

ENS PARIS

LABORATOIRE DE PHYSIQUE STATISTIQUE

INTERNSHIP REPORT

Lifting algorithms in the XY model

Author:

Johannes MAYER

j.mayer@stud.uni-heidelberg.de

Supervisor:

Dr. Werner KRAUTH

werner.krauth@ens.fr

April 27, 2015

Contents

1	Introduction	2
2	The 2D XY Model	3
2.1	Magnetic susceptibility χ	4
3	Markov Chain Monte Carlo Simulations	5
3.1	Monte Carlo Simulations	5
3.2	Markov Chain and Metropolis Filter	5
3.3	The factorized Metropolis filter	7
3.4	Autocorrelation functions	9
3.5	The Wolff Cluster Algorithm	9
4	Lifting in Markov chains	11
4.1	Particle in 1 D box	11
4.2	Two discrete XY spins	12
4.3	N -particle systems	15
4.4	Rejection-free algorithms	18
5	Simulation of the XY model	22
5.1	Measurements of χ	22
5.2	Measurement of τ_χ	23
5.3	Concluding Remarks	28
	Bibliography	i

1 Introduction

The following report is a summary of the work done during the internship in the group of Werner Krauth at the ENS Paris. The so-called factorized Metropolis filter has recently shown to be of great use in Monte Carlo simulation of the interaction between soft disks in two dimensions. [1] [2]

The goal of the internship was to familiarize myself with new techniques in Monte Carlo simulations used in the algorithm developed by the group of Werner Krauth and to test the applicability of the algorithm to a spin system.

The system to which the algorithm was applied is the two dimensional XY model. The system displays a phase transition, a point at which local Markov Chain algorithms often have difficulties in producing independent samples. To test the performance of the algorithm the speed with which it produces independent samples was measured and compared to two other algorithms used in Monte Carlo Simulations.

2 The 2D XY Model

The 2D XY model describes an ensemble of two-component spins of unit length that are placed on the grid of a two-dimensional lattice whose interaction is invariant under a global rotation. Each spin j is defined by an angle ϕ_j , $\vec{S}_j = (\cos(\phi_j), \sin(\phi_j))$. The spins interaction is given by the following Hamiltonian:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \vec{S}_i \vec{S}_j = -J \sum_{\langle i,j \rangle} \cos(\phi_i - \phi_j) \quad (2.1)$$

In this report only the case of $J > 0$ is considered, called the ferromagnetic XY model. The ferromagnetic XY model does not possess any spontaneous ordering at any given temperature. This is because the continuous spins can align in waves such that neighboring spins don't differ much at nearly no energy cost. However the nonexistence of spontaneous ordering does not imply the absence of a phase transition. The model undergoes a so-called Kosterlitz-Thouless transition at a temperature T_{KT} . [3] The transition manifests in the unbinding of pairs of so called vortices of spins. Spin vortices are configurations in the spin system where spins along a closed path \mathcal{C} make a multiple of 2π rotation, i.e.

$$\int_{\mathcal{C}} \vec{\nabla} \phi(\vec{x}) d\vec{s} = n 2\pi \quad (2.2)$$

for $n \in \mathbb{Z}$.

Below the transition temperature vortices with positive and negative n are bound in pairs. This way there is local order in the system. Above the transition temperature these pairs unbind, free vortices can exist and local order is destroyed.

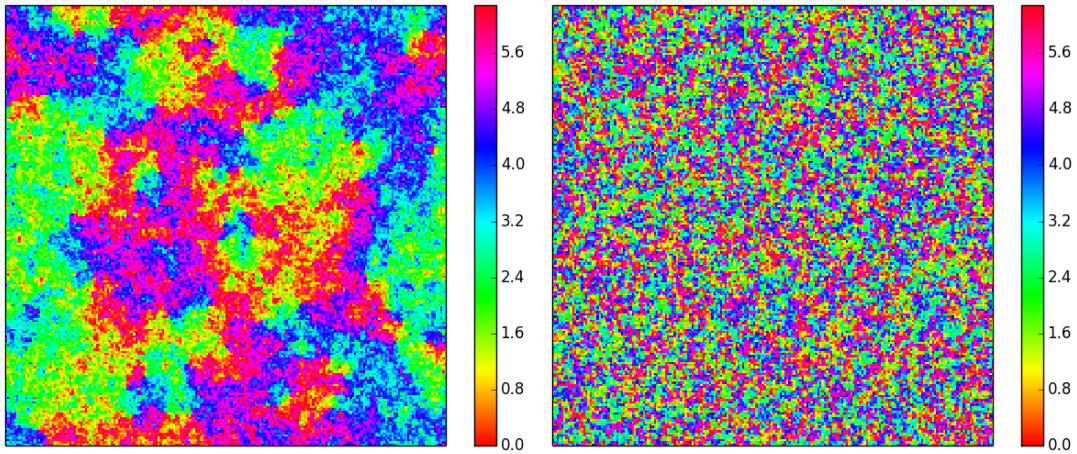


Figure 2.1: Two equilibrium configurations of a 200x200 square lattice. The spins directions are displayed in color. Left: configuration below the transition temperature. Right: above critical temperature. One can clearly see that local order is destroyed above T_{KT} . The configurations were created with the algorithm described below.

2.1 Magnetic susceptibility χ

The observable of the XY model examined in this report is the magnetic susceptibility χ which is defined via:

$$\chi = \frac{|\vec{M}^2|}{N} \quad (2.3)$$

where N is the number of spins and $\vec{M} = \sum_i \vec{S}_i$ the sum of all spins. Above the KT temperature there is no order in the system which means that the expectation value of $|\vec{M}|^2$ is close to zero. The distribution of χ due to fluctuations around the mean value is a very narrow peak around zero. Below the critical temperature however the system is locally ordered and the magnetic susceptibility not only gains a finite value but also the distribution becomes very broad. For the simulation of this system with local Monte Carlo methods this poses the problem of critical slowing down, as will be explained below. So in order to test the speed of a local algorithm on the XY system this observable depicts a good measurement device.

3 Markov Chain Monte Carlo Simulations

3.1 Monte Carlo Simulations

The present chapter is based on Ref. [4]. The Monte Carlo simulation method is introduced and current algorithms used to simulate the XY model are presented. The Monte Carlo method is a statistical method to calculate integrals by evaluating random positions (in configuration space) that follow a chosen distribution. This method is very powerful in statistical physics since in order to calculate partition functions and expectation values one often has to evaluate high dimensional integrals of the type:

$$\mathcal{Z} = \sum_{\{\sigma\}} e^{-\beta E(\sigma)} \quad (3.1)$$

where $\beta = \frac{1}{k_B T}$ the Boltzmann factor and the sum running over all possible configurations σ with energy $E(\sigma)$. The goal of a Monte Carlo simulation is to sample a probability distribution, in the following called π . In statistical physics this means sampling many different states of a statistical system, distributed according to the Boltzmann distribution imposed by the interaction between the elements of the system.

3.2 Markov Chain and Metropolis Filter

Since it is often not possible to generate these states directly (by drawing random numbers), an important tool in Monte Carlo is the so called Markov Chain Monte Carlo (MCMC). Its idea is to start from a valid configuration and evolve the system by changing it only little at each step in time. In the limit of infinitely many little changes, the

ensemble of the visited configurations together forms the desired distribution again. The recipe describing how to evolve the system such that it samples the desired distribution is an algorithm. An MCMC algorithm is the set of probabilities that tells its user how to move in configuration space depending on the state the system is in.

$$\{p(\alpha \rightarrow \beta)\} = \{\text{probability to move from state } \alpha \text{ to } \beta\} \quad (3.2)$$

Finding an algorithm now means finding a set of probabilities which will result in the desired distribution.

With the use of Markov Chain methods, problems arise: Since the application of the algorithm evolves the state of the system from one to another, the produced samples are not independent. One only accesses the underlying probability distribution in the limit of long running times. Furthermore, the state one starts the simulation with might not be from of the equilibrium distribution of the system, so one has to wait for a certain amount of steps until the algorithm has converged to equilibrium until one can start sampling.

Of course the probabilities have to obey certain conditions in order for the Markov Chain to converge to the stationary distribution. A necessary condition is the so called global balance condition. Once the system has reached its stationary distribution the algorithm should not leave it again. This is manifested in the stationarity condition:

$$\frac{d}{dt}\pi_\alpha = \sum_\beta \{\pi_\beta p(\beta \rightarrow \alpha) - \pi_\alpha p(\alpha \rightarrow \beta)\} = 0 \quad (3.3)$$

where π_α is the stationary probability to be in state α . The flow of probability out of one configuration α needs to level the flow into the configuration. The other two conditions are ergodicity (each configuration of the system can be reached from each other one in a finite number of steps) and aperiodicity (there is no state which the algorithm maps to itself after a finite number of steps bigger than one.)

For a physical system, the stationary distribution $\pi(\alpha)$ is given by the states' Boltzmann weights:

$$\pi(\alpha) = \frac{1}{\mathcal{Z}} e^{-\beta E(\alpha)} \quad (3.4)$$

with $\beta = \frac{1}{k_B T}$.

A very simple algorithm to sample this distribution was proposed by Metropolis et al. [5]: the acceptance probability is given by

$$p^{\text{Met}}(\alpha \rightarrow \beta) = \min\left(1, \frac{\pi(\beta)}{\pi(\alpha)}\right) \quad (3.5)$$

This algorithm applied to the XY model takes the following form:

1. Start from a valid state of the system α
2. sample a proposed move $\delta\varphi \in [-\xi, \xi]$
3. Pick a random spin ϕ_i of the system and apply the move $\phi_i \rightarrow \phi_i + \delta\varphi$, thereby sending the system from state α to state γ
4. accept the move with the probability $p^{\text{acc}}(\alpha \rightarrow \gamma) = \min(1, e^{-\beta(E_\gamma - E_\alpha)})$
5. if the move is accepted then add γ to the ensemble of samples and repeat the algorithm with state γ as initial configuration, in case of refusal add α and repeat with

By simply plugging in the Metropolis acceptance probabilities together with the Boltzmann weights in (3.3) one can see that this algorithm not only satisfies global balance, but that every term in the sum vanishes by itself. This is called "detailed balance".

Setting up a working MCMC algorithm by using the Metropolis filter will therefore for sure lead to the correct equilibrium as long as the two other conditions for convergence are also fulfilled. However it has a downside to it: The goal of an MCMC simulation is to produce as many independent samples as possible. Since the detailed balance keeps all probability flows between all states in equilibrium, the probability of changing from $\alpha \rightarrow \beta$ is just the same as in the next move to go from $\beta \rightarrow \alpha$. So the detailed balance causes no quick movement in phase-space, it is rather a random walker, the algorithm behaves "diffusive".

3.3 The factorized Metropolis filter

The Metropolis filter is not the only set of probabilities that fulfill the global balance condition. The standard Metropolis filter can be changed in a non-intuitive way. In the

case of a system of N interacting particles the acceptance probability is given by:

$$p^{\text{Met}}(\alpha \rightarrow \gamma) = \min(1, e^{-\beta \sum_{i,j} \Delta E_{ij}}) = \min(1, \prod_{i,j} e^{-\beta \Delta E_{ij}}) \quad (3.6)$$

where the sum/product runs over all pairs and the $\Delta E_{ij} = E_{ij}(\gamma) - E_{ij}(\alpha)$ are the energy changes between each pair interaction. The so called factorized Metropolis filter now uses the acceptance probability

$$p^{\text{fac}}(\alpha \rightarrow \gamma) = \prod_{i,j} \min(1, e^{-\beta \Delta E_{ij}}) = e^{-\beta \sum_{i,j} \max(0, \Delta E_{ij})} \quad (3.7)$$

These probabilities are not the same as before in Eq. (3.6). For $N = 2$ the acceptance probabilities agree trivially but for $N > 2$ in general $p^{\text{fac}} \leq p^{\text{Met}}$. Nonetheless the factorized filter also respects detailed balance. This can be seen by applying

$$\frac{\pi(\alpha)}{\pi(\gamma)} = e^{\beta \sum_{i,j} \Delta E_{ij}} = e^{\beta \sum_{i,j} (\max(0, \Delta E_{ij}) - \max(0, -\Delta E_{ij}))} \quad (3.8)$$

to equation (3.7) and finding :

$$\frac{\pi(\alpha)}{\pi(\gamma)} p^{\text{fac}}(\alpha \rightarrow \gamma) = p^{\text{fac}}(\gamma \rightarrow \alpha) \quad (3.9)$$

This means that if used in an algorithm, these acceptance probabilities will also sample the correct distribution even though the rejection probability is higher. But simply replacing $p^{\text{Met}} \rightarrow p^{\text{fac}}$ in the algorithm scheme described above does not give an advantage, it rather slows down the algorithm since rejection means no movement in phase space at all.

However the big advantage is, that this filter takes each pair interaction into account individually instead of considering their sum. Connected to another technique called lifting in Markov chain, this filter allows to construct rejection free algorithms discussed in the next chapter.

3.4 Autocorrelation functions

An important tool in order to judge the independence of the data is the so called autocorrelation function. Monte Carlo simulations produce an array of samples on which observables can be evaluated. It is defined for an observable of a dataset via:

$$C_{\mathcal{O}}(dt) = \frac{\langle \mathcal{O}(0)\mathcal{O}(dt) \rangle - \langle \mathcal{O} \rangle^2}{\langle \mathcal{O}^2 \rangle - \langle \mathcal{O} \rangle^2} \quad (3.10)$$

This function is normalized to 1 for $dt = 0$ and smaller than one for $dt \geq 1$. If all the samples in an array are independent, then by definition the term

$$\langle \mathcal{O}(0)\mathcal{O}(dt) \rangle = \langle \mathcal{O}(0) \rangle \langle \mathcal{O}(dt) \rangle = \langle \mathcal{O} \rangle^2 \quad (3.11)$$

factorizes and the autocorrelator drops to zero for $dt \geq 1$. If the data is not independent then one expects to see an exponential decay. The rapidity of this decay is what defines the time after which two samples can be regarded as independent from each other.

Of course there are several observables for each model. The slowest one is defining the speed of the algorithm.

3.5 The Wolff Cluster Algorithm

The currently fastest algorithm to simulate spin systems with nearest neighbor interaction is called the Wolff Cluster algorithm [6]. As mentioned before the local MCMC methods take very long to produce independent samples in the temperature regime of the phase transition. This is because the distribution of the observable χ becomes very broad. The local MCMC however only moves one single spin at a time and the change in χ is very little. As it fulfills detailed balance the inverse move has the same probability so it takes very long for the algorithm to explore the whole distribution of χ and so the produced samples are all correlated on a long timescale. This is known as critical slowing down.

The Wolff algorithm tackles this problem: instead of moving one spin at a time it constructs clusters of spins that are all moved collectively. This way there can be much

larger changes in χ with one cluster movement and the algorithm produces independent samples much faster than the local MCMC algorithm. The algorithm takes a random vector \vec{r} from the unit circle, constructs a cluster of spins starting from a random point on the grid. After the construction is finished it flips the component of the cluster spins which is parallel to \vec{r} across the plane rectangular to \vec{r} .

A cluster C is constructed and flipped in the following way (set $J = 1$):

1. Choose a random vector from the unit sphere \vec{r} . Pick a random site i on the lattice as a starting point for the cluster
2. Flip the spin $\vec{S}_i \rightarrow \vec{S}_i - 2(\vec{S}_i \cdot \vec{r})\vec{r}$, mark spin i as flipped and add site i to C
3. Visit all the sites j connecting site i and add activate the bond $\langle i, j \rangle$ with probability $p(\vec{S}_i, \vec{S}_j) = 1 - \exp[\min\{0, 2\beta(\vec{S}_i \cdot \vec{r})(\vec{S}_j \cdot \vec{r})\}]$. If it is accepted flip $\vec{S}_j \rightarrow \vec{S}_j - 2(\vec{S}_j \cdot \vec{r})\vec{r}$, mark spin j and add site j to C .
4. Continue this procedure with each bond of newly added spins until no more unmarked spin can be flipped and the process stops.

This algorithm respects detailed balance. Yet it makes big steps in the distribution of χ and is not affected by critical slowdown. One sample of this algorithm is taken after every cluster flip.

4 Lifting in Markov chains

The the following the concept of lifting in a Markov chain is introduced with the help of the example of a single particle in a 1-dimensional box with hard walls followed by the application to two interacting XY spins. The source of the algorithms presented in the following chapter is Ref. [1].

4.1 Particle in 1 D box

Consider a particle in a one dimensional box with discretized positions $\alpha = \{\Delta, 2\Delta, \dots, L\Delta\}$ and hard walls. In order to sample a uniform distribution on this set of states the standard Metropolis algorithm would consist in sampling moves uniformly from the set $\{-\Delta, \Delta\}$. A proposed move that keeps the particle in the box is accepted with probability 1, a move that would take the particle out of the box is rejected. It would fulfill the detailed balance $\varphi_{\alpha \rightarrow \alpha + \Delta} = \varphi_{\alpha + \Delta \rightarrow \alpha}$ for all α and converge to the stationary distribution.

Lifting in this example consists of duplicating every configuration by introducing a lifting variable p : The states then obtain a second degree of freedom $(\alpha) \rightarrow (\alpha, p = \pm)$. The weights of the new states in the stationary distribution are unaffected by this, only the normalization changes: $\pi((\alpha, \pm)) = \pi(\alpha)$. The lifting variable now fixes the direction of the next move, something which in regular MCMC would be sampled from an a priori distribution. The only proposed moves then are:

$$\begin{aligned}(i\Delta, +) &\rightarrow ([i + 1]\Delta, +) \\(i\Delta, -) &\rightarrow ([i - 1]\Delta, -)\end{aligned}$$

Clearly this algorithm does not respect detailed balance any more because

$$p[(\Delta, +) \rightarrow (2\Delta, +)] \neq 0 \quad (4.1)$$

$$p[(2\Delta, +) \rightarrow (\Delta, +)] = 0 \quad (4.2)$$

Now in case of a rejection in the physical variable α instead of having no flow out of the configuration, one changes the lifting variable and thereby introduces a flow between the two duplicates:

$$(i\Delta, \pm) \rightarrow (i\Delta, \mp) \quad (4.3)$$

The next proposed move will therefore go into the other direction. In this way it is possible to explore all L sites in $\mathcal{O}(L)$ operations instead of the usual $\mathcal{O}(L^2)$ random walk behavior and therefore increase the speed of the algorithm by breaking the detailed balance of the transition probabilities.

4.2 Two discrete XY spins

The same thing can be done for two XY spins. Consider two XY spins 0 and 1 which can take only L discrete values, $\phi \in \{0, \frac{2\pi}{L} \dots \frac{2\pi(L-1)}{L}\}$. The two spins interact via a potential that only depends on the difference in angle between the two. Each state is duplicated into two:

$$(\alpha) \rightarrow (\alpha; l) \quad (4.4)$$

where $l \in \{0, 1\}$ is the lifting variable which defines which spin is proposed to move.

The lifting algorithm takes the following scheme:

1. Split each state of the XY system into two by introducing a lifting variable which identifies one of the two spins as the moving one. .
2. Randomly initialize a valid state. This means one of the two spins is lifted.
3. Propose to move the lifted spin ϕ_{lift} by an amount $\delta = \frac{2\pi}{L}$ in a fixed direction (for example counterclockwise): $\phi_{\text{lift}} \rightarrow \phi_{\text{lift}} + \delta$

4. Accept the move with the Metropolis algorithm probability $p^{\text{Met}} = \min(1, \exp[-\beta\Delta E])$ with ΔE the energy difference $\Delta E = E_{\text{after}} - E_{\text{before}}$ before and after the move
5. If the move is refused don't move the spin but instead change the lifting variable to the other spin.
6. Back to step 3

The event of a rejected move is in the following also referred to as a collision, as the spin fails to cross the potential barrier created by the other spin and "collides" with it.

Of course in order to converge to the stationary distribution one needs to fulfill the global balance condition.

$$\sum_{\beta} \varphi(\beta \rightarrow \alpha) = \sum_{\gamma} \varphi(\alpha \rightarrow \gamma) \quad (4.5)$$

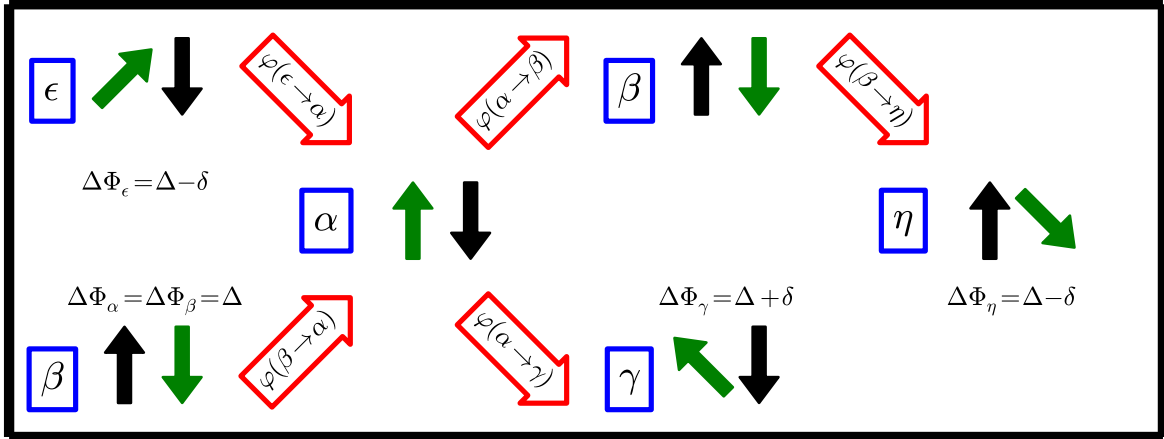


Figure 4.1: The flows in the between different states of the system. The states are labeled by greek letters, the red arrows represent the probability flows between them. The green spin has the lift and the algorithm proposes to move it.

In Figure 4.1 the scheme of such a system is shown. Note that the spins only turn in one direction, ergo the represented scheme are all possible in and outgoing flows for the state α . The stationary flows are given by:

$$\varphi(\alpha \rightarrow \gamma) = \pi(\alpha) p^{\text{acc}}(\alpha \rightarrow \gamma) \quad (4.6)$$

where $\pi(\alpha)$ is the stationary probability to be in the state α and p^{acc} is the Metropolis acceptance probability. The stationary probability is given by the state's Boltzmann weight. The states Boltzmann weights are given by:

$$\pi(\alpha) = e^{-\beta E_\alpha} = e^{-\beta E_\beta} = \pi(\beta) \quad (4.7)$$

$$\pi(\gamma) = e^{-\beta E_\gamma} \quad (4.8)$$

$$\pi(\epsilon) = e^{-\beta E_\epsilon} \quad (4.9)$$

$$\pi(\eta) = e^{-\beta E_\eta} \quad (4.10)$$

The flow equations read:

$$\varphi_{out} = \pi(\alpha)(p^{\text{acc}}(\alpha \rightarrow \gamma) + p^{\text{acc}}(\alpha \rightarrow \beta)) \quad (4.11)$$

$$= \pi(\alpha)[\min(1, e^{-\beta(E_\gamma - E_\alpha)}) + 1 - \min(1, e^{-\beta(E_\gamma - E_\alpha)})] \quad (4.12)$$

$$= \pi(\alpha) \quad (4.13)$$

$$\varphi_{in} = \pi(\epsilon)p^{\text{acc}}(\epsilon \rightarrow \alpha) + \pi(\beta)p^{\text{acc}}(\beta \rightarrow \alpha) \quad (4.14)$$

$$= \pi(\epsilon) \min(1, e^{-\beta(E_\alpha - E_\epsilon)}) + \pi(\beta)(1 - \min(1, e^{-\beta(E_\eta - E_\beta)})) \quad (4.15)$$

Now for the specific choice of a potential which only depends on $\Delta\varphi = \varphi_1 - \varphi_2$ one has that $E_\eta = E_\epsilon$. Recalling that $\pi(\alpha) = \pi(\beta)$ one finds that:

$$\varphi_{in} - \varphi_{out} = \pi(\epsilon) \min(1, e^{-\beta(E_\alpha - E_\epsilon)}) - \pi(\alpha) \min(1, e^{-\beta(E_\epsilon - E_\alpha)}) = 0 \quad (4.16)$$

This means that the proposed algorithm respects global balance for each value of the discretization δ . As soon as one introduces an interaction which does not depend on the distance between the two spins the assumption that the energy of state ϵ and η are the same does not hold any more and global balance is violated. Since the discretization δ is arbitrary it also holds for two continuous spins.

4.3 N -particle systems

The above application of the lifting algorithm to two XY spins (in the following called spins) is generalized straight forward to N particle systems. The lifting variable is still which spin is turning at the moment. Each configuration is characterized by the angles of each XY spin to a fixed axis and a move turns only the lifted spin:

$$\alpha_i = \{\phi_1, \dots, \phi_i, \dots, \phi_N\} \rightarrow \beta_i = \{\phi_1, \dots, \phi_i + \Delta, \dots, \phi_N\} \quad (4.17)$$

However this algorithm does not respect global balance any more. This is because of there might be configurations where it is unclear to which particle should be lifted. This is shown in Figure 4.2

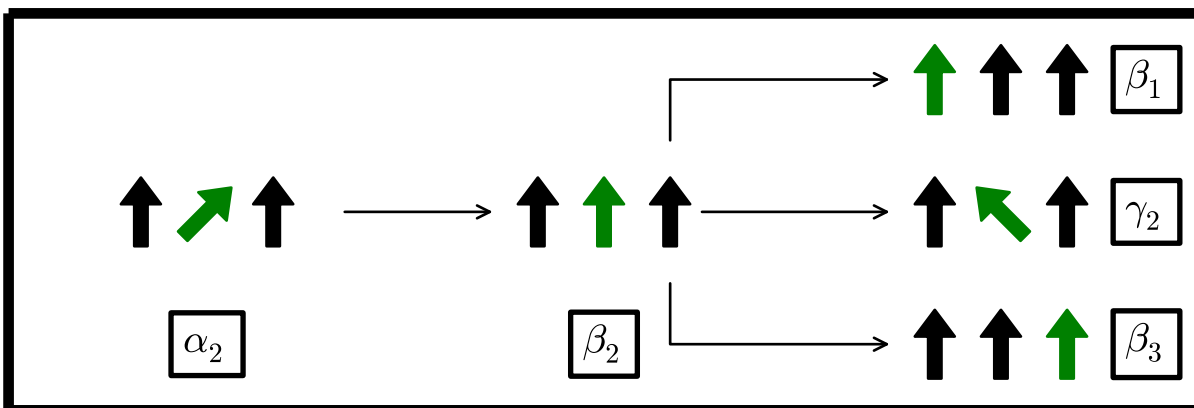


Figure 4.2: The states are labeled by the configuration of the spins and by the lifting variable. If $p^{\text{acc}}(\beta_2 \rightarrow \gamma_2) \neq 1$ in case of a rejection of a move $\beta_2 \rightarrow \gamma_2$ it is not clear to which state the system should be evolved. It is impossible to define flows that respect global balance in this situation.

However this problem does not occur any more in the limit $\Delta \rightarrow 0$ is made. In the continuous case there are no spins that are at the exact same position so global balance is recovered. With the help of the factorized Metropolis filter it is possible to identify a unique lifting partner.

For general two body interactions $E_{ij}(\phi_i - \phi_j)$ the energy change between the two

configurations $\alpha_i \rightarrow \beta_i$ by an infinitesimal turning of spin ϕ_i by $d\phi_i$ is:

$$dE = \sum_{j \neq i} \frac{\partial E_{ij}(\phi_j - \phi_i)}{\partial \phi_i} d\phi_i = \sum_{j \neq i} dE_{ij} \quad (4.18)$$

By application of the factorized filter the move is rejected with probability

$$p^{\text{rej}} = 1 - p^{\text{fac}}(\alpha_i \rightarrow \beta_i) = 1 - \exp(-\beta \sum_{j \neq i} \max[0, dE_{ij}]) = \beta \sum_{j \neq i} \max[0, dE_{ij}] \quad (4.19)$$

where in the last step nonlinear terms of the exponential vanish as the changes are infinitesimal. This means pair interactions that lead to an energetically more favorable state are not contributing to the rejection probability.

The lifting probabilities are then exactly chosen as the pairwise rejection probabilities:

$$p^{\text{lift}}(\alpha_i \rightarrow \alpha_j) = \beta \max[0, dE_{ij}] \quad j \neq i \quad (4.20)$$

The algorithm now displaces the lifted spin by a sequence of infinitesimal moves which lead to a finite turning of the spin until a lifting partner is identified which inherits the lift and is the continued to be turned.

The lifting probabilities define an infinitesimal lifting algorithm. That this algorithm's flows obey global balance is demonstrated along the example displayed in figure 4.3. The flows into the configuration γ_2 are given by two lifting flows and one infinitesimal spin movement and amount to (note that $\pi(\alpha) = \pi(\gamma)$, see caption 4.3):

$$\varphi^{\text{lift}}(\gamma_1 \rightarrow \gamma_2) = \pi(\gamma) \beta \max[0, dE_{12}] \quad (4.21)$$

$$\varphi^{\text{lift}}(\gamma_3 \rightarrow \gamma_2) = \pi(\gamma) \beta \max[0, dE_{32}] \quad (4.22)$$

$$\varphi^{\text{move}}(\alpha_2 \rightarrow \gamma_2) = \pi(\alpha) p^{\text{fac}}(\alpha_2 \rightarrow \gamma_2) \quad (4.23)$$

$$= \pi(\alpha) \exp(-\beta \{ \max[0, dE_{12}] + \max[0, dE_{32}] \}) \quad (4.24)$$

$$= \pi(\gamma) (1 - \beta \max[0, dE_{12}] - \beta \max[0, dE_{32}]) \quad (4.25)$$

Their sum amounts exactly to $\pi(\gamma)$ which is the flow out of configuration γ_2 .

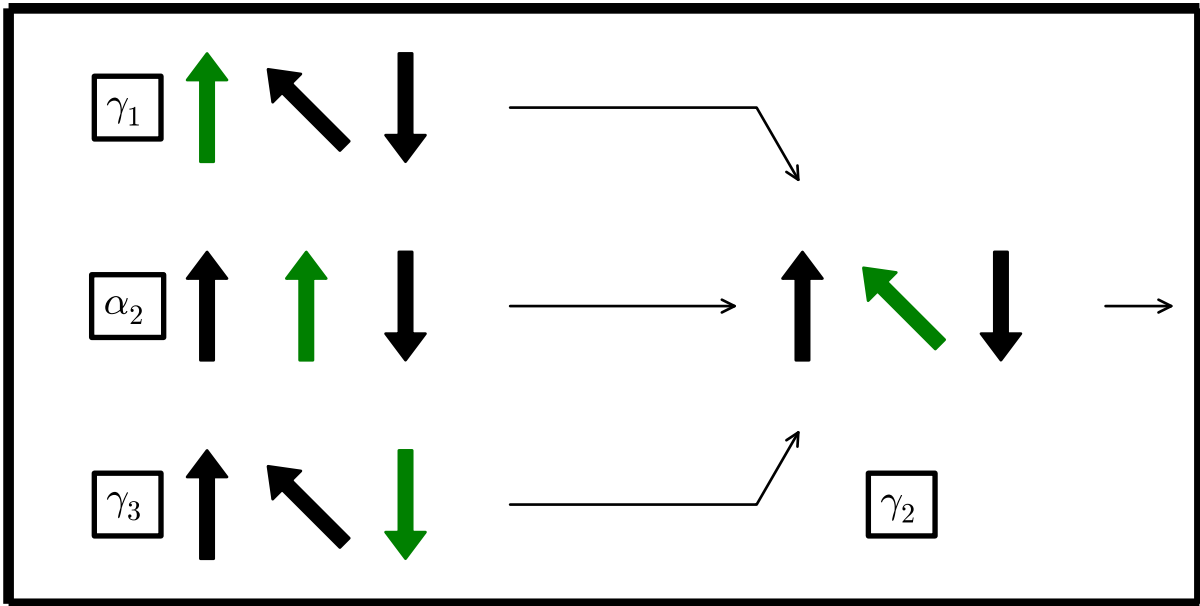


Figure 4.3: Infinitesimal flow between configurations. The central spin in state γ_2 is only infinitesimally different from the central one in α_2 . This is however represented by a finite difference in order to see it in the graphic. Since the spin is only infinitesimally different so is the energy of the configurations. This means their Boltzmann weight agrees.

4.4 Rejection-free algorithms

To implement the infinitesimal algorithm, one does not use an extremely small but finite discretization but constructs a rejection free algorithm. The methods described before in Markov Chain Monte Carlo all relied on the principle: propose a move and accepted or rejected it with a given probability defined by the algorithm.

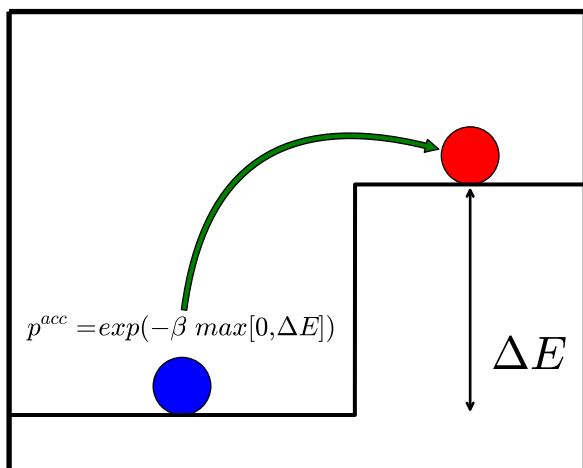


Figure 4.4: A proposed particle move over a potential barrier is accepted with the Metropolis probability

There is an alternative approach [7]. That is to sample the number of moves one can make until a rejection occurs. In the infinitesimal XY case this means by which finite amount the algorithm can turn a spin until a collision occurs, i.e. the lifting variable has to be changed. In this way no rejection ever occurs but rather a continuum of states is created.

To find the collision points consider a spin as a particle in a one dimensional potential created by another fixed spin. For simplicity assume that in order to turn the spin it would have to cross a barrier of height ΔE as shown in figure 4.4.

So by turning the spin from the position Φ_{old} to the position Φ_{new} it would have to cross several such barriers and the probability that no rejection (collision) occurs is given by:

$$p_{\text{no coll}}(\Phi_{old}, \Phi_{new}) = \prod_k \exp(-\beta \max[0, \Delta E_k]) = \exp(-\beta \max[0, \sum_k \Delta E_k]) \quad (4.26)$$

where the sum/product runs over all crossed barriers. Taking the continuum limit and using $\Delta E \rightarrow \frac{\partial E(\Phi)}{\partial \Phi} d\Phi$, the probability that there is no collision after turning a spin by

a finite amount to the new position Φ is

$$p_{\text{no coll}}(\Phi) = \exp\left(-\beta \int_{\Phi_{\text{old}}}^{\Phi} \max\left[0, \frac{\partial E(\phi)}{\partial \phi}\right] d\phi\right) \quad (4.27)$$

So probability that a collision occurs only is collected whilst moving up in the potential landscape, whilst constant or downwards movement does not lead to an increase of collision-probability.

To sample this distribution, i.e. the collision points, one draws a random number from a uniform distribution $\Upsilon \in (0, 1)$ and by this determines the amount of energy the particle can move up in a potential via $E^* = -\frac{1}{\beta} \log \Upsilon$ and then solves the equation

$$E^* = \int_{\Phi_{\text{old}}}^{\Phi} \max\left[0, \frac{\partial E(\phi)}{\partial \phi}\right] d\phi \quad (4.28)$$

for Φ , the upper integral bound.

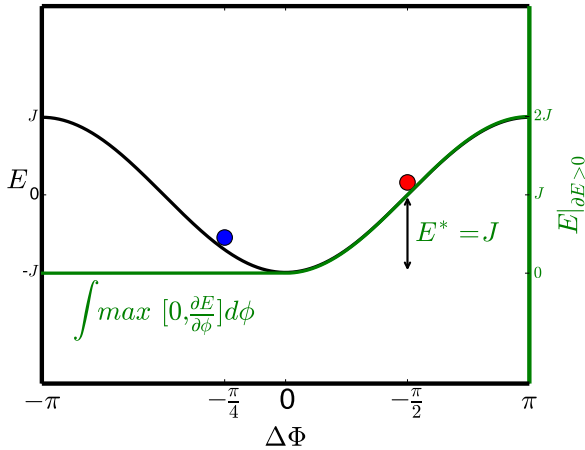


Figure 4.5: Example for solving (4.28) displayed graphically. The initial position is marked by the blue particle and the final position in red. The green curve displays only the part of the potential with positive derivative. The particle only feels this part in order to determine a collision.

The movement between two collisions is called an event. The chain of all subsequent events is called the event chain. At the end of each chain, the lifting variable is resampled.

This usually means the final event must be truncated, even if there has not been the next collision yet. A parameter in the algorithm is the length of this chain ℓ . A small chain length ℓ means that the lifting variable is changed often. For a ferromagnetic XY model with only next neighbor interaction, which will be used to test the algorithm on, this means practically that the algorithm turns spins in different regions of the grid. If ℓ is very big, this

means that the algorithm travels slowly through the grid since the lifting variable can only be given to one of the neighboring sites of the turning spin.

Very important is that the collision points do not sample the equilibrium distribution of the system but only the points where the lifting variable is changed. Since the system goes through a continuum of states one needs to take equidistant samples during the simulation. For example, the end points of each event chain samples the equilibrium distribution since one sample is taken after ℓ movement of the system.

An example for two ferromagnetic XY spins is displayed in figure 4.5.

The potential is given by $E = -J \cos(\Delta\Phi)$. After the lift has been passed for the last time, the spins had $\Delta\Phi = -\frac{\pi}{2}$. The randomly drawn energy was $E^* = J$. Note that the particle does not “gain“ energy by going downwards, it only feels the part of the potential with positive derivative, displayed by the green curve in the figure. This means the turned amount is $\frac{\pi}{4}$ (now it is at the minimum) plus the solution of the equation:

$$E^* = J = -J \cos(\phi) + J \tag{4.29}$$

which is $\phi = \frac{\pi}{2}$. This means the spin with the lift is turned by $\frac{3}{4}\pi$.

The factorized Metropolis filter is able to identify a unique partner because if the turning spin i interacts with the other spins $\{j\}$ via the pair potential E_{ij} the probability that no collision occurs is the product of the probabilities from equation (4.27) for each pair potential E_{ij} and they are all independent. This means in order to obtain the collision points for many interacting spins, one needs draw an admissible energy E_{ij}^* for each pair, solves equation (4.28) for each of them for the upper integral bound. One then determines the minimum of all the solutions, turns the spin i by this amount and the spin whose interaction produced the minimum solution inherits the lift. This allows to generalize a lifting algorithm which works for two particles to an N particle system.

For two particles in the ferromagnetic XY model the distribution of $\Delta\Phi$ can be plotted and it follows:

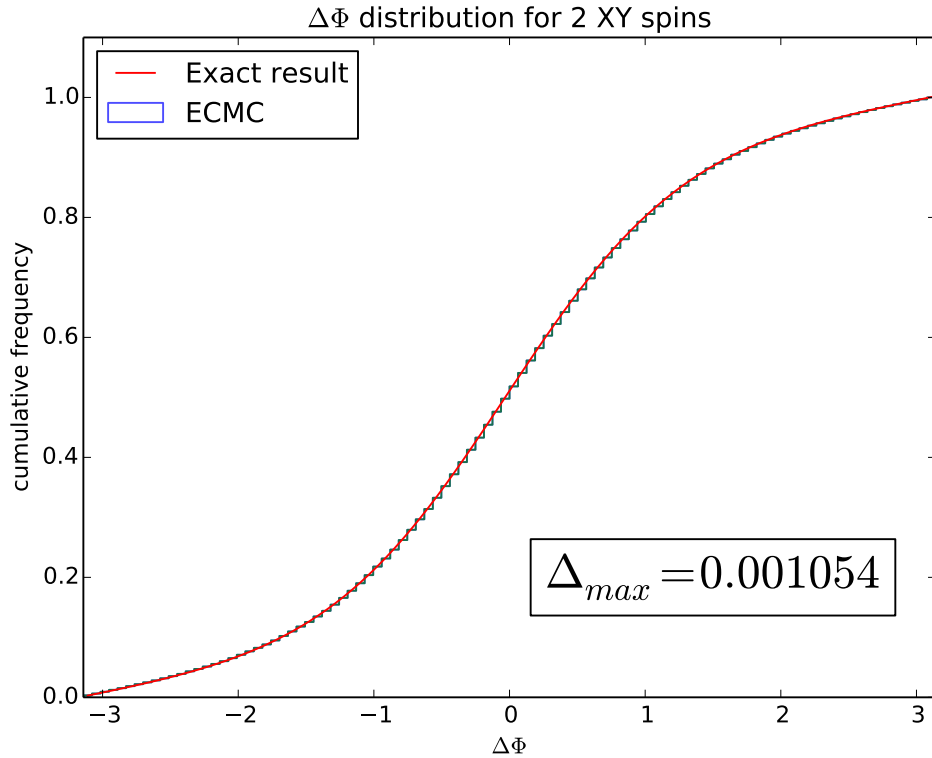


Figure 4.6: Cumulative Distribution for the angular difference $\Delta\Phi$ of two ferromagnetic XY spins simulated with the rejection free algorithm. One can see that the data follows the exact solution perfectly.

$$p(\Delta\Phi) = \frac{1}{\mathcal{Z}} e^{\beta J \cos(\Delta\Phi)} \quad (4.30)$$

with $\mathcal{Z} = \int_0^{2\pi} e^{\beta J \cos(\psi)} d\psi$ which can be evaluated numerically. In figure 4.6 the cumulative distribution of the exact solution and the data for $\beta = J = 1$ is plotted. $n = 10^5$ samples were taken. One can see that they agree perfectly and the maximal distance $\Delta_{max} \leq \frac{1}{\sqrt{n}}$. This means the algorithm can be generalized to N particle systems. Also for the N particle system the spins need only to be turned into one direction only.

5 Simulation of the XY model

In the following chapter the results of the application of the rejection free event chain algorithm to the two dimensional ferromagnetic XY model on a square lattice of length L with periodic boundary conditions are presented. This means the number of particles $N = L^2$. The measurements were taken for $J = 1$ at the critical temperature determined by Hasenbusch $\beta_{crit} = 1.1199$ [8]. The system sizes were chosen as $L = \{16, 32, 64\}$. To make the measurements for each local Markov Chain Monte Carlo (MCMC) simulation 1 million data points were taken at a maximal step width of $\delta = 0.7\pi$. For the Event Chain Monte Carlo (ECMC) one measurement of χ was taken every 100π of total rotation in the system. For each system size different chain lengths ℓ were tested. This means that the measurements were still taken after 100π but after ℓ the lifting variable was resampled. For the system size $L = 16$ $n = 125000$ data points were taken for each chain length, for $L = 32$ $n = 250000$ and for $L = 64$ $n = 350000$ data points. The cluster algorithms had $n = 75000$ data points taken for $L \in 16, 64$ and $n = 250000$ for $L = 32$.

5.1 Measurements of χ

Since in the case of more than two particles it is not possible to calculate the exact angular distribution of the system. If the algorithm produces the correct results was tested on the observable χ and compared to the currently most exact simulations presented in [8].

L	χ_{ECMC}	χ_{MCMC}	χ_{Wolff}	χ_{ref}
16	132.83 ± 0.08	132.88 ± 0.30	132.81 ± 0.21	133.011 ± 0.009
32	452.10 ± 0.45	449.89 ± 1.1	452.11 ± 0.35	452.114 ± 0.031
64	1535.11 ± 3.5	1508.51 ± 12	1528.6 ± 4.0	1536.58 ± 0.11

Table 5.1: Measured magnetic susceptibilities.

One can see that the measurements all agree with each other and with the values obtained by Hasenbusch. To see that the measurements have the same distribution and not only the same mean value the cumulative distributions of the data is plotted in figure 5.1. One can see clearly the data is all from the same distribution.

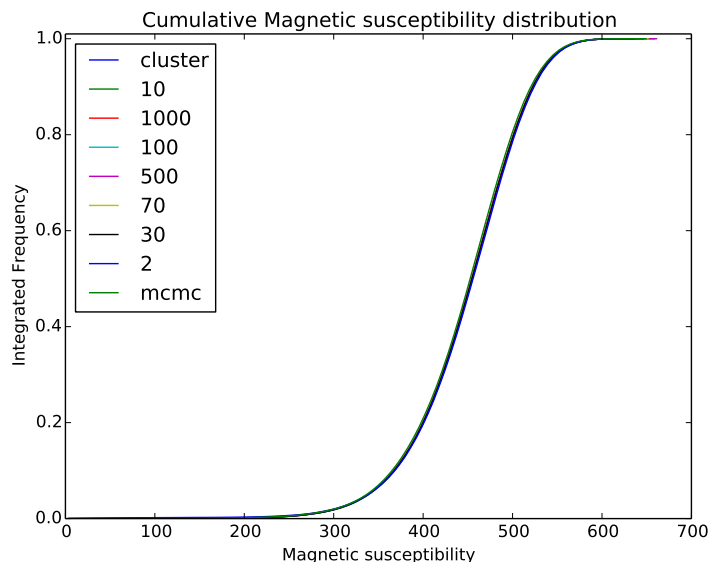


Figure 5.1: Cumulative distribution of χ . One can see that all the simulations follow the same distribution. The different curves are from the MCMC simulation, the Cluster simulation and several different chain lengths. The different chain lengths are given in units of π

5.2 Measurement of τ_χ

In order to compare the autocorrelation functions and the correlation times τ_χ one needs to find comparable units for one time step because all three algorithms work completely different. The reference of time is the number of spins moves attempted by the ECMC algorithm. For the ECMC this means one unit of time were the number of collision

points sampled in 100π movement. This is chosen as a time reference because the average number of moves per fixed measurement length does not depend on the system size but only on the temperature which is held constant for every system size. This is $dt = 354$ attempts. For the MCMC one measurement was taken every $dt = 350$ attempts. Of course in the MCMC algorithm the parameter of the step size is directly responsible for autocorrelation time. However there has been no optimization in this sector during the measurements. The improvement would only be marginal.

The number of operations necessary for one cluster flip strongly depends on the system size. So any measured correlation time for the cluster algorithm should be multiplied by $\frac{\langle \text{cluster size} \rangle}{354}$. The function that was fitted in order to get the correlation time τ_χ was:

$$C(dt; \tau_\chi) = e^{-\frac{dt}{\tau_\chi}} \quad (5.1)$$

For the different sizes the autocorrelation functions are plotted in figures 5.3, 5.2 and 5.4. The average cluster sizes were $\langle c \rangle = 108$ for $L = 16$, $\langle c \rangle = 367$ for $L = 32$ and $\langle c \rangle = 1236$ for $L = 64$.

From the plot one can see several things: first, the speed with which the algorithm produces independent samples is strongly depending on the chain length. A frequent change of lifting variables is less favorable. However there is a chain length above which there is no further improvement. The second thing is that the correlation functions show two different slopes and a knee appears. It is not clear yet if that behavior is due to the lack of data and the correlation function is not computed correctly any more in that regime of dt or that the algorithm possesses two time scales. However the plot of the $L = 64$ data the cluster algorithm shows the same kind of knee in the plot. This autocorrelation function was computed on much less data than the ones on $L = 32$. This hints towards that the knee being an artifact of too little data. But even more important is that the ECMC is able to be as fast or maybe even faster than the Wolff Cluster algorithm if the knee is due to little amount of data.

A major factor in the application of the algorithm to spin systems is that the number of particles with the lifted particle is fix and small. This makes the amount of calculations necessary to find one collision point very little in comparison to systems where the lifted particles is interacting with all $N - 1$ other particles.

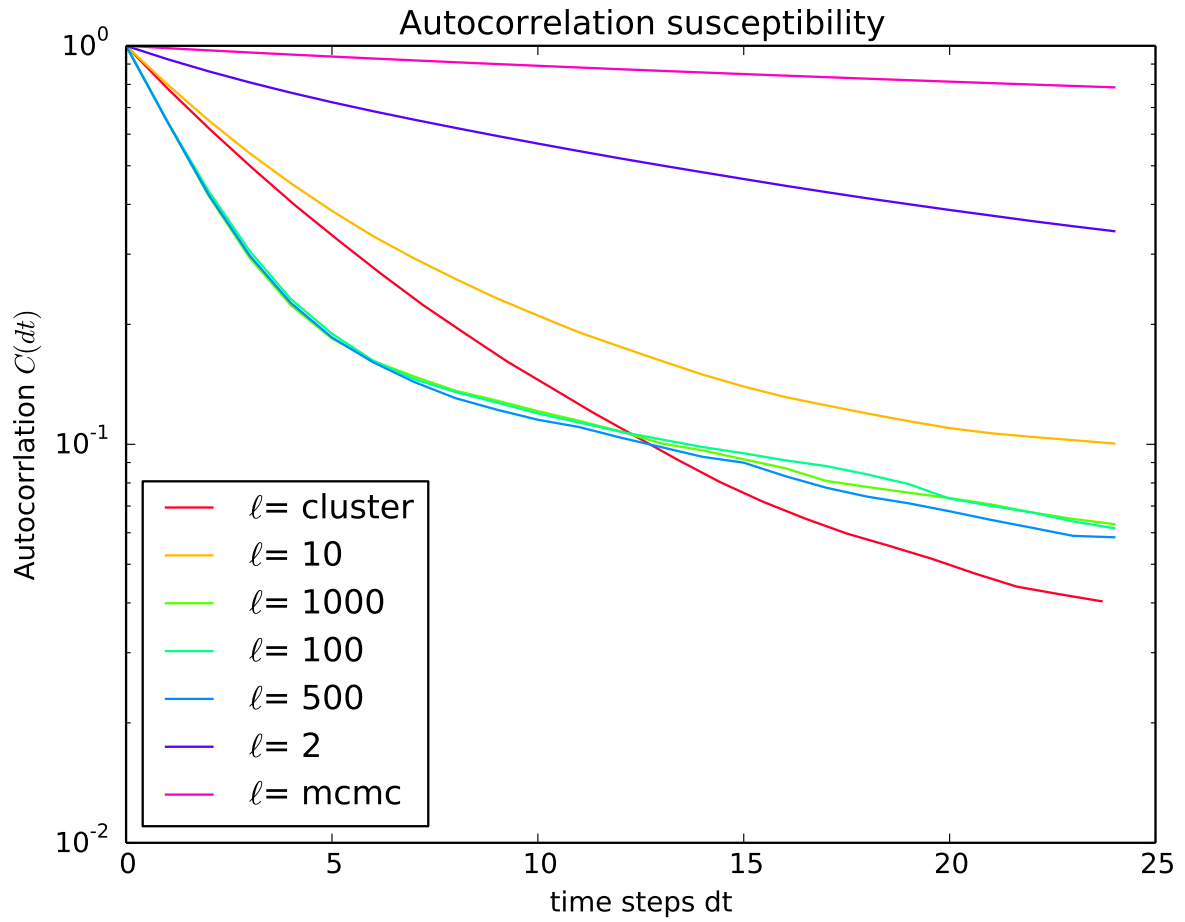


Figure 5.2: Autocorrelation function for the susceptibility χ for $L = 32$. The legend shows the different chain lengths ℓ in units of π .

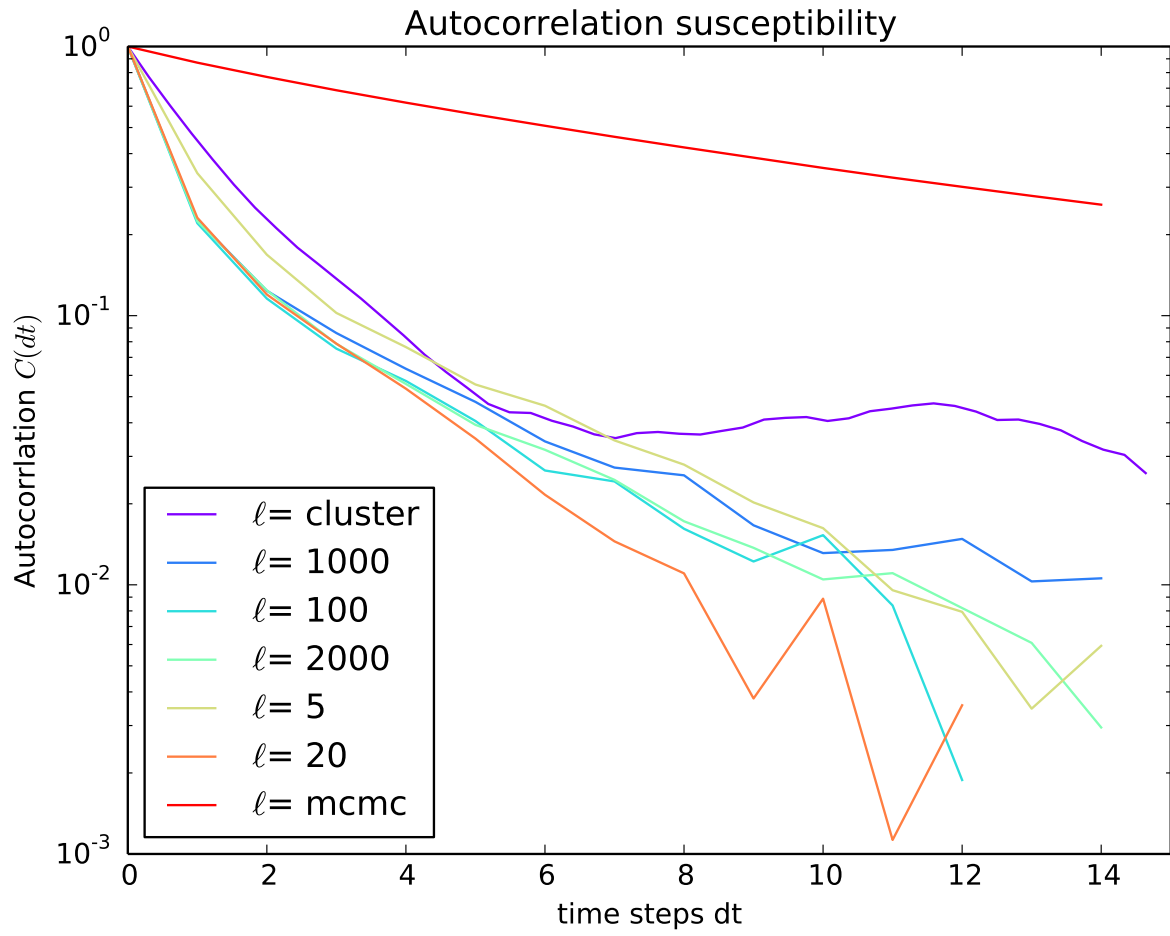


Figure 5.3: Autocorrelation function for the susceptibility χ for $L = 16$. The legend shows different chain lengths ℓ in units of π .

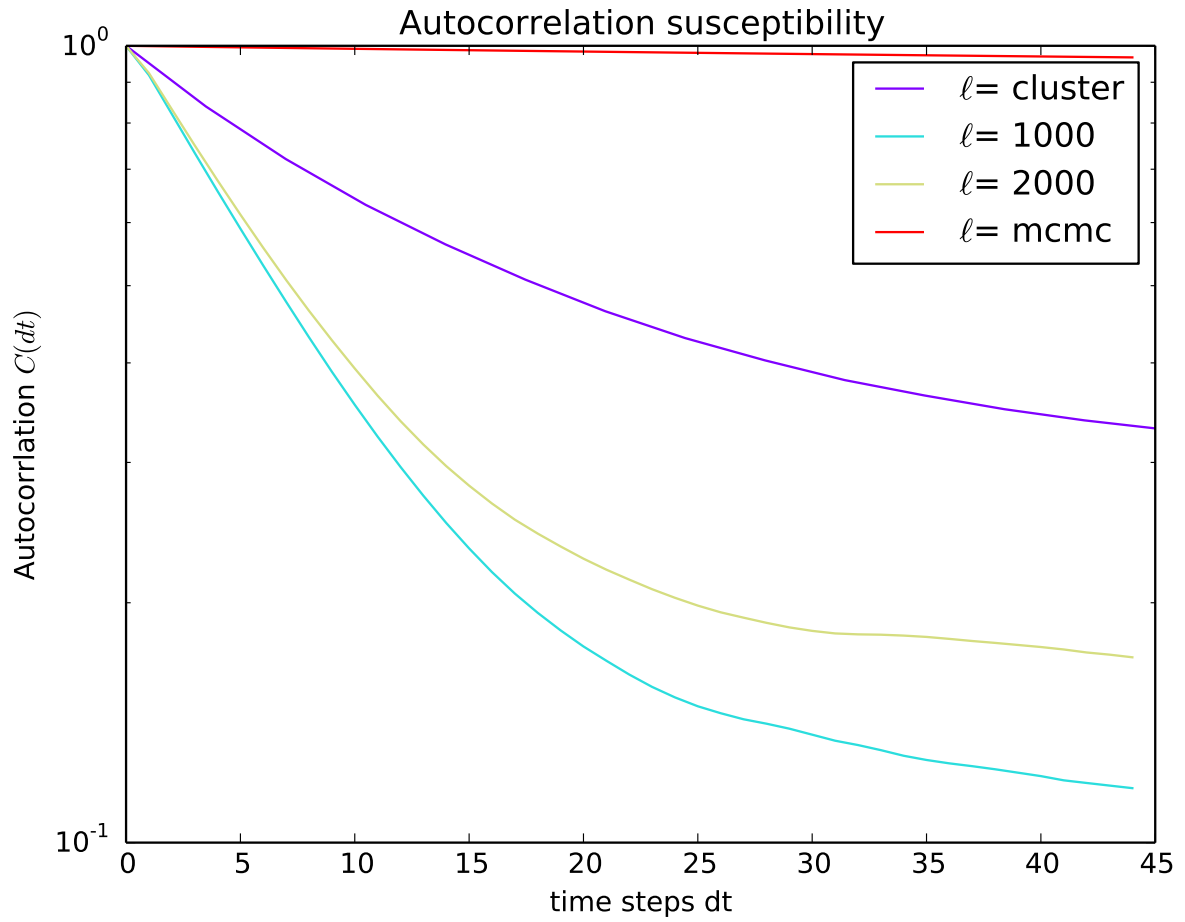


Figure 5.4: Autocorrelation function for the susceptibility χ for $L = 64$. The legend shows different chain lengths ℓ in units of π .

The obtained results were:

L	$\tau_{\chi,ECMC}$	$\tau_{\chi,MCMC}$	$\tau_{\chi,Wolff}$
16	0.7	9.15	1.3
32	2.5	93	4.5
64	9.5	1150	22.7

Table 5.2: Measured correlation times corrected for cluster sizes.

These measured correlation times are only the ones which were measured for the fast decay for small dt .

To see the scaling of the algorithm with the system size in great detail more system sizes with much more data to see the knee disappear would be needed. However from these three points the algorithm scales like $N^{0.95}$, which means linearly with the system size. This is much better than for the local Monte Carlo algorithm who from these three points scales like $N^{1.7}$.

5.3 Concluding Remarks

The Event Chain Monte Carlo algorithm described in [1] can be applied to spin systems with continuous degrees of freedom and pair interaction which only depend on the angular distance between one pair of spins. The algorithm has the potential to be faster than the current fastest algorithm, if there are no two time scales in the algorithm and the curve in the plots is simply an artifact. To find the exact scaling with the system size, simulations with bigger system sizes are needed. The algorithm can also be applied to XY-spinning systems with arbitrary pair coupling J_{ij} and there might also be the possibility for a speed up with respect to the local Markov chain. A paper by the group of W. Krauth about the algorithm is in the making.

I want to thank the members of the group for giving me the opportunity to work with them and for their patience.

Bibliography

- [1] M. Michel, S. C. Kapfer, and W. Krauth. Generalized event-chain monte carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps. The Journal of Chemical Physics, 140(5):–, 2014.
- [2] Kapfer, Sebastian C. and Krauth, Werner. Two-Dimensional Melting: From Liquid-Hexatic Coexistence to Continuous Transitions. Phys. Rev. Lett., 114:035702, Jan 2015.
- [3] Etienne Bernard. Algorithms and applications of the Monte Carlo method : Two-dimensional melting and perfect sampling. Theses, Université Pierre et Marie Curie - Paris VI, September 2011.
- [4] W. Krauth. Statistical Mechanics: Algorithms and Computations. Oxford Master Series in Physics. OUP Oxford, 2006.
- [5] N. Metropolis, A. W. Rosenbluth, M.N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. Journal of Chemical Physics, 21:1087–1092, 1953.
- [6] Ulli Wolff. Collective monte carlo updating for spin systems. Phys. Rev. Lett., 62:361–364, Jan 1989.
- [7] E. A. J. F. Peters and G. de With. Rejection-free monte carlo sampling for general potentials. Phys. Rev. E, 85:026703, Feb 2012.
- [8] Martin Hasenbusch. The Two dimensional XY model at the transition temperature: A High precision Monte Carlo study. J.Phys., A38:5869–5884, 2005.