

Monte Carlo simulations of Potts models with random power-law interactions

By Emma Hagevang



Thesis for the degree of Master of Science

Department of physics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

June 2015

Acknowledgements

I would like to thank Olav Syljuåsen, my thesis supervisor, for his patience and words of encouragement. I would also like to thank my friends and fellow students, Are Flaten and Robert Fauli, for making my time at the university so much more enjoyable.

Emma Hagevang

Oslo, June 2015

Contents

Introduction.....	6
The model.....	6
Monte Carlo method.....	7
Metropolis	10
Cluster algorithms	11
Wolff cluster	11
LB cluster	13
Error treatment and verification of the algorithms	15
Brute force.....	16
Periodic boundary conditions	23
The critical temperature.....	25
Closing remarks	31
Bibliography.....	32

Introduction

We have attempted to simulate a Potts model with random power-law interactions using several different numerical algorithms in an effort to find the critical temperature. We have used the Potts model to model the effective interactions between adsorbates placed at random positions on a graphene surface.

The model

The Potts model Hamiltonian is

$$H = -J \sum_{\langle ij \rangle} \delta_{s_i s_j} \quad (1)$$

Each adsorbate (in the model usually called spin) is assigned an integer value s between 1 and q . Graphene is a hexagonal lattice with a carbon atom at each vertex. Each adsorbate can be located at different microscopic positions on the graphene sheet. If the adsorbate has a tendency to be located above the carbon atoms, there are effectively two different microscopic positions, either it is located above an atom on the A-sublattice or on the B-sublattice. We model this degree of freedom by setting the number of possible s values $q=2$. Figure 1 illustrates this. If on the other hand the adsorbate prefers to be located above carbon-carbon bonds there are three inequivalent positions corresponding to the orientations of the bonds. In that case we use $q=3$. This possibility is illustrated in figure 2. Whether an adsorbate will prefer to be located over a carbon atom or over a carbon-carbon bond depends on what kind of adsorbate it is.

If two neighboring adsorbates have the same s value they contribute $-J$ to the Hamiltonian. In the case of long-range interactions all adsorbates “neighbor” all other adsorbates and the interaction strength J will not necessarily be the same for all the pairs. For our system we take $J = \frac{1}{r^3}$ (ref. [4]). Taking this into account the Hamiltonian becomes

$$H = \sum_{ij} -J_{ij} \delta_{s_i s_j} \quad (2)$$

with $J_{ij} = \frac{1}{r_{ij}^3}$ where r_{ij} is the distance between adsorbates i and j . Note that this means that the microscopic parameters J_0 and a , which are properties of graphene and would otherwise

appear in the Hamiltonian, (or rather the product $J_0 a^3$) have been set equal to one. We also note that in order for the dimensions to make sense we should measure the distance r in terms of the length of the system i.e. $J = \frac{1}{(\frac{r}{L})^3}$. However, we can set the length of the graphene surface, L , equal to one as well. J then becomes dimensionless and if we should wish to use dimensions we must reintroduce J_0 and a .

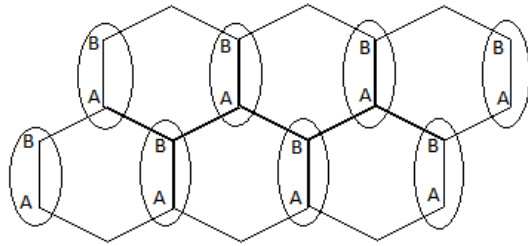


Figure 1 An adsorbate can be located above a carbon atom of type A or type B, giving $q=2$

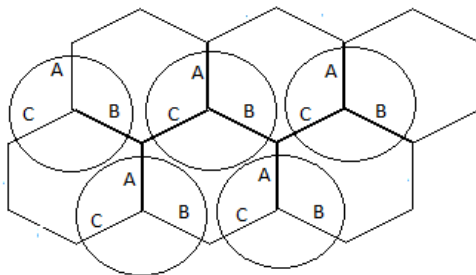


Figure 2 An adsorbate can be located above a carbon-carbon bond with orientation of type A, B or C, giving $q=3$

Monte Carlo method

The expectation value $\langle Q \rangle$ of some observable quantity Q is given by

$$\langle Q \rangle = \frac{\sum_{\mu} Q_{\mu} e^{-\beta E_{\mu}}}{\sum_{\mu} e^{-\beta E_{\mu}}} \quad (3)$$

where the sum is over all states μ of the system. Thus calculating properties of models in statistical mechanics often involves performing sums with very many terms. For large

systems the number of states will be too large and we must limited ourselves to averaging over some subset of states. If we choose M states at random using the probability distribution p_μ the best estimation (often called estimator) of Q is

$$Q_M = \frac{\sum_{i=1}^M Q_i p_i^{-1} e^{-\beta E_i}}{\sum_{j=1}^M p_j^{-1} e^{-\beta E_j}} \quad (4)$$

We want to choose p_μ in such a way that we pick the states which make important contributions to the sum in equation (3) more often. This is called importance sampling. To easily see that not all states contribute equally consider how the system will behave at low temperatures, without the thermal energy needed to excite the system it will spend most of the time in the ground state. The most common form of importance sampling is to sample the states of the system in such a way that the likelihood of any particular state appearing is proportional to its Boltzmann weight, i.e.

$$p_\mu = Z^{-1} e^{-\beta E_\mu} \quad (5)$$

which gives

$$Q_M = \frac{1}{M} \sum_{i=1}^M Q_i \quad (6)$$

How then should we go about generating a random set of states where each of them appears according to its Boltzmann probability? Almost all Monte Carlo schemes solve this by making use of Markov processes. A Markov process is simply put a process which randomly changes the state of the system. The probability of transitioning from one state μ to another state ν is called the transition probability and is denoted by $P(\mu \rightarrow \nu)$. It must always fulfill the three requirements

1. Does not vary over time
2. Depend only on the states μ and ν , and not any other states of the system
3. $\sum_\nu P(\mu \rightarrow \nu) = 1$

Using a Markov process repeatedly produces a Markov chain of states. For Monte Carlo simulations we want to use a Markov process which when run for long enough produces such a chain where the states appear according to the Boltzmann probability distribution,

this is called coming to equilibrium. In order to ensure this we must require two more things of the process; ergodicity and detailed balance.

Ergodicity means that it must be possible for the system to transition between any two states if we repeat the Markov process enough times. This is necessary because in the Boltzmann probability distribution all states have a non-zero probability to appear. Then if it is impossible to reach a state ν from another state μ , no matter how long we run the process, and we start in state μ the probability of state μ appearing in the simulation will be zero, which it should not be.

For a system to be in equilibrium the rate of transition into a state μ must be the same as the rate of transition out of the state μ i.e.

$$\sum_{\nu} p_{\mu} P(\mu \rightarrow \nu) = \sum_{\nu} p_{\nu} P(\nu \rightarrow \mu) \quad (7)$$

since $\sum_{\nu} P(\mu \rightarrow \nu) = 1$ this simplifies to

$$p_{\mu} = \sum_{\nu} p_{\nu} P(\nu \rightarrow \mu) \quad (8)$$

We want to ensure that if we run the process long enough the probability distribution for the system will be p_{μ} no matter what state we start with, however simply choosing transition probabilities that satisfy equation (8) does not guaranty this. While this condition ensures equilibrium it does not guarantee a simple equilibrium as it also permits so called dynamic equilibriums in which the probability distribution will cycle trough a number of values, such a cycle is called a limit cycle.

Detailed balance means that it should be equally likely for the system to transition from μ to ν as from ν to μ , i.e.

$$p_{\mu} P(\mu \rightarrow \nu) = p_{\nu} P(\nu \rightarrow \mu) \quad (9)$$

We see that if detailed balance is fulfilled so is (7). The probability to find the system in a certain state can change, like it will in a limit cycle, but only if there are either more or less transitions out of a state than into the state. Since detailed balance prohibits this it prevents

limit cycles. Setting the probability distribution at equilibrium to the Boltzmann distribution, equation (9) can be written as

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{p_\nu}{p_\mu} = e^{-\beta(E_\nu - E_\mu)} \quad (10)$$

Since this condition only fixes the ratio of the transition probabilities we still have quite a lot of freedom in how we choose them. The transition probabilities can be broken down into two parts like this

$$P(\mu \rightarrow \nu) = g(\mu \rightarrow \nu)A(\mu \rightarrow \nu) \quad (11)$$

The selection probability $g(\mu \rightarrow \nu)$ is the probability that given the initial state μ the state ν will be generated. $A(\mu \rightarrow \nu)$, the acceptance ratio, is the probability that we actually change the system from state μ to ν instead of staying in the state μ .

The condition (10) then becomes

$$\frac{P(\mu \rightarrow \nu)}{P(\nu \rightarrow \mu)} = \frac{g(\mu \rightarrow \nu) A(\mu \rightarrow \nu)}{g(\nu \rightarrow \mu) A(\nu \rightarrow \mu)} \quad (12)$$

The acceptance ratios can have any value between 0 and 1, but if they are low the system will rarely change state and the algorithm will take longer to sample enough states.

Therefore we should try to choose the selection probabilities and the acceptance ratios in such a way that the acceptance ratios are as large as possible within the given constraints.

Metropolis

The Metropolis algorithm (ref. [3]) is characterized by the acceptance ratio

$$A(\mu \rightarrow \nu) = \begin{cases} e^{-\beta\Delta E} & \text{if } \Delta E > 0 \\ 1 & \text{otherwise} \end{cases} \quad (13)$$

If the proposed change lowers the energy of the system we always accept it, otherwise we accept the change with the given probability.

The Metropolis algorithm goes as follows

1. Choose an adsorbate at random

2. Assign it a new s value different from the old. All possible values should be equally probable.
3. Calculate the change in energy, ΔE , that changing the s value in step 2 would cause.
4. Accept or reject the new configuration according to the acceptance probability above.
5. Repeat steps 1-4 n times, where n is the number of adsorbates.
6. Take measurements.

This must be repeated in order for the algorithm to get measurements from enough states. Each such repetition is called a Monte Carlo step. It is also important to allow the algorithm to run for a while before we start taking measurements in order for it to reach equilibrium.

For the Metropolis algorithm all the selection probabilities $g(\mu \rightarrow \nu)$ are the same, the probability of selecting any given spin is always $\frac{1}{n}$ and the selection probability will be $\frac{1}{q-1}$ for all possible s values. Therefore it is up to the acceptance ratios to fulfill the condition of detailed balance. We also need the acceptance ratios to be as close to one as possible. Both of these things can be accomplished by setting the highest acceptance ratio to one and the other to what it needs to be in order to fulfill detailed balance, which is how the acceptance ratio given in (13) was determined. Ergodicity is also fulfilled since we can change any spin to any s value with a non-zero probability.

Cluster algorithms

Wolff cluster

The Metropolis algorithm as described above is a single-flip algorithm because we change a single spin at a time. A cluster algorithm on the other hand changes entire groups of spins with the same s value, called clusters, at once. One such cluster algorithm is the Wolff algorithm (ref. [5]) which goes as follows

1. Choose an adsorbate at random. This adsorbate is called the seed spin.
2. Go through all other adsorbates and if they have the same s value add them to the cluster with probability p_{add} . The adsorbate we are currently attempting to activate bonds from is called the current spin. To activate a bond to an adsorbate is the same as adding it to the cluster.

3. For each adsorbate added in step 2 we then repeat step 2 with this adsorbate as current spin. This is repeated until every adsorbate in the cluster has been the current spin.
4. Change s value for all adsorbates in cluster. Though for practical purposes it is a good idea to change the s value of a spin when we add it to the cluster since that will automatically prevent the adding of the same spin twice.

So we need to determine the probability p_{add} . Consider a state μ that can be transformed into another state ν by changing one cluster. The probability of selecting a spin as seed spin is the same in both states, so is the probability of adding the spins in the cluster to the cluster and so is the probability of choosing the needed new s value to change the cluster to. The only probability that changes is the probability of not adding the spins with the same s value that is not supposed to be a part of the cluster. Say we have n_a spins with the same s value as the cluster that is not to be added to the cluster in state μ , the probability of not adding these n_a spins is $\prod_i^{n_a} (1 - P_{\text{add},i})$. And in state ν we have n_b spins with the same s value that should not be added, the probability of not adding them is then $\prod_j^{n_b} (1 - P_{\text{add},j})$. For the Wolff cluster the condition of detailed balance then becomes

$$\frac{g(\mu \rightarrow \nu) A(\mu \rightarrow \nu)}{g(\nu \rightarrow \mu) A(\nu \rightarrow \mu)} = \frac{\prod_i^{n_a} (1 - P_{\text{add},i}) A(\mu \rightarrow \nu)}{\prod_j^{n_b} (1 - P_{\text{add},j}) A(\nu \rightarrow \mu)} = e^{-\beta(E_\nu - E_\mu)} \quad (14)$$

When we change the cluster we will get n_b times the size of the cluster new spins bonds which will contribute to the energy and we break n_a times the size of the cluster which will no longer contribute to the energy. The change in the energy is then given by

$$E_\nu - E_\mu = \beta \left(\sum_i^{n_a \text{clsize}} J_i - \sum_j^{n_b \text{clsize}} J_j \right) \quad (15)$$

Inserting this into (14) we get

$$\frac{\prod_i^{n_a} (1 - P_{\text{add},i}) A(\mu \rightarrow \nu)}{\prod_j^{n_b} (1 - P_{\text{add},j}) A(\nu \rightarrow \mu)} = e^{-\beta(\sum_i^{n_a \text{clsize}} J_i - \sum_j^{n_b \text{clsize}} J_j)} \quad (16)$$

Some reorganizing gives

$$\frac{A(\mu \rightarrow \nu)}{A(\nu \rightarrow \mu)} = \prod_i^{n_{aclsiz}} e^{-\beta J_i} \prod_j^{n_{bclsiz}} e^{\beta J_j} \prod_i^{n_{bclsiz}} (1 - P_{add,i}) \prod_j^{n_{aclsiz}} (1 - P_{add,j})^{-1} \quad (17)$$

We see that if we set $P_{add,k} = 1 - e^{-\beta J_k}$ the acceptance ratios will be equal to one. Ergodicity will also be fulfilled since it is possible for a cluster to consist of only one spin.

LB cluster

In ref. [1] Luijten and Blöte presents an algorithm inspired by the Wolff algorithm, but instead of attempting to add every single spin one by one as in the standard Wolff cluster algorithm they introduces the cumulative bond probability

$$C(j) = \sum_{n=1}^j P(n) \quad (18)$$

Where

$$P(j) = (1 - p_1)(1 - p_2) \dots (1 - p_{j-1})p_j \quad (19)$$

is the probability that the first spin to be added to the cluster is at a distance j from the current spin (assuming the same s value). With

$$p_n = 1 - e^{-\beta J_n} \quad (20)$$

the probability of adding a spin at a distance n . J_n is the interaction strength for the bond between spins with distance n .

We draw a random number a_r and if a_r is between $C(j-1)$ and $C(j)$ we active bond j , assuming the same s value. The next bond to be activated should be to a spin further away, at a distance k . That means we must change P accordingly which gives

$$P_j(k) = (1 - p_{j+1})(1 - p_{j+2}) \dots (1 - p_{k-1})p_k \quad (21)$$

And the cumulative probability becomes

$$C_j(k) = \sum_{n=j+1}^k P_j(n) \quad (22)$$

Substituting for p_n we get

$$C_j(k) = 1 - e^{-\beta \sum_{n=j+1}^k J_n} \quad (23)$$

We then use the cumulative probability (22) in exactly the same way we used (18) to find the next distance k , then we shift P again. This is repeated until we have been through all possible distances.

The algorithm in ref. [1] is meant for systems in which the spins are positioned at the vertices of a regular lattice. Since the positions of the spins in our system are random some modifications are required. Firstly j/k are no longer the distance to the next spin to be added, instead we sort the bonds by the strength of the interaction and make j/k an index running over this list. That is if we draw a random number a_r between $C(j-1)$ and $C(j)$ we add the j th spin in the list. So when implementing this we need to make a table that for each spin contains

1. The interaction strength with every other spin, sorted from weakest to strongest.
2. The cumulative probability
3. Indices for the vector that stores the s values since the sorting will be different for each current spin

One might think that it is necessary to also store the $C_j(k)$ s for every value of j as well, fortunately this is not necessary as we can calculate them from the $C(j)$ s. We see that

$$C(k) = C(j) + \left(\prod_{i=1}^j (1 - p_i) \right) C_j(k) = C(j) + (1 - C(j)) C_j(k) \quad (24)$$

Which rewritten gives

$$C_j(k) = \frac{C(k) - C(j)}{1 - C(j)} \quad (25)$$

Another thing that we must take into consideration is the fact that for a system with spins located at the vertices of a lattice the minimum distance between them will be one whereas in our system the maximum distance is one. This means that the interaction strength will be larger. Many of them will be so large in fact that the exponentials in the expression for the cumulative probability will be numerically zero and we end up with many cumulative probabilities equal to one, even when they are not supposed to be.

This can be solved by using the LB algorithm for the weaker bonds and then switching to the standard Wolff algorithm for the strong bonds. Since these bonds are so strong almost all the spins will be added to the cluster anyway and there is not much to gain from using the LB algorithm for the strong couplings. This combination of the Wolff and the LB algorithm is the method we have used.

Error treatment and verification of the algorithms

It is important to have some way of estimating the statistical errors in the simulation. One such method is called blocking. Say we have N measurements, these measurements are then divided into N_b groups or blocks. For each block we can then calculate the mean of the measurements in that block. Then we calculate the mean of these “block means”, which is our best estimation of that quantity. If we make the blocks large enough we can treat the “block means” as statistically independent and calculate the standard deviation on the total mean using

$$\sigma = \sqrt{\frac{\frac{1}{N_b} \sum_{i=0}^{N_b} (m_i - \bar{m})^2}{N_b - 1}} \quad (26)$$

We must also of course check that the algorithms work properly. This can be done by comparing the results against an analytical solution. For as few as two and three adsorbates we can find an exact expression for the energy with pen and paper. For two adsorbates and with $q=3$ we have

$$\begin{aligned} \langle E \rangle &= \frac{\sum_s E_s e^{-\beta E_s}}{\sum_s e^{-\beta E_s}} \\ &= -\frac{\frac{3}{r^3} e^{\frac{\beta}{r^3}}}{3e^{\frac{\beta}{r^3}} + 6} \end{aligned} \quad (27)$$

$q=2$ gives

$$\langle E \rangle = \frac{-\frac{2}{r^3} e^{\frac{\beta}{r^3}}}{2e^{\frac{\beta}{r^3}} + 2} \quad (28)$$

And for $q=4$

$$\langle E \rangle = \frac{-\frac{4}{r^3} e^{\frac{\beta}{r^3}}}{4e^{\frac{\beta}{r^3}} + 12} \quad (29)$$

For three adsorbates and $q=3$ we get

$$\begin{aligned} \langle E \rangle &= \frac{\sum_s E_s e^{-\beta E_s}}{\sum_s e^{-\beta E_s}} \\ &= \frac{-3(J_{12} + J_{13} + J_{23})e^{\beta(J_{12}+J_{13}+J_{23})} - 6J_{12}e^{\beta J_{12}} - 6J_{13}e^{\beta J_{13}} - 6J_{23}e^{\beta J_{23}}}{3e^{\beta(J_{12}+J_{13}+J_{23})} + 6e^{\beta J_{12}} + 6e^{\beta J_{13}} + 6e^{\beta J_{23}} + 6} \end{aligned} \quad (30)$$

And $q=2$ gives

$$\langle E \rangle = \frac{-2(J_{12} + J_{13} + J_{23})e^{\beta(J_{12}+J_{13}+J_{23})} - 2J_{12}e^{\beta J_{12}} - 2J_{13}e^{\beta J_{13}} - 2J_{23}e^{\beta J_{23}}}{2e^{\beta(J_{12}+J_{13}+J_{23})} + 2e^{\beta J_{12}} + 2e^{\beta J_{13}} + 2e^{\beta J_{23}}} \quad (31)$$

where, as before, $J_{ij} = \frac{1}{r_{ij}^3}$.

Brute force

As mentioned previously the number of terms in (3) will be too large to handle for large systems, however it is doable for smaller systems. In order to further verify the results from the Monte Carlo algorithms we wish to compare them to a straight forward brute force numerical calculation of this sum. However there is one issue with this approach even for relatively small systems. Since the size of the surface $L \times L$ is constant the more adsorbates we have the closer together they will be. That means that the interaction strength $J_{ij} = \frac{1}{r_{ij}^3}$ will be large for most of the pairs which will give very large energies for many of the states. When we then perform the sum many of the terms will then be very large and we risk getting a result that is too large for the computer to handle. This can be avoided by adding a constant to the energy (we add since the energy will be negative). This can be seen as adjusting the energy of all states by the same amount and leads to no physical changes in the system.

Denoting the new expectation value for the energy $\langle E' \rangle$ and the constant we add E_k , we have

$$\begin{aligned}
\langle E' \rangle &= \frac{\sum_s (E_s + E_k) e^{-\beta(E_s + E_k)}}{\sum_s e^{-\beta(E_s + E_k)}} \\
&= \frac{e^{-\beta E_k} \sum_s E_s e^{-\beta E_s} + e^{-\beta E_k} E_k \sum_s e^{-\beta E_s}}{e^{-\beta E_k} \sum_s e^{-\beta E_s}} \\
&= \frac{\sum_s E_s e^{-\beta E_s}}{\sum_s e^{-\beta E_s}} + E_k \\
&= \langle E \rangle + E_k \quad (32)
\end{aligned}$$

We see that we can simply calculate $\langle E' \rangle$ and then subtract E_k to find the original expectation value $\langle E \rangle$. Similarly for the expectation value for the energy squared

$$\begin{aligned}
\langle E'^2 \rangle &= \frac{\sum_s (E_s + E_k)^2 e^{-\beta(E_s + E_k)}}{\sum_s e^{-\beta(E_s + E_k)}} \\
&= \frac{e^{-\beta E_k} \sum_s E_s^2 e^{-\beta E_s} + 2E_k e^{-\beta E_k} \sum_s E_s e^{-\beta E_s} + e^{-\beta E_k} E_k^2 \sum_s e^{-\beta E_s}}{e^{-\beta E_k} \sum_s e^{-\beta E_s}} \\
&= \langle E^2 \rangle + 2E_k \langle E \rangle + E_k^2 \quad (33)
\end{aligned}$$

We also note that since the constant in the exponential will cancel out, the expectation value of any quantity that is not the energy will not be affected, which is consistent with adding the constant not causing any physical changes in the system.

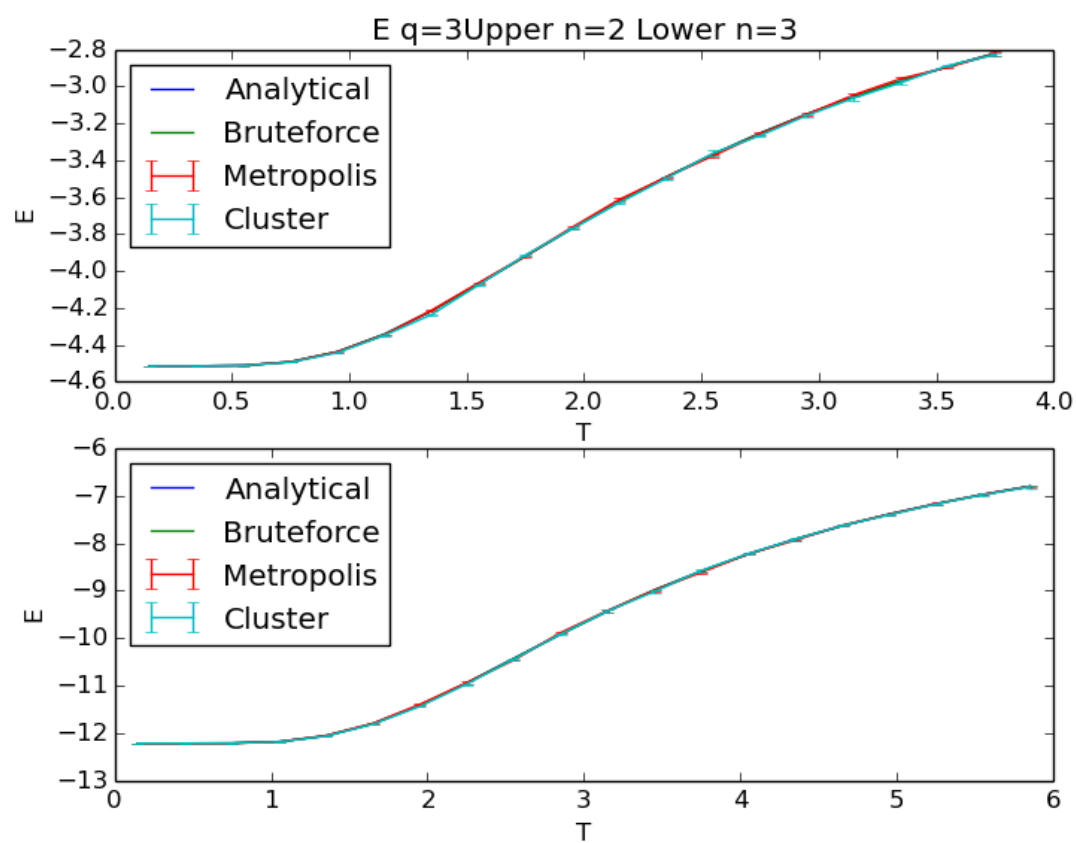


Figure 3 $\langle E \rangle$ calculated using all four different methods for two adsorbates in the upper panel and three adsorbates in the lower panel with $q=3$

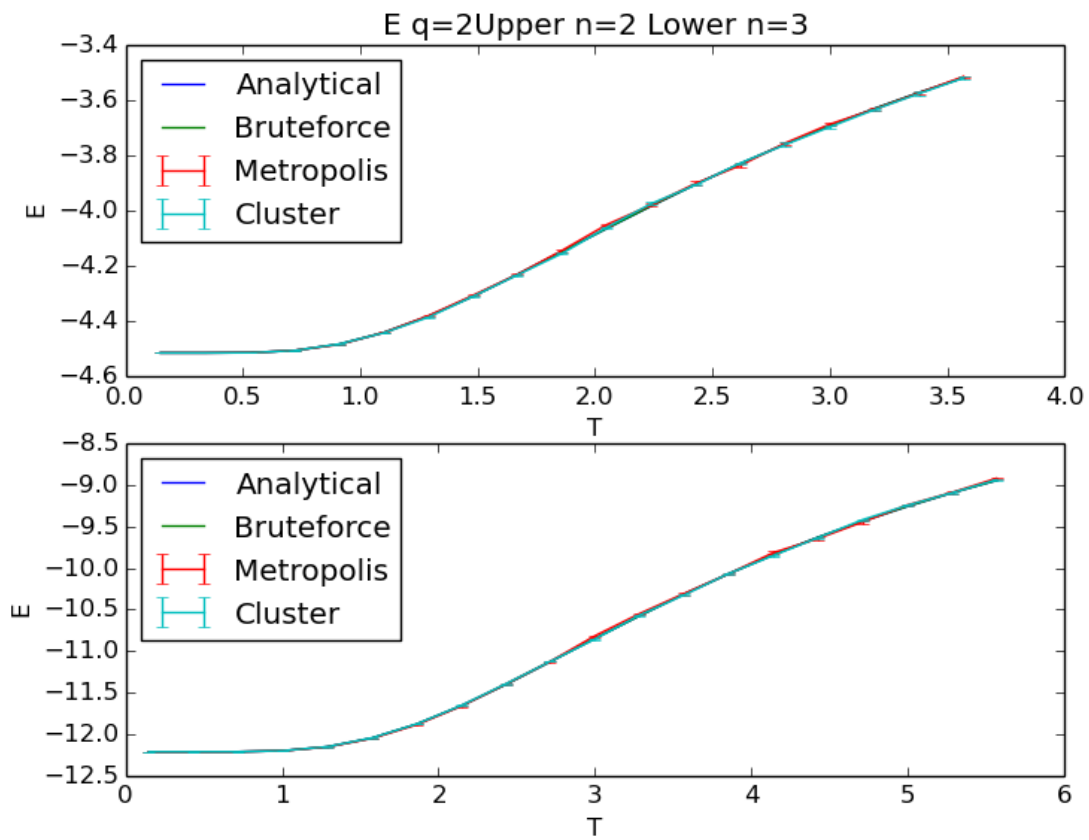


Figure 4 $\langle E \rangle$ calculated using all four different methods for two adsorbates in the upper panel and three adsorbates in the lower panel with $q=2$

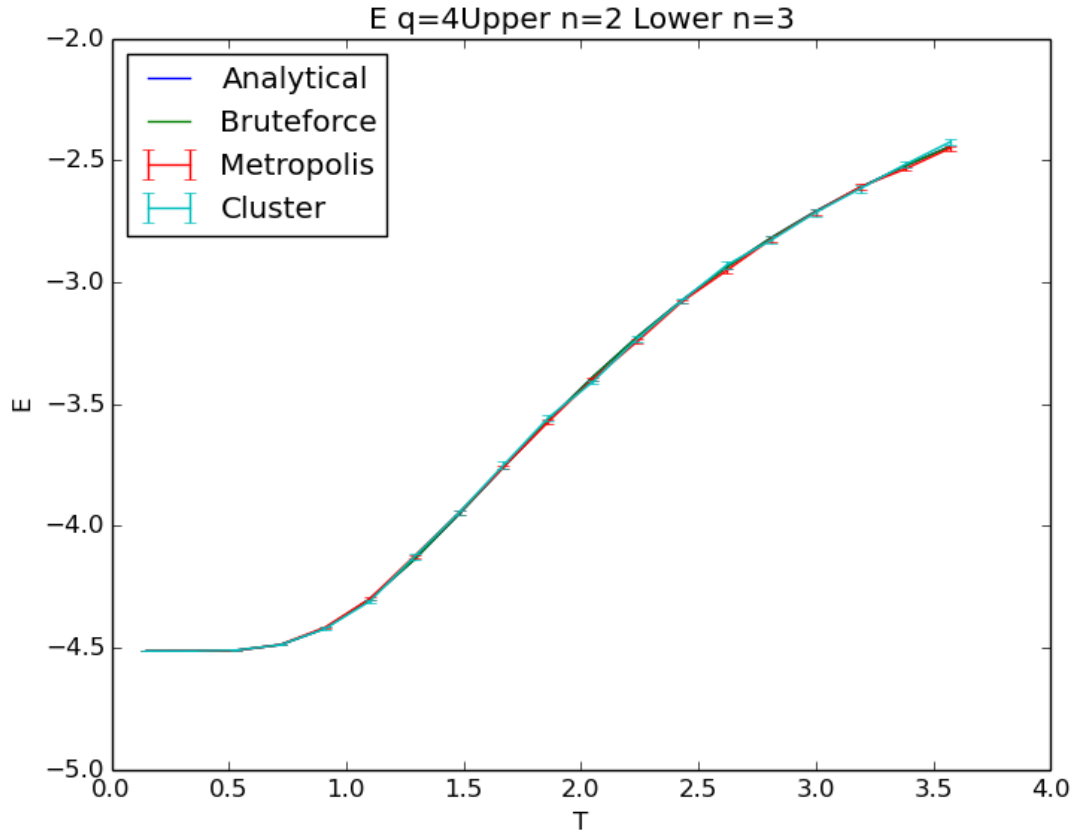


Figure 5 $\langle E \rangle$ calculated using all four different methods for two adsorbates and with $q=4$

In figures 3-5 we see that the results from both the brute force calculation and the two Monte Carlo algorithms are in agreement with the exact analytical expression in equations (27-31). Figures 6 and 7 upper panels show the order parameter $|m|^2$ with m given by

$$m = \sum_{s=1}^q e^{\frac{2\pi i s}{q}} n_s \quad (34)$$

Where $n_s = \frac{N_s}{n}$ is the proportion of spins with value s .

In the lower panels of figures 6 and 7 we show the specific heat C_v given by the expression

$$C_v = \frac{d \langle E \rangle}{dT} \quad (35)$$

$$= \frac{d}{dT} \left(\frac{1}{Z} \sum_s E_s e^{-\beta E_s} \right) = \frac{1}{T^2} (\langle E^2 \rangle - \langle E \rangle^2) \quad (36)$$

For $n=5$, figure 6, we get the same results using all three numerical methods. However, for $n=10$ we see that Metropolis disagrees with the other two. For low temperatures almost all spins will have the same s value and changing the value of a spin will increase the energy, since we have long-range interactions the energy increase will be larger for a higher number of spins. For the Metropolis algorithm the acceptance ratios will be very small for large increases in energy. This gives the algorithm a tendency to get stuck in the ground state. For the cluster algorithm on the other hand the acceptance ratios will always be one and the cluster constructing probabilities only depend on the temperature and the interaction between the spins and not the total energy change of the system, which means that we avoid this problem. Therefore we have used the cluster algorithm and not Metropolis when calculating our results.

We also note that the curves for the specific heats are not smooth. This is caused by the way we calculate the specific heat. Normally we would want to use (36) however in our case that will not work properly due to catastrophic cancellation. Catastrophic cancellation can occur when we compute small numbers by subtracting two nearly equal large numbers. To avoid this we instead use (35) and the centered differencing formula

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} \quad (37)$$

to calculate the specific heat. Unfortunately numerical derivation will introduce some inaccuracies.

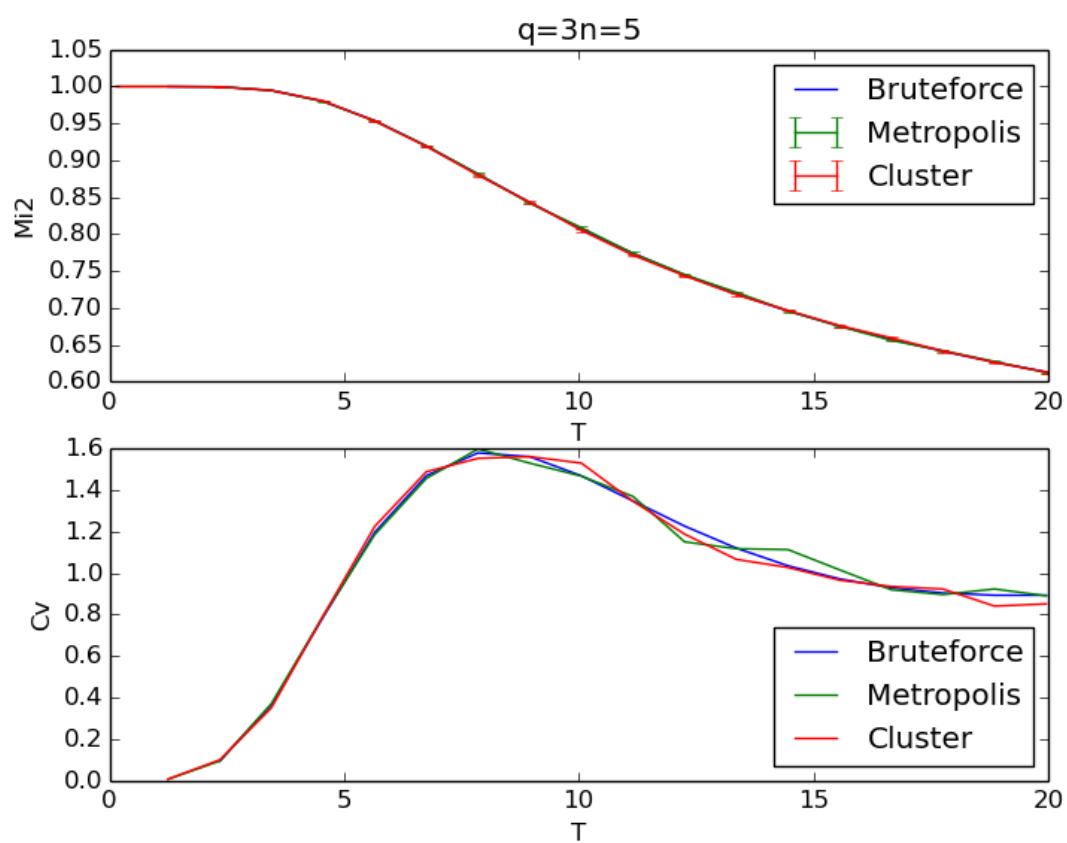


Figure 6 $|m|^2$ and C_v for 5 adsorbates and $q=3$

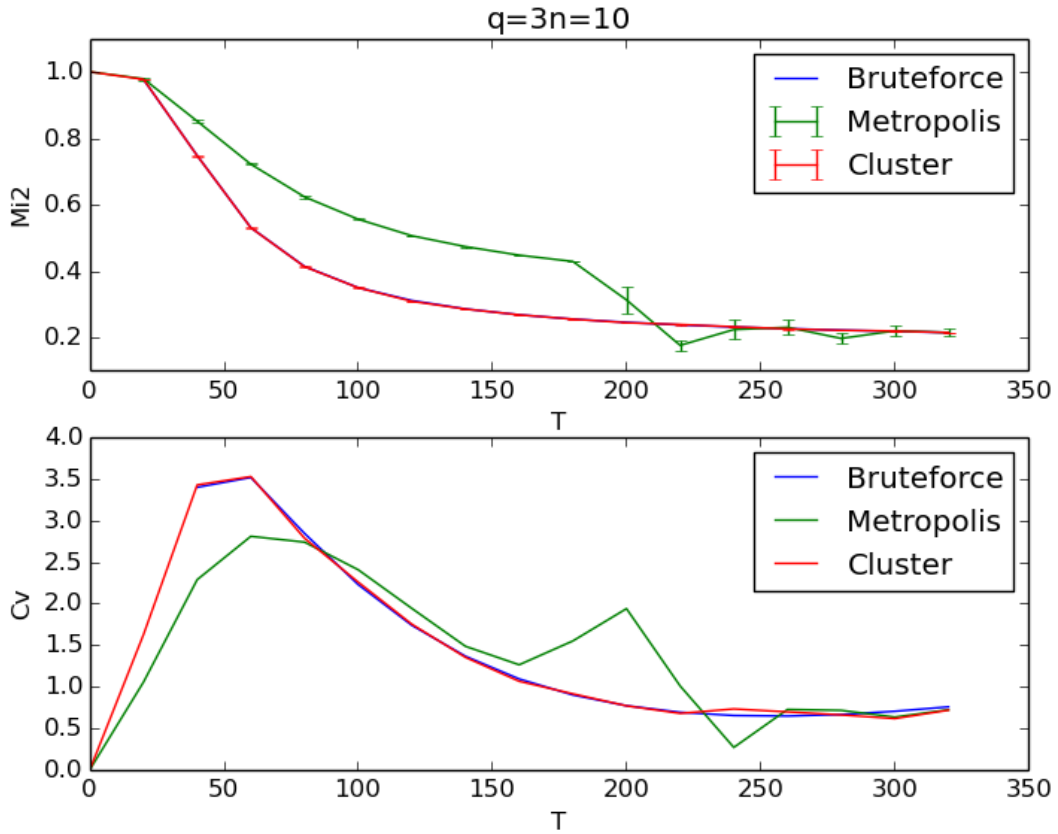


Figure 7 $|m|^2$ and C_v for 10 adsorbates and $q=3$

Periodic boundary conditions

In order to minimize boundary effects we employ periodic boundary conditions. This is done by surrounding our $L \times L$ surface with copies of itself.

Periodic boundary conditions means the energy will not only depend on the interactions between the spins of our original system, now we must also take into account the interactions between the spins of the original system and the copies. This can be implemented by changing the effective interaction from J_{ji} to J'_{ji} given by

$$J'_{ji} = \sum_{n_x, n_y} \frac{1}{\left((x_j - x_i + n_x L)^2 + (y_j - y_i + n_y L)^2 \right)^{\frac{3}{2}}} \quad (38)$$

which is the sum of the interactions between spin j and all version of the spin i ,

where n_x and n_y denotes which copy of i we are considering. For example for the copy located directly to the right of the original surface we will have $n_x=1$ and $n_y=0$.

As we can see in figure 8 the results obtained using the three different methods are in agreement for periodic boundary conditions as well.

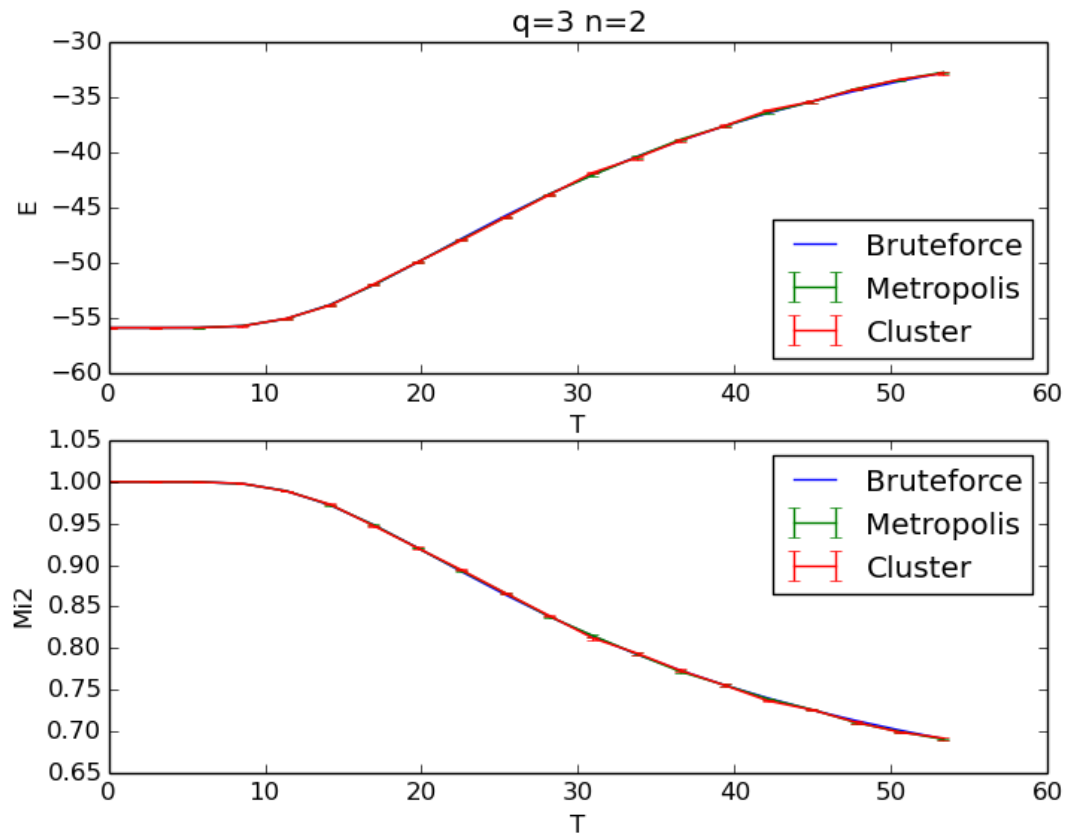


Figure 8 $\langle E \rangle$ and $|m|^2$ calculated using periodic boundary conditions

Figure 9 shows $|m|^2$ and C_v using different numbers of copies. We see that the effect of the interactions with the copies further away than five L is negligible. This means that we only need to include five copies in each direction when calculating results.

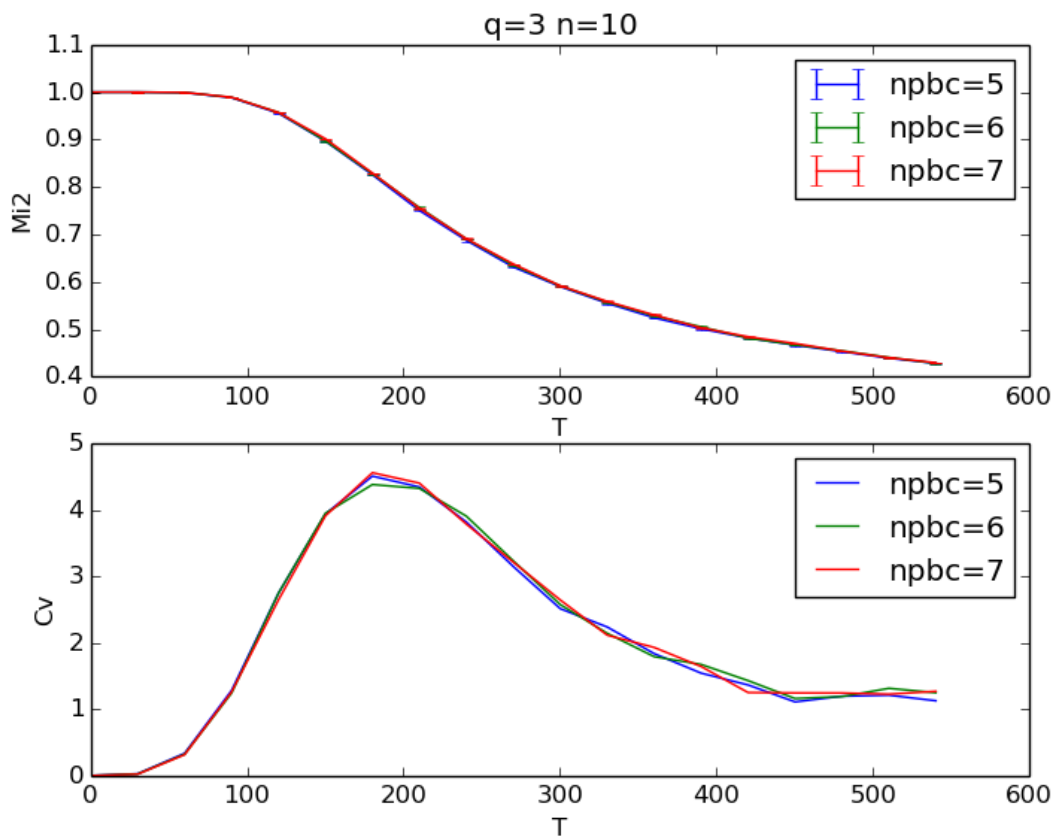


Figure 9 $|m|^2$ and C_v calculated using different numbers of copies. N_{pbc} denotes the number of copies included in each direction

The critical temperature

As previously mentioned, for low temperatures almost all spins will have the same s value whereas for higher temperatures there will be an almost equal number of spins with each possible s value. The transition between these two regimes is sharp, the temperature at which it is at its sharpest is called the critical temperature and such a transition is called a phase transition. As we can see in figure (10) the behavior of the system will be dependent on the positions of the spins, which we expect since the interactions depend on the positions. We also see that this dependence is stronger for fewer adsorbates. This makes sense since adding more spins will have a self-averaging effect on the inter-particle distances.

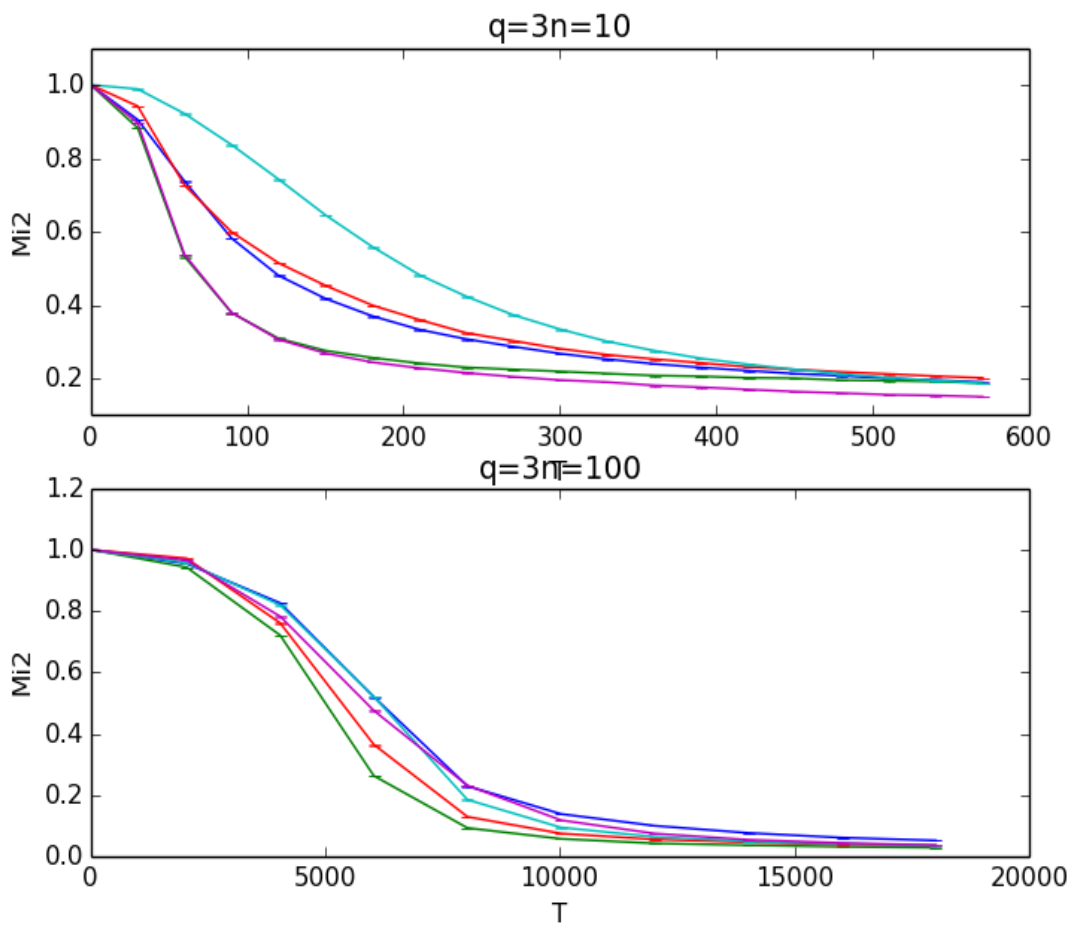


Figure 10 $|m|^2$ for five different sets of positions for the adsorbates

We do not want to limit ourselves to one particular set of positions or number of adsorbates but rather look at a general system. That means that in order to find the critical temperature we must take the average over several different sets of adsorbate positions. And as we can see in figure (11) we must scale the temperature if we wish to compare curves for different values of number of adsorbates n .

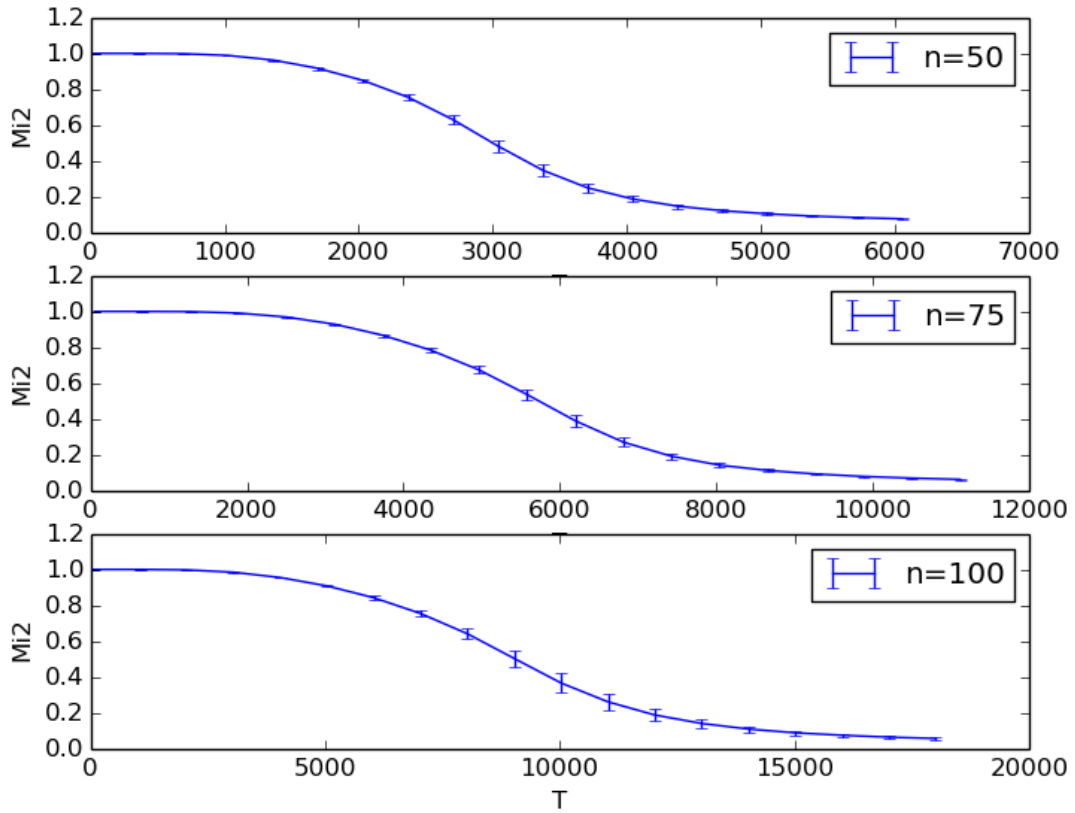


Figure 11 $|m|^2$ for different values of n and $q=3$

The typical distance between adsorbates is

$$r_t = \sqrt{\frac{L^2}{n}} = \frac{L}{\sqrt{n}} \quad (40)$$

And the typical interaction strength is then

$$J_t = \frac{1}{r_t^3} = \frac{n^{\frac{3}{2}}}{L^3} \quad (41)$$

The temperature will scale with this and in order to compare various values of n we must then divide the temperature we are plotting against by $n^{\frac{3}{2}}$.

Doing this gives us the results shown in figures 12-14, where in addition to $|m|^2$, we show C_v and the Binder cumulant (ref. [8]) given by

$$U_M = 1 - \frac{\langle m^4 \rangle}{3 \langle m^2 \rangle^2} \quad (42)$$

From these figures we find that the critical temperature T_c is approximately 7.5 for $q=2$, 8.5 for $q=3$, and 11 for $q=2$.

Figure 15 shows T_c plotted as functions of q . We see that T_c decreases for increasing q which is reasonable as that simply means that it is easier to disturb the order when we have a greater number of possible s values. Based on this limited number of data points T_c appears to be close to inversely proportional to q .

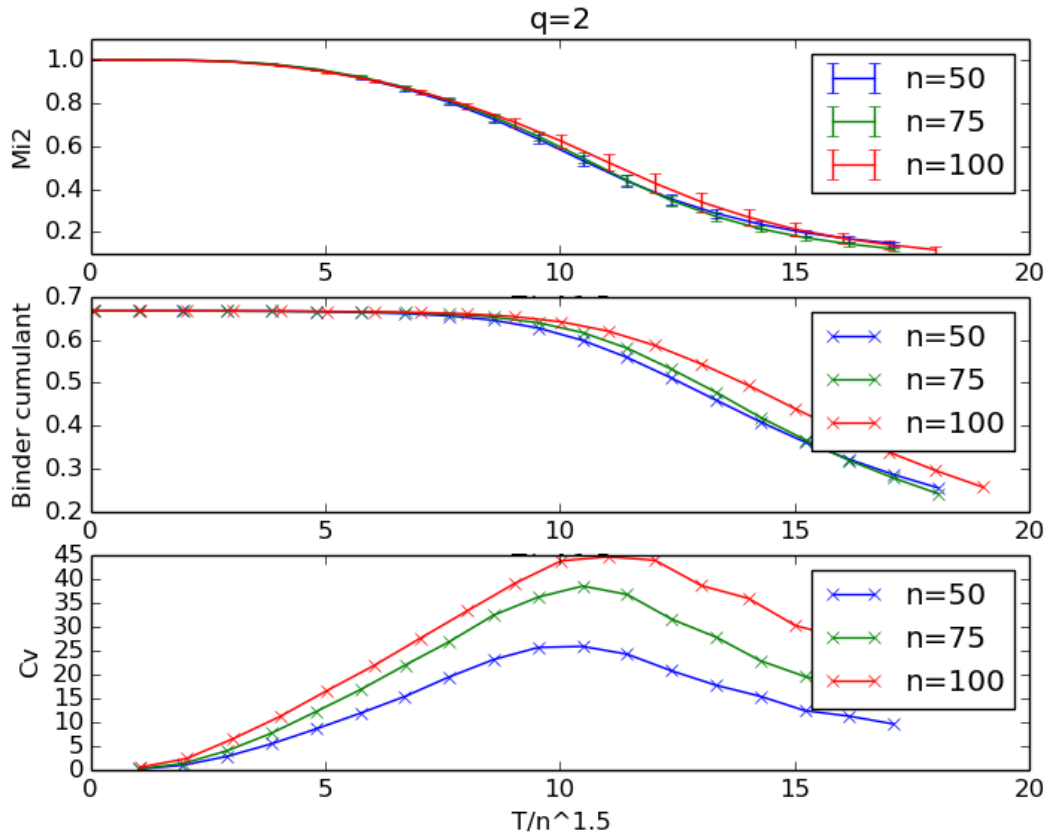


Figure 12 $|m|^2$, the Binder cumulant, and C_v for different n with the temperature scaled for $q=2$

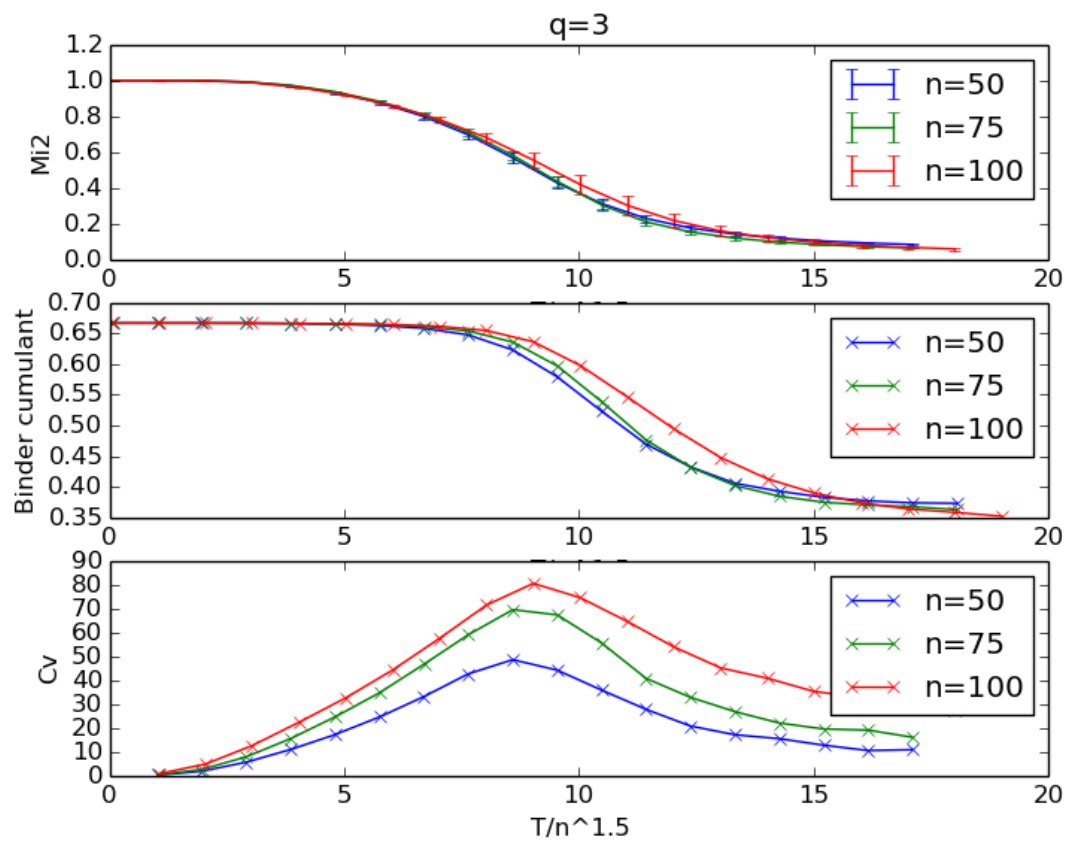


Figure 13 $|m|^2$, the Binder cumulant, and C_v for different n with the temperature scaled for $q=3$

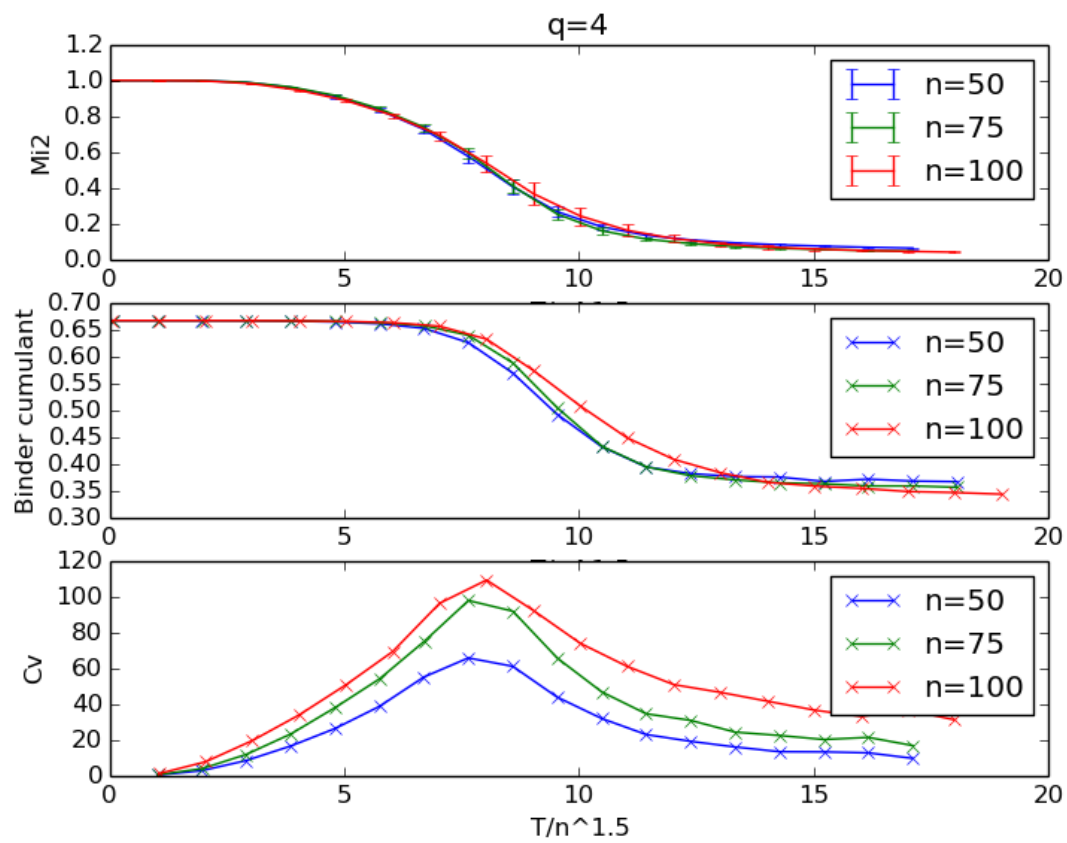


Figure 14 $|m|^2$, the Binder cumulant, and C_v for different n with the temperature scaled for $q=4$

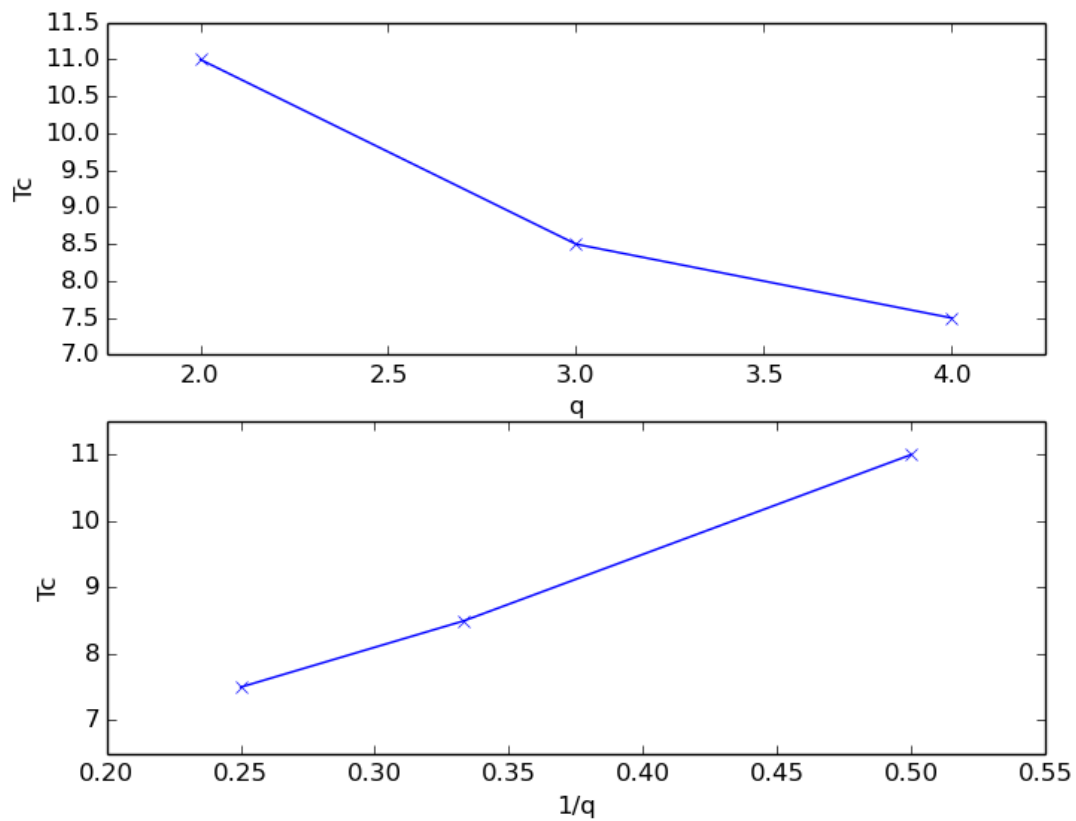


Figure 15 The approximate values of T_c as functions of q

Closing remarks

We have found that a cluster algorithm is a good fit for simulating this kind of system. Although the particular algorithm we have used would probably be even more suited for larger systems, as they will have a greater number of weak bonds we could potentially skip. We have also determined an approximate value for T_c for three different values of q . In order to get more accurate estimates of T_c we would need to run the program for larger values of n , as well as increasing the number of data points close to T_c .

Bibliography

- [1] E. Luijten and H.W.J. Blöte. *Monte Carlo method for spin models with long range interactions*. Int. J. Mod. Phys. C6, 359-370, (1995).
- [2] M.E.J. Newman and G.T. Barkema. *Monte Carlo methods in statistical physics*. Oxford University Press, (1999).
- [3] N. Metropolis et al. *Equation of State Calculations by Fast Computing Machines*. J. Chem. Phys. 21, 1087, (1953).
- [4] V.V. Cheianor, V.I. Falko, O. Syljuåsen, and B.L. Altshuler. *Partial Kekule Ordering of Adatoms on Graphene*. Solid State Comm. 149, 1499, (2009).
- [5] U. Wolff. *Collective Monte Carlo Updating for Spin Systems*. Phys. Rev. Lett. 62, 361, (1989).
- [6] E. Luijten and H.W.J. Blöte. *Classical critical behavior of spin models with long-range interactions*. Phys. Rev. B, 56, 8945-8958, (1997).
- [7] M.H. Jensen. *Computational Physics Lecture Notes Fall 2012*. University of Oslo, (2012).
- [8] K. Binder. *Finite scaling analysis of Ising model block distribution functions*. Zeitschrift für Physik B, 43, 2, 119-140, (1981).
- [9] H.W.J. Blöte, J.R. Heringa, E. Luijten. *Cluster Monte Carlo: Extending the range*. Comp. phys. Comm. 147, 58-63, (2002).
- [10] D.V. Schroeder. *An introduction to thermal physics*. Addison Wesley Longman, (2000).