Esempio LaTeX

Alberto Paolini

December 18, 2024

1 Lattice gauge theory

The main difference between Ising lattice and gauge lattice is that the variables of the second one are the links instead of the sites. In an Ising gauge lattice the variables over the links can be only ± 1. We are interested to 4-D Ising gauge lattice, so we will have three space-like coordinates and one time-like.

2 Monte Carlo method

The goal of a numerical simulation is to estimate the expectation value of some functions $A[\phi]$ of the field variables $[\phi] = \phi_{x\alpha}$ ($\phi_{x\alpha}$ denotes a real fields component with index α at lattice site x).
This is given by path integral as

$$\langle A \rangle = Z^{-1} \int [d\phi]\, e^{-S[\phi]} A[\phi]$$

$$Z = \int [d\phi]\, e^{-S[\phi]}$$

$S[\phi]$ is the lattice action, which is assumed to be a real function of the field variables. In the case of lattice field systems of interest, the number of integration variables in $[d\phi] = \prod_{x,\alpha} d\phi_{x\alpha}$ is very large.
Apart from trivial system, usually the only possibility is to try a Monte Carlo intgration.
To perform that computation, let's observe the analogy between classical statistical mechanics and lattice QFT. Let's take as example Wilson action of a U(N) lattice gauge field theory with $\phi \rightarrow$U.

$$S[U] = \beta \sum_p (1 - Tr[U_p])$$

As all the typical lattice action, The Wilson's one contain a summation over the lattice. In order to display only the essential statistical mechanics features of the lattice field systems, [**?**] suggest to write the action as

$$S[\phi] = \beta E[\phi] = \Omega \beta \epsilon[\phi]$$

Also Montvay and Müster in [**?**] to rewrite the statistical mechanic's tools in the following way in order to use the analogy with lattice field theory for our Monte Carlo integration.
In statistical mechanics, $\beta = 1/(K_B T)$, but in lattice field theory it's some overall parameter in the action. $E[\phi]$ is the analogous of the energy, instead $\epsilon[\phi]$ is the analogous of the average energy of each plaquette for gauge fields.
As consequence of that analogy, is possible define the density of states $D[\epsilon]$ as follow

$$D[\epsilon] = e^{-\Omega\epsilon} = \int [d\phi]\, \delta\left(\epsilon - \frac{S(\phi)}{\beta\Omega}\right)$$

Here $s(\epsilon)$ is a sort of entropy density. Let's so write the partition function $Z = Z_\beta$

$$Z_\beta = \int d\epsilon D(\epsilon) e^{-\Omega\beta\epsilon} = \int d\epsilon D(\epsilon) e^{\Omega[s(\epsilon) - \beta\epsilon]}$$

This show the probability distribution in ϵ is

$$\rho_\beta = \frac{D(\epsilon)}{Z_\beta} e^{-\Omega\beta\epsilon} = \frac{e^{\Omega[s(\epsilon)-\beta\epsilon]}}{Z_\beta}$$

$s(\epsilon) - \beta\epsilon$ will be the free energy density. Since Ω is sufficiently large, only a small vicinity of the minimum of the free energy density contribute in the path integral. The Monte Carlo simulation have to take this in account by imposing the Boltzmann distribution $exp(-S[\phi])$ to the lattice configurations.
So, for a generical observables O, it's possible compute the mean value as follow

$$< O >= \frac{1}{Z} \sum_i O_i e^{-\beta E[\phi_i]}.$$

The sum is over all the microstates ϕ_i's , $E[\phi_i]$ the internal energy of the microstate ϕ_i, O_i are the value of the observable on the i-th configurations and Z is the partition function, defined as [**?**]

$$Z = \sum_i e^{-\beta E[\phi_i]}.$$

In this way, the computation take into account the Boltmann distribution and all the lattice fields functions are rewritten in terms of statistical tools.

2.1 Markov chain

A Markov chain is defined by the transition probability p_{ij} and by the constraint (probability normalization) [**?**]

$$\sum_i p_{ij} = 1.$$

The result does not depend on the j index because, if there isn't transition, when i=j p_{jj}=1, when there are transitions the sum of all the possible transitions from i to fixed j have to be =1. If we want as stable point of the chain we have to impose the Boltzmann distribution, defined as ... We define $\pi(j)$ as the probability in the Boltzmann distribution. The following condition must be fulfilled:

$$\sum_j \pi(j) p_{ji} = \pi(i)$$

Usually this condition is substituted by another one which is more conservative but simpler to handle:

$$\pi(j)p_{ji} = \pi(i)p_{ij} \quad "Detailed \quad balance"$$

This condition implies the previous one and it is equivalent to ask the chain to be reversible. At this point we must find a way to construct the p_{ij} which have $\pi(i)$ as steady state.

2.1.1 Markov Chain procedure

Starting point: an arbitrary matrix p^0_{ij}. The (still unknown) correct matrix p_{ij} will be obtained from p^0 as follows:

$$p_{ij} = p^0_{ij} a_{ij} \qquad i \neq j$$

$$p_{ii} = p^0_{ii} + \sum_{i \neq j} p^0_{ij}(1 - a_{ij})$$

The idea is to use p^0 to propose a transition from i to j which will be accepted with probability a_{ij} and refused with probability $(1 - a_{ij})$. If the transition is refused we have a "null transition" $i \to i$.
Thus we end up with the following algorithm:

1. Let us start with a configuration i and choose another one j with probability p^0_{ij}
2. Evaluate E_i and E_j (energies of the presente config. and of the proposed one)
3. If $E_j - E_i \leq 0$ (i.e. if we have a gain in energy) we always accept the new configuration
4. If $E_j - E_i \geq 0$ (i.e. if we have a loss in energy) we accept the proposed configuration with probability $e^{-\beta(Ej - Ei)}$

2.1.2 Equilibrium

Usually, in Monte Carlo simulations, "equilibrium" means that the average probability of finding our system in any particular state p_n, is proportional to the Boltzmann weight $e^{-\beta E_n}$. A system at equilibrium spends most of its time in a small subset of states in which its internal energy and other properties take on a narrow range of values.
We might then ask ourself "How can we reach one of these states in the subset"? In [**?**] Newman and Barkema, for Ising model simulations, suggest flipping one spin at time, chosen randomly, until the simulation finds the correct sequence of spins flips to bring the system into one of the desired states. The algorithm must be designed to simulate equilibrium by imposing the Boltzmann distribution.
Since our theory is pure gauge, the variables are defined over the link. Hence, we flip these variables with a probability $e^{-\beta \Delta E}$, where $\Delta E = E_{iniziale} - E_{finale}$. We won't choose the links to split randomly. Thanks to the ergodicity of the system, we can select a fixed sequence for flipping the variables, and this will be equivalent to any random sequence.
For semplicity, we take the link in the same order in which they are stored. Once all the links are flipped with a probability $e^{-\beta \Delta E}$, the code saves the internal energy and repeat this process for all links until equilibrium is reached.

Newman and Barkema [**?**] call the time required to reach the equilibrium τ_{eq}. For 100x100 Ising lattice obtain, they obtained the following graph for magnetization m and internal energy u:

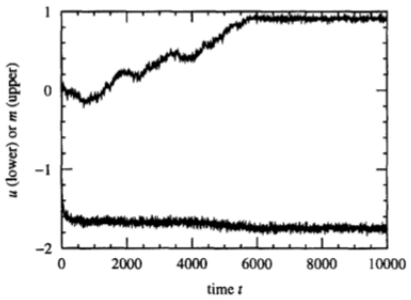

Figure 1: The magnetization (upper curve) and internal energy (lower curve) per site of a two-dimensional Ising model on a square lattice of 100 x 100 sites with J = 1 simulated using the Metropo- lis algorithm

To lighten the code, we don't save internal energy after each individual split, but only after all the links have been processed. We define an "iteration" as the process of flipping all the links once.
A typical graph obtained for the internal energy density (internal energy/ number of links) by using this procedure is the follows:

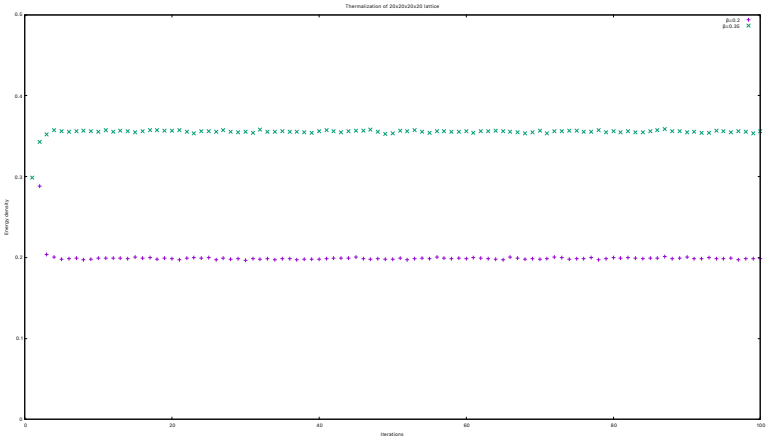

Figure 2: Internal energy density at β=0.2 (purple line) and β=0.35 (green line) for a 20x20x20x20 lattice gauge.

We define this sequence of iteration to reach equilibrium "thermalization".
Newman and Barkema [**?**] also suggest to change the initial condition of the

system, in order to avoid "false equilibrium".
Because of ergodicity, the subset of equilibrium states depend only to the system, so changing the initial condition change only the time to reach on of these states.
We can watch where the internal energy reach the same approximately constant value in two system where only the initial state is different and obtain the τ_{eq}. For semplicity, one of the initial condition we will is the " freeze" one, in which all the variables are 1, and a random one. In the follow graph, we can see the results obtained by 2 systems not correctly thermalized and then 2 correctly thermalized.

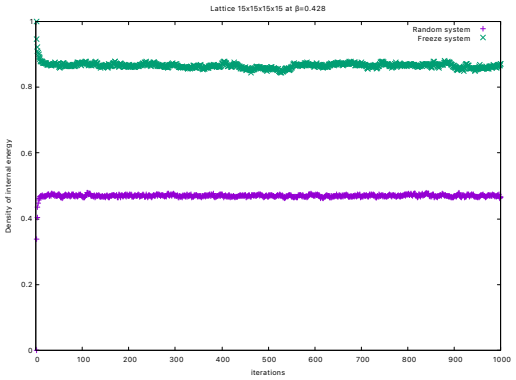

Figure 3: As we can observe, the density of internal energy (internal energy/ number of links) for the same β=0.428 is different if we start from a freeze system (green dots) or a random one (purple dots). At least one of them is in a "false equilibrium".

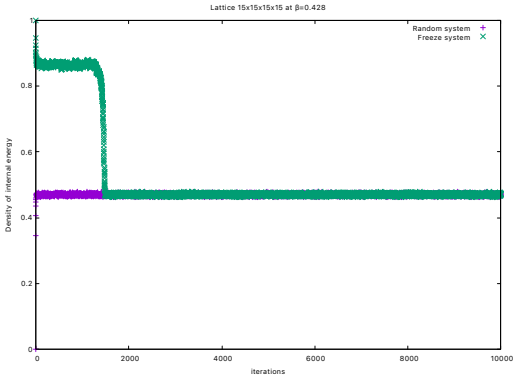

Figure 4: Increasing the iterations of the systems in fig. 3, we can see that the freezed system was in a false equilibrium.

3 Error

The naive computation of the variance requires to calculate $< E^2 >$ and $< E >$ of a certain observable E in order to obtain

$$Var(E) =< E^2 > - < E > \cdot < E >$$

However, to use this formula, the data in the dataset must be completely uncorrelated, a condition that is not entirely satisfied. The code generates a random number, each split of a given link's value is accepted only if the generated number is $< exp(-\beta \Delta E)$ in order to get the Boltzmann distribution. So all the data depend to the pseudo-random number generator in C++ srand(time(NULL)). Because of that, the data in our dataset cannot be completely uncorrelated. Therefore, we applied the blocking method.

3.1 Blocking method

The assumption is that there is a correlation length in the data, so two elements of the dataset separated by a sufficient distance are uncorrelated. To determine this correlation length, we divide our dataset into blocks and compute the variance using the mean of each block as independent data.

What can be observe is that the variance increases as the number of data points in each block grows, up to a certain point, after which it stabilizes. The block size at which this stabilization occurs corresponds to the correlation length.
At this point, we consider the variance to be the true variance, computed using only independent data.
We must strike a balance, if the blocks are too large, we have fewer data points, if the blocks are too small, our data will remain correlated. With the variance obtained, is then possible compute the standard deviation that will be our error by using

$$\sigma = \sqrt{Var(E)}$$

Because of we desire only states at the equilibrium, we have to reject all the data before equilibrium time. For example, in the follow graph we must choose only data after the 1000-th iteration.

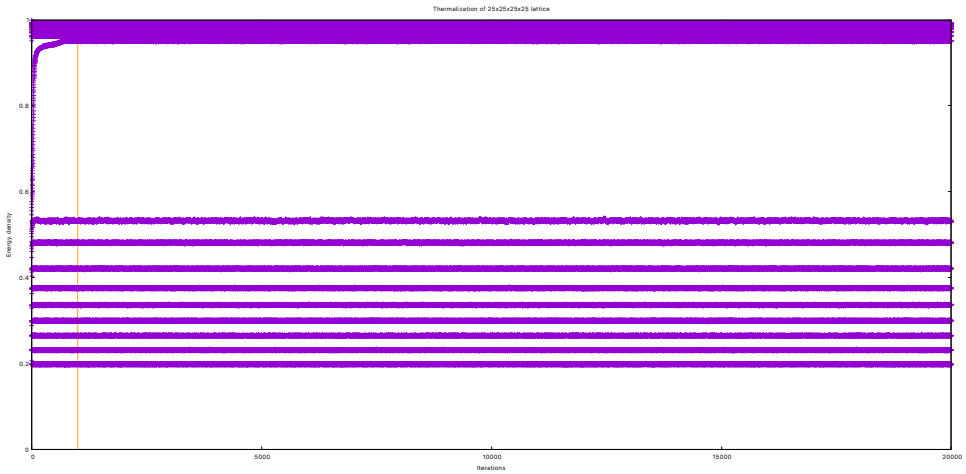

Figure 5: Thermalization of 25x25x25x25 lattice gauge system. We will consider "good data" only the right ones with respect the yellow vertical line which represent the 1000-th iteration.

3.2 Results from blocking

As we can observe in fig. 6, fig. 7 and 8 the standard deviation behaves exactly as we predicted.

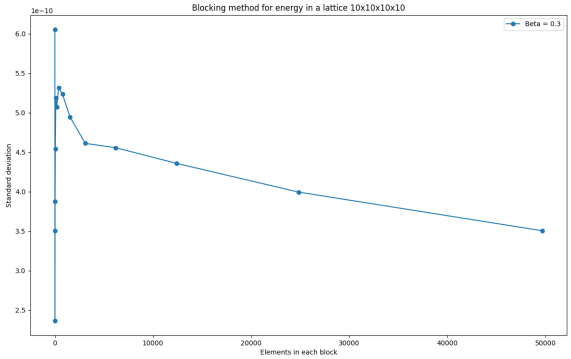

Figure 6: Standard deviation for energy density at $\beta = 0.3$ by using blocking method.

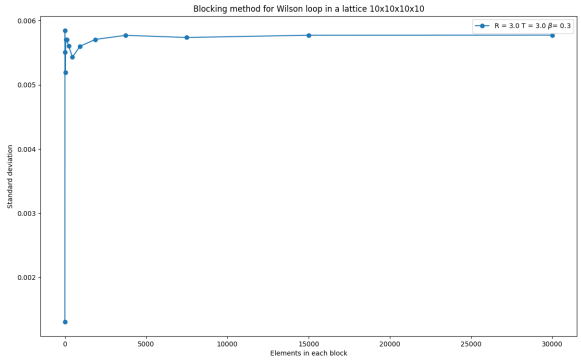

Figure 7: Standard deviation for Wilson loop RxT at $\beta = 0.3$ with time distance T=3 and R=3 using blocking method.

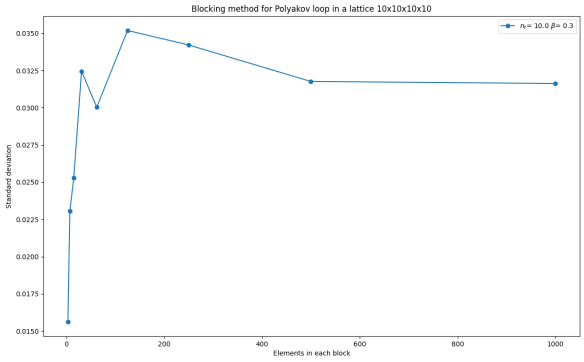

Figure 8: Standard deviation for Polyakov loop at $\beta = 0.3$ by using blocking method.

Too choose the best balance, we used the find_optimal_block command in the pyblock library of Python.
However, find_optimal_block command choose the best balance in the blocks we get from our dataset, if there are problems in our data, as poor statistics, data too much correlated,.. the best block is still not good.
To ensure the goodness of the standard deviation, is important observe if it's properly stabilized by changing the size of the blocks in each dataset.

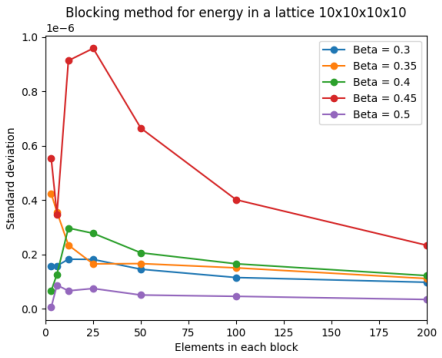

Figure 9: Standard deviation of density of energy in a 10x10x10x10 lattice. Because of the poor dataset, some standard deviations are not yet properly stabilized.

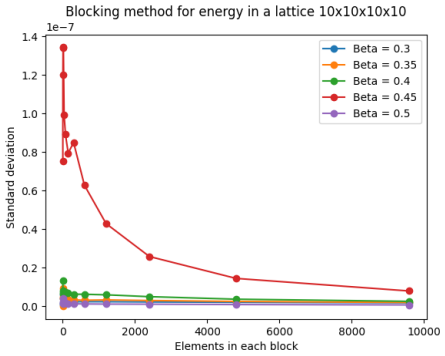

Figure 10: Standard deviation of density of energy in a 10x10x10x10 lattice. In this case we have more data in each dataset than the fig. 9 case. In β=0.45, the standard deviation is higher because it's the dataset in which there is the transition.

As we can see in fig. 9 and in fig. 10, the dataset which contain the transition is characterized to an higher error.
The reason of that are the fluctuations, when we approach the transition, the fluctuations increase (the subset of equilibrium states became bigger) until allow the transition. Those fluctuations are clearly visible in the observables,

which oscillates as a consequence. Now it's clear the reason for the higher variance and higher standard deviation as consequence of the dataset near of transitions than the far ones.
For consider this fact without overly burdening the code, we will increase the iterations near the transition and the number of dataset, in order to have sufficient statistics also after blocking procedure and confining the transition with more precision.

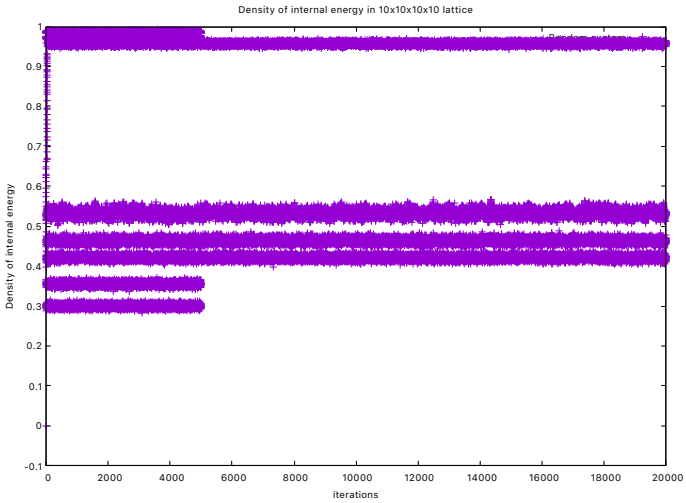

Figure 11: In this graph we can observe the 2 lower dataset (density of internal energy for β= 0.3 and β=0.0.35) which contain 5000 iterations, the richer ones in the middle are β=0.4, 0.425, 0,45, 0.475 and the upper ones β=0.5 and β=0.55.

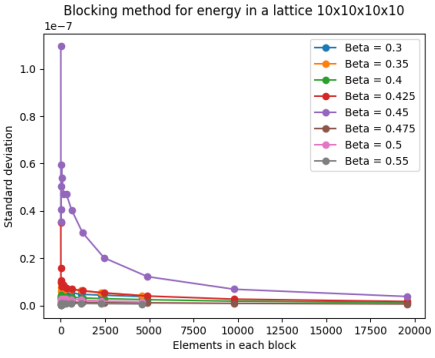

Figure 12: Blocking method applied to the dataset represented in fig. 11.

4 Results

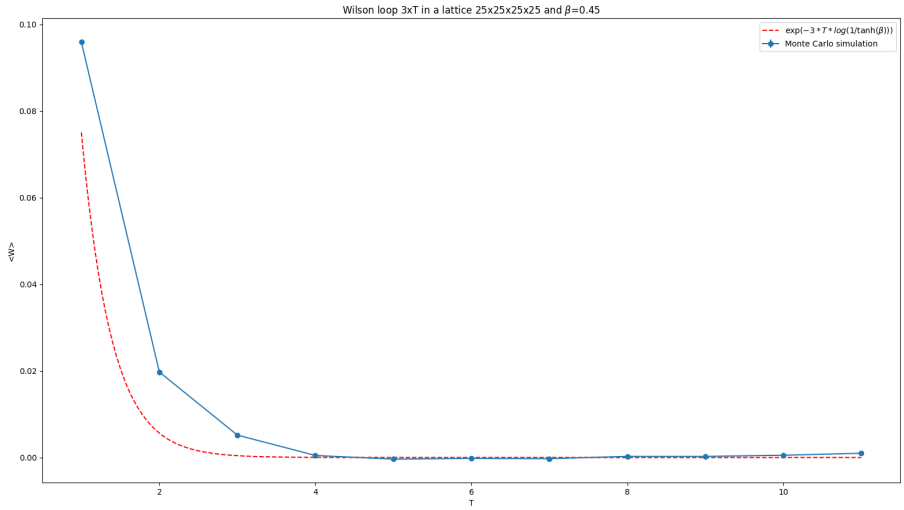

Figure 13: Variance for energy density at $\beta = 0.575$ using blocking method.

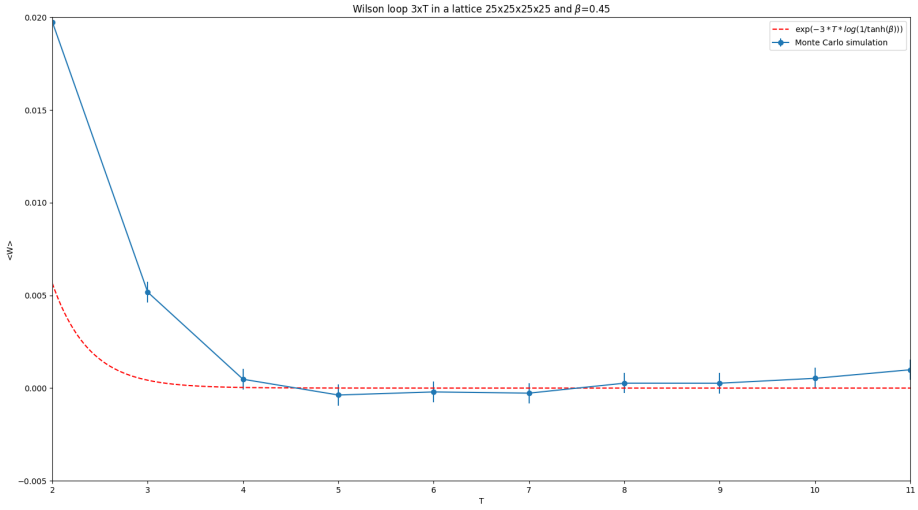

Figure 14: Variance for Wilson loop at $\beta = 0.2$ and time distance T= 9 using blocking method.