

Simulations on Infinite Lattices

Momchil Ivanov
IPSP Student Nr. 9918005
Email: momchil@xaxo.eu

April 26, 2011

1 Introduction

The Monte Carlo method is usually used to simulate systems of finite size. The results from these simulations depend on the size of the system. One needs to do finite size scaling in order to obtain results for infinite systems, provided one knows the correct scaling laws. It is therefore sometimes necessary to be able to obtain data for systems where the scaling laws are not known. An algorithm that gives direct results for infinite systems and thus overcomes the limitations of simulations on finite systems has been proposed recently in [1]. The authors claim that it produces results for infinite systems using observables in the bond representation of the Ising model on a 2-dimensional square lattice without external magnetic field. This work describes an implementation of the proposed method and compares the obtained results with exact analytic solutions from [2] and simulations on finite systems. A discussion is made about the dependence of the results on the system initialisation, the temperature and the number of Monte Carlo steps.

2 Theoretical background

2.1 The Ising model

The Ising model is a simple model of a magnet. The model consists of discrete objects called spins which can take one of two different states. The spins are placed on a lattice or graph and each spin interacts only with its nearest neighbours. The energy E of the system without an external magnetic field is defined as

$$E = - \sum_{\langle ij \rangle} J_{ij} s_i s_j, \quad (1)$$

where the sum is over nearest neighbours i and j , J_{ij} are the coupling constants and $s_i = \pm 1$ are the spin states. The partition function Z is defined as

$$Z = \sum_{\{s\}} \exp(-\beta E), \quad (2)$$

where the sum is over all possible spin configurations s and β is the inverse temperature ($\beta = 1/k_b T$). The model considered in this work is on a 2-dimensional square lattice with the isotropic case $J_{ij} = J$. In this case the partition function can be rewritten in the form:

$$Z = \sum_{\{s\}} \sum_{\{n_{ij}\}} \prod_{\langle ij \rangle} e^{\beta J [(1-p)\delta_{n_{ij},0} + p\delta_{s_i s_j} \delta_{n_{ij},1}]} \quad (3)$$

with

$$p = 1 - e^{-2\beta J} \quad (4)$$

where n_{ij} are bond variables which can take values 0 or 1, interpreted as “deleted” or “active” bonds. One can use this Fortuin-Kasteleyn [3] representation to construct bonds between equal neighbouring spins with probability p , thus forming a cluster, and then flip the whole cluster. This is the basis of multiple cluster update algorithms. For temperatures T near the critical temperature T_c this update results in a greater phase space move than the one of the Metropolis algorithm, where single spins are flipped. This results in almost complete elimination of the critical slowing down observed with the Metropolis algorithm.

2.2 The Wolff cluster update algorithm

Wolff [4] has proposed a cluster update algorithm based on the work of Swendsen and Wang [5]. Using his simpler method one constructs and flips only one cluster at a time, therefore it is often called single-cluster update. The algorithm works as follows:

1. Start with an empty cluster.
2. Choose a random spin s_i .
3. Add every neighbour s_j of s_i to the cluster with probability

$$P_{add} = 1 - e^{-2\beta J} \quad (5)$$

if $s_j = s_i$ and s_j is not in the cluster.

4. For each spin s_j added to the cluster repeat step 3.
5. Flip all spins in the cluster.

Using this simple update procedure, one can generate different configurations on every Monte Carlo (MC) step. An example of such update is shown on fig. 1. Observables can be measured on every iteration using either the spin or the bond representation.

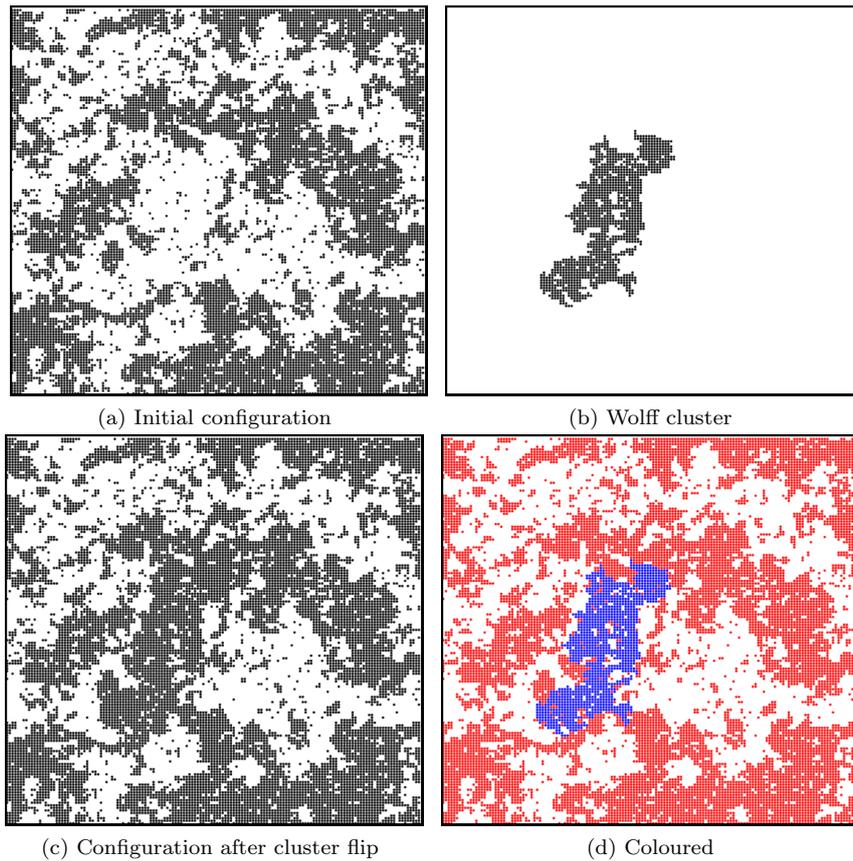


Figure 1: An example spin configuration is shown in fig. 1a - white and dark dots represent the two different spin states. A cluster (fig. 1b) has been constructed using the Wolff algorithm. The resulting configuration after the cluster has been flipped is shown in fig. 1c. A coloured picture of the system is shown on fig. 1d. The change of the system (the cluster being flipped) is represented with blue and the two different spin states are represented with white and red colour.

2.3 The proposed cluster method

A new method for obtaining quantities in the infinite size limit has been proposed in [1]. It introduces a small modification to the Wolff cluster update algorithm:

1. Start with an empty cluster.
2. ~~Choose random spin s_i~~ Always choose spin $s_i = s_{seed}$.
3. Add every neighbour s_j of s_i to the cluster with probability

$$P_{add} = 1 - e^{-2\beta J} \quad (5)$$

when $s_j = s_i$ and s_j is not in the cluster.

4. For each spin s_j added to the cluster repeat step 3.
5. Flip all spins in the cluster.

Using this modification one needs to store only the finite part of the lattice around s_{seed} that the cluster algorithm has reached so far and allows the system to grow whenever new points are to be reached. The stored system grows with the number of MC steps and the growth rate depends on the cluster sizes, hence β . An example update is shown on fig. 2 where one recognises the circular shaped area that the clusters have reached.

2.4 The two-point correlation function

Due to the translation invariance of the Ising model the two-point correlation function $\langle s_i s_j \rangle$ can be considered as a function $C(\vec{r})$ with \vec{r} being the vector connecting s_i and s_j . It has been exactly solved for the 2-dimensional Ising model on a square lattice without external magnetic field. The expressions for the axial and diagonal directions, namely $\langle s_{0,0} s_{0,N} \rangle$ and $\langle s_{0,0} s_{N,N} \rangle$, have been given for large N in [2, 6]. Here $s_{0,0}$ is the spin sitting on the lattice point $(0,0)$, respectively $s_{0,N}$ is the spin sitting on the lattice point $(0,N)$ and $s_{N,N}$ sits on (N,N) . The following equalities have been obtained from [2]:

$$\lim_{\substack{\mathcal{M} \rightarrow \infty \\ \mathcal{N} \rightarrow \infty}} \langle s_{0,0} s_{0,N} \rangle_{\mathcal{M},\mathcal{N}} = \langle s_{0,0} s_{0,N} \rangle \quad (6)$$

$$\lim_{N \rightarrow \infty} \langle s_{0,0} s_{0,N} \rangle = \lim_{N \rightarrow \infty} \langle s_{0,0} s_{N,N} \rangle \quad (7)$$

where $\mathcal{M} \times \mathcal{N}$ are the dimensions of the lattice. For $T > T_c$ the expressions computed in [2] are given as:

$$\begin{aligned} \langle s_{0,0} s_{0,N} \rangle \sim & \frac{1}{(\pi N)^{1/2} \alpha_2^N} \left(\frac{1 - \alpha_1^2}{1 - \alpha_2^{-2}} \right) \frac{1}{(1 - \alpha_1 \alpha_2)^{1/2}} \\ & \times \left[1 + \frac{1}{4} \frac{A_1 \rangle}{N} + \frac{3}{16} \frac{A_2 \rangle}{N^2} - \frac{5}{6} + \frac{15}{64} \frac{A_3 \rangle}{N^3} - \frac{7}{6} \frac{A_1 \rangle}{N^3} + \dots \right] \quad (8) \end{aligned}$$

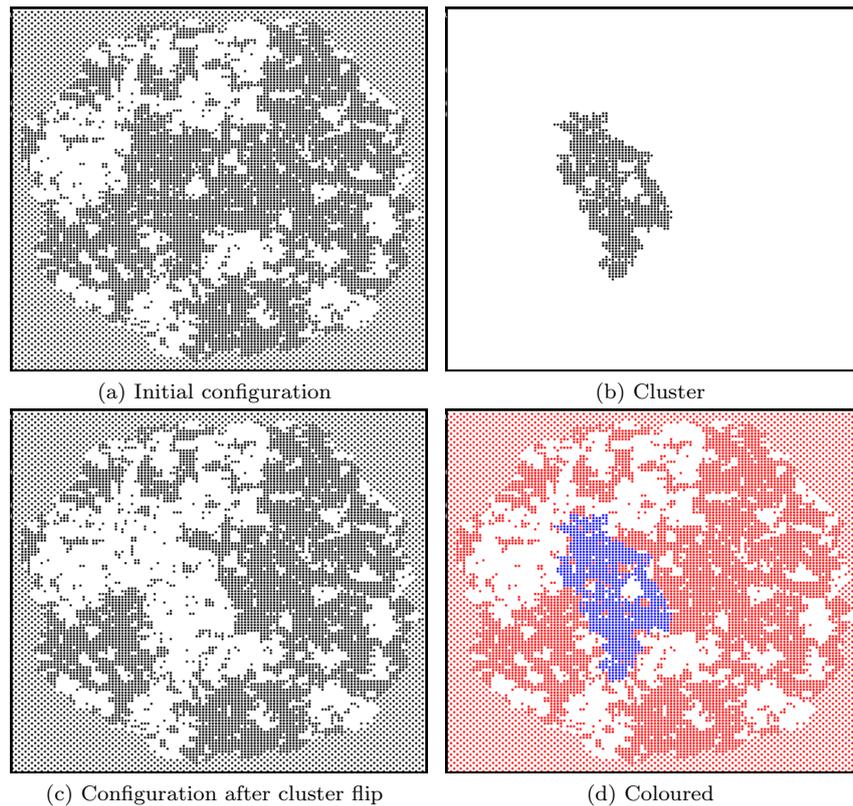


Figure 2: An example spin configuration is shown in fig. 2a - white and dark dots represent the two different spin states. A cluster (fig. 2b) has been constructed using the proposed algorithm. The resulting configuration after the cluster has been flipped is shown in fig. 2c. A coloured picture of the system is shown on fig. 2d. The change of the system (the cluster being flipped) is represented with blue and the two different spin states are represented with white and red colour. The circular area in the centre represents spins that have already been reached, hence flipped, at least once by a cluster update. The outer area represents the initial spin configuration, which in this case is a chess board.

$$\begin{aligned}
\langle s_{0,0} s_{N,N} \rangle &\sim \frac{1}{(\pi N)^{1/2}} \frac{[\sinh 2\beta J_1 \sinh 2\beta J_2]^N}{(1 - (\sinh 2\beta J_1 \sinh 2\beta J_2)^2)^{1/4}} \\
&\times \left[1 - \frac{1}{8} \frac{x'_3}{N} + \frac{1}{128} \frac{9x'_3{}^2 - 8}{N^2} \right. \\
&\left. + \frac{5}{1024} \frac{x'_3(-15x'_3{}^2 + 16)}{N^3} + \dots \right], \tag{9}
\end{aligned}$$

with

$$A_{0>} = 1, \tag{10}$$

$$A_{1>} = -\frac{1}{2}(x_1 - x_2 + x_3), \tag{11}$$

$$A_{2>} = \frac{3}{8}(x_1^2 + x_2^2 + x_3^2) - \frac{1}{4}(x_2 x_3 - x_3 x_1 + x_1 x_2), \tag{12}$$

$$\begin{aligned}
A_{3>} &= -\frac{5}{16}(x_1^3 - x_2^3 + x_3^3) + \frac{3}{16}(x_1^2 x_2 + x_1 x_2^2 - x_2^2 x_3 + x_2 x_3^2 - x_3^2 x_1 - x_3 x_1^2) \\
&+ \frac{1}{8} x_1 x_2 x_3, \tag{13}
\end{aligned}$$

$$x_1 = \cosh 2\beta J_1, \tag{14}$$

$$x_2 = \coth 2\beta J_2, \tag{15}$$

$$x_3 = \frac{\alpha_2^2 + 1}{\alpha_2^2 - 1}, \tag{16}$$

$$\alpha_1 = \tanh \beta J_1 \exp(-2\beta J_2), \tag{17}$$

$$\alpha_2 = \tanh^{-1} \beta J_1 \exp(-2\beta J_2), \tag{18}$$

$$x'_3 = \frac{(\sinh 2\beta J_1 \sinh 2\beta J_2)^{-2} + 1}{(\sinh 2\beta J_1 \sinh 2\beta J_2)^{-2} - 1}. \tag{19}$$

J_1 and J_2 are the coupling constants between $s_{j,k} s_{j,k+1}$ and $s_{j,k} s_{j+1,k}$ respectively. The values of $\langle s_{0,0} s_{N,N} \rangle$ for small N deviate from what is expected, namely the curves should go to 1 for $N = 0$. Moreover as $\beta \rightarrow \beta_c$:

$$\beta_c = \frac{1}{2} \ln(1 + \sqrt{2}) \approx 0.440686, \tag{20}$$

the range for which the function produces wrong values for small N gets longer (see fig. 3).

The difference $O_1(N)$ between $\langle s_{0,0} s_{N,N} \rangle$ and $\langle s_{0,0} s_{0,N} \rangle$ plotted in fig. 4 gets smaller when approaching the critical point ($\beta \rightarrow \beta_c$). In the range $N \in [0; 150]$ that is being considered in this work, the relative difference $O(N)$ is less than 1%, therefore only one of the functions will be discussed later.

$$O_1(N) = \frac{|\langle s_{0,0} s_{N,N} \rangle - \langle s_{0,0} s_{0,N} \rangle|}{\langle s_{0,0} s_{0,N} \rangle} \tag{21}$$

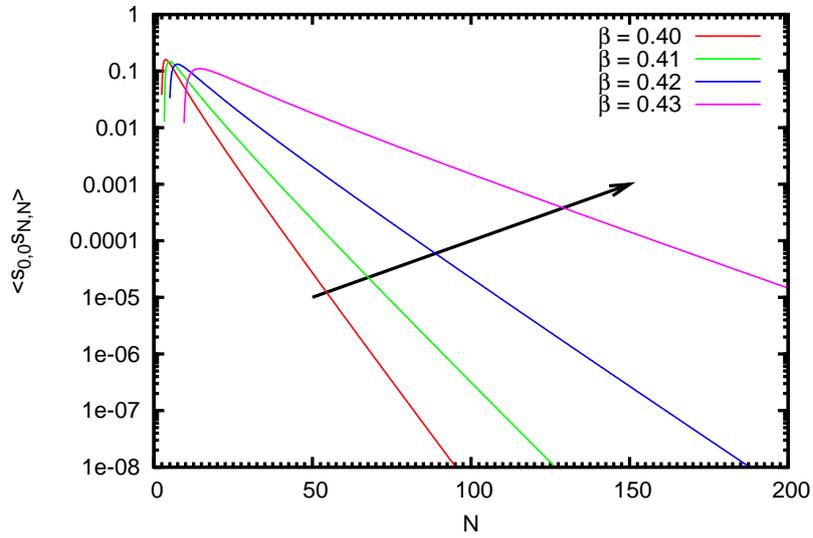


Figure 3: Dependence of $\langle s_{0,0}s_{N,N} \rangle$ on β , the arrow indicates increasing values of β .

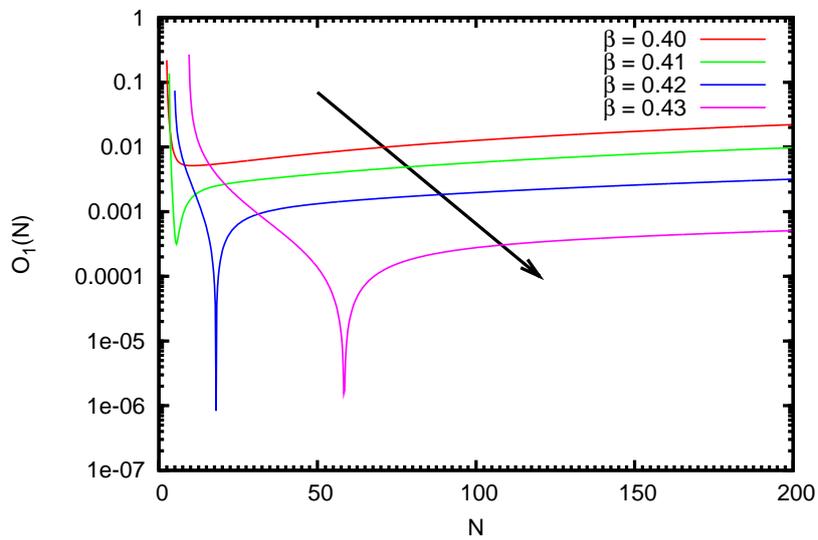


Figure 4: Dependence of $O_1(N)$ on β , the arrow indicates increasing values of β .

3 Simulations

3.1 Wolff cluster update and effects on finite size lattices

When simulating the Ising model, one takes a finite size lattice with dimensions $L \times L$ and uses toroidal periodic boundary conditions. Under these conditions one can measure the correlation functions $\langle s_{0,0}s_{0,N} \rangle$ and $\langle s_{0,0}s_{N,0} \rangle$ only for $N \in [0, L]$. Moreover, because of the boundary conditions, the measured functions become symmetric around the point $N = \frac{1}{2}L$ as one can see on fig. 5. They are symmetric at $r = \frac{1}{2}L$ for the axial directions and at $r = \frac{1}{\sqrt{2}}L$ for the diagonal direction. Here \vec{r} is the vector connecting the two spins s_i and s_j for which the correlation function $\langle s_i s_j \rangle$ is being considered and $r = |\vec{r}|$ is the length of the vector. The values for the simulated $\langle s_{0,0}s_{0,N} \rangle$ and $\langle s_{0,0}s_{N,0} \rangle$ lay on top of each other as expected due to the 90° rotational invariance of the 2-dimensional Ising model. Their difference $O_2(N)$ (plotted in fig. 6) is about 1% and indicates that there is no error in the implementation of the Wolff algorithm.

$$O_2(N) = \frac{|\langle s_{0,0}s_{0,N} \rangle - \langle s_{0,0}s_{N,0} \rangle|}{\langle s_{0,0}s_{N,0} \rangle} \quad (22)$$

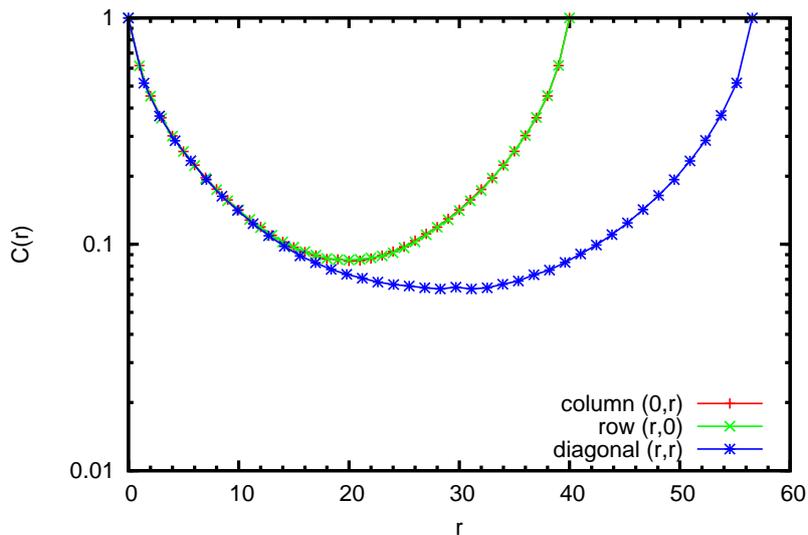


Figure 5: Measured correlation function $C(\vec{r})$ in different lattice directions, namely column $\langle s_{0,0}s_{0,N} \rangle$, row $\langle s_{0,0}s_{N,0} \rangle$ and diagonal $\langle s_{0,0}s_{N,N} \rangle$ vs. r on a 40×40 lattice with periodic boundary conditions at $\beta = 0.42$ (note: data points for directions $(0, N)$ and $(N, 0)$ lay on top of each other).

In order to get proper values of $\langle s_i s_j \rangle$ for infinite systems one has to do finite size scaling. Extrapolation is generally straight forward as one sees in fig. 7 because the data points deviate for large r ($r \sim \frac{1}{2}L$) due to finite size effects.

In the following part the proposed algorithm for obtaining results for infinite systems is implemented and the obtained values are compared with the exact solution and the results on finite size systems.

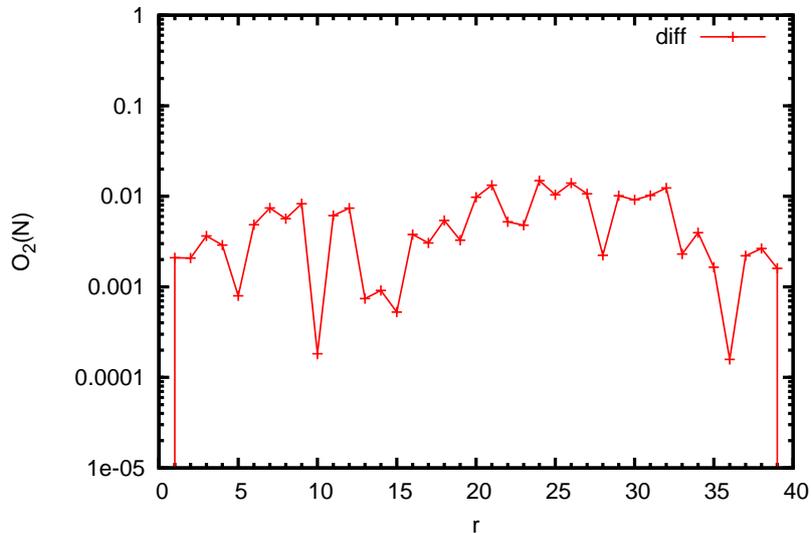


Figure 6: Difference $O_2(N)$ between measured $\langle s_{0,0}s_{0,N} \rangle$ and $\langle s_{0,0}s_{N,0} \rangle$ from simulation on a 40×40 lattice at $\beta = 0.42$.

3.2 Algorithm

1. Generate an initial configuration.
2. Construct a cluster using the proposed method and expand the system if necessary.
3. Measure observables:

$$C(\vec{r}) = \langle s_i s_{i+\vec{r}} \rangle = \left\langle \frac{1}{V_{\text{cl}}} \sum_{i \text{ in cluster}} \delta(i + \vec{r} \text{ in cluster}) \right\rangle. \quad (23)$$

4. Flip the cluster.
5. Go to step 2

3.3 System initialisation

The first point of consideration is the initial lattice configuration - choosing all spins equal results in the system growing infinitely with the first few clusters. Therefore, one needs a configuration that limits the growth rate. I have considered two different initializations:

- the chessboard configuration - it suppresses the system growth (in units of MC steps) at most because every spin needs to be flipped at least once before its neighbours can be considered for insertion in the cluster,
- the random configuration with equal distribution of spin states.

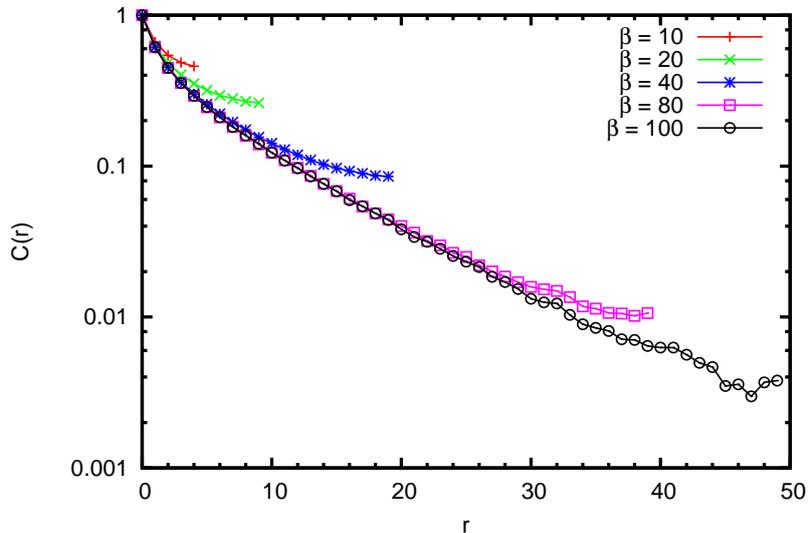


Figure 7: Measured correlation functions $C(\vec{r})$ for the axial direction on lattices of size $L = 10, 20, 40, 80, 100$ at $\beta = 0.42$. Values have been considered only for distances $r < \frac{1}{2}L$.

3.4 System growth and implementation

In the beginning I create a lattice A with dimensions $A_x \times A_y$ and initialise it. After some MC steps the cluster wants to reach a point outside of A , therefore the lattice needs to expand. I have implemented it as follows:

1. Generate a new lattice B with dimensions $B_x \times B_y$, where $B_x = A_x + 2dx$, $B_y = A_y + 2dy$ (fig. 8). Here dx and dy are the chosen growth steps, typically about 10.
2. Initialise the new lattice B .
3. Copy the spin configuration from A onto B .
4. Discard A and use B .

I use recursion to implement the Wolff cluster update algorithm, therefore I use a coordinate system that maps coordinates (x, y) to spins s_i and preserves the mapping upon lattice growth. The mapping $s_{x,y} \leftrightarrow s_i$ that I use is:

$$i = x - yx + A_c \quad (24)$$

$$x = i \bmod A_x - A_c \bmod A_x \quad (25)$$

$$y = A_c \operatorname{div} A_x - i \operatorname{div} A_x \quad (26)$$

with A_c being the spin s_{A_c} mapped to $(0, 0)$. The cluster is then represented as an array of coordinates that always point to the same spins using the mapping $A_c \mapsto B_c$:

$$B_c = A_c + 2dxdy + A_x dy + dx + 2dx(A_c \operatorname{div} A_x) \quad (27)$$

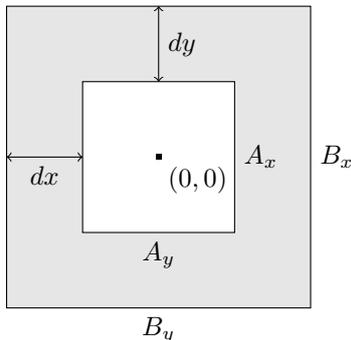


Figure 8: Lattice A expands by placing it onto a new larger lattice B such that the origins $(0,0)$ of both coordinate systems coincide.

3.5 Measurements of $C(\vec{r})$

The measurements are taken using the bond representation: after the cluster has been created but before it has been flipped. A spin is considered equilibrated if it has been flipped at least $n_{eq} = 20$ number of times. This means that a measurement is done only if all the required spins have been flipped at least n_{eq} number of times. For the correlation function $C(\vec{r})$ these are all the spins i in the cluster and all the spins $i + \vec{r}$ reached by the correlation vector.

Figure 9 shows that the results obtained from the proposed simulation method fit quite well the data obtained from simulations on finite lattices. The results for finite lattices have been obtained using the spin representation.

3.5.1 Dependence on system initialisation

Simulations have been done with the two different initialisation algorithms: chess board and random with equal state distribution. As expected, the random initialisation lets the system expand faster (in MC steps) than the chess board initialisation (see fig. 10). Although it shows the same exponential behaviour for large r , it is a bit better than the chess board for measuring the correlation function with the estimator from eq. (23). As the system expands faster one needs less computational time (MC steps) to get more measurements for some r than when using the chess board initialisation.

3.5.2 Dependence on the number of MC steps

The results of the simulations converge to the exact solution as the number of MC steps (cluster flips) increases. Moreover the accuracy of the measured values depends on the distance r from the origin s_{seed} as one sees on fig. 11. The resulting curve deviates more from the exact solution for larger r , therefore one needs to perform more MC steps $N_2 > N_1$ in order to get the same accuracy of $C(r_1)$ and $C(r_2)$ for $r_2 > r_1$. This is a result of the construction of the estimator for $C(\vec{r})$, since it converges from below due to the zero contributions to the expected value (23) from clusters with no point at distance r from the origin s_{seed} . Therefore the number of MC steps one needs to get good values

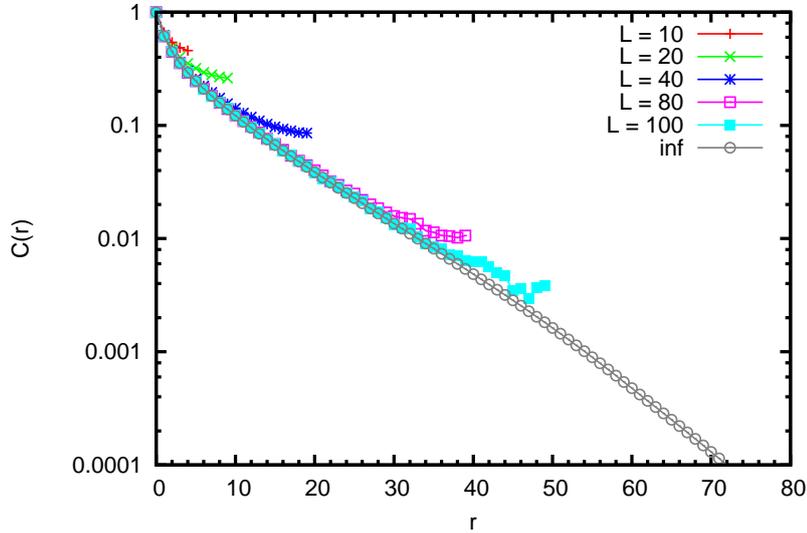


Figure 9: $C(r)$ vs. r obtained from simulations at $\beta = 0.42$ (about $3e7$ MC steps) on finite size lattices and the proposed method, \vec{r} is an axial vector.

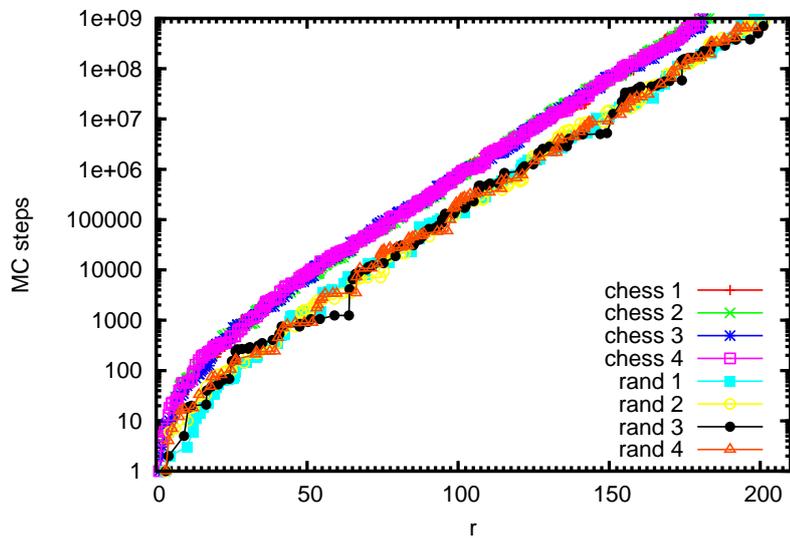


Figure 10: The number of MC steps needed for the cluster to reach a point at distance r from the starting spin s_{seed} for the first time vs. r at $\beta = 0.42$. The data points are from simulations with different random number sequences using the two different initialisation algorithms: the chess board and the random with equal distribution of states.

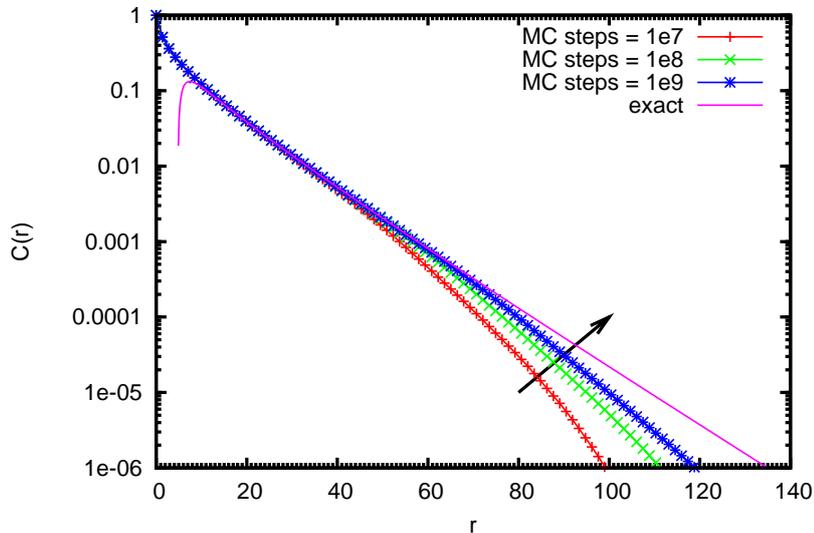


Figure 11: Comparison of the measured correlation function $C(r)$ with different number of cluster flips and the exact solution with \vec{r} being a diagonal vector at $\beta = 0.42$. The arrow indicates increasing number of MC steps.

for $C(\vec{r})$ depends directly on the cluster sizes, hence β .

Moreover the accuracy

$$O_3 = \frac{C(\vec{r}) - \langle s_{0,0}s_{N,N} \rangle}{\langle s_{0,0}s_{N,N} \rangle} \quad (28)$$

of the measured $C(r)$ does not depend linearly on the number of MC steps as one sees on fig. 12. One needs to do about 10 times more MC steps in order to reduce the deviation by a factor of 2.

3.5.3 Dependence on β

The results of the simulations deviate from the exact solution for larger values of r as one approaches β_c . This might be explained with the estimator for the correlation function as follows: near β_c the clusters get larger, therefore the number of clusters that contain spins at distance r increases. This results in more contributions to the estimator that are different from 0 and therefore increases the expected value. It means that results obtained from the simulations for a distance r get better as one approaches β_c , which can be seen on fig. 13.

3.6 Comparison with the results from [1]

Finally, a comparison has been done between the results from my simulations, the exact solution for $\langle s_{0,0}s_{0,N} \rangle$ and the plot from [1] in fig. 14. The simulations from [1] have obviously been done for r being an axial vector. The results from the paper show the same systematic deviation for large r as described above, namely that it lies a bit below the exact solution. A comparable accuracy with the results from [1] has been obtained with $1e9$ MC steps.

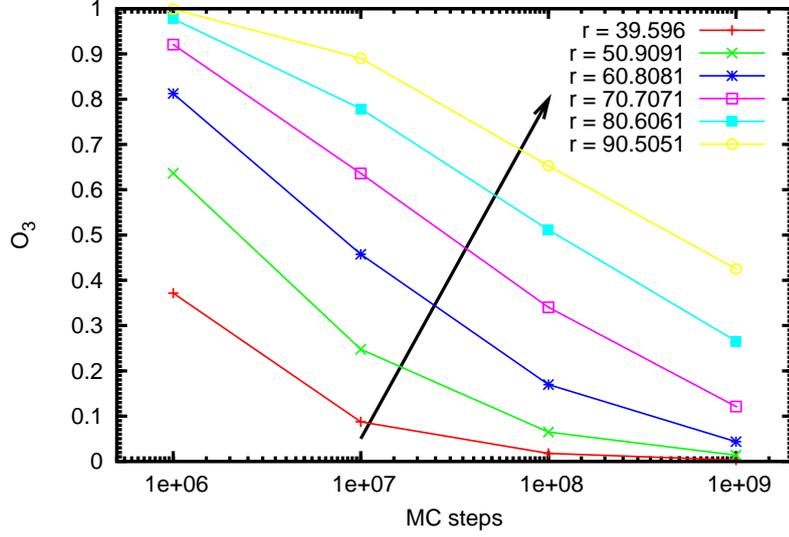


Figure 12: Comparison of the deviation (eq. 28) of the measured $C(\vec{r})$ from the exact solution $\langle s_{0,0}s_{N,N} \rangle$ in dependence of the MC steps with \vec{r} being a diagonal vector at $\beta = 0.42$. The arrow indicates increasing r . The data points have been obtained from fig. 11.

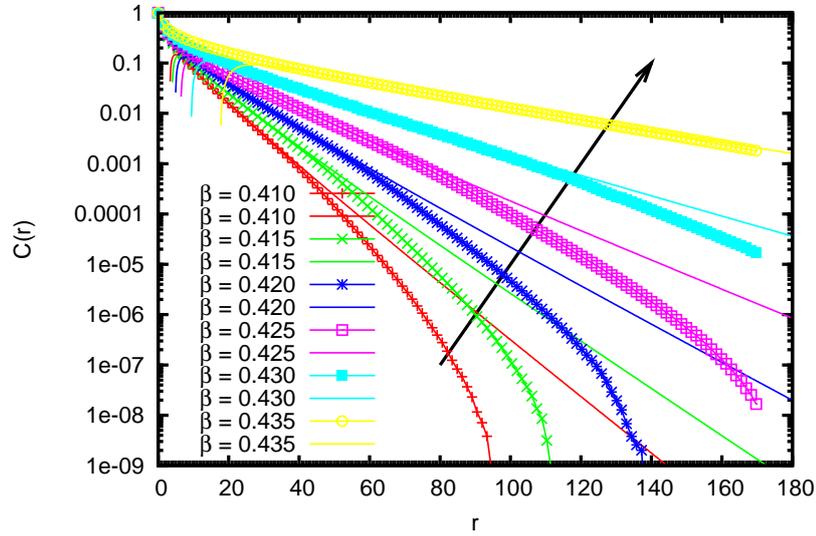


Figure 13: Comparison of $C(\vec{r})$ obtained using the simulation (lines with points) and the calculated values for $\langle s_{0,0}s_{N,N} \rangle$ (solid lines) for different values of β , \vec{r} being a diagonal vector. The simulations have been done with fixed number of $1e8$ MC steps. The arrow indicates increasing values of β .

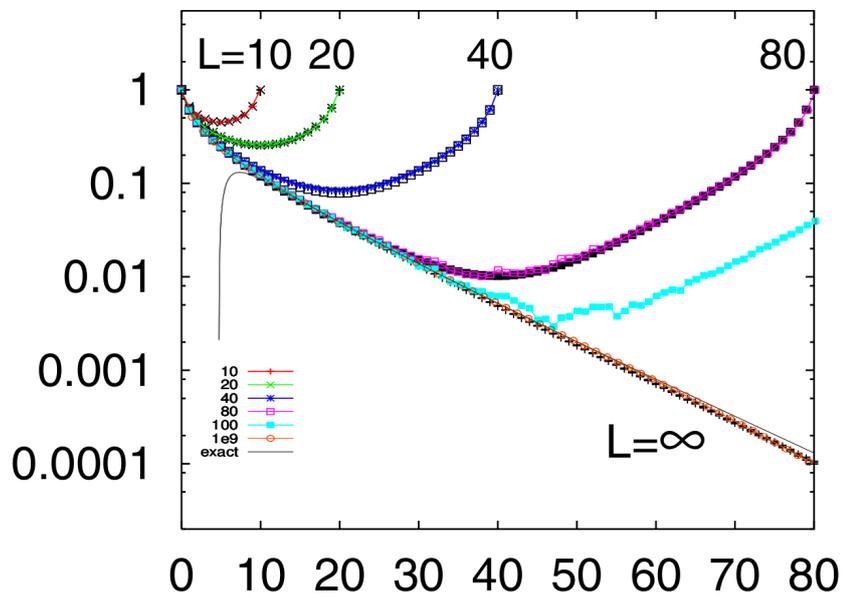


Figure 14: Correlation functions $C(r)$ vs r of the Ising model on a 2-dimensional square lattice with $L = 10, 20, 40, 80, 100$ at $\beta = 0.42$ and the results from the simulations on an infinite lattice from [1] (black), the results of my simulations on top (colour) and the exact solution for $\langle s_{0,0}s_{0,N} \rangle$ (grey), r being an axial vector and $N = r$.

4 Conclusion

In this work I have successfully managed to reproduce the results for the correlation function on the 2D Ising model without an external magnetic field and also successfully implemented the algorithm proposed in [1]. A comparison of the results from the simulations with the exact solutions for large N from [2] for the axial and diagonal directions for different lattice sizes and at different temperatures T showed that the estimator for the correlation function produces increasing systematic deviation with the proposed algorithm. The deviation increases as the distances r from the origin s_{seed} increases. This deviation depends also on the temperature, the system initialisation and the length (in MC steps) of the simulation. Moreover the authors of [1] have the same systematic deviation for large r . Nevertheless the proposed algorithm makes the estimator for the correlation function converge from below to the exact solution and can be used to obtain results from other models where one can make use of the bond representation and the Wolff cluster update algorithm. Other estimators might work better with this algorithm, but have not been discussed in this work.

References

- [1] H. G. Evertz and W. von der Linden. Simulations on infinite-size lattices. Phys. Rev. Lett., 86(22):5164–5167, May 2001.
- [2] Barry M. McCoy and Tai Tsun Wu. The Two-Dimensional Ising Model. Harvard University Press, 1973.
- [3] C. M. Fortuin and P. W. Kasteleyn. On the random-cluster model : I. introduction and relation to other models. Physica, 57(4):536–564, 1972.
- [4] Ulli Wolff. Collective monte carlo updating for spin systems. Phys. Rev. Lett., 62(4):361–364, Jan 1989.
- [5] Robert H. Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in monte carlo simulations. Phys. Rev. Lett., 58(2):86–88, Jan 1987.
- [6] Tai Tsun Wu, Barry M. McCoy, Craig A. Tracy, and Eytan Barouch. Spin-spin correlation functions for the two-dimensional ising model: Exact theory in the scaling region. Phys. Rev. B, 13(1):316–374, Jan 1976.