Cluster Algorithms for 2D Ising Model

Lara Turgut

20th May 2025

Abstract

The performance of the Metropolis, Wolff, and Swendsen–Wang algorithms for simulating the 2D Ising model is compared by analyzing runtime, linear autocorrelation times, and overall efficiency. Binning analysis applied to magnetization data provides estimates of linear autocorrelation times at various temperatures and system sizes. The cluster algorithms (Wolff and Swendsen–Wang) substantially reduce autocorrelation times near the critical temperature, thereby mitigating the effects of critical slowing down. In contrast, the Metropolis algorithm exhibits significantly larger autocorrelation times. These results highlight the superior efficiency of cluster algorithms, particularly in simulations near criticality.

1 Introduction

This study focuses on simulating the 2D Ising model using cluster algorithms to investigate critical slowing down and evaluate the efficiency of various Monte Carlo methods.

The Potts model is a generalization of the Ising model in which each site on a lattice can occupy one of q distinct states. The Hamiltonian of the system is given by

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \delta_{\sigma_i \sigma_j} - H \sum_i \sigma_i, \tag{1}$$

where $\sigma_i \in \{1, \ldots, q\}$, and the Kronecker delta $\delta_{\sigma_i \sigma_j}$ ensures that only neighboring sites in the same state contribute to the interaction energy. The Potts model exhibits a first-order phase transition for $q > 4$ in two dimensions and for $q > 2$ in higher dimensions.

To efficiently sample configurations, the **Swendsen–Wang** and **Wolff** algorithms are employed, both of which are cluster algorithms designed to alleviate critical slowing down. In the Swendsen–Wang algorithm, bonds between neighboring sites in the same state are formed with probability

$$p = 1 - e^{-\beta J}, \tag{2}$$

and clusters are identified using the Hoshen–Kopelman algorithm. These clusters are then flipped collectively to generate new configurations. The Wolff algorithm, in contrast, constructs a single large cluster starting from a randomly chosen site and grows it recursively by adding neighboring sites in the same state with probability p.

The algorithms are implemented and their performance compared by examining observables such as magnetization, magnetic susceptibility, and the Binder cumulant near the critical temperature T_c. Efficiency is also assessed via runtime measurements and autocorrelation time analysis.

2 Results

I simulated the 2D Ising model using the Metropolis, Wolff and Swendsen-Wang algorithms. For all algorithms, the system is initialized with a random configuration and thermalized over 10^4 steps.

In the cluster algorithms, a single step involves flipping a cluster (Wolff) or multiple clusters (Swendsen-Wang) of approximately $\sim N$ spins. For a fair comparison, the Metropolis algorithm is executed with $\sim N$ sub-sweeps per measurement step to achieve a comparable number of spin updates.

I use 10^4 measurement steps for all algorithms. Notably, no sub-sweeps are performed for the cluster algorithms since their linear correlation lengths remain manageable even without them. Moreover, introducing sub-sweeps to these algorithms would significantly increase their runtime.

The results for the magnetization, magnetic susceptibility, and Binder cumulant for $L = 16$ are presented in **Fig. 1**. The agreement between the algorithms is evident. Additionally, I have included results from the Metropolis algorithm **without sub-sweeps** to highlight the detrimental impact of correlations on the measurements. As the plot shows, the Metropolis algorithm without sub-sweeps yields significantly poorer results.

Furthermore, the susceptibility peaks observed in all three algorithms are located close to the critical temperature, as expected.

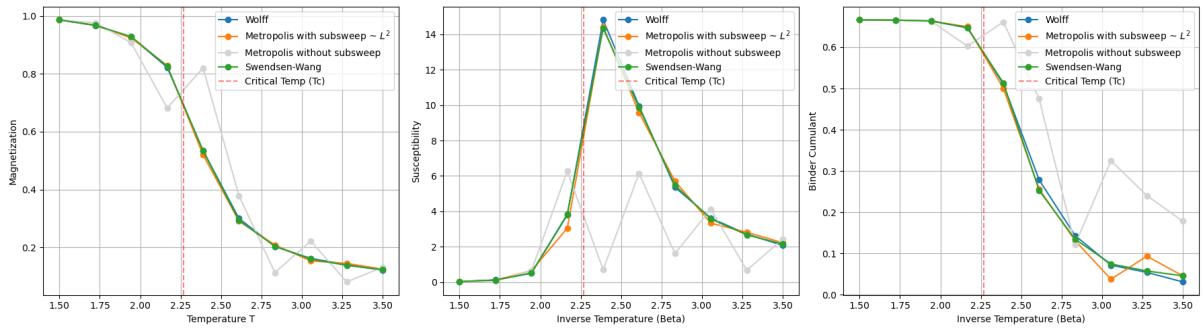

Figure 1: Magnetization, magnetic susceptibility and binder cumulant calculated by the Wolff, Metropolis (with and without subsweep) and Swendsen-Wang algorithm.

The performance of the algorithms is compared based on **runtime**, **linear autocorrelation time**, and MC_{speed}. Runtime measurements are conducted exclusively during the sampling phase, following thermalization. The linear autocorrelation time is estimated using binning analysis applied to magnetization data.

Figure 2 displays the results of the binning analysis for the Metropolis algorithm under two scenarios: with sub-sweeps (upper plot) and without sub-sweeps (lower plot). When sub-sweeps are included, the statistical errors converge to a stable plateau as the binning level increases. The x-axis represents the binning level, corresponding to successive rounds of data averaging intended to suppress statistical fluctuations, while the y-axis indicates the associated statistical error. More pronounced plateaus could be obtained by increasing the number of measurement steps and extending the binning levels, although this would significantly increase the overall runtime.

In contrast, omitting sub-sweeps leads to a continuous divergence of the statistical errors with increasing binning levels, failing to reach a stable plateau. This highlights the detrimental effect of neglecting sub-sweeps in the Metropolis algorithm.

Furthermore, the linear autocorrelation time shows a pronounced dependence on temperature: it increases in the vicinity of the critical temperature and decreases away from it. This trend reflects the expected influence of **critical slowing down**, a phenomenon in which correlation times are enhanced near criticality.

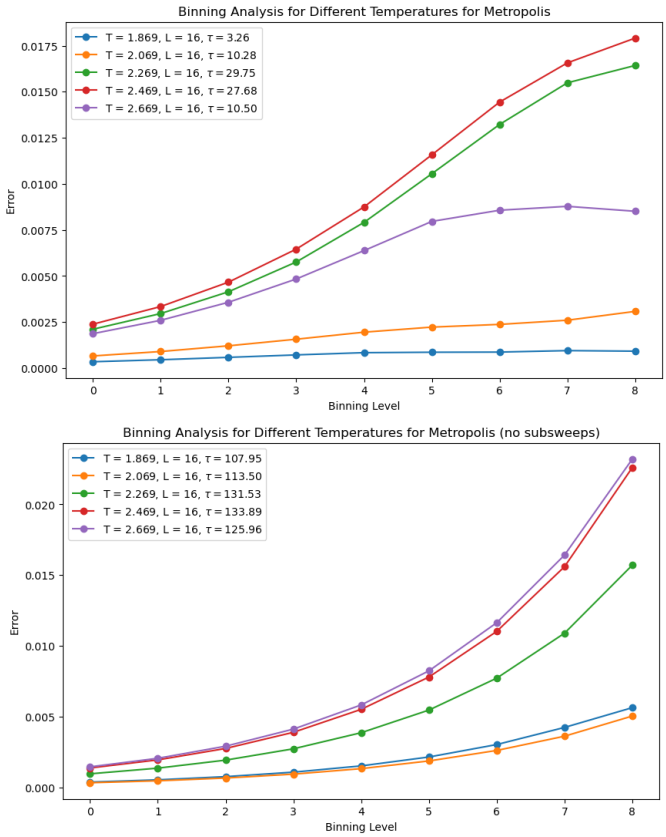

Figure 2: The plots show binning analysis of magnetization data using the Metropolis algorithm at different temperatures. The **upper plot** includes $N = L^2$ sub-sweeps, while the **lower plot** omits them. Linear autocorrelation times τ are indicated in the legends. With sub-sweeps, errors converge to clear plateaus as binning levels increase, enabling reliable estimation of τ. Without sub-sweeps, errors diverge and no stable plateau forms, making accurate estimation of τ infeasible. The reported τ values in this case are approximate, taken from the peak of the diverging curves, and likely underestimate the true autocorrelation time.

The **Figure 3** presents the binning analysis of magnetization data obtained using the Wolff algorithm at various temperatures. The Wolff algorithm, which employs cluster flipping, effectively reduces autocorrelation times by updating a large cluster of spins simultaneously.

The binning analysis demonstrates that the errors reach well-defined plateaus at higher binning levels, indicating efficient sampling and reliable estimation of linear autocorrelation times, τ.

Compared to the Metropolis algorithm, the Wolff algorithm shows significantly shorter autocorrelation times, particularly near the critical temperature. This efficiency highlights the algorithm's robustness against critical slowing down, making it highly suitable for simulations near criticality.

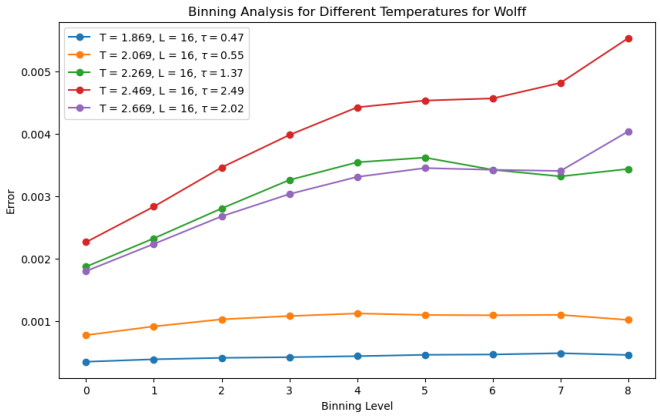

Figure 3: Binning analysis of magnetization data obtained using the Wolff algorithm at various temperatures

The **Figure 4** presents the binning analysis of magnetization data obtained using the Swendsen-Wang algorithm at various temperatures. The Swendsen-Wang algorithm, which updates configurations by flipping multiple clusters simultaneously, effectively reduces autocorrelation times compared to the Metropolis algorithm.

In comparison with the Metropolis algorithm, the Swendsen-Wang algorithm exhibits significantly shorter autocorrelation times, particularly close to the critical temperature. This demonstrates substantial robustness against critical slowing down.

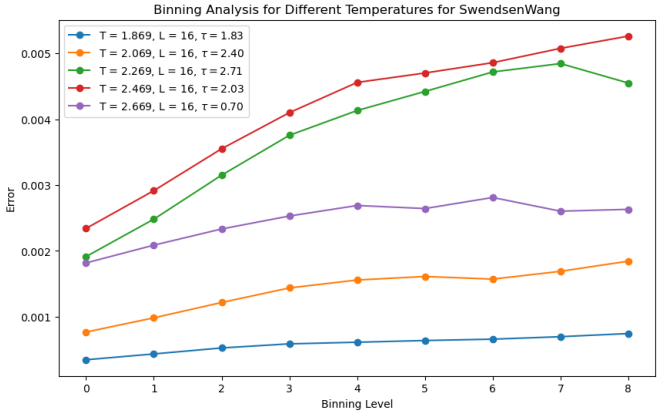

Figure 4: Binning analysis of magnetization data obtained using the Swendsen-Wang algorithm at various temperatures.

In the **Tables 1, 2, 3**, the run-time, linear autocorrelation time τ and the MC_{speed} for all three algorithms at various temperatures computed. As mentioned above, one can see that the autocorrelation time τ is very large at the critical temperature and it has a poor MC_{speed} overall. We see that the Wolff algorithm has large MC_{speed} values overall due to its small τ's and short run-times. Similary, Swendsen-Wang give short τ's compared to Metropolis, however it needs a large runtime due to handling multiple clusters at the same time, which results in a MC_{speed} lower than the one of Wolff but larger than the one of Metropolis.

T	L	runtime	τ	MC_{speed}
1.869	16	11.0	3.26	278
2.069	16	12.0	10.28	81
2.269	16	12.9	29.75	26
2.469	16	13.2	27.68	28
2.669	16	14.2	10.50	67

Table 1: Runtime, linear autocorrelation time and MC_{speed} of M as a function of temperature for the Metropolis algorithm with $N = L^2$ sub-sweeps.

T	L	runtime	τ	MC_{speed}
1.869	16	8.1	0.47	2621
2.069	16	7.9	0.55	2297
2.269	16	4.6	1.37	1591
2.469	16	2.0	2.49	2015
2.669	16	0.9	2.02	5494

Table 2: Runtime, linear autocorrelation time τ and MC_{speed} of M as a function of temperature for the Wolff algorithm.

T	L	runtime	τ	MC_{speed}
1.869	16	12.0	1.83	458
2.069	16	14.8	2.40	281
2.269	16	14.8	2.71	249
2.469	16	17.3	2.03	284
2.669	16	17.0	0.70	841

Table 3: Runtime, autocorrelation time and MC_{speed} of M as a function of temperature for the Swendsen-Wang algorithm.

Binning analysis was also conducted for different lattice sizes using magnetization data across all three algorithms. As expected, all methods exhibit increased slowdown with larger system sizes due to the greater number of spins that must be updated.

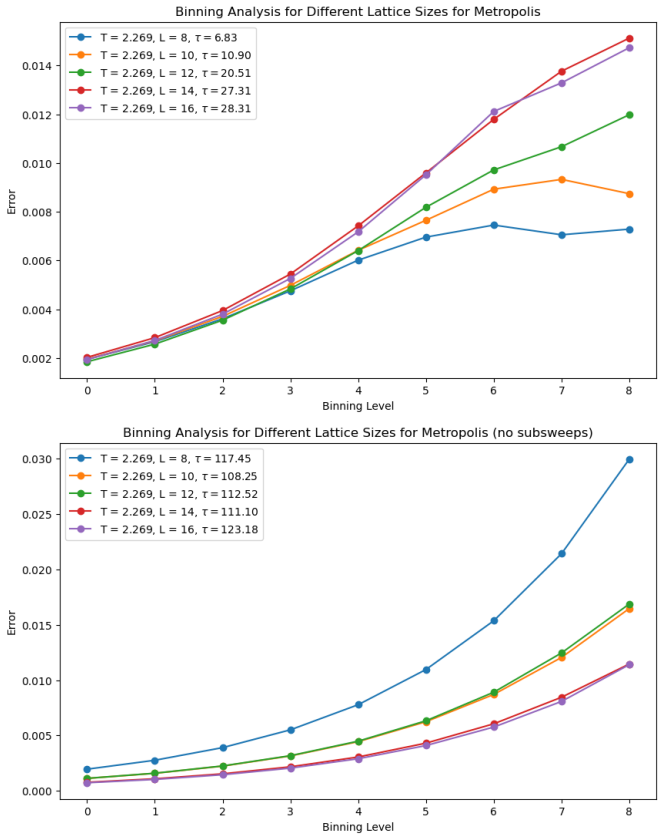

Figure 5: Binning analysis of magnetization for the Metropolis algorithm at different lattice sizes. The **upper plot** includes N sub-sweeps, while the **lower plot** omits them. Linear autocorrelation times τ are shown in the legend. For the no-sub-sweep case, τ values are only indicative, estimated from the peak of diverging curves due to the absence of plateaus. The true τ is likely larger and requires more steps and higher binning levels for accurate estimation.

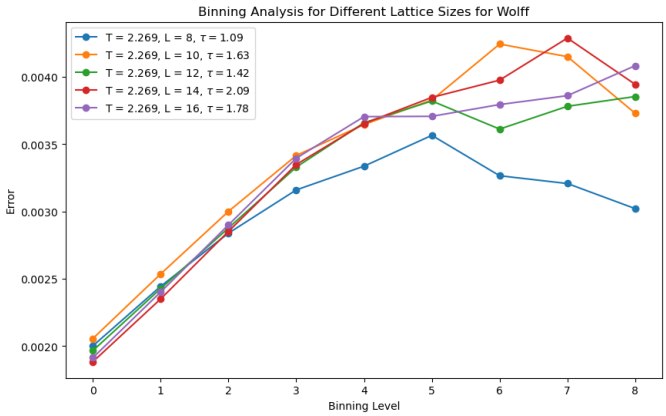

Figure 6: Binning analysis for different lattice sizes using magnetization for Wolff algorithm.

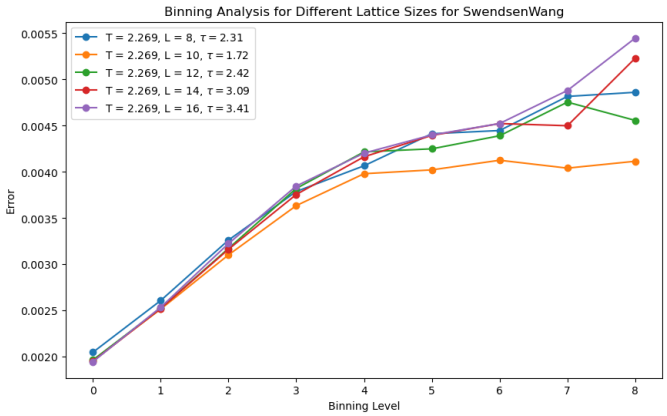

Figure 7: Binning analysis for different lattice sizes using magnetization for Swendsen-Wang algorithm.

In the **Tables 4, 5, 6**, the run-time, linear autocorrelation time τ, and the MC_{speed} for all three algorithms at the critical temperature for different system sizes are computed. As expected, the autocorrelation time τ increases with system size. The Wolff algorithm maintains relatively high MC_{speed} values overall due to its consistently small τ and shorter run-times, even as the system size increases. Similarly, the Swendsen-Wang algorithm produces shorter τ values compared to Metropolis, but requires longer run-times due to managing multiple clusters simultaneously. This results in MC_{speed} values lower than those of the Wolff algorithm but still superior to those of the Metropolis algorithm.

T	L	runtime	τ	MC_{speed}
2.269	8	3.7	6.83	393
2.269	10	5.5	10.90	168
2.269	12	7.3	20.51	67
2.269	14	9.9	27.31	37
2.269	16	14.1	28.31	25

Table 4: Runtime, linear autocorrelation time and MC_{speed} of M as a function of system size for the Metropolis algorithm with $N = L^2$ sub-sweeps.

T	L	runtime	τ	MC_{speed}
2.269	8	1.5	1.09	6005
2.269	10	2.8	1.63	2224
2.269	12	3.2	1.42	2188
2.269	14	4.5	2.09	1055
2.269	16	4.7	1.78	1207

Table 5: Runtime, linear autocorrelation time τ and MC_{speed} of M as a function of system size L for the Wolff algorithm.

T	L	runtime	τ	MC_{speed}
2.269	8	3.9	2.31	1102
2.269	10	6.5	1.72	899
2.269	12	8.6	2.42	479
2.269	14	10.8	2.09	300
2.269	16	15.1	3.41	193

Table 6: Runtime, autocorrelation time and MC_{speed} of M as a function of system size for the Swendsen-Wang algorithm.

Figure 7 presents a comparison of binning analysis results for three algorithms—Wolff, Metropolis with N subsweeps, and Swendsen–Wang—at temperature $T = 2.269$ and lattice size $L = 16$. The Wolff algorithm shows the slowest error growth, indicating a short correlation time ($\tau = 1.77$) and highly efficient sampling. The Swendsen–Wang algorithm displays moderate error growth with a correlation time of $\tau = 2.94$, suggesting reasonable efficiency, though not as optimal as Wolff. In contrast, the Metropolis algorithm exhibits substantial error increase with binning level, reflecting strong autocorrelations and poor sampling efficiency with a long correlation time ($\tau = 40.52$).

This comparison highlights the superior performance of the Wolff algorithm in reducing autocorrelations and improving measurement accuracy near the critical temperature. Although the Swendsen–Wang algorithm was expected to perform comparably—due to its global cluster updates—this was not observed in the current simulation. The relatively weaker performance of the Swendsen–Wang algorithm will be further investigated by examining the implementation in detail.

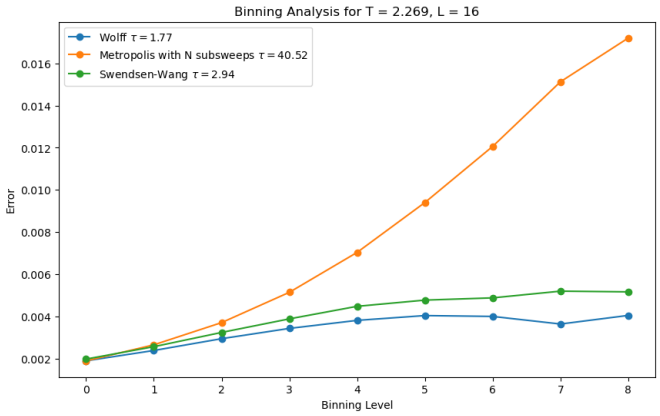

Figure 8

3 Conclusion

The Wolff and Swendsen-Wang algorithms outperform the Metropolis algorithm near the critical temperature by significantly reducing autocorrelation times and mitigating critical slowing down. The Metropolis algorithm, especially without sub-sweeps, shows poor convergence and unreliable results, underscoring the efficiency of cluster algorithms for simulations

near criticality.