1 Introduction

The Swendsen-Wang [2] and the Metropolis algorithm were implemented for the 3d Ising system to study autocorrelation behavior around the critical temperature and Monte Carlo speed.

2 Algorithm Description

2.1 Swendsen-Wang algorithm

On the highest level, the Swendsen-Wang algorithm for the Ising model works as follows.

- Consider bonds between two sites active
  - if the sites have the same spin state
  - with probability $p = 1 - e^{2\beta J}$
- Identify clusters
- Go through all clusters and flip each cluster with probability $p = 0.5$
- Repeat until desired number of sweeps

The task of identifying clusters can be considered a subalgorithm: Historically, the Hoshen-Kopelman algorithm is often used for identifying clusters in the lattice. As it is somewhat cumbersome to implement in three dimensions and its main benefit of efficient memory usage is no longer relevant these days, clusters were identified using a generic Union-find Forest algorithm (of which Hoshen-Kopelman is a special case) with path compression [4]. The cluster identifying algorithm works as follows:

- Go through all lattice sites
- For each site, consider the bonded neighbors established (see above)
- Perform a union-find step on the site with each neighbor in turn:
  - Find respective cluster roots
  - Find respective cluster masses
  - Adopt the lighter of the two clusters into the heavier cluster

2.2 Metropolis algorithm

The Metropolis algorithm was implemented as previously described in the report for exercise 1 of this course.

3 Results

The program was implemented as described above and submitted with this report. For all experiments, the coupling constant J was fixed to the simplest ferromagnetic value of $J = 1.0$. Experiments where run for system side lengths $L \in \{8, 12, 16, 20\}$ and temperatures $T \in \{3, 4, 4.4, 4.515, 4.6, 5, 6\}$. For both algorithms, 20 samples each were taken and averaged (figures 1 and 2).

Also, the autocorrelation ρ_{MM} of magnetization M was calculated once thermalization was reached (figure 3 and 4).

Finally, autocorrelation time τ at critical temperature $T_c \approx 4.515$ then was extracted from a semilog plot of ρ_{MM} versus Monte Carlo time t measured in steps per site using the relationship $\rho_{MM} \sim e^{-t/\tau}$ (figure 5 and 6).

A comparison of actual simulation runtimes gives $0.141s$/sweep for the Swendsen-Wang algorithm at $T = 4.515$ and $L = 20$ and $0.232s$/sweep for the Metropolis algorithm also at $T = 4.515$ and $L = 20$. For this temperature and lattice size, we thus get $\mathrm{MC}_{\text{speed:S-W}} = \frac{1 \text{ sweep}}{0.141\text{s}} \frac{1}{0.87 \text{ sweep}} = \frac{8.152}{s}$ and $\mathrm{MC}_{\text{speed:Metropolis}} = \frac{1 \text{ sweep}}{0.232\text{s}} \frac{1}{2.486 \text{ sweep}} = \frac{1.731}{s}$.

4 Discussion

The plots of the magnetization, the magnetic susceptibility and the Binder cumulant seem as expected and in line with literature. As expected, the autocorrelation behaviour shows critical slowing down for the Metropolis algorithm around the critical temperature while the Swendsen-Wang algorithm remains largely unaffected. The comparison of Monte Carlo speed shows that the Swendsen-Wang algorithm beats the metropolis algorithm by a factor of 4.7 near critical temperature.
Qualitatively, all observations work out as expected from literature. However, the measured autocorrelation times fall short of published values by a linear factor. In spite of spending a relevant amount of time looking for possible errors, I could not establish the reason.

References

[1] Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E.
Equations of State Calculations by Fast Computing Machines,
Journal of Chemical Physics. 21 (6): 1087
1953.

[2] Swendsen, R. H.; Wang, J.-S.
Nonuniversal critical dynamics in Monte Carlo simulations
Phys. Rev. Lett., 58(2):86–88
1987

[3] Boettcher, L.,
Computational Statistical Physics - Lecture Notes,
ETH Zurich,
2019.

[4] Galler, B. A.; Fischer, M. J.
An improved equivalence algorithm,
Communications of the ACM. 7.: 301–303,
1964

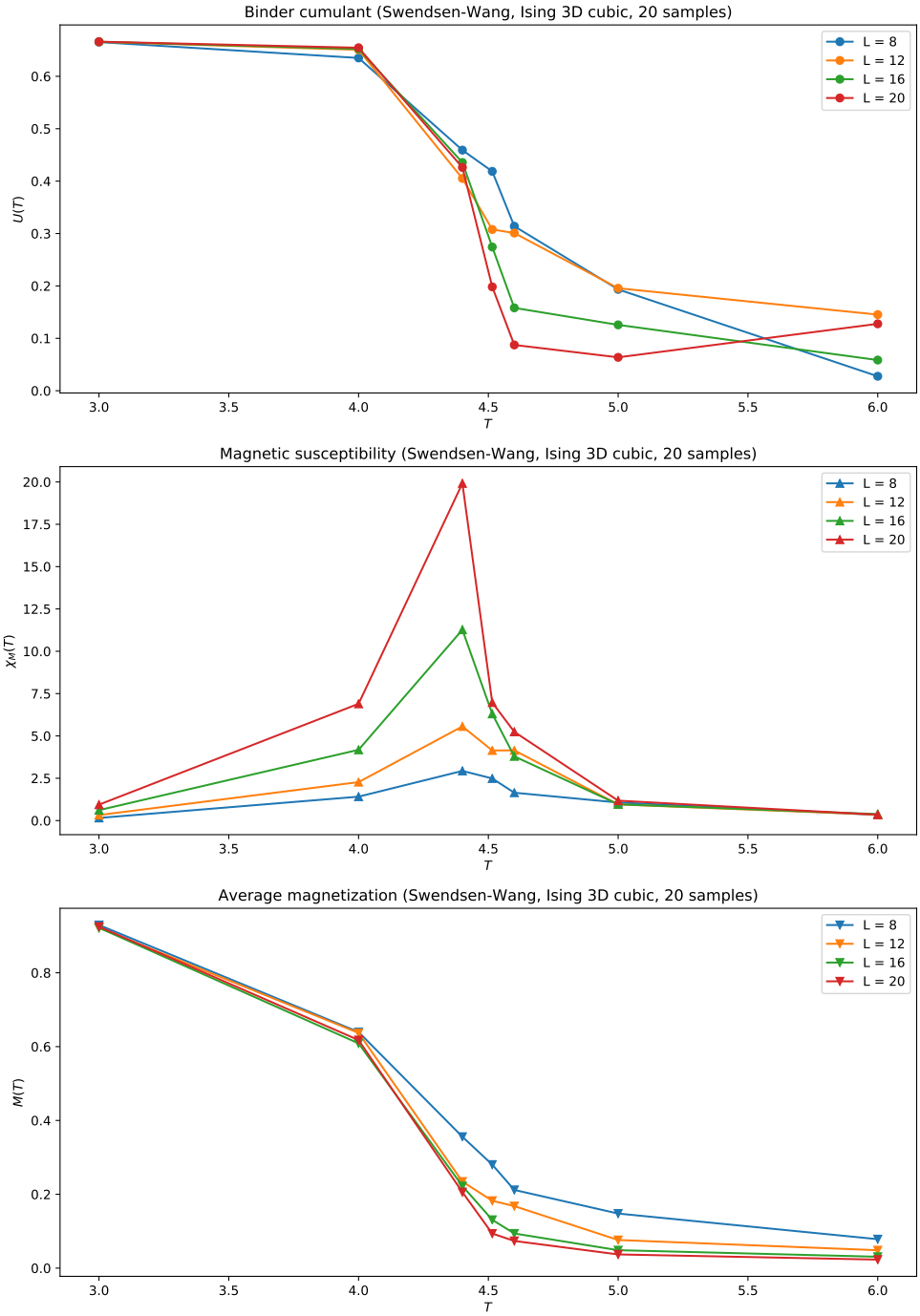

Figure 1: Results for Swendsen-Wang algorithm

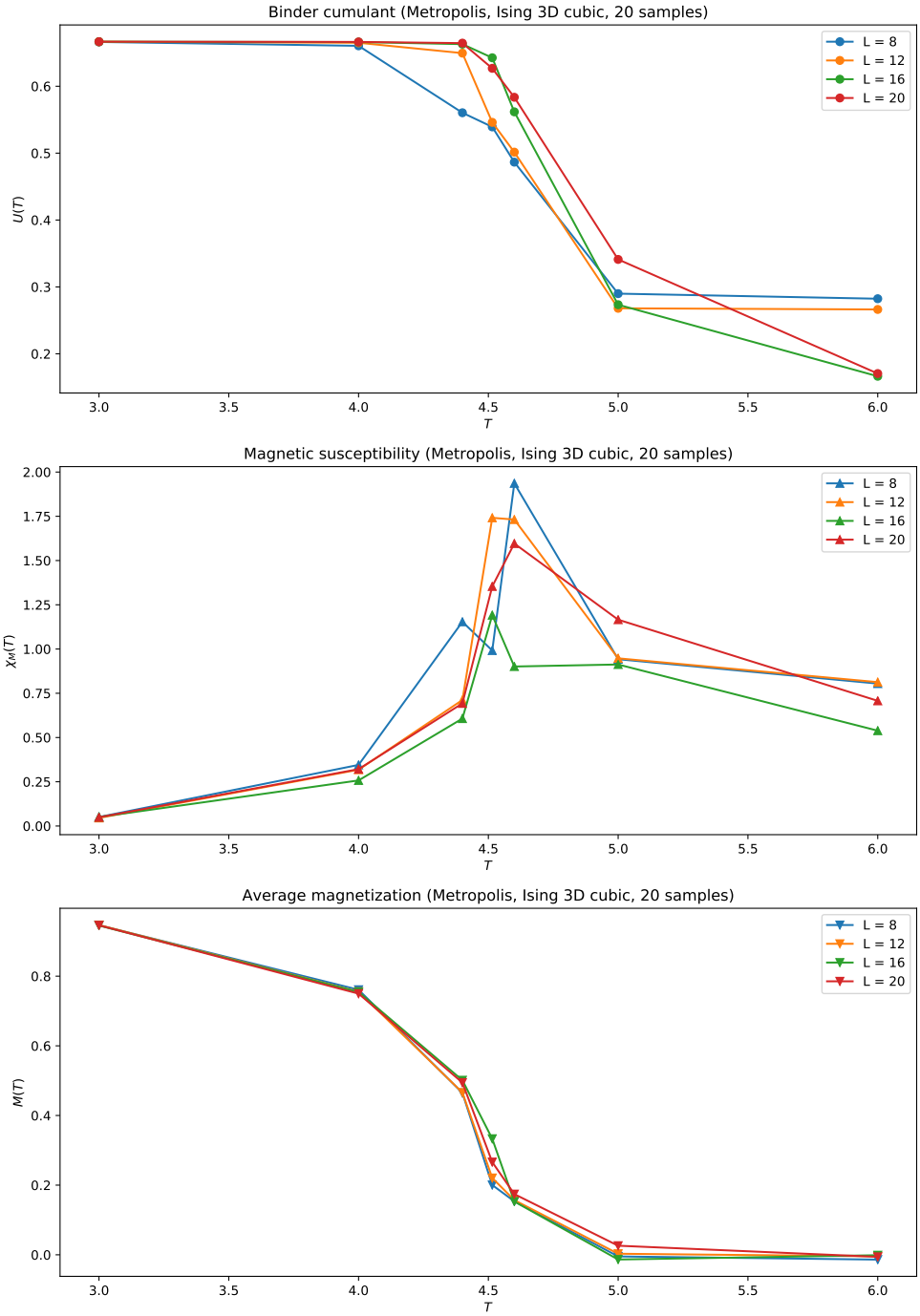

Figure 2: Results for Metropolis algorithm

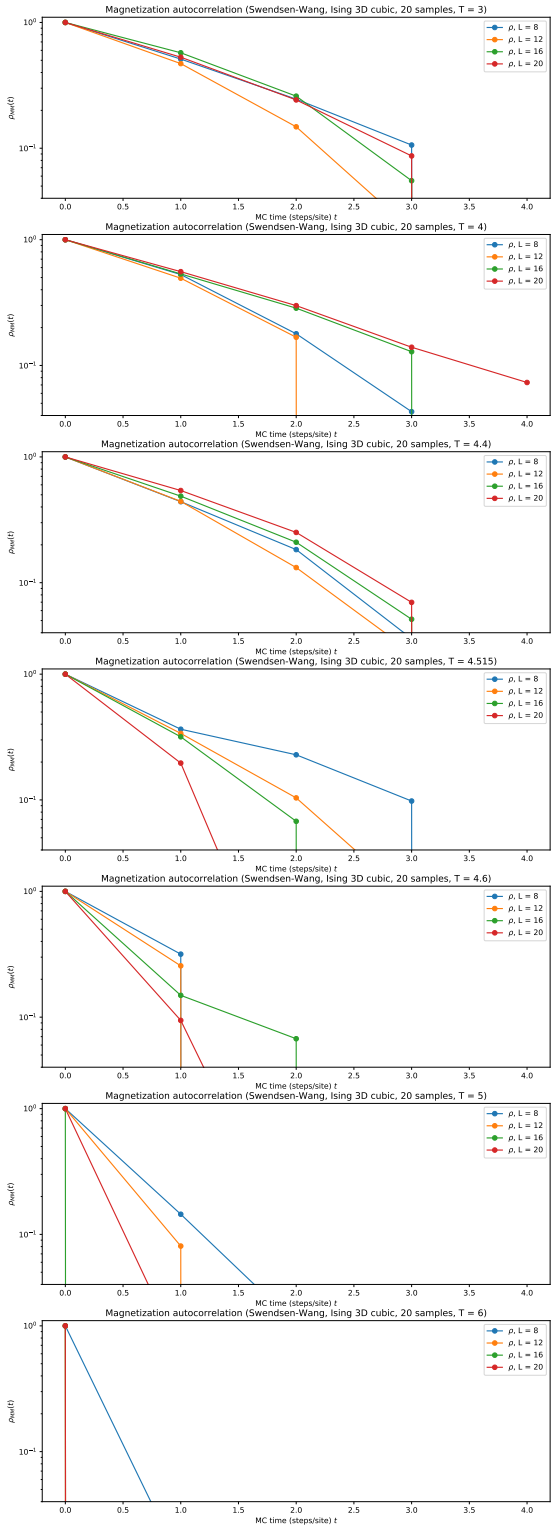

Figure 3: Autocorrelation behavior for Swendsen-Wang algorithm

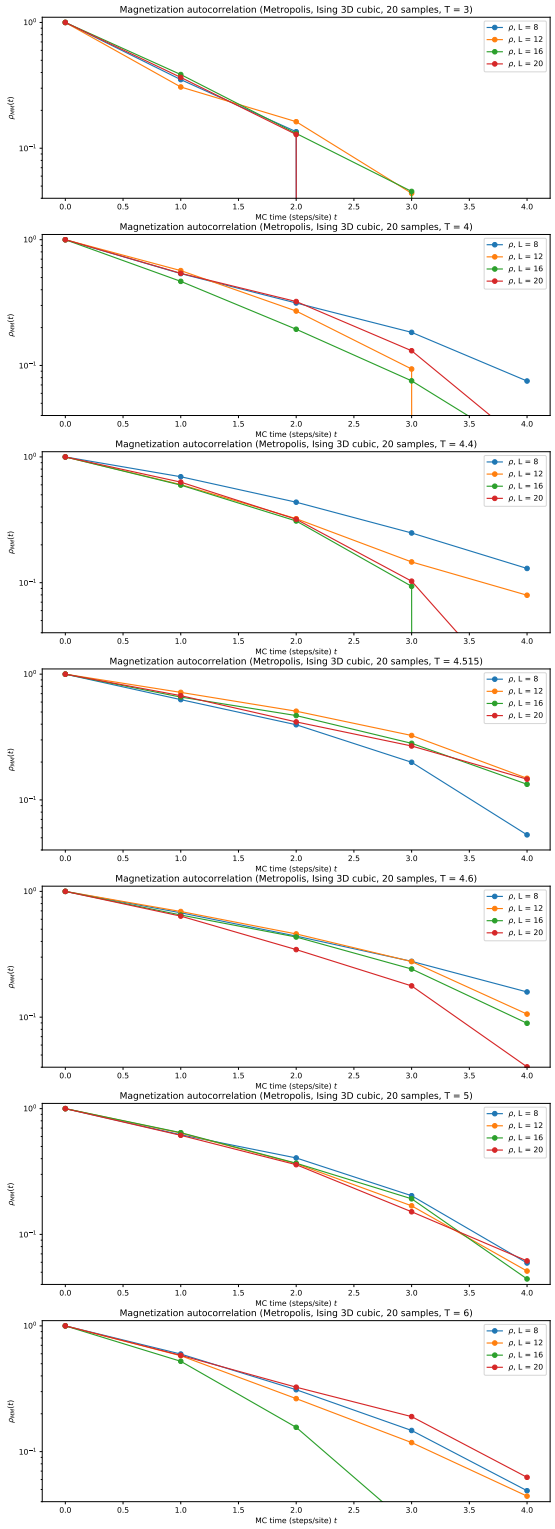

Figure 4: Autocorrelation behavior for Metropolis algorithm

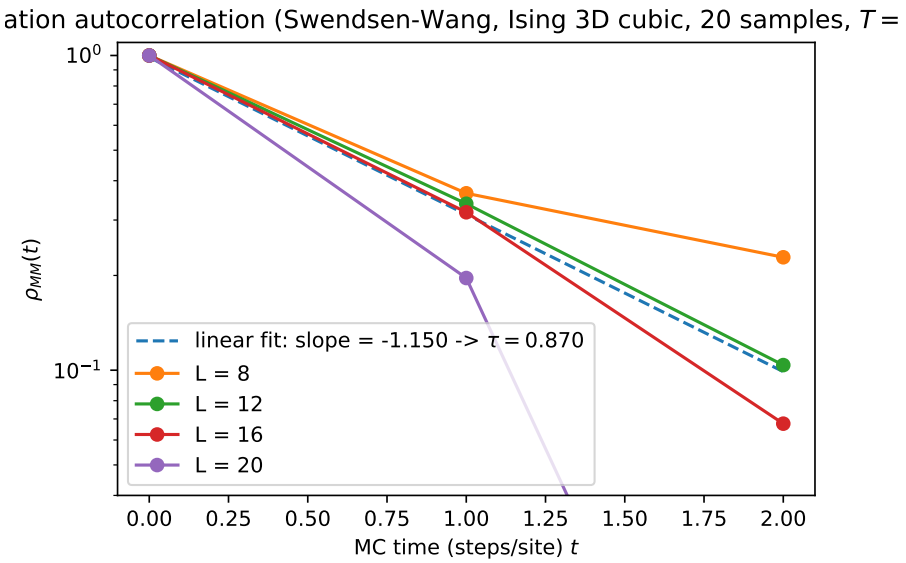

Figure 5: Autocorrelation time for Swendsen-Wang algorithm

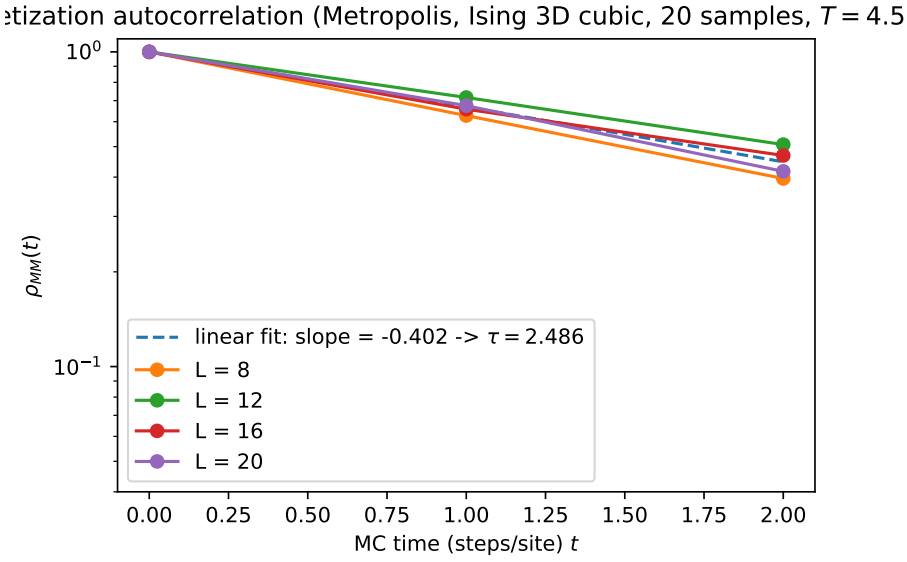

Figure 6: Autocorrelation time for Metropolis algorithm