University *of Ljubljana*
Faculty *of Mathematics and Physics*

Department of Physics

Metropolis-Hastings Algorithm

8. Task for Model Analysis I, 2023/24

Author: Marko Urbanč
Professor: Prof. Dr. Simon Širca
Advisor: doc. dr. Miha Mihovilovič

Ljubljana, July 2024

Contents

1 Introduction

We're continuing our exploration into random numbers and their applications. Previously we had a look at Monte Carlo sampling. Today we're going to delve into the Metropolis-Hastings algorithm, which can be thought of as Monte Carlo sampling with a few extra steps. Since our end goal is to simulate the relaxation of a lattice of spins in a magnetic field, we'll take the Ising model as our physical context.

We know from statistical physics that the 2D Ising model relaxes to a state of minimum energy. We can simulate this relaxation by flipping spins at random and accepting or rejecting the new state based on the change in energy. We can have a negative change of energy which we can call a *good move* or a positive change of energy which we can call a *bad move*. The added twist with this algorithm is that while we always accept the new state if we make a good move, we also sometimes accept a new state after a bad move. This is the key to the Metropolis-Hastings algorithm. Given our system the probability of accepting a bad move is given by the Boltzmann factor and the temperature of the system:

$$P_{\text{bad accept}} = \exp\left(-\frac{\Delta E}{kT}\right) , \tag{1}$$

where ΔE is the change in energy, k is the Boltzmann constant and T is the temperature. Why exactly this works is a bit more involved and probably out of the scope of this report however a dedicated reader can find more information in this well written blog post by Gregory Gundersen of Princeton University [1]. We can define multiple different exit conditions for the algorithm, such as a fixed number of iterations, a fixed number of accepted moves or an ε tolerance for ΔE. Really that is all there is to this method from a theory standpoint.

2 Task at Hand

2.1 Molecular Chain

The instructions given demand that we first explore the Metropolis-Hastings algorithm on a simple molecular chain. The molecular chain is made up of 17 molecules that can go from a state of 0 to a depth of -18. The deeper a molecule is the lower its potential energy. But we also have a positive energy contribution if neighboring bonds are very stretched. The final result is determining the equilibrium energy as a function of the temperature.

2.2 Ising Model

As mentioned in the intro our main goal is to simulate the relaxation of a lattice of spins in a magnetic field. We can describe a ferromagnetic material with the following Hamiltonian:

$$\mathcal{H} = -J \sum_{\langle i,\, j \rangle} s_i s_j - H \sum_i s_i \,, \tag{2}$$

where s_i is the spin of the i-th site, H is the external magnetic field and J is the coupling constant, which is positive for ferromagnetic materials and negative for antiferromagnetic materials. In the absence of a magnetic field the critical temperature for the transition from a paramagnetic to a ferromagnetic state is given by the Onsager solution approximately as:

$$T_c \approx 2.269185 \, \frac{J}{k_B} \,. \tag{3}$$

The instructions demand that we determine the average energy $\langle E \rangle$ and eigen-magnetization $\langle S \rangle$ as functions of temperature. The magnetization of the system is defined as:

$$S = \frac{1}{N} \sum_i s_i \,, \tag{4}$$

where N is the number of spins. We can also have a look at spin susceptibility χ and heat capacity c at different external magnetic field strengths. Spin susceptibility is defined as:

$$\chi = \frac{\langle S^2 \rangle - \langle S \rangle^2}{N k_B T} \,, \tag{5}$$

and heat capacity is defined as:

$$c = \frac{\langle E^2 \rangle - \langle E \rangle^2}{N k_B T^2} \,. \tag{6}$$

3 Solution Overview

This report marks a change in the way I approach these tasks for Model Analysis I. Up until now I've given heavy focus on the quality, optimization and performance of my code and have neglected the actual purpose of the subject, which is the analysis of physical models. The idea was that more advanced code or methods would allow me to do better analysis. However, the overhead of time needed to implement these methods was not worth the benefit. I don't think I've ever managed to show the results of my code since I've always been too focused on the code itself. It's time to change that. As much as it pains me I will try to employ a *good-enough* approach (aka. the *KISS* principle) and focus on the analysis of the models. I will still try to write clean and efficient code but I will not spend too much time on it, especially on paralelization and optimization. From a utilitarian perspective it's better to have a working solution that is not optimal than to have an optimal solution that took to long to implement.

3.1 Molecular Chain

I stared writing the code for the molecular chain quite a long time ago before I burnt out. I threw all that old code away and started anew. As has become a bit of a habit I created a 1D Metropolis-Hastings solver class `Metropolis1` which I can then also reuse for the Ising model with minimal adaptations. The advantage of a class-based approach is that I can more easily do parameter sweeps as I can just create a new instance of the class with different parameters. Since I wanted to get decent statistics for the molecular chain I had quite strict exit conditions and also many repeat runs for each temperature. This produced quite a massive amount of data which I conveniently stored in a `HDF5` file. I ♡ `HDF5`. It is also worth mentioning that I implemented a rudimentary form of **Simulated Annealing** by slowly decreasing the temperature on each good move. The so-called annealing factor was set to 0.9999 by default. Results of the sweep of this parameter are displayed in the results section.

I started of by just running the code and plotting a few results to see if everything is working as expected. A nice result is displayed in Figure (1) where the average of 100 runs is shown for the final state of the molecular chain as well as the scatter of the final temperature and energy across runs where

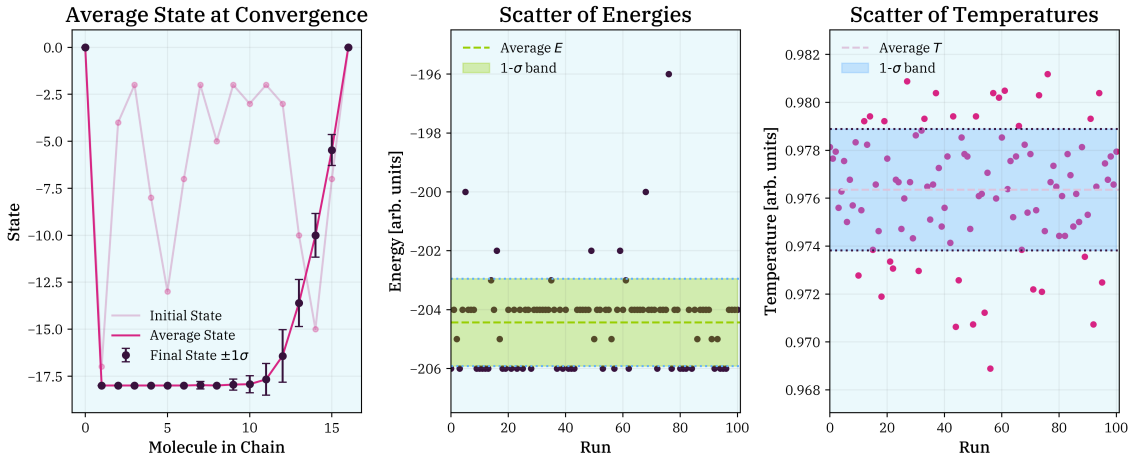

Figure 1: Final state, Temperature, Energy scatter over runs for the molecular chain at a fixed initial state.

the initial state is fixed. The aim of this plot is to display the working of the Metropolis-Hastings algorithm and its convergence to a final state.

It was here that I also experimented slightly with the parameters of the chain, such as length or allowed molecule depth. For example I managed to find a state that optimized almost to a chain link but since one of the molecules was on the top level it tried to make two chain links, as displayed in Figure (2).

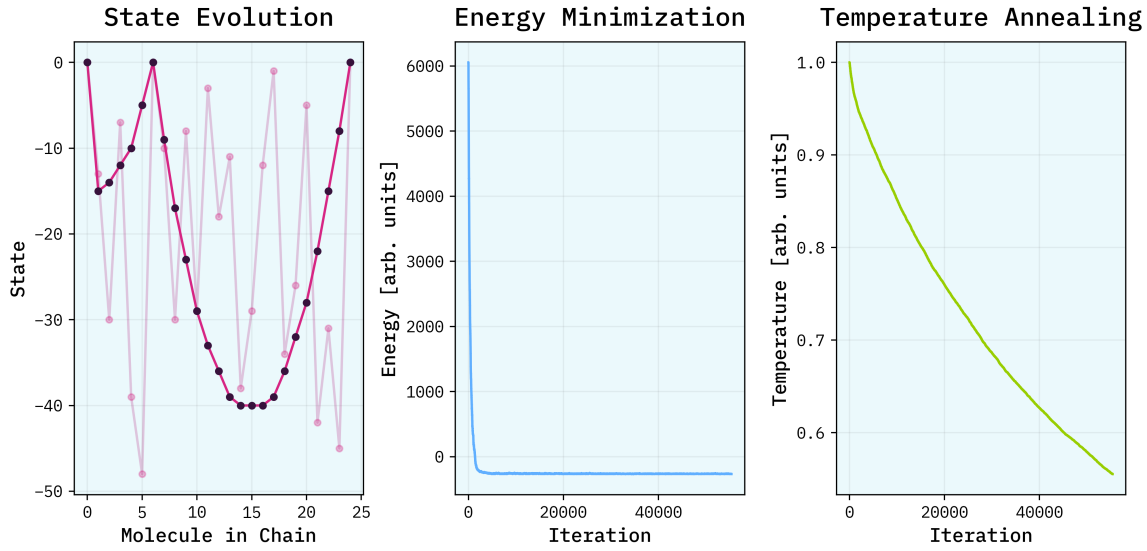

Figure 2: Optimized state of the molecular chain that almost forms a double chain link.

I also trid maximize the *curlyness* of the chain by modifiying the coefficients of the energy function. One of my better attempts is plotted in Figure (3), where the elastic energy term is set to be 10 times larger.

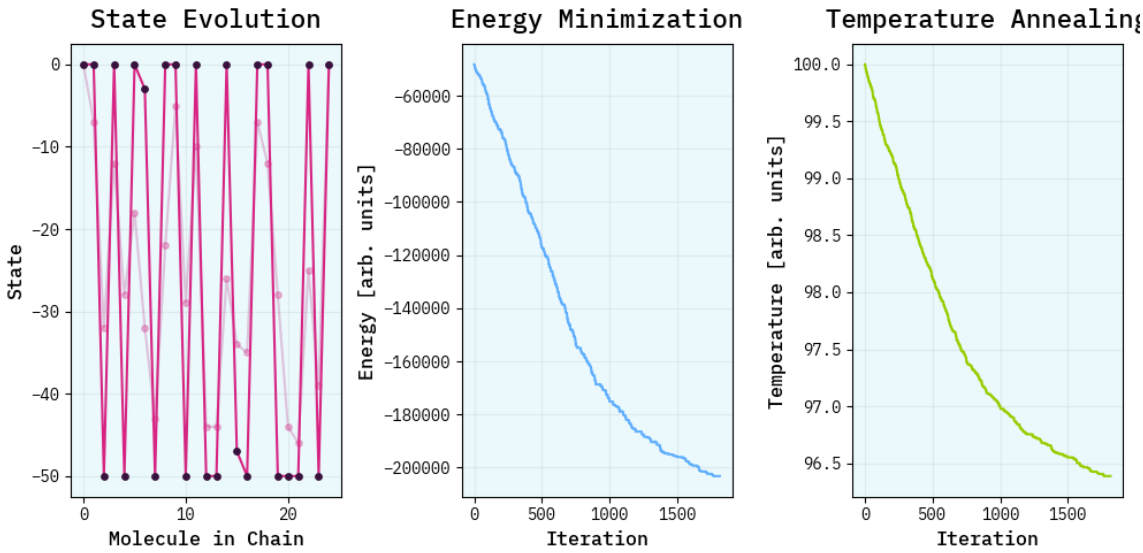

Figure 3: Optimized state of the molecular chain that is maximally curly.

The rest of the results for the molecular chain are displayed in the results section as I think they are more than just intermediate results or random observations.

3.2 Ising Model

After (re)writing the code for the molecular chain I adapted it for the Ising model, creating a solver class `Metropolis2`. I had a few problems with the scale of energy and temperature but I think I managed to solve those in the end and the results seem to be reasonable. Data was also stored in a `HDF5` file, due to the sheer volume as I wanted a decent sample size for my statistics. The main challenge was the code being relatively slow per lattice and I wanted to simulate long times. As much as I promised not to focus on optimization I did to single-computer parallelization by utilizing the `ProcessPoolExecutor` from the `concurrent.futures` module of Python. As stated above, a class-based approach allowed me to easily spawn multiple instances of the solver class with different parameters on different processes which greatly sped up the gathering of data. I think this style of parallelization is reasonable as it takes very little time to implement and yields significant speedups. Here is a brief overview of how `ProcessPoolExecutor` is used:

```
import ...

class Worker:
    def __init__(self,
                 temperature: float,
                 run: int,
                 ):
        self.T = temperature
        self.run = run

    def process(self):
        dim = (50, 50)
        m = Metropolis2(dim, self.T)
        # Set parameters for the simulation
        s_init, s_final, en = m.run()

        return self.run, s_init, s_final, en, m.temperatures

# Create function that'll get the workders to work
def worker_task(params):
    worker = Worker(*params)
    return worker.process()

# Run process pool and save on thread execution
with concurrent.futures.ProcessPoolExecutor(max_workers=16) as executor:
    futures = [executor.submit(worker_task, params) for params in param_list]

    # Step 4: Collect the results
```

```
    for future in concurrent.futures.as_completed(futures):
        run, s_init, s_final, en, temps = future.result()
        # Save data etc.
```

I have to mention somewhere that I accidentally kept Simulated Annealing on for some of the temperature dependance studies, which meant I had to recalculate a lot of data. Anyways, let's get to some images. Like before some debug style plots at the beginning definitely do not hurt. I wanted to get a feel for where the critical temperature is and how long it takes to reach equilibrium. For the purpose of nicer images I ran the simulation for a 500×500 lattice, as the professor commented that this size would give nice-looking results. Figure (4) shows the initial and final states of the lattice at a subcritical temperature of $T = 1$.

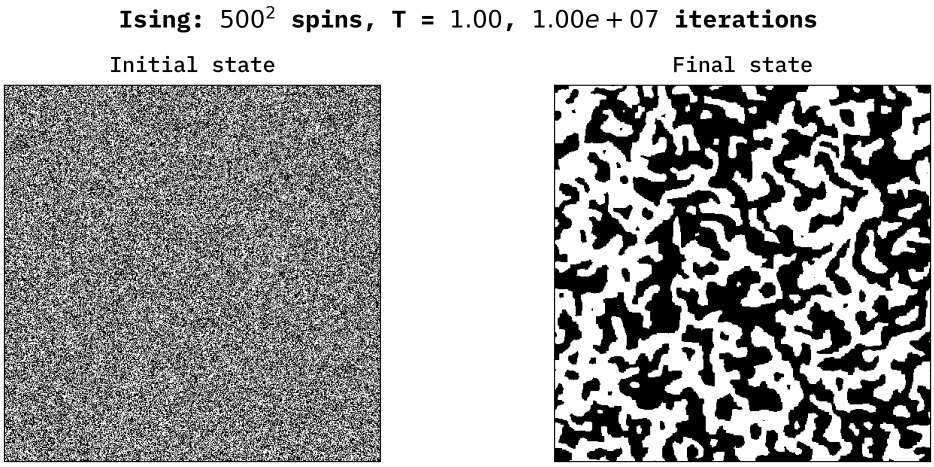

Figure 4: Initial and final states of the Ising model lattice at a subcritical temperature of $T = 1$.

We can see that the lattice is in a ferromagnetic state and that magnetic domains have nicely formed. The lattice is drastically different from the initial state. It now only makes sense to take a look at a supercritical temperature of $T = 10$ which is displayed in Figure (5).

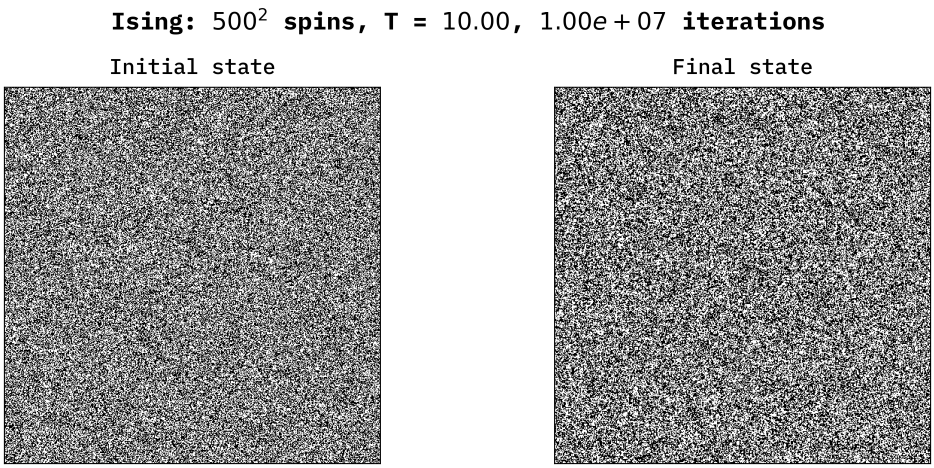

Figure 5: Initial and final states of the Ising model lattice at a supercritical temperature of $T = 10$.

The lattice looks very simmilar to the initial state which is expected as we do not expect spontaneous magnetization at supercritical temperatures. I was also interested in how the lattice behaves at the critical temperature, which is forcasted by the Onsager solution (3) to be $T_c \approx 2.269185$. The result is displayed in Figure (6).

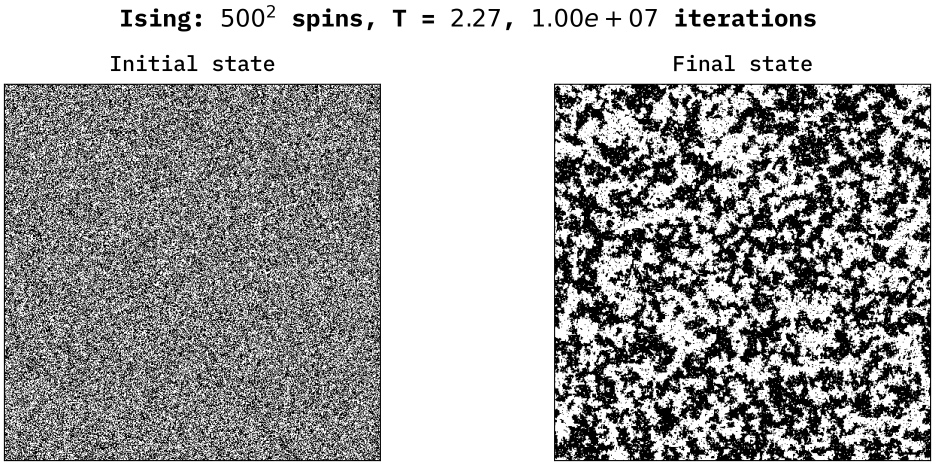

Figure 6: Initial and final states of the Ising model lattice at the critical temperature of $T \approx 2.269185$.

The lattice is sort of a mix between the subcritical and supercritical states, which is exactly what we expect. Visually it looks like a grainyer version of the ferromagnetic state. As a final little debug plot I wanted to see what a state would look like if I allowed more than just $|\uparrow\rangle$ and $|\downarrow\rangle$ states. As an attempt at a generalization to $s = 3/2$ particles I allowed for states to go from -2 to 2. The result of optimizing such a lattice with the same Hamiltonian at a subcritical temperature is displayed in Figure (7).

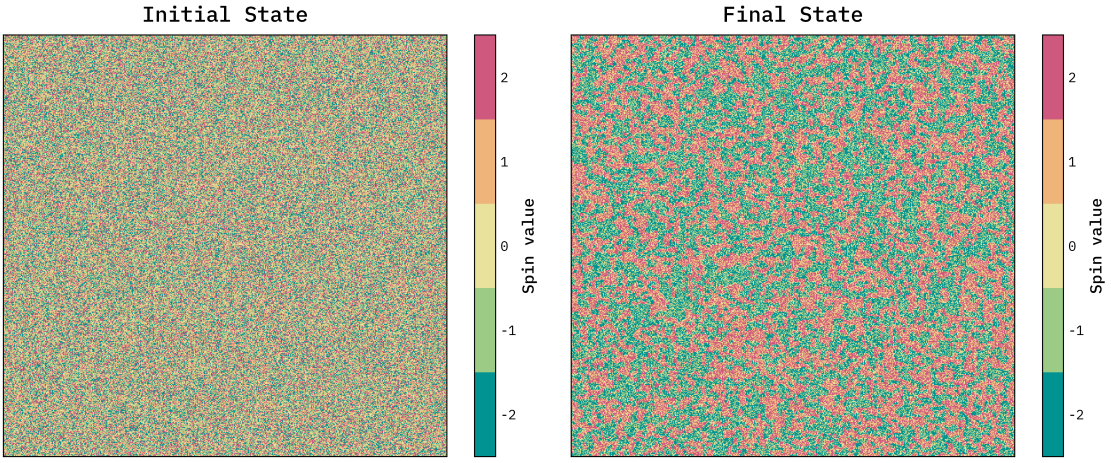

Figure 7: Initial and final states of the Ising model lattice with $s = 3/2$ particles at a subcritical temperature of $T = 1$.

The lattice looks very similar to the $s = 1/2$ lattice however with more variation. I find it interesting that patches of states 2 and -2 surround themselves with states 1 and -1. I have no idea if this model has any real life applications however. I wish I had enough time to explore this further. Analogous to $s = 1/2$ particles we can expect interesting behavior at the critical temperature. The result is displayed in Figure (8).

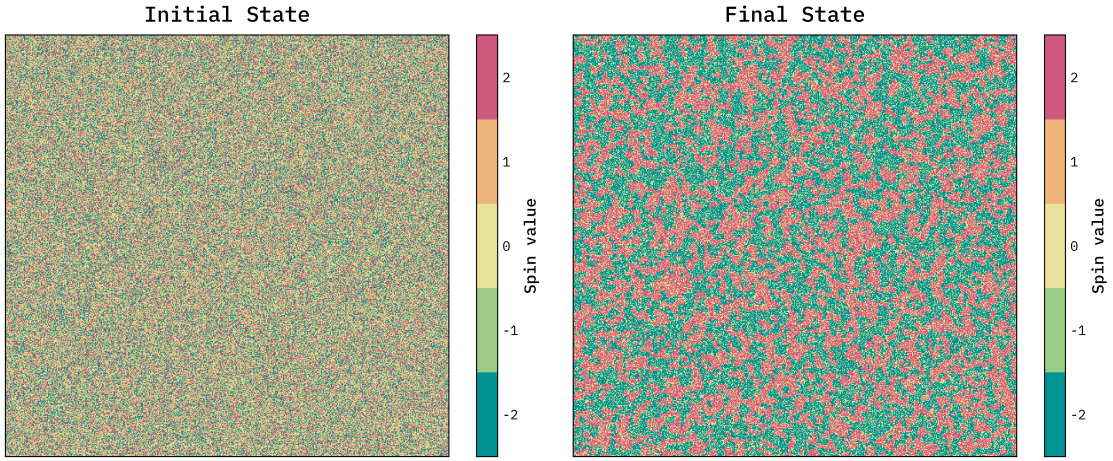

Figure 8: Initial and final states of the Ising model lattice with $s = 3/2$ particles at the critical temperature of $T \approx 2.27$.

I'm not going to lie, the results look very simmilar to the subcritical temperature lattice with $s = 3/2$ particles. Maybe even less *dusty* which is odd because I would expect the opposite. Due to a lack of time I did not have the liberty of exploring this further by, for example, gathering more data and doing a statistical analysis instead of eesentialy an empirical one. For completeness I also ran the simulation at a supercritical temperature of $T = 10$ which is displayed in Figure (9) where we get exactly what we expect, random noise more or less. Although it does appear somewhat more structured.

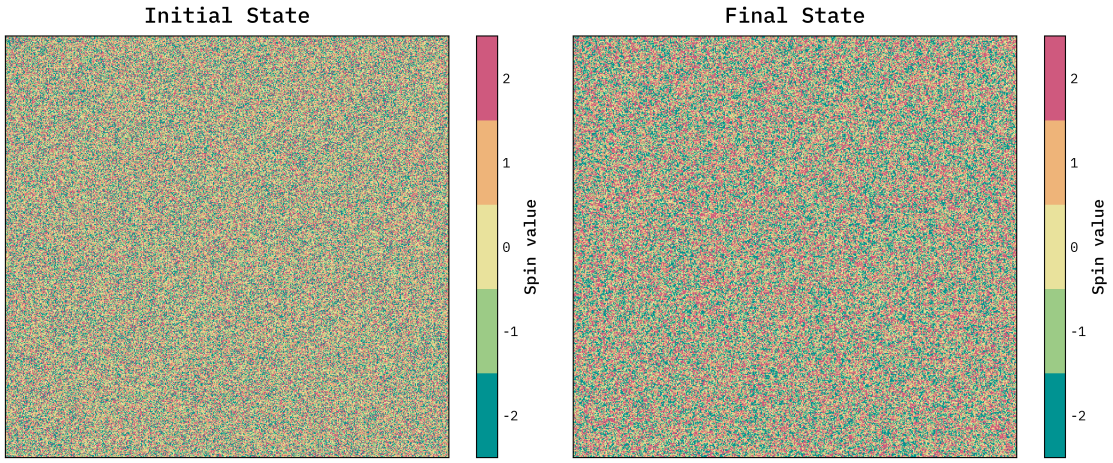

Figure 9: Initial and final states of the Ising model lattice with $s = 3/2$ particles at a supercritical temperature of $T = 10$.

Since I already had the code at hand I also further expanded the state space to include the range -4 to 4 which would by my previous logic be $s = 7/2$ particles for which I found some interesting papers [2]. The results are displayed in Figure (10) for a subcritical temperature of $T = 1$.

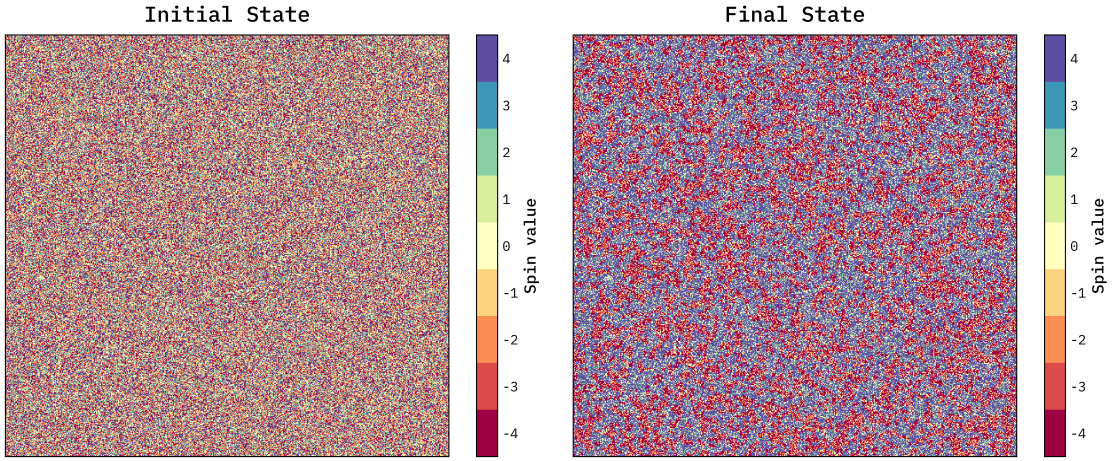

Figure 10: Initial and final states of the Ising model lattice with $s = 7/2$ particles at a subcritical temperature of $T = 1$.

The lattice looks very similar to the $s = 3/2$ lattice however with even more variation. So much so that the fine detail is hardly discernable. I also ran the simulation at the critical temperature of $T \approx 2.27$ which is displayed in Figure (11). I left this simulation to run for very long 10^8 steps at a reduced size of 100×100 to see what happens when the lattice is allowed to really relax. The smaller lattice also helps see the fine detail. The result is a nice pattern of what I'm guessing could be called magnetic domains. I find it fascinating that the domains are almost entirely made of 4 and -4 states. As a matter of fact there are little other states present besides a scatter of states 0. This *extremization* to the edge states is exactly why I let the simulation run for so long.

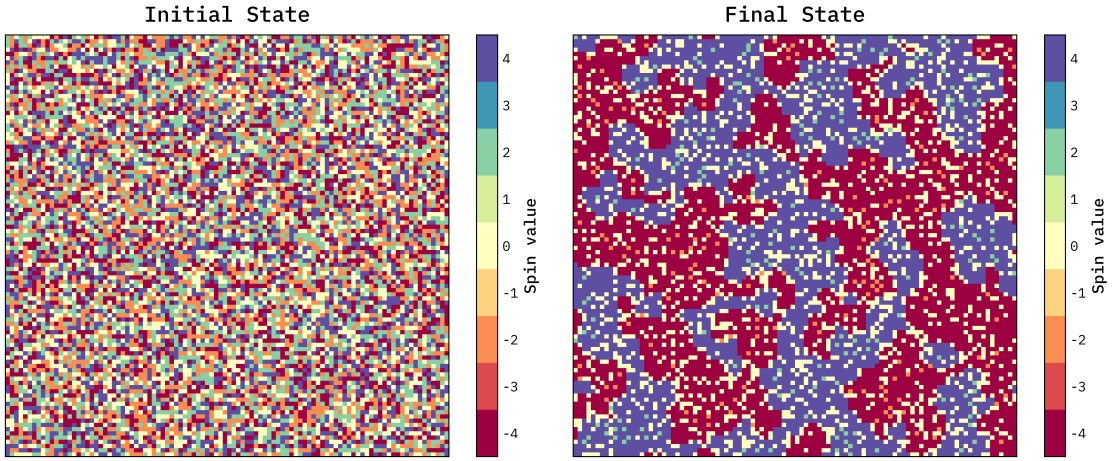

Figure 11: Initial and final states of the Ising model lattice with $s = 7/2$ particles at the critical temperature of $T \approx 2.27$.

And of course how could one forget the anti-ferromagnetic phase (where we set $J = -1$) which is displayed in Figure (12). The lattice is in a checkerboard pattern which is exactly what we expect.

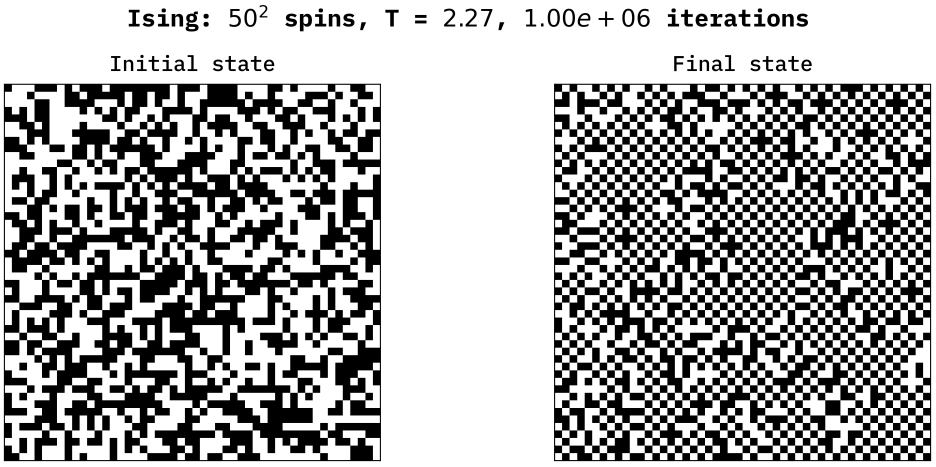

Figure 13: Initial and final states of the Ising model lattice in the anti-ferromagnetic phase at a critical temperature of $T \approx 2.27$.

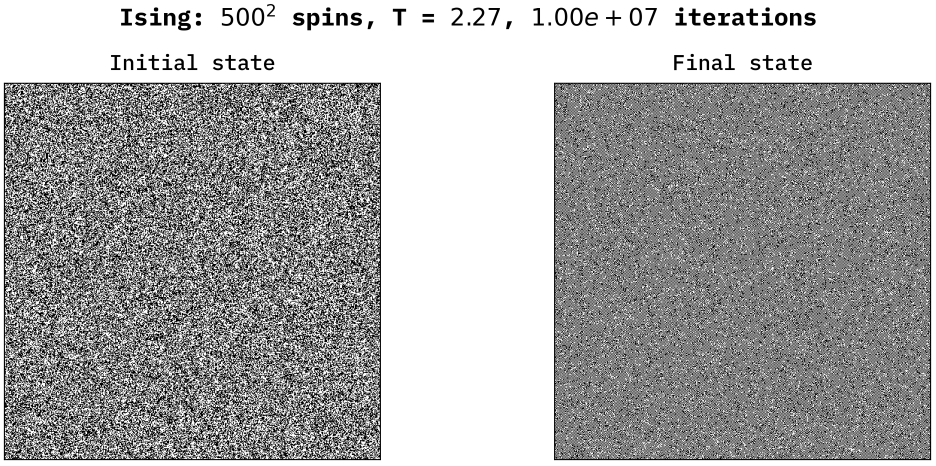

Figure 12: Initial and final states of the Ising model lattice in the anti-ferromagnetic phase at a critical temperature of $T \approx 2.27$.

This image is nice to see the omni-presence of the checkerboard pattern however I plotted a smaller lattice of 100×100 spins to make the pattern more visible. This can be seen in Figure (13).

The rest of the relevant plots for the Ising model are displayed in the results section.

4 Results

4.1 Molecular Chain

The instructions of the task demanded that we explore the temperature dependance of the equilibrium energy, which is presented in Figure (14). For better statistics due to the stochastic nature of the Metropolis-Hastings algorithm I ran the simulation 100 times for each temperature. This resulted in 10000 data sets of multiple thousand points each, hence the need for `HDF5` storage. As we can see from the image the equilibrium energy of the molecular chain decreases with temperature as is expected. The standard deviation of the energy also somewhat decreases with temperature, which is likewise to be expected. I thought it would be interesting to also see how many steps of the Metropolis algorithm were needed to reach my convergence criteria which was an ε tolerance of 10^{-14} averaged out over the last 50 steps. Below are also spreads of the final temperature and energy across the runs. I thought it was a neat alternative way to visually display the scatter of the data for each temperature.

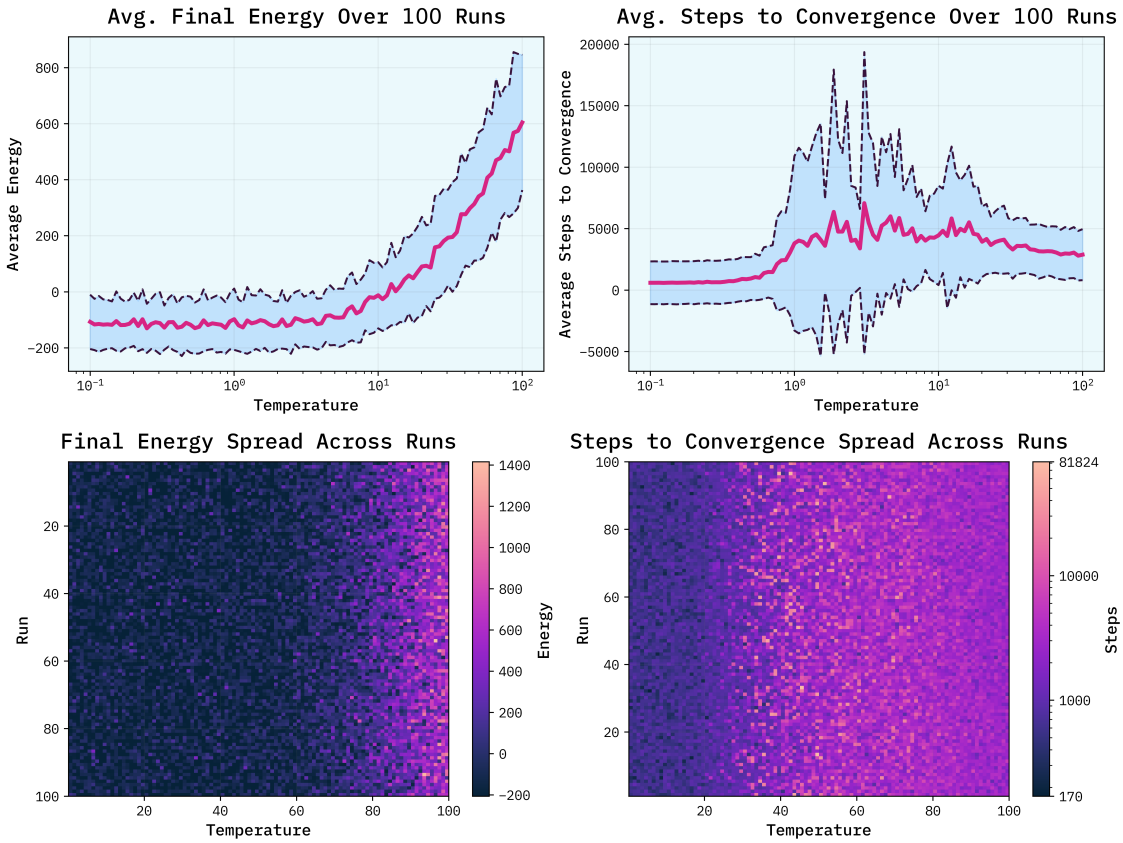

Figure 14: Equilibrium energy of the molecular chain as a function of temperature.

Since I had the data already gathered it was very simple to plot the average shape of the molecular chain across all temperatures. I also tried to display the average shapes over repeat runs for each temperature. The results are displayed in Figure (15). It is interesting to note albeit expected that the shape optimizes best at medium to low temperatures. At high temperatures the chain can still jump around a lot and at low temperatures it can get stuck in local minima. Overall though we get quite a well defined shape that is close to the optimal. I find it insteresting that the left portion of the chain is always well defined while the right position has some uncertainty.

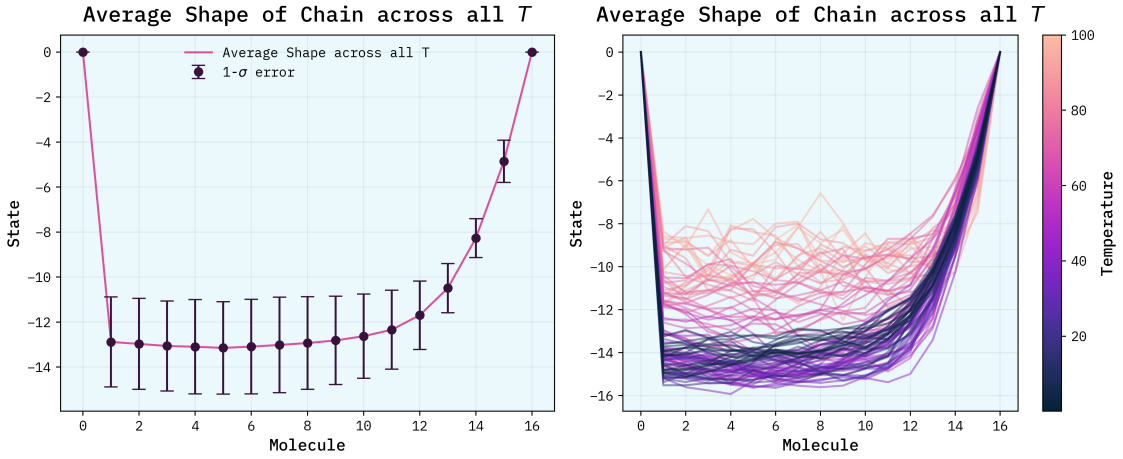

Figure 15: Average shape of the molecular chain across all temperatures and average shapes over repeat runs for each temperature.

As promised I also ran a sweep of the annealing factor in combination with the temperature. The results are displayed in Figure (16).

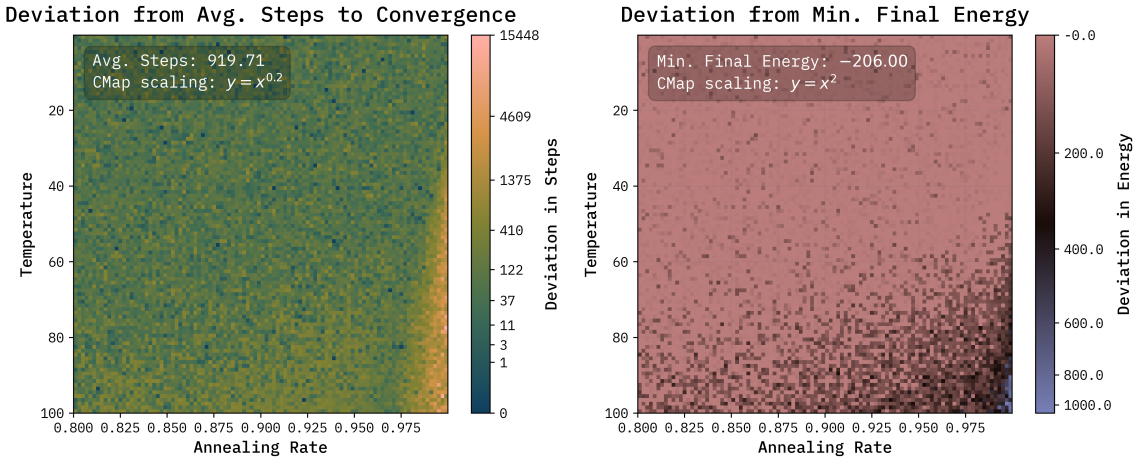

Figure 16: Sweep of the annealing factor in combination with the temperature.

From the plot we can see that very high annealing factors combined with high temperatures are not very effective as they take a long time to converge and produce results that are not the most optimal.

4.2 Ising Model

I find the images or rather patterns generated by the Ising model to be quite beautiful. The instructions demaded we explore the temperature dependance of the average equilibrium energy and magnetization of the lattice. The results are displayed in Figure (17), where we're again looking at averages over 100 runs at each temperature. I'm not sure whether it is due to an error or just the nature of numerics but the tranition from a paramagnetic to a ferromagnetic state is not quite as pronounced as I would have expected. It is quite drawn out as a matter of fact. The rest is more or less as expected. We can clearly see the appearence of spontaneous magnetization below the critical temperature.

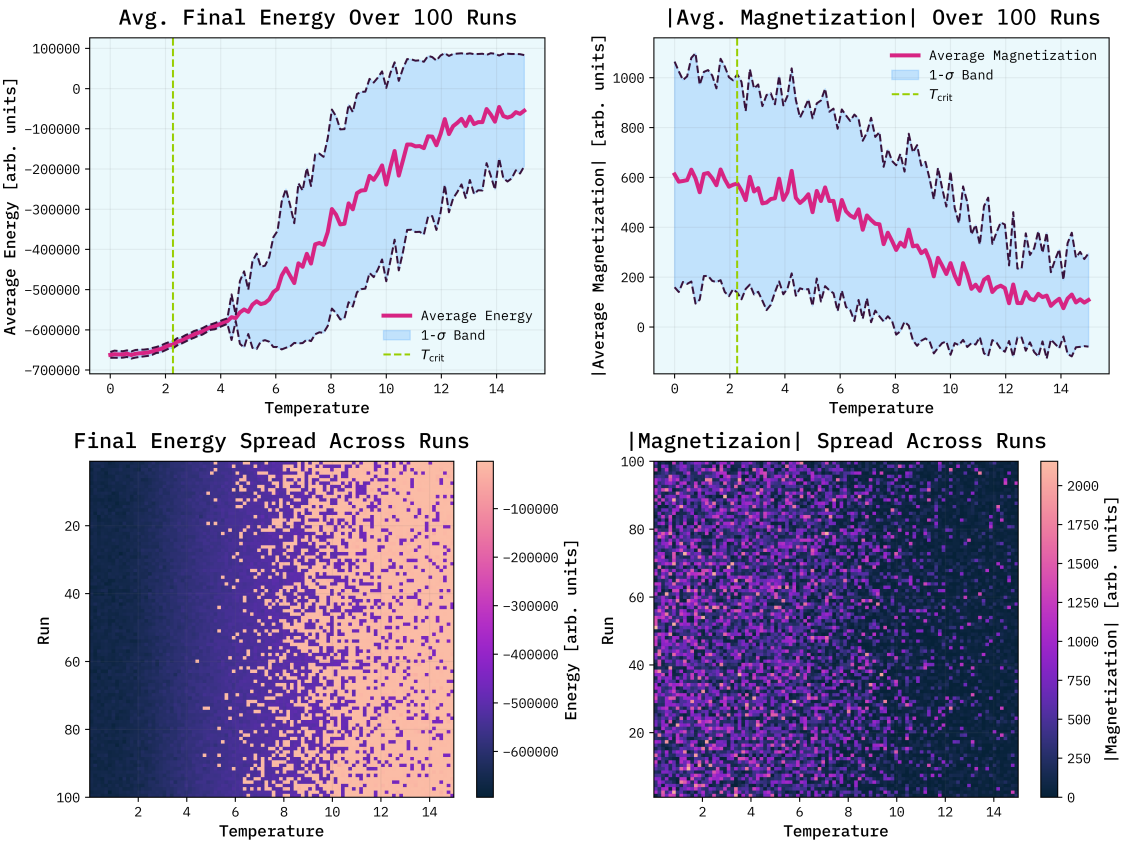

Figure 17: Temperature dependance of the average energy and magnetization of the Ising model lattice.

I thought it would be interesting to also see the steps needed for convergence for each temperature, which is displayed in Figure (18). The number of steps needed for convergence is quite high at low temperatures which is to be expected.

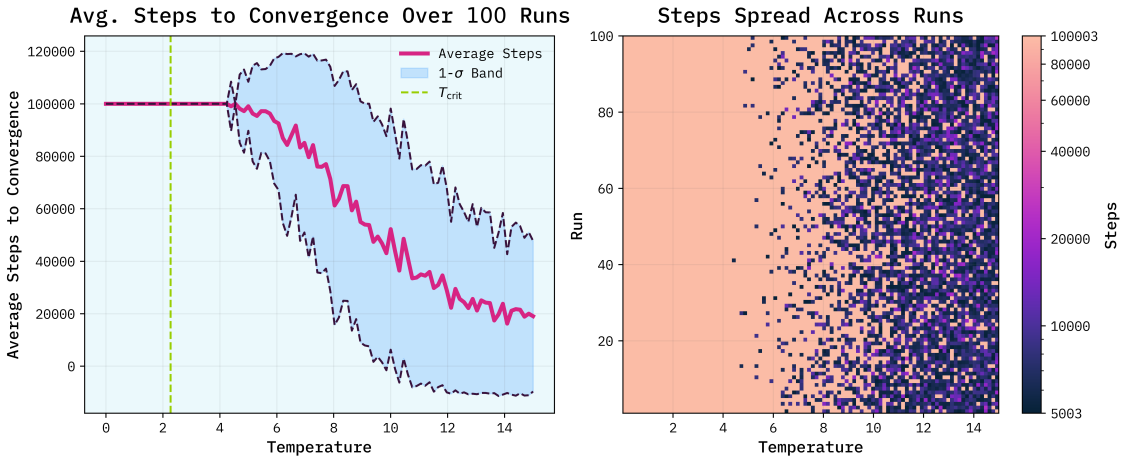

Figure 18: Number of steps needed for convergence for each temperature.

I also thought it would be interesting to see the temperature dependence of the average equilibrium energy and magnetization for my attempt at a generalization to $s = 3/2$ particles. The results are displayed in Figure (19), where the averages are now over 10 runs at each temperature. I find the dip in equilibrium energy around the critical temperature particularly interesting and I cannot find a suitable explaination for it.

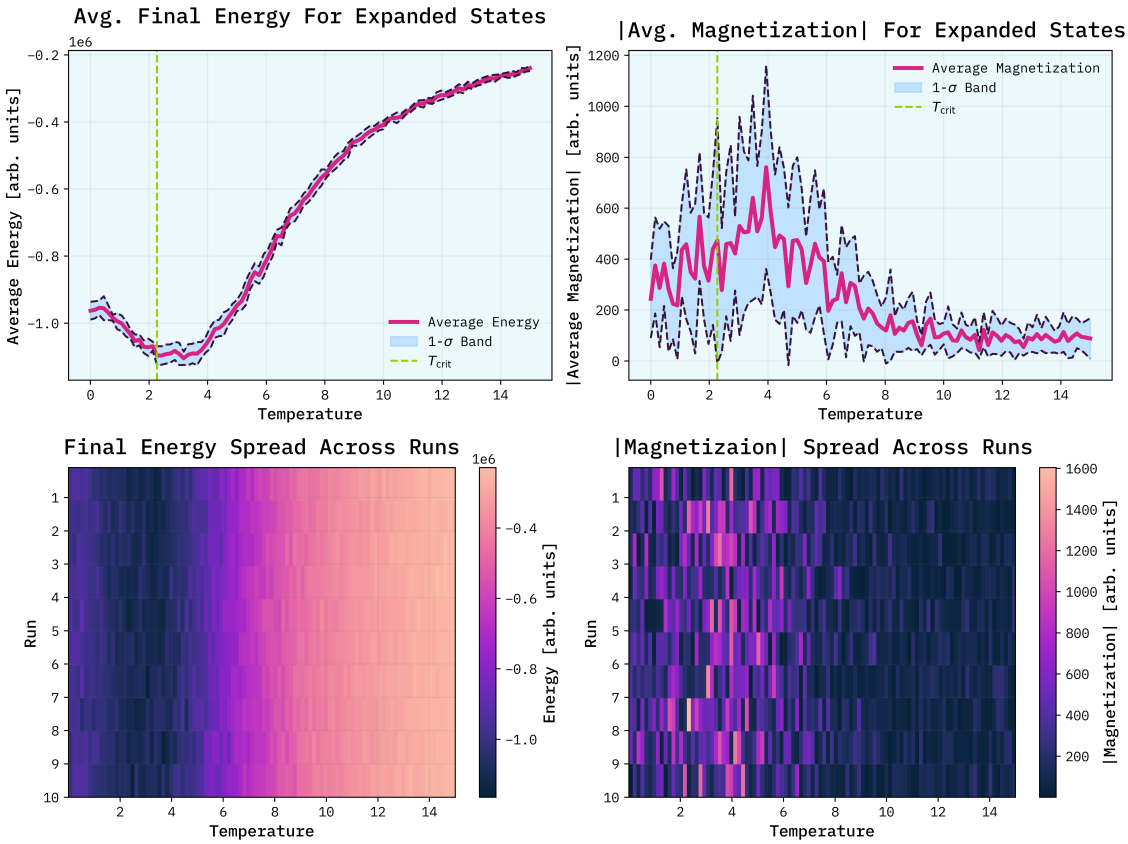

Figure 19: Temperature dependance of the average energy and magnetization of the Ising model lattice with $s = 3/2$ particles.

And as the final result, per the instuctions, I also explored the temperature dependance of the spin

susceptibility and heat capacity of the lattice at different external magnetic field strengths both for $s = 1/2$ and $s = 3/2$ particles, which is plotted in Figure (20). I expected smaller magnetic field strengths to produce more pronounced results hence the clumps of data on the lower end of the magnetic field strength scale. What I find fascinating is the fact that the $s = 3/2$ and $s = 1/2$ particles produce nearly identical results. I'm almost sure this is due to an error somewhere or maybe it is just the nature of statistical variance which may not necessarily be intuitive.

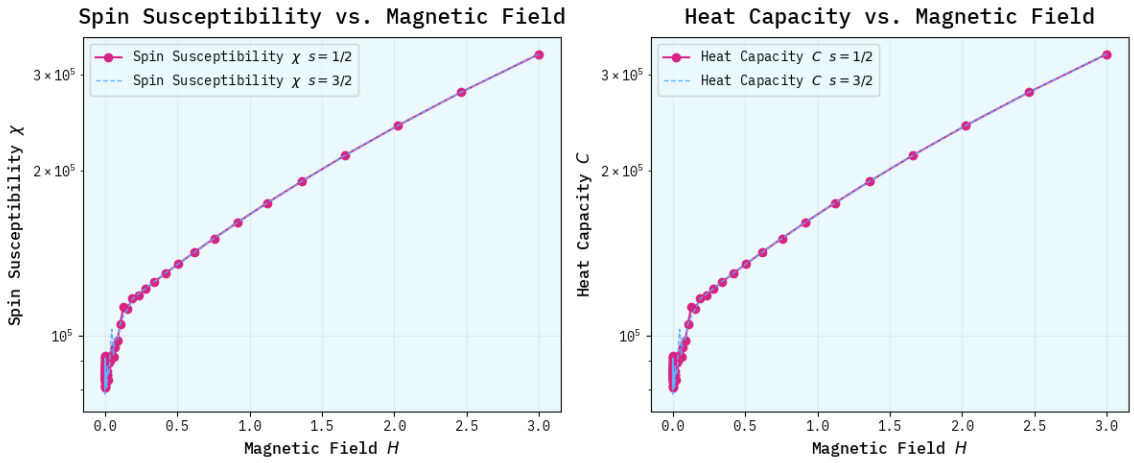

Figure 20: Temperature dependance of the spin susceptibility and heat capacity of the Ising model lattice at different external magnetic field strengths.

5 Conclusion and Comments

Man this report took way longer to write than I expected. I've just read online that my attempt at a generalized Ising model is close to what is known as the **Potts** model, however I kept the Ising model's simple Hamiltonian which still lead to some interesting results. If I had more time I would definitely explore this topic further as I find it quite fascinating. Especially the fact that nature really does tend to order itself into neat patterns. Or maybe we think they're neat because they are a part of nature. The way I wanted to explore many posibilites of the models really made me generate quite a lot of data. I hope this report is a bit more in the spirit of the subject and less about the technical approach, although I did include that short segment on thread pools. I'm quite happy with the results I got and I think I managed to explore the models quite well.

References

[1] Gregory Gundersen. Why Metropolis-Hastings Works. https://gregorygundersen.com/blog/2019/11/02/metropolis-hastings/#bishop2006pattern, Nov 2019. [Accessed 24-07-2024].

[2] Vladimir Yu. Osipov, Danil W. Boukhvalov, and Kazuyuki Takai. Isolated spin-7/2 species of gadolinium (iii) chelate complexes on the surface of 5-nm diamond particles. *Nanomaterials*, 13(13), 2023.