get closer and closer to the actual behviour if we keep increasing the lattice size as shown in Figure 4. Figure 4 also shows the comparison between the Metropolis and Wolff algorithms. It is noted that the Metropolis algorithm gives more fluctuations in the behaviour for the same number of iterations. In Figure 3 we have plotted the magnetic

susceptibility as a function of temperature. We expect it to diverge at T_C (see equation 5). Again, due to the finite size of the lattice, the susceptibility cannot exceed the lattice size and so the divergence shows up as a peak. Nevertheless, we can clearly see a difference in slopes below and above T_C which seems to indicate that our code works properly. Again, as a function of L, we see an improvement in the behaviour of χ, getting closer to the true one with increasing L (Figure 5).

The difference between the responses of the two algorithms towards critical slowing down is best seen when the number of steps in Metropolis algorithm is allowed to vary to maintain a particular acceptance rate. The time per iteration is plotted in Figure 6. We have not used a variable step number for the production of this graph nevertheless, the effects of the Wolff algorithm can be clearly seen.

Because of the critical slow down in Ising model, we can see that Monte Carlo methods under perform near the critical temperature. In order to understand this phenomena better we need to look at the autocorrelation function(ACF) near the critical temperature and compare it with the ACF away from the critical temperature. From the figure we can see that near the critical temperature the lag time is very low when compared to away from critical temperature. As you can see in the table I and figure 12 at the critical temperature the the characteristic time high and as seen in the table II and figure 13 the deviation in this characteristic time keeps going up as the lattice size increases. This problem can be solved using cluster-flip mechanism instead of single-flip mechanism. As seen in the figure 21 and 20 Metropolis and Wolff algorithms have the same equilibrium energy but from the graphs we can see that Metropolis algorithm thermalizes in fewer steps than Wolff-cluster. Where as for magnetisation Wolff Cluster oscillates between the states m=1 and m=-1 at much higher frequency than Metropolis this leads to shorter autocorrelation time as we observe in figures 16 and 17.

Binders Cumulant[10] is an observational tool that we used for estimating the critical temperature of the 2d and 3d models, we can estimate the critical temperature by looking for the intersection of the Binders Cumulant vs Temperature curves for various lattice sizes, the temperature corresponding to the point of intersection is the critical temperature. As you can see from the figure for the 3D model estimate temperature at the point of intersection is approximately 4.5(+/- 0.020) (Figure 14) and for the 2D model it is intersecting at approximately 2.25(+/- 0.056) (Figure 8) which are very reasonable estimates to the correct value. Apart from this we have also calculated the critical temperature by fitting the susceptibility with the power law($\chi \sim |T - T_c|^{\gamma}$) as seen in the figure 15 Critical temperature and Gamma exponent found through that process turned out to be 2.24 ± 0.03 and 1.32 ± 0.2. Using the same process we tried to find critical exponents α and β which turned out to be 0.34 ± 0.2 and 1.73 ± 0.6 which unlike γ are not reasonable estimates. A better method which can be used to estimate these two exponents are finite-size scaling/analysis.

Application to 3D. Since we are confident in our Wolff algorithm, we can take the ideas from 2D and without too much of a hassle, apply to the 3D case. As seen (see Section IV), we have very precisely determined the value of T_C.

VI. SUMMARY

We have seen that the Wolff cluster algorithm proves to be superior to the run of the mill Metropolis algorithm because it drastically reduces the effect of critical slowing down. As a result of the correctness of our code for 2 dimensions, we were able to match T_C to the literature value for a higher dimensional lattice as well. Where susceptibility and magnetization fail, the binders cumulant proves to be a very useful tool for the accurate determination of T_C.

We would like to thank Professor Stefan Krieg for his support and clearance for us to use the super computers in Jülich.

VII. FIGURES APPENDIX

An appendix for all the graphs that we plotted which did not deserve their place in the main results section.

A. 2 Dimensional Case

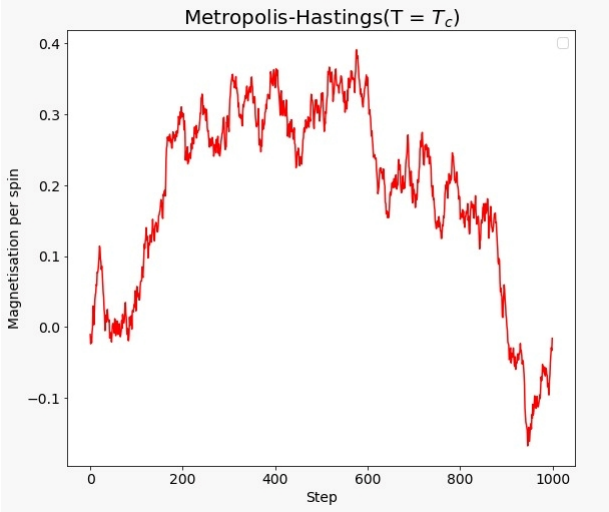

FIG. 16: Magnetization per spin, Metropolis

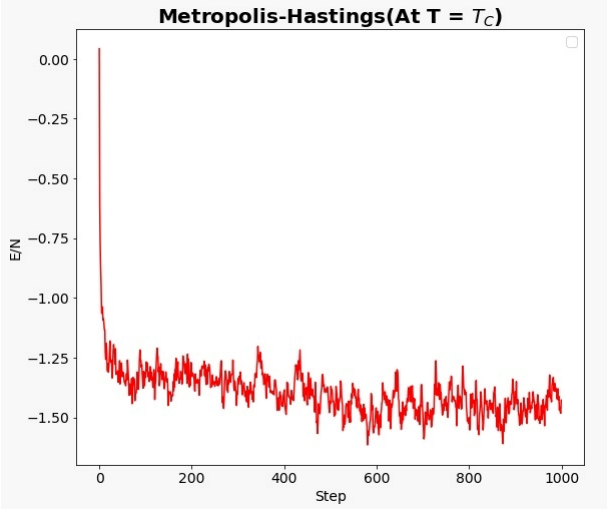

FIG. 18: Average Energy, Metropolis

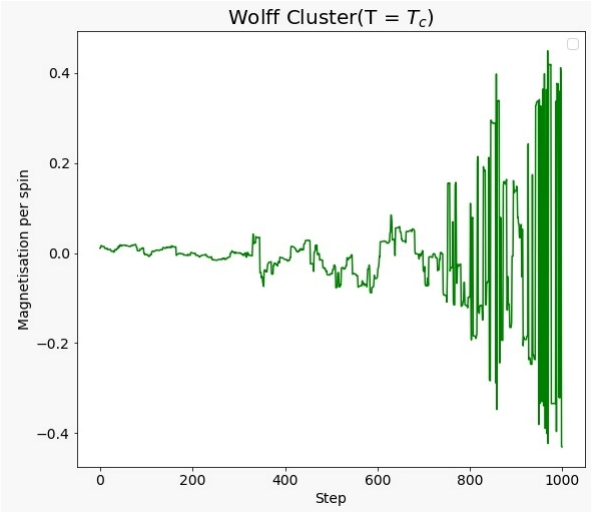

FIG. 17: Magnetization per spin, Wolff Cluster

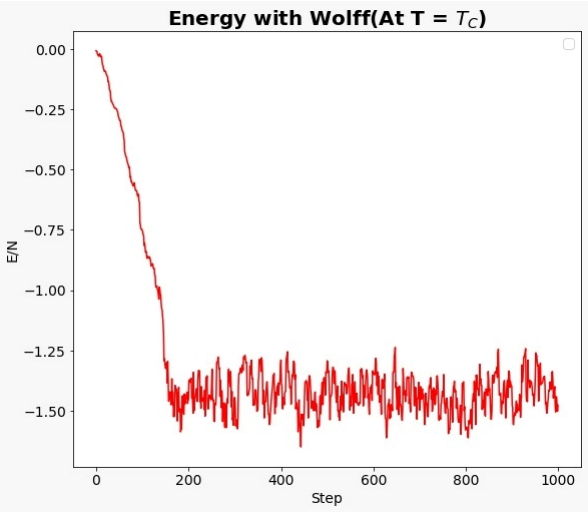

FIG. 19: Average Energy, Wolff Cluster

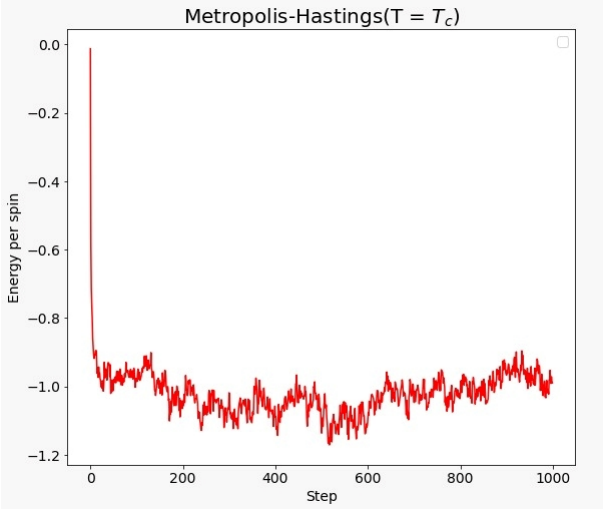

FIG. 20: Energy per spin, Metropolis

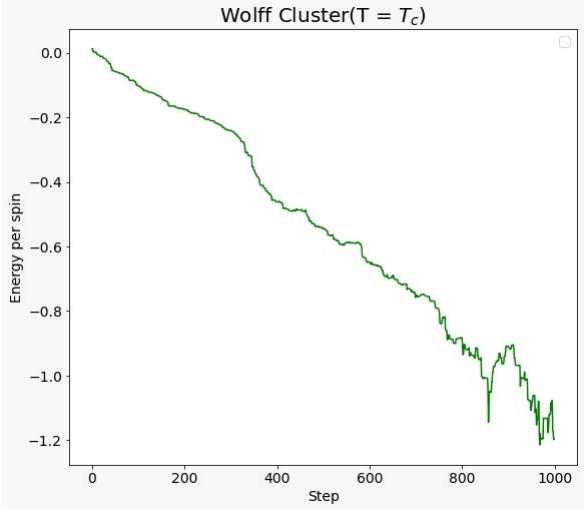

FIG. 21: Energy per spin, Wolff Cluster

B. 3 Dimensional Case

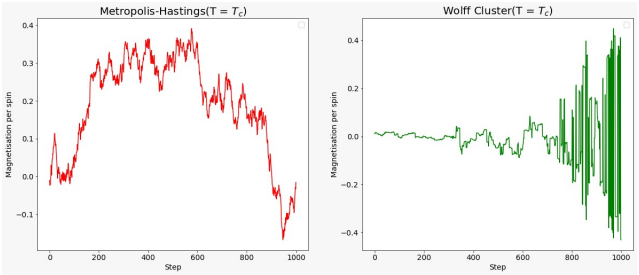

FIG. 22: Magnetization per spin for both algorithms

[1] M. Hohnisch, S. Pittnauer, S. Solomon, and D. Stauffer, *Socioeconomic interaction and swings in business confidence indicators*, Physica A 345, 646–656 2005.

[2] `https://guava.physics.uiuc.edu/~nigel/courses/569/Essays_Fall2010/Files/ccchang.pdf`

[3] Haken, H., Kelso, J.A.S. & Bunz, H. *A theoretical model of phase transitions in human hand movements*. Biol. Cybern. 51, 347–356 (1985)

[4] Onsager, L. 1944. Crystal Statistics. I. *A Two-Dimensional Model with an Order-Disorder Transition*. Phys. Rev., 65, p.117–149.

[5] Plischke, M. and Bergersen, B., 1994. *Equilibrium statistical physics*. 3rd ed. Singapore: World Scientific.

[6] `https://en.wikipedia.org/wiki/Square_lattice_Ising_model`

[7] Krauth, W. (2012). *Statistical mechanics: Algorithms and computations*. Oxford University Press.

[8] Wolff, U. 1989. *Collective Monte Carlo Updating for Spin Systems*. Phys. Rev. Lett., 62, p.361–364.

[9] Werner Krauth. (2003). *Cluster Monte Carlo algorithms*.

[10] Sonsin, A.F., *Computational analysis of 3d ising model* - iopscience. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/630/1/012057 [Accessed March 4, 2022].

[11] Rick Astley. (2010). NGGYU

[12] `https://iopscience.iop.org/article/10.1088/1742-6596/630/1/012057/pdf`

[13] Binder, K. (1981). *"Finite size scaling analysis of ising model block distribution functions"*. Zeitschrift für Physik B: Condensed Matter. 43 (2): 119–140

[14] The magnetization is shown positive in this figure due to us taking the absolute value in the code.