CSP Ex01

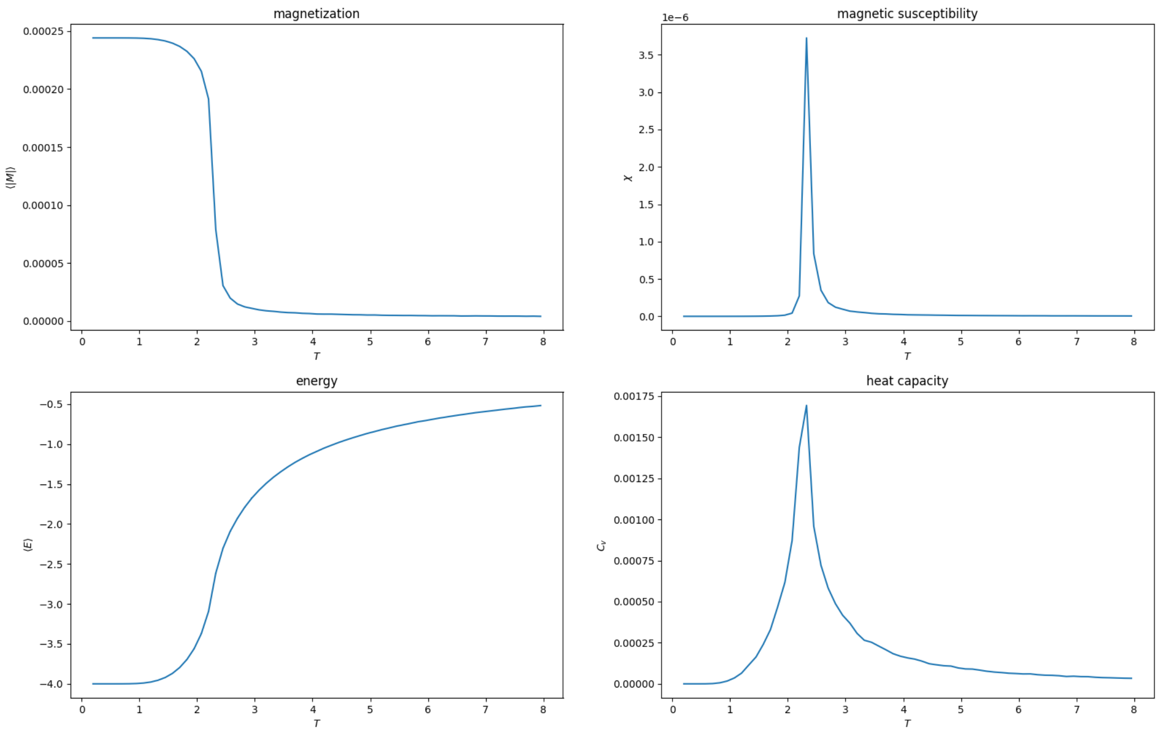
Preface: I created two different code versions: cpp-version (ising2d.cpp) and py-version (ising2d.py). To run the cpp version type: “source run.sh”. For the py-version type: “python3 ising2d.py”.

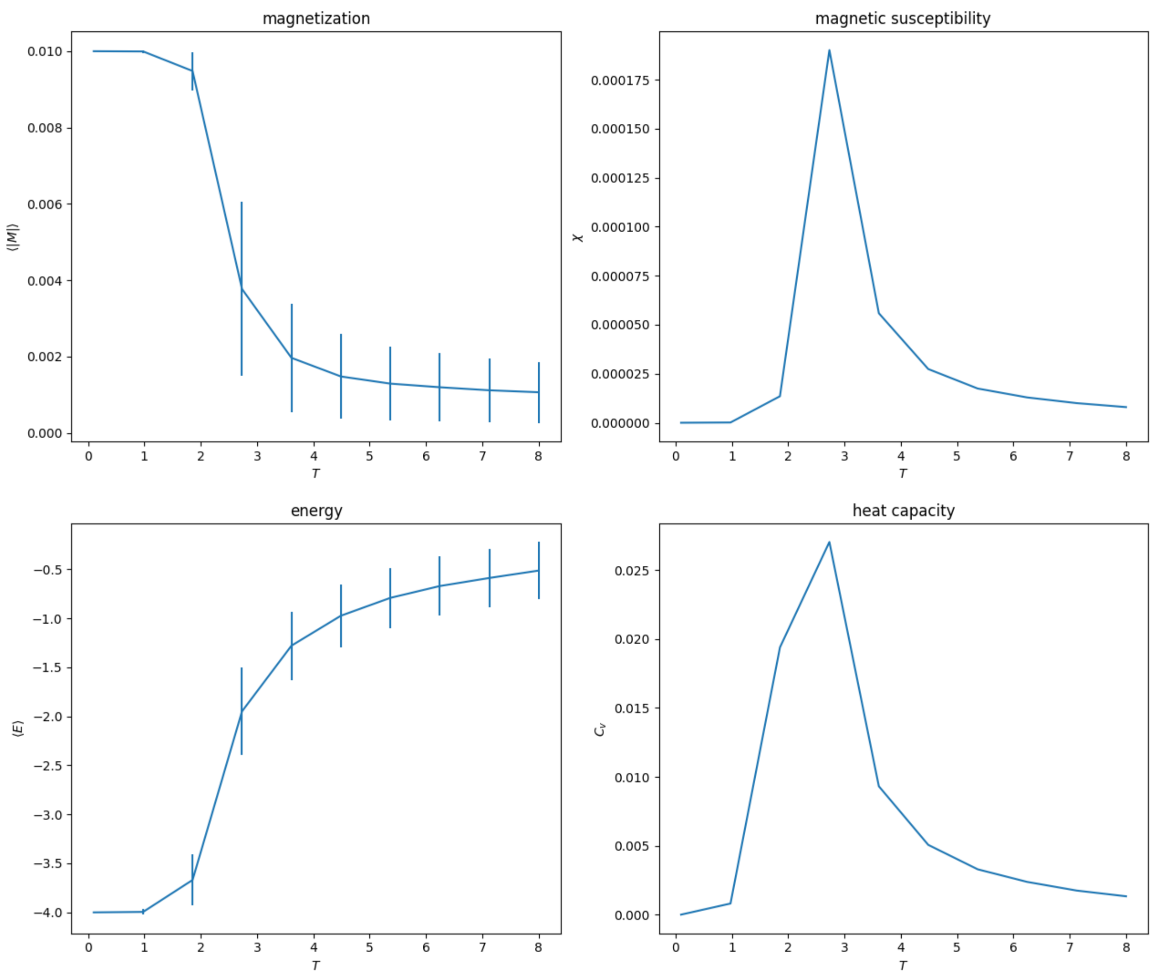
Discussion: The py-script is an optimised version of the given skeleton python script. I tried to improve it as much as possible with my knowledge. Unfortunately only an improvement up to 3x from “intended/simplified implementation”. I did not use a precomputed list of random numbers due to two reasons. 1. The list will get immensely big and would be more of a loading problem than generating. 2. Cpp’s mt19937 is known to be very fast and good enough. If speed is still a problem then I would do a precomputed list. But now, I am happy with the performance (especially with the cpp version). But I precomputed the boltzmann factors. For the cpp version, do you guys know the Eigen3 library? It is very very fast and cool. I used every array like data eigen to store data. The cpp version is literally the same as the python script. The only difference is the execution. We use shell scripting to parallelize the execution with different temperatures. For plotting we used the matplotlib like before.

Task1: The first 4 plots are created with the cpp version. L = 64, Ntherma = 10e9, Nsample = 5000, Nsubsweep = 10N. The last 4 plots are done with my python script with L=10, Ntherma=10e5, Nsample=5000, Nsubsweep = 10N

Task2: In magnetic susceptibility from plot_cpp.png we see it is around T_c = 2.35. I could make a better approx, but it would take longer to run.

Task3: Interesting. If I increase the lattice size, then there is more noise in the system, which can be eliminated with increasing the thermalization time. We also get closer to the critical temperature the bigger the lattice.


magnetization
⟨|M|⟩
T
magnetic susceptibility
1e−6
χ
T
energy
⟨E⟩
T
heat capacity
C_v
T


magnetization
⟨|M|⟩
T
magnetic susceptibility
χ
T
energy
⟨E⟩
T
heat capacity
C_v
T