Exercise 1. Cluster algorithms

Goal: Cluster algorithms can be used to reduce the critical slowing down substantially. This week you have to simulate either the 3D Ising model or the Potts model with $q = 2$ using a cluster algorithm. You may choose either the Swendsen-Wang or the Wolff algorithm.

Task 1: *Find the connection probability in the Ising model.*

Hint: In the Potts model the probability for connecting sites in the same state is $p = 1 - \exp(-\beta J)$.

Task 2: *Implement either the Swendsen-Wang or the Wolff algorithm. Check your code by plotting the binder cumulant or the magnetic susceptibility around T_c.*

Task 3 (OPTIONAL): *Implement the other algorithm as well.*

Task 4: *Compare the performance of the algorithms and show that in the cluster algorithms the critical slowing down is substantially reduced. Measure the runtime and compute the linear autocorrelation time τ (relative to the observable E or M) of the cluster algorithm and compare it to the Metropolis algorithm. Report a table with τ and the Monte Carlo speed defined as*

$$MC_{speed} = \frac{sweeps}{time} \cdot \frac{1}{\tau}$$

for at least three temperature values, say T_c, $T_c + 1$, $T_c - 1$, and fixed size. Make another table with the temperature fixed at T_c and varying lattice size. Interpret the results.

Hint: Remember that you can extrapolate τ from the autocorrelation series

$$\rho_{XX}(\Delta t) = \frac{\left\langle (X_t - \bar{X})(X_{t+\Delta t} - \bar{X}) \right\rangle_t}{\sigma^2}.$$

Solution. In this exercise we compare the Metropolis algorithm with a cluster algorithm. In the following we will investigate the Wolff algorithm but Swendsen-Wang would give a similar behavior.

The connection probability for the Ising model is $p = 1-\exp(-2\beta J)$ since the spin configurations are ± 1 instead of 0, 1 as in the Potts model.

For both algorithms we initialize the system in a random configuration and thermalize the system. One (sweep) step in the Wolff algorithm will flip a cluster of $\sim N$ spins. For comparability a sweep in the Metropolis algorithm should also flip $\sim N$ spins, hence we pick randomly N spins and attempt to flip them. The system is thermalized over 100 sweeps and then measured 100000 times with a single sweep between each measurement.

The results for the binder cumulant and the magnetic susceptibility for $L = 10$ are shown in Fig. 1. We see that the results agree.

Next we want to compare the performance of the algorithms in terms of calculation time, autocorrelation time and $\mathrm{MC}_{\mathrm{speed}}$. For this we measure the calculation time of the sampling after the thermalization. The autocorrelation ρ_{XX} is calculated from the measurements of M or E. The autocorrelation time τ can be extracted from the logarithm of the autocorrelation ρ_{XX}. Be aware that the numerically calculated autocorrelation series always sums up to $1/2$ (or $-1/2$ if

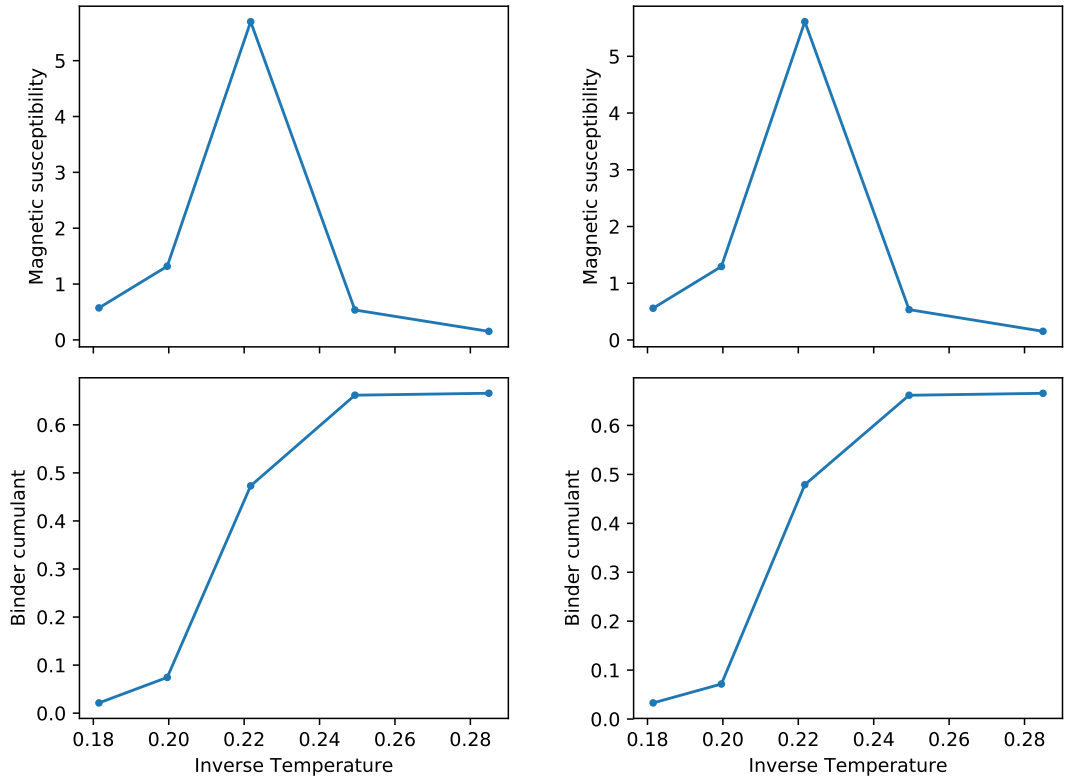

Figure 1: Magnetic susceptibility and Binder cumulant calculated by the Metropolis (left) and the Wolff algorithm (right).

β	L	runtime	τ	MC_{speed}
0.28	10	141.3	2.4	294
0.25	10	125. 2	5.9	163
0.22	10	171. 0	30.4	19
0.20	10	161.2	7.2	86
0.18	10	160.5	3.3	188

β	L	runtime	τ	MC_{speed}
0.28	10	102.6	0.96	1017
0.25	10	74.2	1.74	777
0.22	10	17.6	5.61	1016
0.20	10	2.3	4.81	8855
0.18	10	1.2	7.57	10721

Table 1: Runtime, autocrorrelation time and $\mathrm{MC}_{\mathrm{speed}}$ of M as a function of temperature for Metropolis (left) and Wolff algorithm (right).

β	L	runtime	τ	MC_{speed}
0.22	8	77.3	19.1	68
0.22	10	171.0	30.4	19
0.22	12	217.6	41.6	11
0.22	16	534.5	74.4	2

β	L	runtime	τ	MC_{speed}
0.22	8	14.0	4.57	1558
0.22	10	17.6	5.61	1016
0.22	12	25.6	6.48	603
0.22	16	53.5	8.78	215

Table 2: Runtime, autocrorrelation time and $\mathrm{MC}_{\mathrm{speed}}$ of M as a function of system size at T_c for metropolis (left) and Wolff algorithm (right).

the 0th element is omitted). For this reason we need to cut off the autocorrelation series when we calculate the slope of $\log(\rho_{XX})$ which gives us $-1/\tau$.

In table 1, 2 the results for different temperatures and different system size are shown. We can observe that τ is increasing at the critical temperature for the Metropolis algorithm (critical slowing down). For the Wolff algorithm there is only a small increase when $\beta < \beta_c$, since here the clusters are small and hence less spins are flipped in a single (sweep) step. However, smaller clusters are built faster and hence the runtime at this temperatures is lower. Looking at the $\mathrm{MC}_{\mathrm{speed}}$ we find that the calculation is slowed down for $\beta > \beta_c$.

In table 2 we look at the dependence on the system size at the critical temperature. We find that both methods slow down as expected since more cells need to be updated. Looking at $\mathrm{MC}_{\mathrm{speed}}$ we find that the Wolff algorithm outperforms Metropolis.