

Cluster algorithms

Devashish Tiwari (20097)

Computational Physics Project

1 Introduction

The 2D Ising model stands as a momentous model in physics, showcasing a genuine phase transition. It was solved exactly by Ernst Ising (1D) in 1920 and in 2D analytically solved by Onsager(2D)⁴ in 1944. We here present the methods to solve Ising model using three different approaches. Due to its lack of intrinsic dynamical evolution, its simulation relies on Monte Carlo algorithms. Particularly near its phase transition, the Metropolis algorithm's efficiency declines sharply, a limitation circumvented by the SW and Wolff method. Critical exponents describe this phase transition, their determination and comparison to theoretical values form the bases of this report. Finally, the performance of all three algorithms is compared. We then apply the use of these cluster algorithms to Hard sphere model, and presented its comparison with theoretical description, concluding this study which is conducted as part of the Computational Physics course (PHY612N) taught by Dr. Sunil Pratap Singh at the Department of Physics, IISER Bhopal.

2 Ising model

Ising model or the Heisenberg model describes the ferromagnetic interaction between the neighbouring spins by the given hamiltonian,

$$H = -J_{ij} \sum_{i,j} \sigma_i \sigma_j - h \sum_i \sigma_i \quad (1)$$

where J_{ij} is the interaction strength (or spin coupling constant) and is non-zero only for the nearest neighbours. The spins have only two possible orientations along a chosen axis “up” (+1) or “down” (-1). and h is the magnetic field strength. There is also a plethora of other physical situations that can be mapped onto Ising models with various forms of the interaction J_{ij} and the field h in binary alloys (where σ_i correspond to the two species of atoms) and atoms adsorbed on surfaces (where σ_i correspond to the presence of absence of an atom on a surface).

Ising found the solution to this problem for a 1D lattice (‘einfache lineare Kette’). The solution to the 2D Ising model for a zero magnetic field was found by L. Onsager. He saw that there was a phase transition present characterised by the vanishing of the magnetisation of the system. The 3D model still unsolved analytically stands as the numerically solved model with a similar phase transitions.

In our case, we will simulate the model in 2D for zero magnetic field.

$$H = -J_{ij} \sum_{i,j} \sigma_i \sigma_j \quad (2)$$

3 Phase Transitions

The 2D Ising model has a second order phase transition from a magnetic (ordered) phase to a nonmagnetic (disordered) phase for which the average magnetisation is nonzero and zero, respectively. The temperature at which this transition occurs is the called critical temperature T_c . The exact solution by Onsager,

shows:

$$\text{Sinh}(J\beta)\text{Sinh}(J'\beta) = 1, \quad (3)$$

where J and J' are spin coupling constant in x and y direction respectively and $\beta = \frac{1}{k_B T_c}$. In this project, we kept the coupling to be constant(=1) in both the perpendicular directions, and thus, $K_B T_c$ in units of J was calculated to be:

$$T_c = \frac{2}{\ln(1 + 2\sqrt{2})} \quad (4)$$

3.1 Critical exponents

In the neighbourhood of the critical temperature the thermodynamic quantities(TDQ) show a behaviour which can be described by so-called critical exponents¹. These exponents theoretically calculated for Ising model are shown in table 1.

Relations of TDQ	Critical Exponent
$c_v(T) \propto T - T_c ^{-\alpha}$	$\alpha = 0$
$m(T) \propto (T_c - T)^\beta, T < T_c$	$\beta = \frac{1}{8}$
$\chi(T) \propto T - T_c ^{-\gamma}$	$\gamma = \frac{7}{4}$
$\xi(T) \propto T - T_c ^{-\nu}$	$\nu = 1$

Table 1: Thermodynamic quantities (TDQ) and there relation to T_c .

4 Simulations

4.1 Monte carlo simulations

The simulation for the model is done using monte carlo methods. Monte Carlo (MC) simulation is a very important class of stochastic methods for calculating thermal properties of many-particle systems—arguably these are the most important numerical techniques in statistical physics. These are based on efficient non-uniform sampling schemes, named importance sampling (Markov chain). By using importance sampling, the configurations (spins (Metropolis), or bonds between the spins (SW and Wolff)) of a finite but large many-body system can be generated according to the Boltzmann distribution, so that thermal expectation values are obtained as simple arithmetic averages of functions “measured” on the configurations.

4.1.1 Detailed balance

Markov chains generate each configuration based on a probability distribution that depends on the previous configuration, thereby improving efficiency. This efficiency is characterized by the transition probability $P(C_j \rightarrow C_i)$, representing the likelihood of transitioning from configuration C_j to C_i . To ensure accurate results, MCMC simulations must satisfy the detailed balance equation (derivation of this is given on my GitHub project repository):

$$\frac{P(C_i \rightarrow C_j)}{P(C_j \rightarrow C_i)} = \frac{P(C_j)}{P(C_i)}, \quad (5)$$

In statistical mechanics the configuration probabilities $P(C_i/)$ is given by:

$$P(C_i) = \frac{W(C_i)}{Z}, \quad W(C_i) = e^{-E(C_i)/T} \quad (6)$$

4.2 Building up the algorithms

We have used two types of algorithm, one is traditional Metropolis algorithm³ and the other one belongs to the group of algorithms called cluster algorithms. We present first one quite shortly as it has been discussed already in deep detail in lectures. The very good description of this is also given in these lecture notes by Professor Anders W. Sandvik.⁵

4.2.1 Metropolis algorithm

The 2D Ising model is analysed by simulating its behaviour for a square $N \times N$ lattice (N^2) lattice sites. In order to mimic the infinite 2D lattice, periodic boundaries are applied, and also every spins is assigned either “up” or “down” configuration.

In Metropolis algorithm (MA), a spin is selected at random location (x, y) in a specific configuration $(P(C_i))$ and is flipped with a probability 1 if the energy difference between the configuration $P(C_i)$ and $(P(C_{i+1}))$ is negative, or if it is positive then it is flipped with a probability $P(C_i \rightarrow C_{i+1})$ equal to the Boltzmann distribution $e^{\beta \Delta E(C_i \rightarrow C_{i+1})}$. The energy for the given configuration $P(C_i)$ is calculated according to the formula presented in 2 and the interaction is considered with nearest four neighbours in 2D ising case.

$$\frac{W(C')}{W(C)} = \exp\left(\frac{J}{T} \sum_{i,j} (\sigma_{i+1} \sigma_{j+1} - \sigma_i \sigma_j)\right), \quad (7)$$

Critical slowing down: MA gives good result as long as the temperature is less than the critical temperature, once it reaches near the critical temperature, the correlation time becomes very large and it takes long time for the configuration to becomes statistically independent of each other. Mathematically arguing the fact that, correlation between the spins goes proportional to the length of the lattice.

$$\tau \propto L^z \quad (8)$$

also correlation length diverges near the critical temperature as theoretically derived¹,

$$\tau \propto \zeta^z \quad \text{and thus} \quad \tau \propto |T - T_c|^{-\nu} \quad (9)$$

The coefficient z called as dynamical exponent changes its value depending upon the type of algorithm one uses, and hence called dynamical. This coefficient z calculated for MA is 2.2.

4.2.2 Cluster algorithms

The phenomenon of critical slowing down significantly hampers the investigations of systems near critical points. Even when systems are distant from critical points, auto-correlation times are quite long or in naive terms, the configurations is not completely independent from the past configurations in a short time interval. However, in certain instances, these challenges can be mitigated, or entirely eradicated, by employing cluster algorithms. These algorithms allow for the simultaneous flipping of a large number of spins, leading to accelerated evolution of configurations.

It should be noted that simultaneous flips of several spins in the Metropolis method are also possible in principle, but then the weight ratio in equation 16 becomes very small on average, reflecting the fact that the new energy is very likely to go up if several spins are flipped, and very few such updates will be accepted. In a cluster algorithm one constructs clusters of spins in such a way that the whole cluster can be flipped with a high probability (1/2 or 1 depending on the formulation). The aspect that have made the cluster algorithm very popular is the fact that they do have local correlation but here we do not have

the acceptance-rejection criteria unlike MA. This causes the correlation time to become less, and hence decreasing the value of dynamical exponent.

We have used 2 cluster algorithms namely Swendsen-Wang (SW) and Wolff algorithm to generate those same results for Ising model and then compared them.

Swendsen-Wang (SW) algorithm

Swendsen and Wang came up with a different algorithm than single flips methods in 1987 which is based on spin clusters instead of individual spins. In contrast to the MA, the SW method does not scan the spin sites but the bonds in the Ising lattice in lexicographic order. These bonds connect two neighbouring spins and represent the interaction between the spins. The bonds can be deleted (no interaction) or frozen (infinitely strong interaction).

In SW algorithm, starting from the initial configuration of our random choice ($P(C_0)$), we set up the coordinates to each spin point in the lattice, and introduce the bond index b , corresponding to the pair of interaction spins $\sigma_{i(b)}\sigma_{j(b)}$, $b = 1, 2, \dots, N_b$, where the number of bonds $N_b = dN$ for a d -dimensional cubic lattice with N -sites and periodic boundary conditions.

Thus, the energy of the ferromagnetic Ising model is:

$$E(\sigma) = -|J| \sum_{i=1}^{N_b} \sum_{j=1}^{N_b} [\sigma_{i(b)}\sigma_{j(b)} + 1] \quad (10)$$

A constant $-|J|$ has been added to the energy of each bond, such translation of energies are used many-where in physics, and it sometimes creates significant helps, like the case discussed here. Now, using the bond energies E_b , we can write the partition function as:

$$Z = \sum_{\sigma} e^{E(\sigma)/T} = \sum_{\sigma} \prod_{b=1}^{N_b} e^{E_b/T} = \sum_{\sigma} \prod_{b=1}^{N_b} [1 + (e^{E_b/T} - 1)] \quad (11)$$

We now define a bond function with arguments 0, 1 corresponding to the two terms on the right hand side above:

$$F_b(0) = 1 \quad (12)$$

$$F_b(1) = e^{E_b/T} - 1 \quad (13)$$

and then write the partition function as,

$$Z = \sum_{\sigma} \prod_{b=1}^{N_b} [F_b(0) + F_b(1)] \quad (14)$$

The bond function F_b depends implicitly on the spins connected by bond b ;

$$\begin{aligned} F_b(0) &= 1, \text{ independent of } \sigma_{i(b)}, \sigma_{j(b)} \\ F_b(1) &= e^{E_b/T} - 1 = \begin{cases} e^{2|J|/T} - 1, & \text{if } \sigma_{i(b)} = \sigma_{j(b)}, \\ 0, & \text{if } \sigma_{i(b)} \neq \sigma_{j(b)}. \end{cases} \end{aligned} \quad (15)$$

For a non-vanishing contribution to the partition function 14, the bond is allowed only when the spins are same as can be predicted directly from equation 15, and we refer such bonds as filled bonds. In this combined space of spins and bonds, the configuration weight in the partition function, in equation 14 is:

$$W(\sigma) = \prod_{b=1}^{N_b} F(b) \quad (16)$$

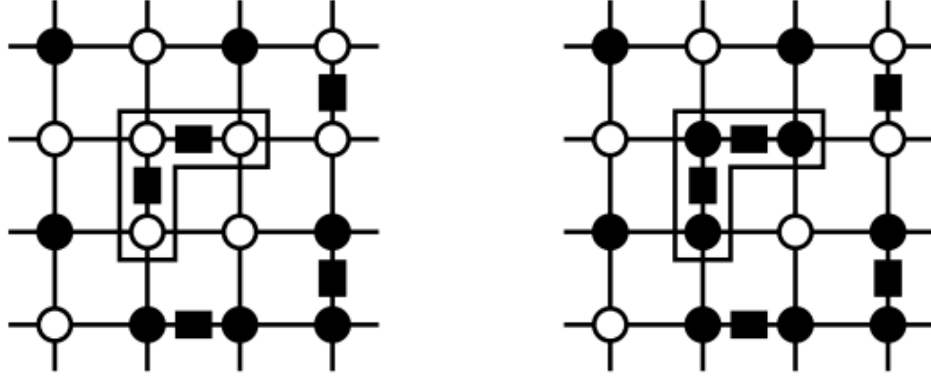


Figure 1: A spin configuration (circles) in which filled bonds (thick lines between circles) have been cast between equal spins according to the appropriate probability

Hence the spin configuration affects the weight only by imposing restrictions on where the filled bonds can be placed. There is a huge difference between the “filled bonds” and simply the “bonds”, which is what I am trying to emphasize in this algorithm.

This scheme relies critically on the added constant $-J$ in each bond energy in equation 10. Without this term there could be (with a different probability) filled bonds also between anti-parallel spins, and the weight function would have a more complex dependence on this spins. As we will see, the key feature of the Swendsen-Wang scheme is that the weight is exactly zero if a filled bond is placed between anti-parallel spins, and that if no such illegal bonds are present the weight is independent on the spin configuration as can be seen from equation 15.

For a given spin configuration, we define the probability of a bond configuration corresponding to the weight in equation 16:

$$P = \frac{F(i, j)}{F(0) + F(1)}, \quad (17)$$

Hence, the probability of the filled bond is:

$$P = 1 - e^{-2J/T}, \quad \text{if } \sigma_{i(b)} = \sigma_{j(b)} \quad (18)$$

$$P = 0, \quad \text{if } \sigma_{i(b)} \neq \sigma_{j(b)} \quad (19)$$

Now, based on the above mathematical formulation, we can present how to run up the SW algorithm, starting from the initial configuration of our random choice ($P(C_0)$):

1. Specifying the coordinates to each lattice sites and then one form a bond between every pairs of nearest neighbours, with a probability $p_{ij} = 1 - e^{-2\beta J}$.
2. All spins that are connected, directly or indirectly, via bonds belong to a single cluster. Thus, the bond assignment procedure divides the system into clusters of parallel spins (a so called cluster decomposition). The bond probability (and hence the typical cluster size) grows with increasing coupling strength βJ (decreasing temperature). For a finite temperature, $p_{ij} < 1$, and hence a cluster is generally a subset of all spins of a given sign. In common words, two spins of the same sign need not belong to the same cluster, even if these spins are adjacent on the lattice.
3. All spins in each cluster are flipped collectively with a probability $\frac{1}{2}$. The reason for choosing half is one's symmetric choice than any physical intuition. Of-course one can choose any number between 0 to 1, and the algorithm works well, provided we do not biased the aspect by choosing specifically 1 or 0.

4. All bonds are erased and the “cluster move” is complete; a new spin configuration has been created, and then one proceeds to step -1 again.

The clusters created in the Swendsen–Wang algorithm have a structure that is very efficient at destroying non-local correlations (*as all the spins are correlated to each other, because when one measure energy of one spins, it requires information about the other nearest 4, and this chain continues upto the length of the lattice, and this correlation is destroyed here*). As a result, the dynamical critical exponent z is lowered to a much smaller value and independent configurations can be generated at a much faster rate than with a single-spin flip algorithm. This advantage only holds in the vicinity of the critical temperature, which also happens to be the most interesting point in the study of lattice spin models such as the Ising model.

This method ofcourse satisfy the detailed balance condition⁶ and also the dynamic critical exponent z for the SW method is much smaller, $z = 0.35$ and our calculated value is 0.377.

Wolff Algorithm

Wolff⁷ introduced a so-called single-cluster variant of this algorithm. In the SW algorithm, small and large clusters are created. While the destruction of critical correlations is predominantly due to the large clusters, a considerable amount of effort is spent on constructing the smaller clusters. In Wolff’s implementation, no decomposition of the entire spin configuration into clusters takes place. Instead, only a single cluster is formed, which is then always flipped. If this cluster turns out to be large, correlations are destroyed as effectively as by means of the large clusters in the SW algorithm, without the effort of creating the smaller clusters that fill up the remainder of the system. If the Wolff cluster turns out to be small, then not much is gained, but also not much computational effort is required. As a result, critical slowing down is suppressed even more strongly than in the SW algorithm, and the dynamical critical exponent z is even smaller.

1. A spin i is selected at random.
2. All nearest neighbors j of this spin are added to the cluster with a probability $p_{ij} = 1 - \exp(-2\beta J)$, provided spins i and j are parallel and the bond between i and j has not been considered before.
3. Each spin j that is indeed added to the cluster is also placed on the stack. Once all neighbors of i have been considered for inclusion in the cluster, a spin is retrieved from the stack and all its neighbors are considered in turn for inclusion in the cluster as well, following step (2).
4. Steps (2) and (3) are repeated iteratively until the stack is empty.
5. Once the cluster has been completed, all spins that belong to the cluster are flipped with $\frac{1}{2}$ probability.

This method ofcourse satisfy the detailed balance condition⁷ and also the dynamic critical exponent z for the Wolff is even much smaller than the previous 2 methods $z = 0.20$ and our calculated value shows promising result of 0.19.

4.3 Cluster algorithms for off-lattice model

After the success of SW and Wolff algorithm, these models found huge applications in off-lattice model. The geometric cluster algorithm described is formulated for particles that interact via hard-core repulsion’s only. We have discussed the case of that hard core repulsion’s only between the spheres.

$$V(r_1, r_2) = \begin{cases} 0 & \text{if } |r_1 - r_2| > \sigma, \\ \infty & \text{if } |r_1 - r_2| < \sigma. \end{cases} \quad (20)$$

To devise a general cluster flipping method for off-lattice systems, e.g. fluids, was not immediately obvious. The first step was taken by Dress and Krauth in 1995 who used geometric operations to design a cluster algorithm for hard spheres.

1. In a given configuration C , a “pivot” is chosen at random.
2. A particle i at position r_i is selected as the first particle that belongs to the cluster. This particle is moved via a point reflection with respect to the pivot. In its new position, the particle is referred to as i' , at position r'_i .
3. Each particle j that interacts with i or i' is now considered for addition to the cluster. A particle j that interacts with i both in its old and in its new position is nevertheless treated once. Unlike the first particle, particle j is point-reflected with respect to the pivot only with a probability $p_{ij} = \max[1 - \exp(-\beta \Delta_{ij}), 0]$, where $\Delta_{ij} = V(|r'_i - r_j|) - V(|r_i - r_j|)$.
4. Each particle j that is indeed added to the cluster (i.e., moved) is also placed on the stack. Once all particles interacting with i or i' have been considered, a particle is retrieved from the stack and all its neighbors that are not yet part of the cluster are considered in turn for inclusion in the cluster as well, following step 3.
5. Steps 3 and 4 are repeated iterative until the stack is empty. The cluster move is now complete.

The very good description of the GCA (Geometric cluster analysis) is given in these lecture notes.²

4.4 Bootstrapping

The ensemble averages of the thermodynamic quantities listed in section 5 are determined by the bootstrapping technique. For a given data set d_{MC} of the MC simulations consisting of K data points, the bootstrapping technique randomly picks K data points from this set. From these randomly picked K data points the quantity Q is calculated in the same manner as it would have been calculated from the original data set. This process is repeated many times (say M) times for the same data set d_{MC} . The M different values of Q are then used to determine the average and the standard deviation of Q .

4.5 Trial function for obtaining the exponents

The critical exponents corresponding to the quantities mentioned in table 1 are determined by fitting a trial function to the simulated data for C_v, χ, m . The trial function for m and χ is given by:

$$f_{trial} = a|\tilde{T}|^b \quad (21)$$

where, $\tilde{T} = T - T_c$ and a, b are the fitting parameters. In the case of magnetization, the absolute sign of temperature is just replaced by \tilde{T} .

The dynamic critical exponent z can be extracted from the peak position of a quantity. In general the peak position scales as $T_{peak} \propto L^z$. Hence, data for multiple system sizes is needed. This simulated data is fitted to the following trial function:

$$f_{trial,z} = L^{-a} + c, \quad (22)$$

where L is the length of the lattice and a, c are the fitting parameters. We will choose the magnetic susceptibility chosen to determine z since its data shows clearly defined peaks unlike specific heat.

5 Thermodynamics quantities

In the simulation, we have computed the following thermodynamic quantities from there standard formula.

$$m = \frac{1}{N} \langle \sum_i \sigma_i \rangle \quad (23)$$

Magnetization, this is the ordered parameter for the ferromagnetic phase transition. This becomes zero after the critical temperature, and was aliged in the same direction below critical temperature.

The fluctuations of the energy of the system are used to calculate the specific heat of the system.

$$C_v = \frac{k_B \beta^2}{N} (\langle E^2 \rangle - (\langle E \rangle)^2) \quad (24)$$

The magnetic susceptibility (χ) is a measure of magnetic properties of a material. It indicates whether a material is attracted to or repelled by a magnetic field.

$$\chi = \beta N (\langle m^2 \rangle - (\langle m \rangle)^2) \quad (25)$$

Since the magnetic susceptibility over temperature is a measure of the change of magnetisation over temperature, a sharp peak is expected at the critical temperature due to the sudden change of the magnetisation at the phase transition.

6 Results

The results presented in this section are for a 2D Ising lattice of sides $L = 40$ unless specifically stated otherwise. The temperature interval chosen starts from 1.5 going up to 1.5 in 80 equidistant steps. Simulations start with all spins in the up position. At every temperature the simulation runs for 5100 MC steps of which the last 5000 MC steps are used to determine the quantities. The first 100 MC steps are needed for the system to adapt to the new temperature (this is called as thermalization). Furthermore, uncertainties are calculated using the bootstrap method with $M = 1000$.

6.1 Thermodynamic quantities against temperature

The graph for the various thermodynamic quantities using SW algorithm are presented here. The similar graphs are also generated for Wolff as well as Metropolis algorithm, just the difference is which algorithm takes what amount of time, and how far in time correlation dies. The graphs for magnetization, specific heat per spin and magnetic susceptibility per spin is presented in figure 2, 3 and 6 respectively.

As expected the peak in the spectrum becomes less flattened when the system size increases and will eventually reach the relation stated in table 1 near the critical temperature. It can also be seen that the location of the peak shifts to lower temperatures when L increases and will eventually go asymptotically to the theoretical value 2.269 for $L \rightarrow \infty$. Similarly for the magnetic susceptibility, the peak within the spectrum becomes more apparent for larger system sizes and also shifts to lower temperatures, and in the theoretical limit of $L \rightarrow \infty$ the peak will be located at the critical temperature. This is as expected since it should behave as the variance of the magnetisation.

The critical temperature is determined from the χ peak at $L = 40$, because this has a more distinctive peak than the specific heat. From figure 6, the highest peak is found to be at $T = 2.30$

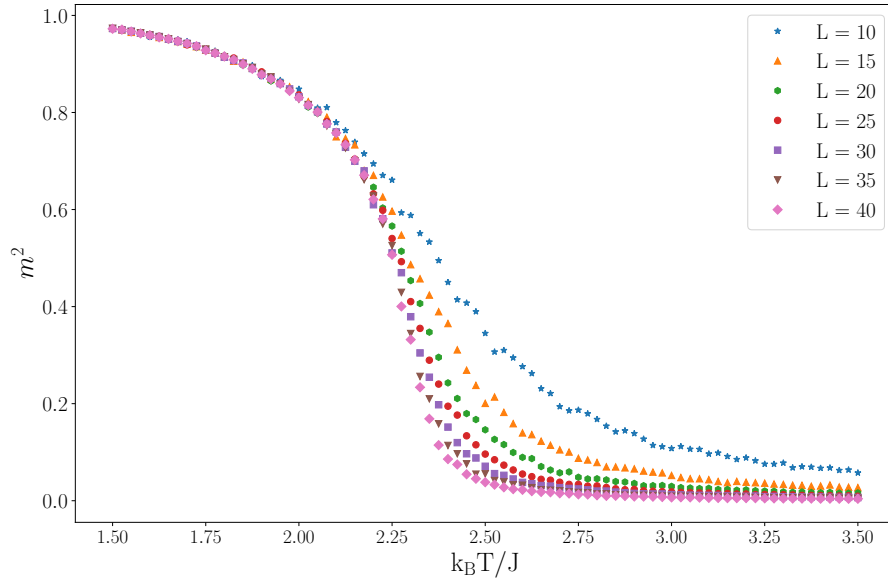


Figure 2: Graph showing the magnetization (actually it's square for symmetry reason) per spin against temperature using SW algorithm for different lattice sizes.

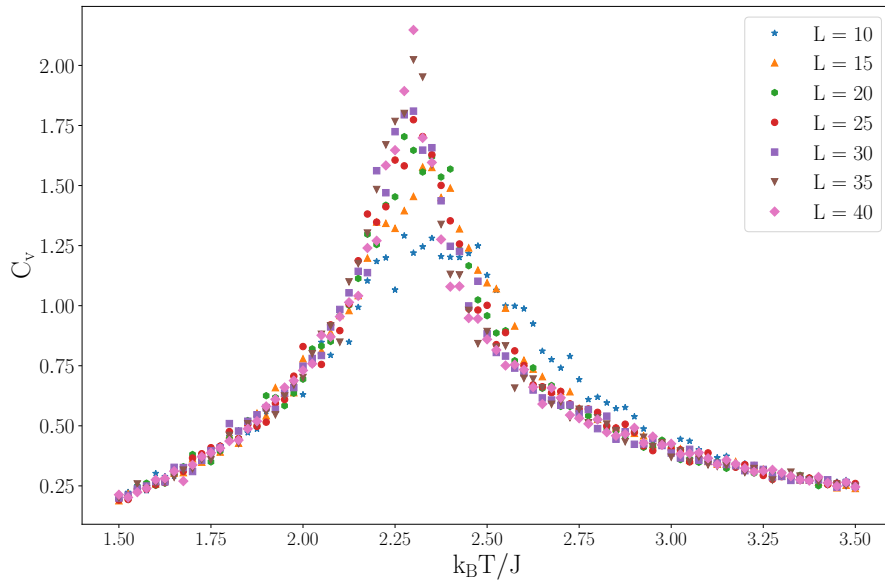


Figure 3: Graph showing the specific heat per spin against temperature using SW algorithm for different lattice sizes.

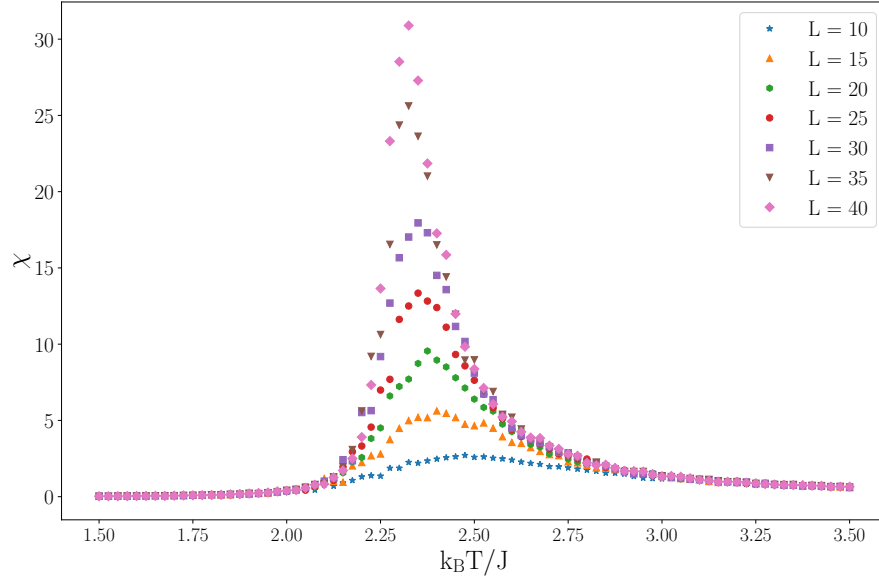


Figure 4: Graph showing the magnetic susceptibility against temperature using SW algorithm for different lattice sizes.

6.2 Exponents

Already presented theoretically, these exponents are then calculated through simulations. We calculated both the dynamic as well as critical exponents for all the three algorithms.

6.2.1 Dynamical exponent

To acquire data for multiple system sizes, L is varied from 10 to 40 in steps of 5. This is done for 2100 MC steps in stead of 5100 to save on computation time. Fitting to equation 22 results for the SW algorithm in $z_\chi = 0.377$ and for Wolff $z_\chi = 0.19$, which agrees from there papers.⁷⁶ We tried to compute z_χ for metropolis as well, but the errors are as big as the quantity itself, so nothing can be predicted. This problem might be overcome if we increase the length of the lattice but this is not feasible option is quite a less time.

The graph for the auto-correlation time against lattice size is shown, from where we extracted z for SW.

6.2.2 Critical exponents

The critical exponents for the magnetisation and the magnetic susceptibility are determined by fitting the trial function in equation 21 to the acquired data. The trial function is defined with respect to $T - T_c$ while the simulated data is defined with respect to T , so we translated all the T 's to, $T - T_c$, which results in an axis with both $T - T_c > 0$ and $T - T_c < 0$. The fit to the magnetic susceptibility is shown in figure 7

From figure 8, one can note that the flattening of the curve for $T - T_c$ close to 0, and also different slopes of the curve for $T - T_c > 0$ and $T - T_c < 0$. The reason for such is just the finite dimensions of the system. The corresponding critical exponents obtained from wolff and SW are also shown in the table 2 and 3.

One can now note from table 2, exponents corresponding to $\langle m \rangle$, i.e., β shows very accurate answer as comparison to theoretical value in table 1. But, this is not the perfect case with γ , the reason being the finite size effect of the system is getting dominated, but actually, theoretical value lays between the two

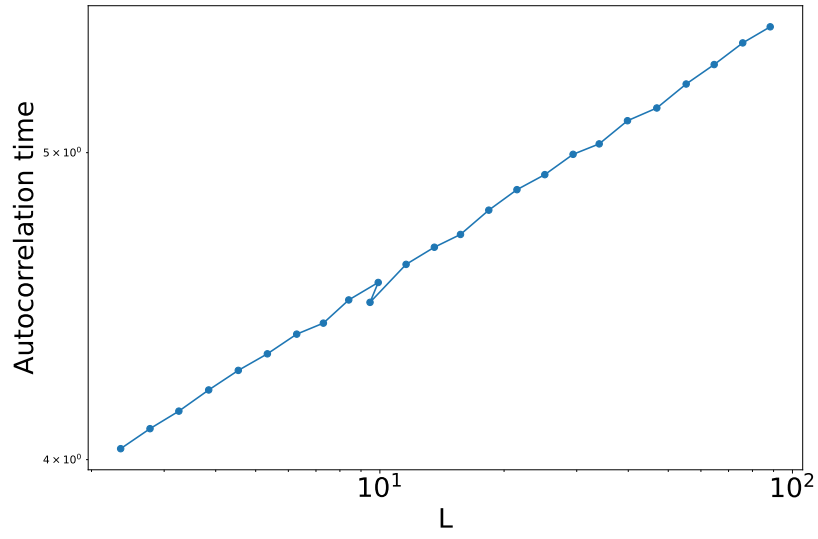


Figure 5: Graph showing the autocorrelation time against lattice size for SW algorithm. Form here, value of z was found to be 0.37 against the SW value 0.35⁶

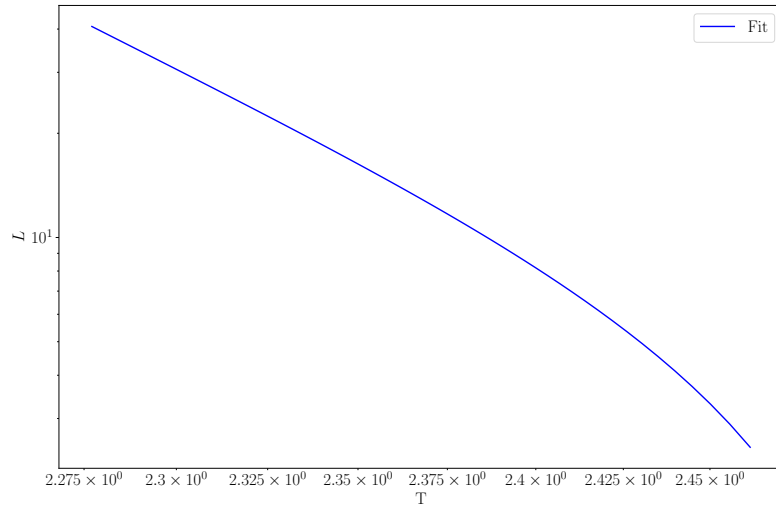


Figure 6: Graph showing the lattice size against critical temperature for wolff algorithm. Form here, value of z was found to be 0.19 against the value 0.20 obtained by Wolff⁷

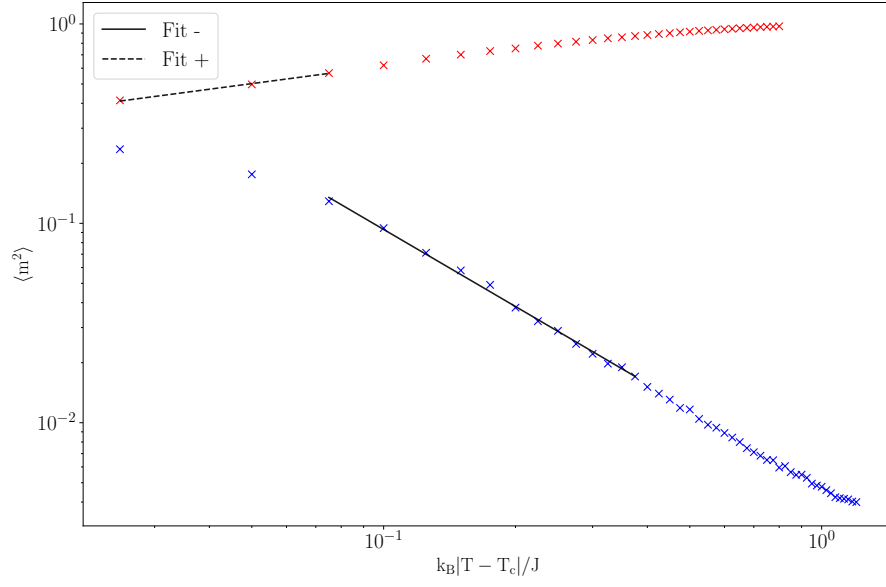


Figure 7: Logarithmic plot of $\langle m^2 \rangle$ including fit. For $T - T_c > 0$ is in blue with accompanying fit 'Fit +' and for $T - T_c < 0$ in red with 'Fit -'.

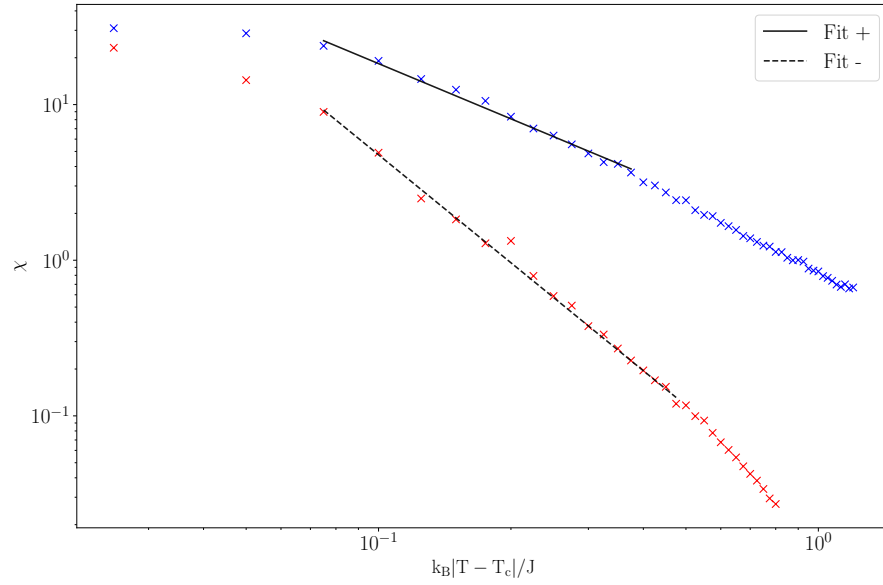


Figure 8: Logarithmic plot of χ including fit. For $T - T_c > 0$ is in blue with accompanying fit 'Fit +' and for $T - T_c < 0$ in red with 'Fit -'.

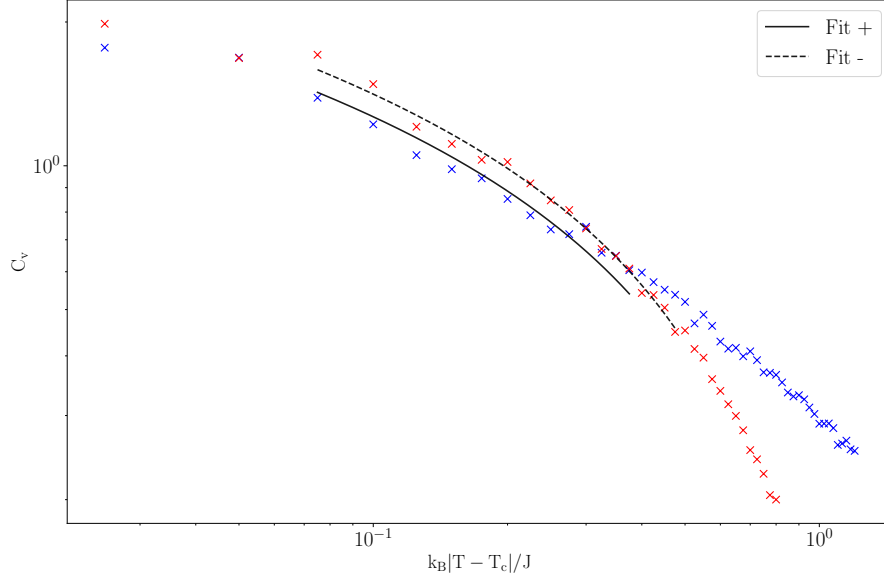


Figure 9: Logarithmic plot of c_v including fit. For $T - T_c > 0$ is in blue with accompanying fit ‘Fit +’ and for $T - T_c < 0$ in red with ‘Fit -’.

found exponents with some relative difference, as can be seen in table for 2 and 3. The relative difference is less for Wolff, and high for SW, showing the former is better than the later.

Table 2: Critical exponents obtained from SW algorithm

	$(T - T_c) < 0$	$(T - T_c) > 0$
m	$\beta = 0.112$	—
χ	$\gamma = 2.27$	$\gamma = 1.37$

Table 3: Critical exponents obtained from Wolff algorithm

	$(T - T_c) < 0$	$(T - T_c) > 0$
m	$\beta = 0.123$	—
χ	$\gamma = 2.01$	$\gamma = 1.47$

Also, the graphs for specific is not perfect because it shows the logarithmic divergence, and the finite size of the system is actually responsible for this. We are not able to extract perfect the exponents corresponding to it, but the logarithmic fit is given in figure 9.

6.3 Comparison of three algorithm

We have compared the time for all the three algorithms, and fitted the time against the lattice size, whose values are shown in the figure 10 itself. The time is then normalized with respect to the highest value. The least time is for the Wolff algorithm making it the more promising and significant over other two. Fitting to normalised simulation time we learn that all three algorithms scale quadratic in system size, though Wolff is not very good fit.

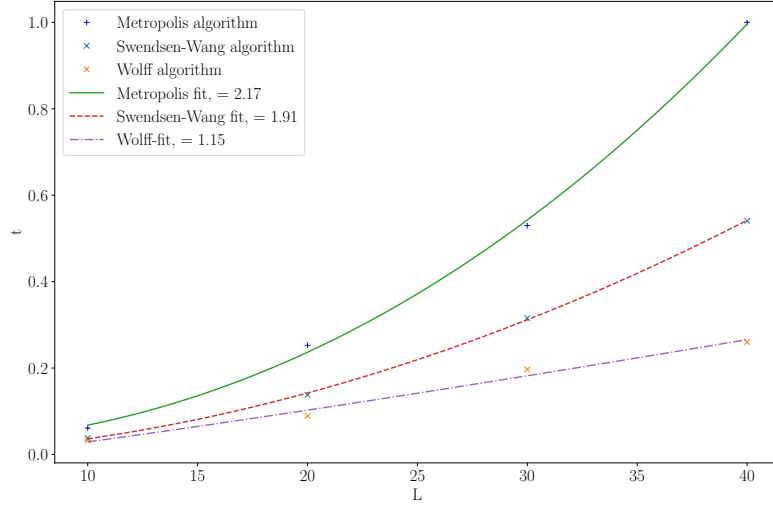


Figure 10: Simulation performance (normalized run time) for Metropolis, SW and Wolff algorithm against different lattice size.

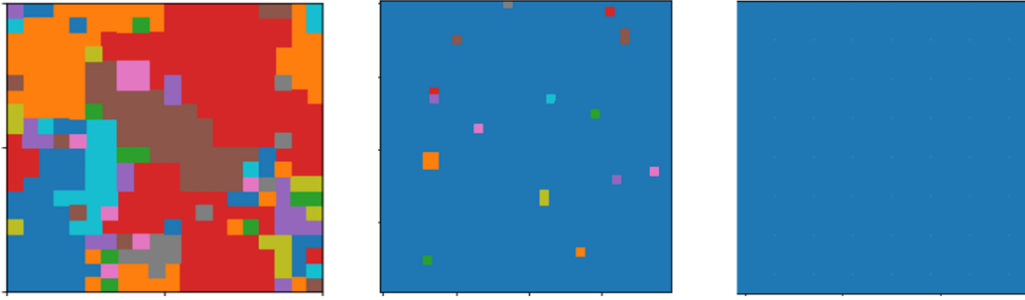


Figure 11: Cluster view for SW algorithm at different MC steps for $L = 40$ at $T = 2.0$

This view on the simulation performance is still omitting the fact that the Wolff algorithm has a much smaller dynamical critical exponent than the SW, which is better over MA. This means that the simulation can be run for a shorter time period to determine the physical quantities and thus makes the Wolff as well as SW algorithm much more efficient over MA.

6.4 Clusters for Wolff and SW algorithms

The view of how cluster looks in SW and Wolff algorithm is shown in the figure 11 and 12 respectively below critical temperature at $T = 2.0$. After the critical temperature, the cluster just appears like, it do appears in step 1 of both SW and wolff.

6.5 Geometric cluster

The GCA as explained in theory part is applied on hard sphere model. The cluster is shown for three random mc steps at different time in figure 13.

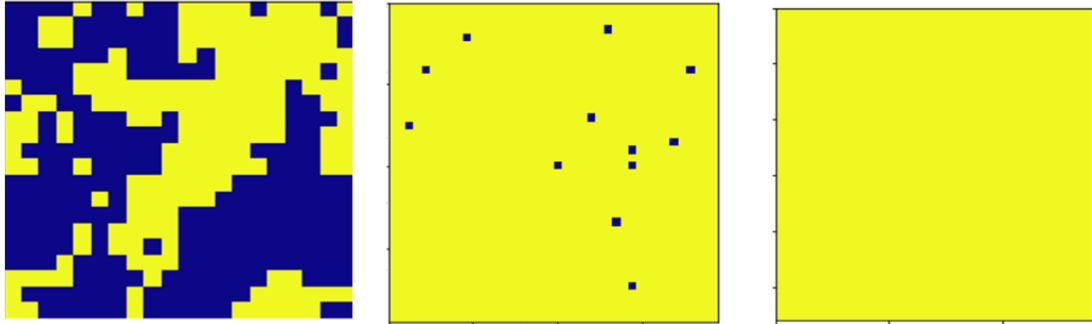


Figure 12: Cluster identification for Wolff algorithm at different MC steps for $L = 40$, namely 100, 700, 1400 below critical temperature $T = 2.0$

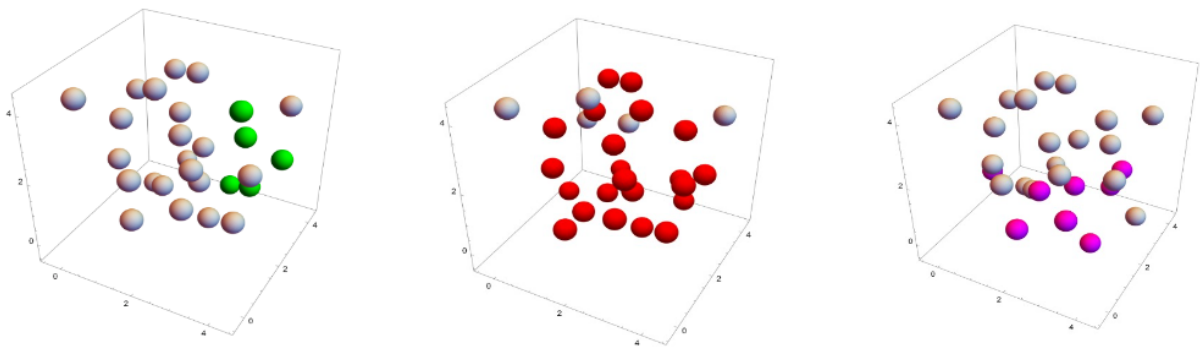


Figure 13: GCA on Hard sphere in shown for three random MC steps. At every step some pivot is chosen and then they are flipped with respect to that pivot.

7 Conclusion

The results presented in this report suggest that the simulation exhibits appropriate behavior with increasing system size, as all physical quantities appear to converge towards the limit of an infinite system. The critical exponent for magnetization closely matches the theoretical value for an infinite system. However, the critical exponents for other physical quantities display deviations from the infinite system limit, likely due to the finite dimensions of the system and the somewhat arbitrary selection of the fitting region. Further simulations with larger system sizes are necessary to validate these observations regarding system size. The selection of the fitting region was primarily based on visual inspection and lacks a solid theoretical foundation. Additional understanding of statistical methods is required to establish this theoretical background. The determination of the dynamical critical exponent aligns with existing literature.

The GCA, which is the application of cluster algorithms in off lattice models, is also presented. The project was aimed to understand how cluster algorithms are more efficient and general. This can have application to many models, where dynamics is hampered by critically slowing down. We both aimed to apply this for LJ potential, but unfortunately we are not able to because of the lack of time.

Bibliography

- [1] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics*. London: Academic Press, 1982.
- [2] Jiwen Liu and Erik Luijten. “Rejection-Free Geometric Cluster Algorithm for Complex Fluids”. In: *Phys. Rev. Lett.* ().
- [3] Nicholas Metropolis et al. “Equation of State Calculations by Fast Computing Machines”. In: *Journal of Chemical Physics* (1953). DOI: [10.1063/1.1699114](https://doi.org/10.1063/1.1699114).
- [4] Lars Onsager. “Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition”. In: *Phys. Rev.* (1944).
- [5] Anders W. Sandvik. *Monte Carlo Simulation Lecture Notes*. Lecture notes. Boston University. URL: <https://physics.bu.edu/~py502/lectures5/mc.pdf>.
- [6] Robert H. Swendsen and Jian-Sheng Wang. “Nonuniversal critical dynamics in Monte Carlo simulations”. In: *Phys. Rev. Lett.* 58 (1987), pp. 86–88.
- [7] Ulli Wolff. “Collective Monte Carlo Updating for Spin Systems”. In: *Phys. Rev. Lett.* 62 (1989).