Simulationsmethoden 1 January 31, 2011

Tutorial 6: *The Ising model*

Michael Kopp (2439093)

1 ising2d

I wrote my own program `ising2d.cpp`. Run `./ising2d -h` to get an overview of its features.

It uses a class `board` which provides a simple 2D array of integers – either 1 or -1 as required by the Ising model.

If a board-instance is created, the board is given a specific size ($L \times L$) and the spins[1] are set to either 1 or -1 – depending on some possibility $\_deg \in [0, 1]$ which is (probably[2]) the ratio of spins equal to 1 to spins equal to -1. Finally since the spins are set randomly, one can pass a seed to the newly created board, thus allowing for several different initial settings.

To initially calculate the energy of the system, the function `evaluate_energy()` iterates over every field (with value σ) and computes the interaction energy[3] $h = -\sigma \cdot \sigma'$ with it's top and left neighbour (with value σ'). Periodic boundary conditions are assumed thus the furthest to the left and the uppermost lines are treated separately: In their case they interact halfway with the fields on the opposite boundary.

The real energy is the sum of all h as h only regards every interaction exactly once.

To check the results of `evaluate_energy()`, 2000 boards with $\_deg = 0.5$ of size $L = 1000$ were initialized and the energy was computed. The results can be seen in figure 1. As you can see, it's a very nice centered Gaussian which is exactly what we would expect.[4]

Another test is to initialize a board with $\_deg = 0$, so for all fields $\sigma = 1$. Since there are L^2 fields with four interactions each, the energy of this system should be $-4 \cdot L^2$. This anticipation verified, too.

To not loose computation time by flipping a spin, checking the energy, and eventually flipping it back, I use the function `Delta_E` which will compute

[1] sometimes I'll refer to them as "fields"

[2] for large L the ratio will approach $\_deg$

[3] To keep units out of the simulation, the interaction energy is actually $J \cdot (\sigma \cdot \sigma')$. See also footnote 6

[4] For technical reasons, $2 \cdot E$ is plotted there instead of E – the important aspect in this figure is the center at 0.

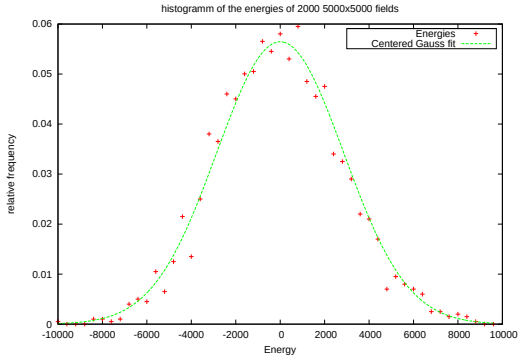


Figure 1: Energy-Histogram for 2000 1000x1000 boards with $\_deg = 0.5$. See footnote 4

the difference in total energy, ΔE, if one spin was flipped. Based on that value, the spin is either flipped (the corresponding integer is multiplied by (-1)) or not – if it's flipped, the energy of the system is altered by ΔE, the magnetisation by $2\,\sigma_f$ where σ_f denotes the final (flipped) spin.

It additionally features to save the data – thus each single spin – after the simulation. This way, one can load the final state of a previous simulation and go on simulating with this. The idea is to simulate once until an equilibrium was reached and from then on use this equilibrium for further analysis.

In order to be able to check whether the system has yet equilibrated, the program outputs not only final mean values for m and e defined via

$$e = \frac{\langle H \rangle}{L^2} \text{ with } H = \sum_{\substack{\text{nearest neighbours } \sigma' \\ \text{each spin } \sigma}} -\sigma \cdot \sigma' \tag{1}$$

and

$$m = \langle |\mu| \rangle \text{ with } \mu = \frac{\sum_i \sigma_i}{L^2} \tag{2}$$

[5]but also *current* values: The file `current.data` contains for each time step information about m, e and the number of spins flipped in the current time step (yet unfortunately this last data does not help to determine whether in equilibrium or not).

[5]In this case, the brackets $\langle \cdot \rangle$ denote a mean over the time steps $t \in \{t_{\text{equ}}, \ldots, t_{\max}\}$.

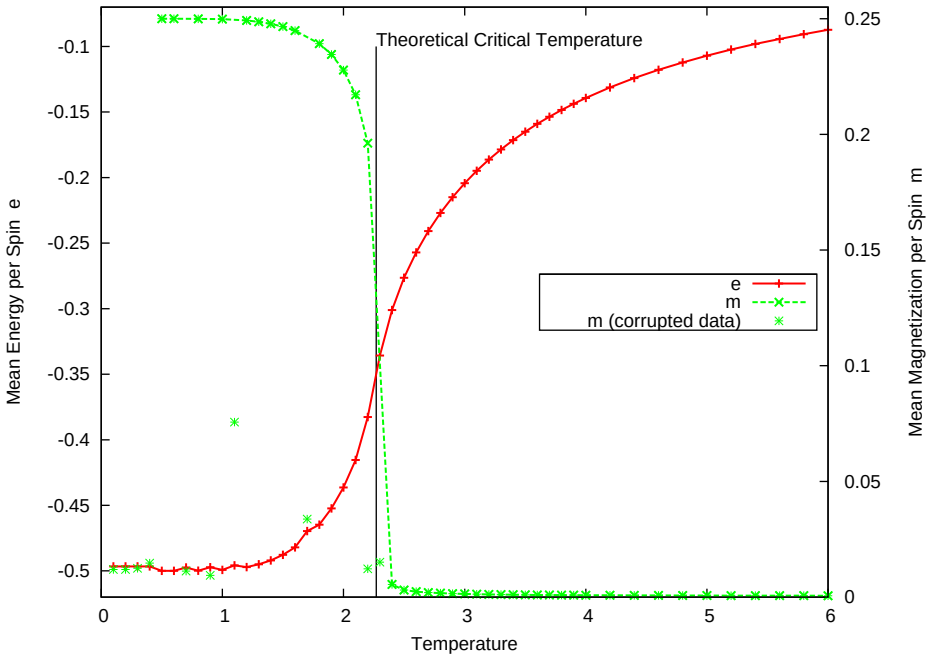


Figure 2: m and e in equilibrium for different Temperatures.

2 Equilibrium for different temperatures

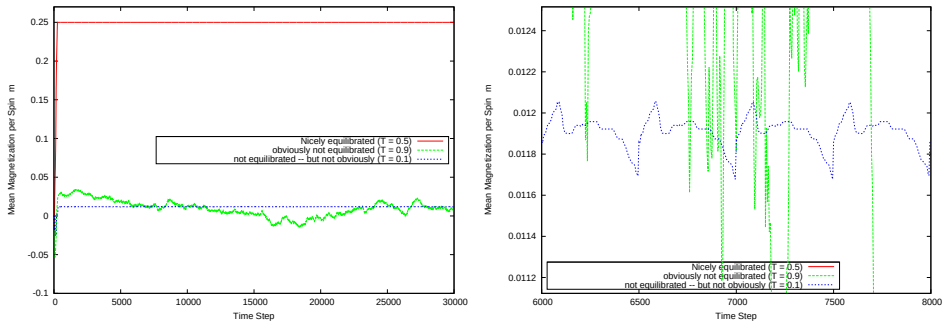
I performed several simulations, each time using identical initial conditions: A board with 50% of it's spins 1 and 50% being -1 randomly distributed. Each time, the magnets have the size $L = 500$ and the simulation is run 30 000 time steps, while the last 10 000 are (hopefully) in an equilibrated system – so only in these time steps values for m and e are ascertained. The temperatures for the different runs in between 0.1 and 4.[6]

In figure 2 you can see the results of m and e – and as you can see they're beautiful. One can observe the leap of the Magnetisation to 0 at the theoretically expected critical temperature of

$$T_c = \frac{2}{\ln(1+\sqrt{2})} \approx 2.27\ . \tag{3}$$

As you can see, some of the points for the magnetisation are not connected to the line bur lie quite far apart. This is due to the fact that they have not been in an equilibrium state. In figure 3 I provide examples of a system in

[6]Temperature unit is in this simulation uniform as instead of the real temperature the program uses the thermal energy: $T \mapsto \mathrm{k}_B T/J$.



(a) Large scale (b) Small scale

Figure 3: Examples for equilibrated and not equilibrated systems.

equilibrium and *not* in equilibrium. Obviously, there exist some states for a system, in which it behaves periodically. These states are pictured in figure 4. Observe the little edge walking horizontally.

For crucial points (thus points near the critical temperature) these simulations were conducted again, using another *seed* for the random number generator. Unfortunately the program refused to yield equilibrated results too near to the critical temperature...

In figure 5 you find several equilibrated states for several (interesting) temperatures. You can see there, that for larger temperatures, the system will be totally disorganized. The more the temperature tends to the critical temperature, the larger domains of common spin orientation grow. But as we can see in figure 2 there exist as much up as down spins as the total magnetisation vanishes.

Since the colder systems tend to lower energies they try to align each single spins in one direction – due to the "$-$" sign in equation (1) this is the state of lowest energy. For systems with $T > T_c$ also want to get into a state of minimal energy but have to maintain a zero global magnetisation, they constitute large (the lower the temperature the larger) domains as within these domains of common spin orientation the interaction energy is the lowest possible. Only on the margins of the domain there is a larger interaction energy. So since the ratio of margin to area will be the smallest (and thus the most "energy efficient") for larger domains, the domains grow with lower temperatures.

Below the critical temperature, the restriction of $m = 0$ does not hold any more so the spins may orient in the most convenient manner – the parallel aligned one.

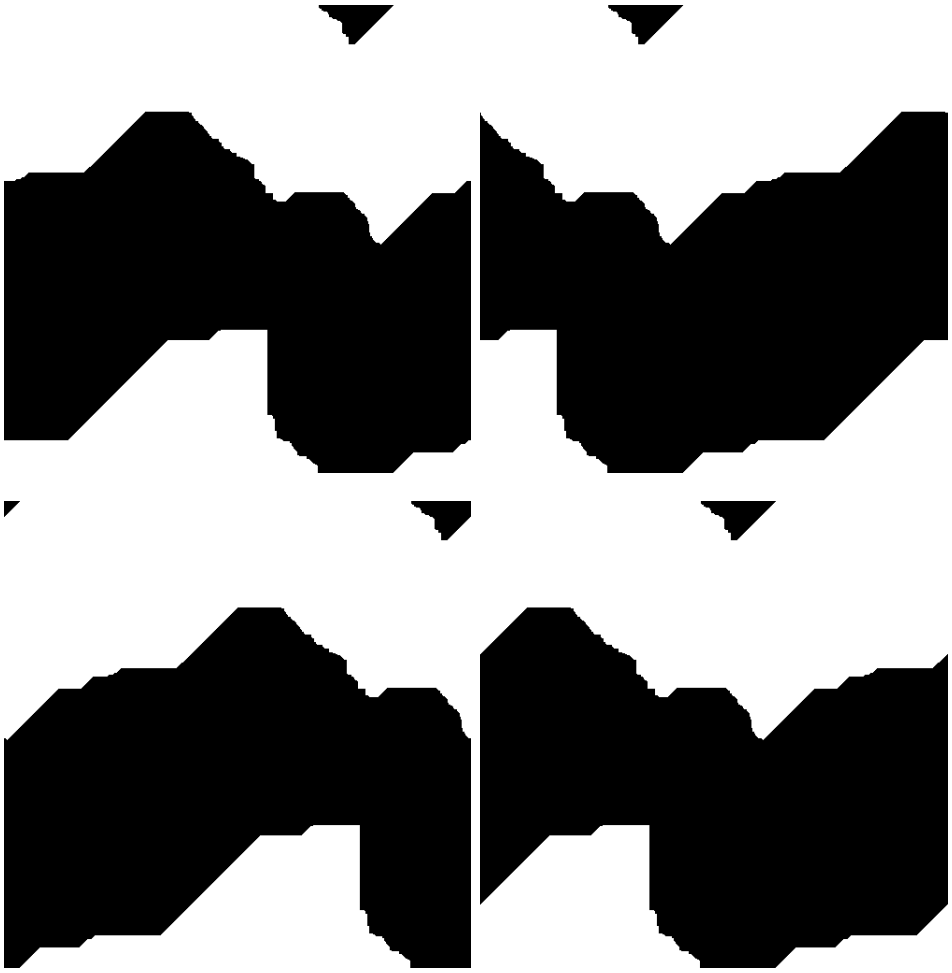
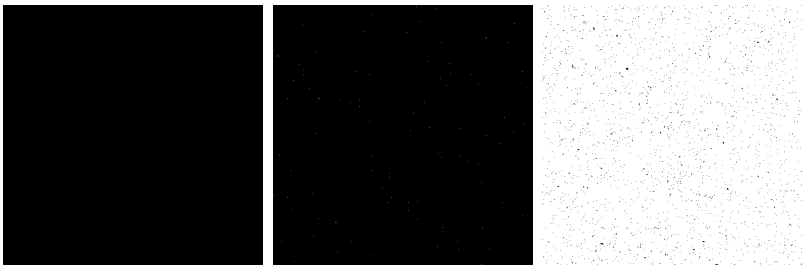


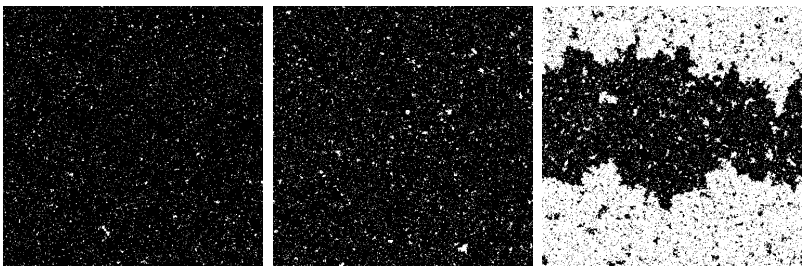
Figure 4: Periodic behaviour at $T = 0.1$ – these images are 200 time steps apart (top left to bottom right).

At $T = 1.6$ you can see quite nicely how the system first builds domains, then one single domain then this one gets round (as again the circle has the best margin to area ratio) and then vanishes completely, letting the system go into one (nearly) ground state. Cf figure 6.

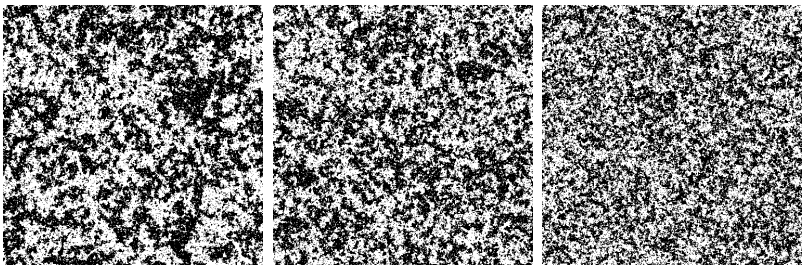
While viewing these images, you must bear in mind that there exist two ground states for the system: The "white" and the "black" one. So in figure 5 the $T = 1.6$ system seeks the white and the other $T < 2.2$ systems seek the black one.



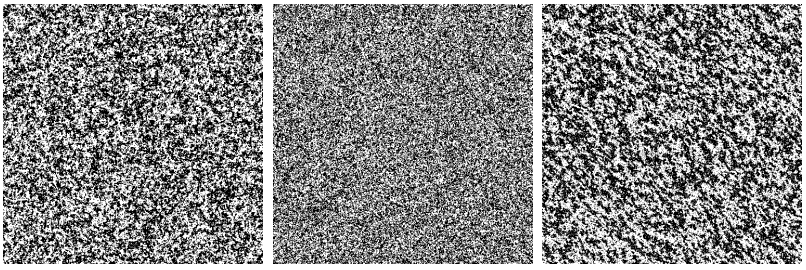
(a) 0.5 (b) 1.0 (c) 1.6



(d) 2.0 (e) 2.1 (f) 2.2



(g) 2.4 (h) 2.5 (i) 2.7



(j) 3.0 (k) 5.0 (l) initial

Figure 5: Equilibrated states for different temperatures (given in captions). The lower right one depicts the initial state – every simulation started in this state.

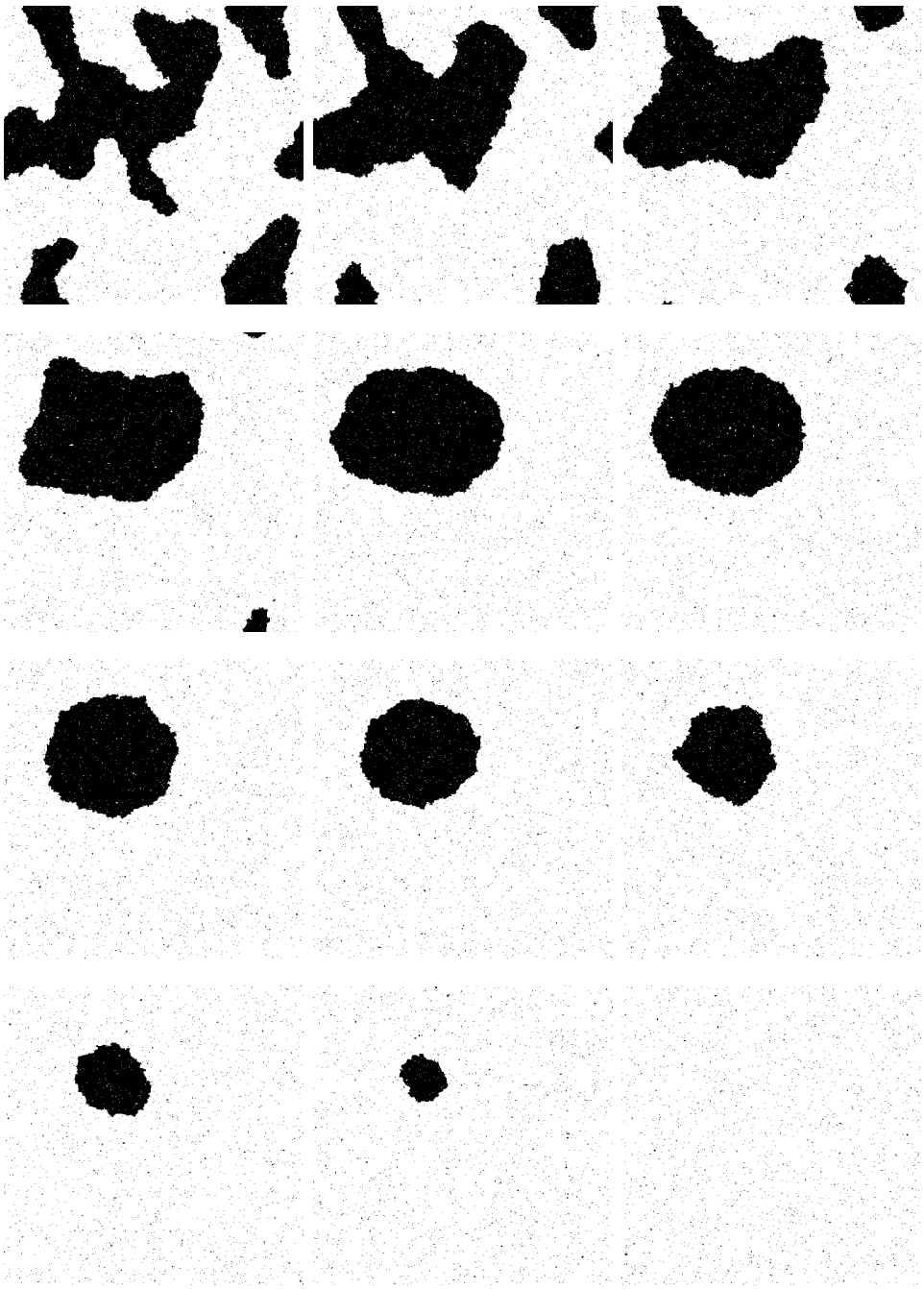


Figure 6: The decay of a domain. From top left to bottom right. Between each image there are 200 time steps.