Homework 2

Francesco Andreuzzi (andreuzzi.francesco@gmail.com)
Data Science and Scientific Computing – Computational Science
Academic Year 2022-2023

May 2, 2023

1 Exercises (a), (b)

We simulate a 30×30 system of spins for $T = 2$ and $T = 4$. In Figure 1 we plotted the instantaneous value of E/N (energy per spin) and M/N (magnetization per spin). For $T = 2$ we observed that the simulation reaches an equilibrium after approximately 200 MC steps; it takes much less to reach an equilibrium for $T = 4$, the system is subject to intense instantaneous fluctuations but on average it's quite stable around the mean value.

In Figure 2 we plotted the final configuration of the two systems taken into account. For $T = 2$ we observe a prevalence of down spins, while for $T = 4$ the system is much more chaotic.

Finally, we computed quantities based on time averages measured during the simulations. In particular we added to the initial script the calculation of the

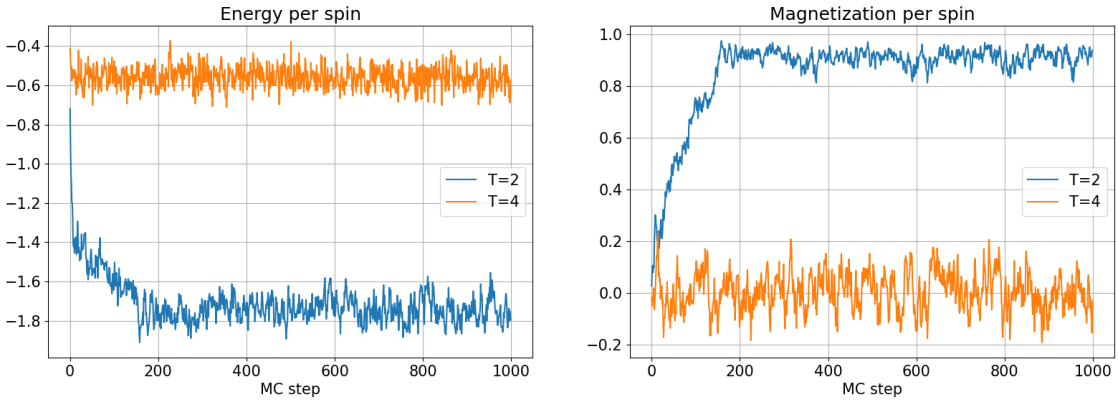

Figure 1: Instantaneous energy per spin (left) and magnetization per spin (right) for initial $T = 2$ (blue) and $T = 4$ (blue).

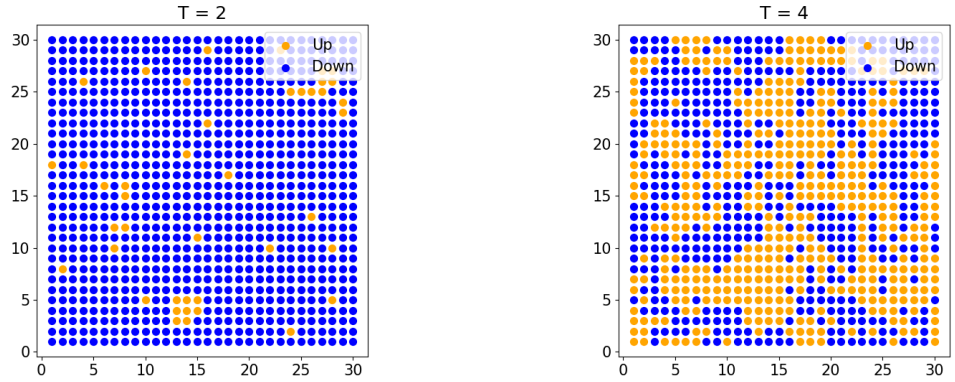

Figure 2: Scatter plot of the final state of the system for $T = 2$ (left) and $T = 4$ (right).

heat capacity c and the *magnetic susceptibility* χ. The results are summarized in the table below:

T	MC Eq. steps	MC steps	c	χ
2	200	1000	0.88	0.43
4	50	1000	0.16	0.99

2 Exercises (d), (e)

In Figure 3 and 4 we run two Ising model simulation with temperature varying between 1.0 and 4.0 (with $\Delta T = 0.5$) and L set respectively to 30 and 4. For each simulation we measured and plotted the time average values of four quantities of interest. In particular for the *thermal capacity* c we've also computed the value as a numerical derivative using the method of *forward finite differences*, in addition to the common interpretation as a function of the fluctuation of the average energy of the system.

We were also interested in identifying the *critical temperature* T_C of the system based on a qualitative examination of the plots, knowing that the estimation coming from the theory is $T_C = 2.26J/k_b$ for $L \to \infty$. In the first case ($L = 30$) the estimation is pretty well respected by empirical measurements, since we can observe pretty drastic changes in both $\langle E \rangle / N$ and $\langle M \rangle / N$, while c and χ are clearly subject of singularities in proximity of T_c^{th}. The situation is quite different in the second case taken into examination ($T = 4$), and this could be also motivated by the fact that the estimation holds only for $L \to \infty$. Indeed the variation for $\langle E \rangle / N$ and $\langle M \rangle / N$ is much less drastic (former) or visible at all (latter).

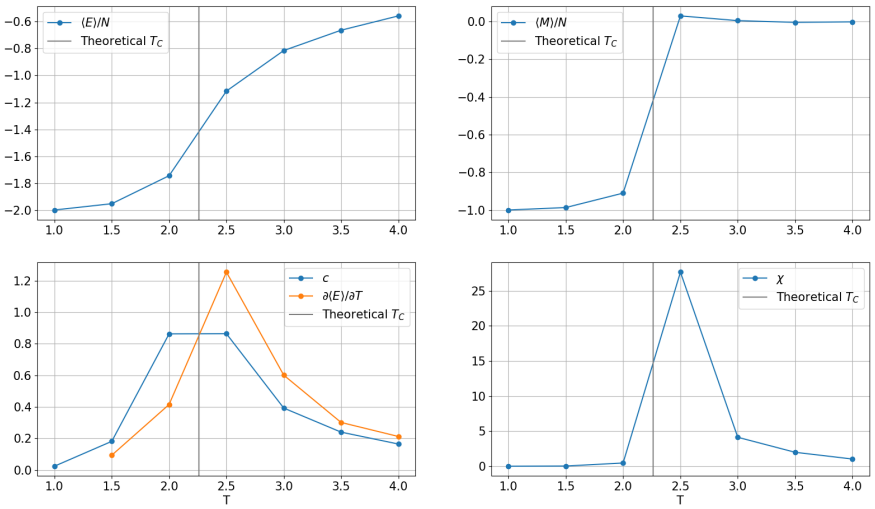

Figure 3: Temperature dependency for four quantities of interest ($L = 30$).

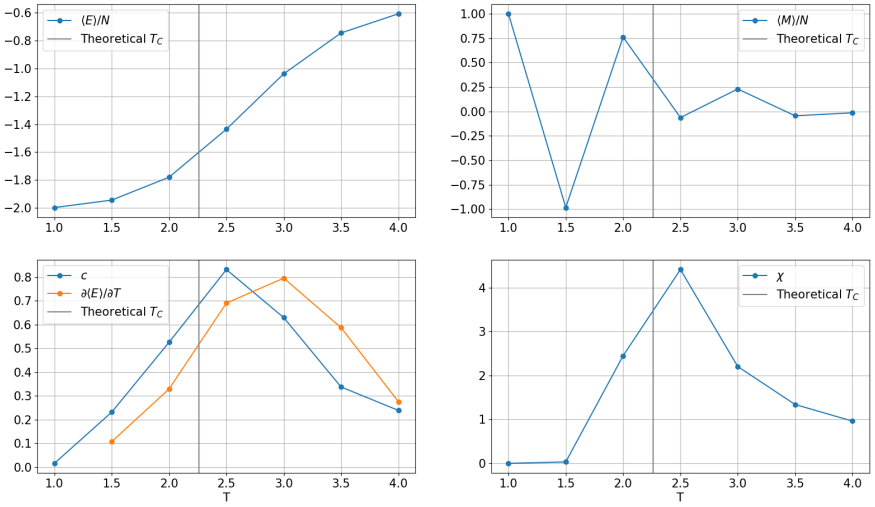

Figure 4: Temperature dependency for four quantities of interest ($L = 30$).

3 Exercise (f)

Before presenting the results we obtained by introducing *open boundary conditions*, we briefly describe our implementation in the simulation code.

3.1 Implementation

In order to implement *open boundary conditions* in the code we slightly changed how boundary conditions were handled. The byproduct of this refactoring was a simplification of the code which handles the original boundary condition in the simulation code (PBC).

The idea is pretty similar to that presented in *Finite-size behavior of the Ising square lattice* by D.P. Landau. The lattice is augmented by a pair of fictitious rows and a pair of fictitious columns which represent the value of the spins on the opposite side of the grid. The idea is better explained with an image: in Figure 5 we plotted two systems with the same parameter as Exercise (b). The semi-transparent dots represent the fictitious boundary, which clearly follows the value of the (solid) spins on the other side of the grid.

In order to keep the fictitious boundary up-to-date with the actual values in the grid it's necessary to change its values appropriately whenever one of the spins in the non-fictitious boundary is switched. For instance, if `spin(1, 1)` was switched we would need to update both `spin(L+1, 1)` and `spin(1, L+1)`.

This new implementations makes it much easier to implement *open boundary conditions*: indeed, it suffices to set the fictitious boundary to 0, such that it does not affect the computation of the energy during the simulation. Moreover, the code looks much simpler (less nested `if` statements) and its apparently faster, even though we haven't conducted an appropriate study on computational performance.

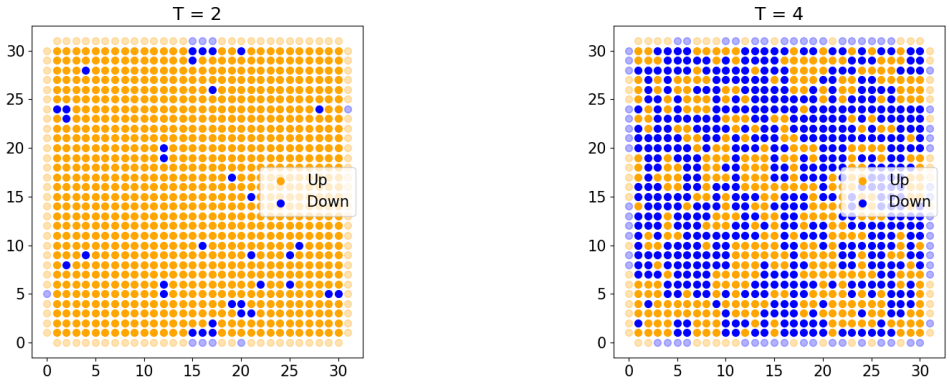

Figure 5: PBC simulation with fictitious rows and columns (semi-transparent) to simulate periodicity.

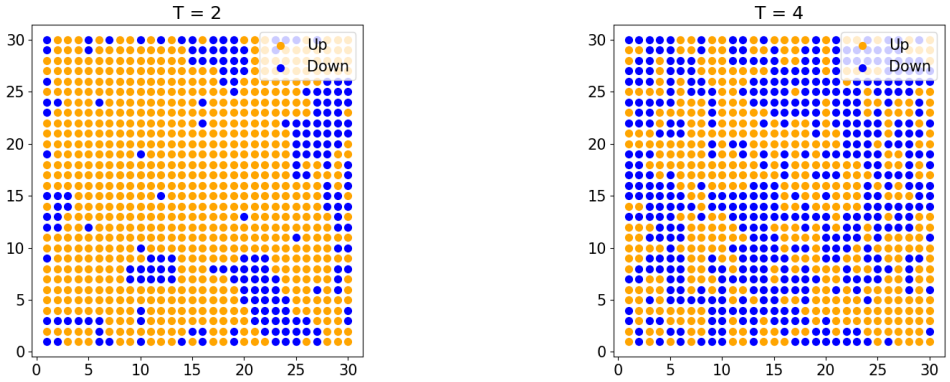

Figure 6: Scatter plot of the final state of the system of Exercise (b) with *open boundary conditions*.

3.2 Open boundary conditions

After implementing the open boundary condition in the code as a CLI flag we ran some simulations. We observed that the final configuration for $T = 2$ and

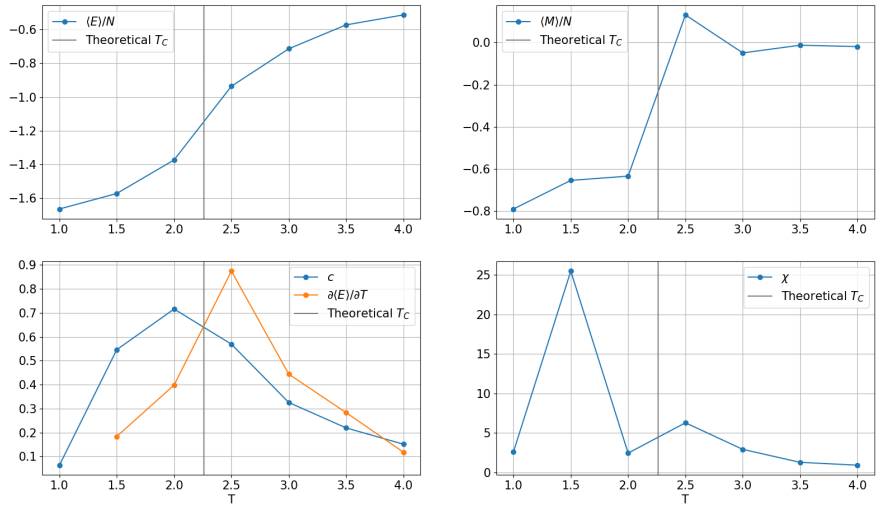

Figure 7: Temperature behavior for four quantities of interest ($L = 30$) with *open boundary conditions*.

$T = 4$ (Figure 6) is somewhat chaotic with respect to the final configuration with PBC.

In Figure 7 we operated the same analysis proposed in Exercise (d) for $L = 30$. We observed that not only the theoretical estimation for the critical temperature is not exactly respected, but there are also some inconsistencies between the position of singularities of c and χ on the temperature axis, thus making identifying the phase transition much more difficult. The reason for this might be the introduction of unphysical phenomena due to the interface effect as a consequence of the open boundary condition.

4 Software

All the simulations in this report were carried out using Fortran 90. Data analysis and visualization were carried out using Python 3 together with the libraries NumPy (for data collection and aggregation when necessary), Matplotlib and scikit-learn. All script files not provided with the delivered homework are available on my GitHub repository for the course *Computational Physics Lab*.