Worm Algorithm for the 6-Vertex Model

Term Project Report

Matthias Pawlik
July 5, 2025

Introduction

In this coding project, we aim to compute two-point correlation functions and extract the corresponding decay exponents for the 6-vertex model. To this end, the configuration space is explored with the Worm Algorithm allowing for classical Monte Carlo sampling. We assume an infinite temperature system in which all micro-configurations respecting the so-called ice rule are equally likely.

The 6-Vertex Model

In statistical mechanics, the so-called ice-type or six-vertex models are a family of vertex models for crystal lattices with hydrogen bonds. Originally proposed by Linus Pauling in 1935, they were invented to model the local environment of water molecules in ice. Importantly, it is known that each oxygen atom is connected by a bond to four hydrogen atoms. By identifying the oxygen atoms with lattice vertices and the hydrogen atoms with the respective directed edges, six micro-configurations are physically possible in the two-dimensional formulation of the model. This is commonly called the ice rule.

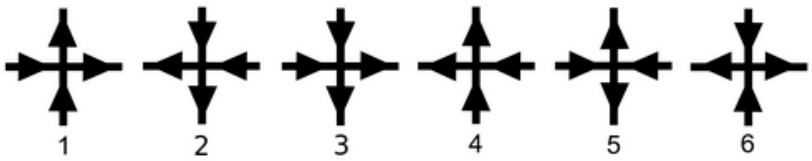

Figure 1: Micro-configurations of model vertices obeying the ice rule.

In the infinite temperature setting, it is assumed that all these vertex configurations are energetically equivalent and hence statistically appear with equal probability. In order to compute correlations of the orientation of two selected arrows at different sites, we have to uniformly sample over all possible micro-configurations subject to the ice rule.

The Worm Algorithm

The Worm Algorithm is a Markov Chain Monte Carlo (MCMC) approach tailored to explore the above mentioned configuration space. The idea is to start from a stable (ice-type) configuration on the square lattice with open boundary conditions (OBC) such as depicted in figure [2].

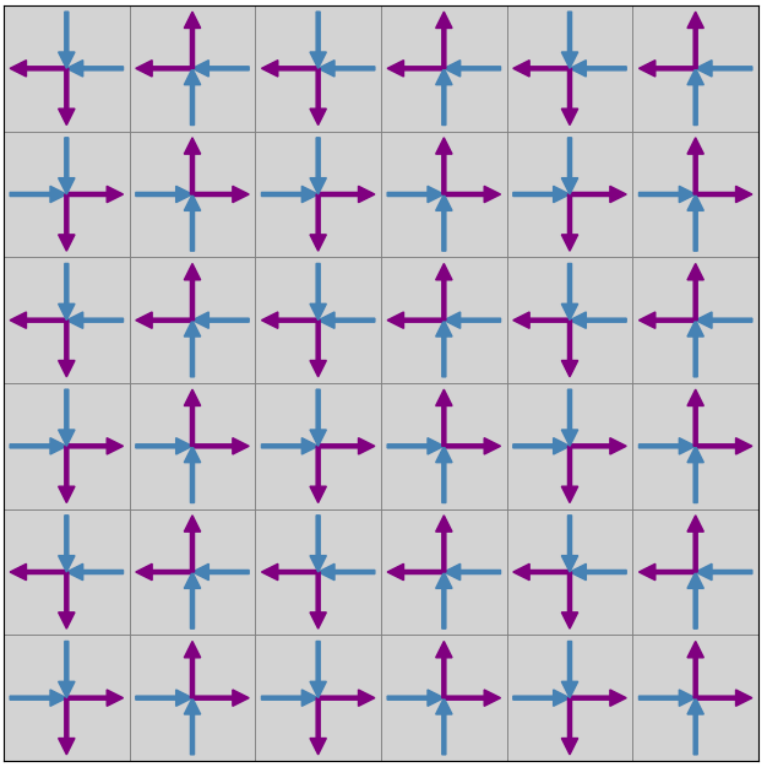

Figure 2: Initial configuration of the Worm Algorithm on a square lattice ($L = 7$) with open boundary conditions.

In the next step, two defects are introduced by randomly choosing a single vertex and randomly flipping one of its bond arrows. One of the defects is then propagated through the lattice in a Markov chain process until the defects meet again and annihilate. At each step, the nature of the defect is evaluated, i.e. its charge and the previous propagation direction. The algorithm then randomly chooses to flip one of the bonds that will restore the vertex to a valid configuration without going back. In this manner, a number of new stable configurations are realized after each run while ensuring both detailed balance and ergodicity. In order to avoid any bias on the initial configuration, the system needs to be thermalized before the correlation measurements are performed.

Methods of the Implementation

For the implementation we choose an object-oriented approach. The class `six_vertex_model` contains the vertical and horizontal sublattices with the respective bond configurations. The method `run_worm_simulation` initially calls `thermalize` which brings the system into an unbiased state after the respective autocorrelation time. The function `propagate_defect` runs the single worms through the lattice before `measure_correlations` evaluates all two-point correlations after each iteration. The individual vertex updates are handled by the methods `flip_arrow`, `flip_next` and `get_next_vertex` whereas `get_vertex_in_out` provides the necessary information about the arrow configuration at the current vertex.

Computation of Correlation Functions

The overarching goal is to extract the decay exponent a from the correlation function $\langle s_0 s_r \rangle \propto 1/r^a$. Here the arguments s_0 and s_r encode the arrow directions at the two selected vertices. We considered all four possible types of correlations between horizontal or vertical arrows along horizontal or vertical paths on the edges of the lattice. In a first instance, we verified rotational invariance of the correlations under discrete rotations of the lattice. For a system of size $L = 10$ and a number of $320,000$ iterations, we plotted the correlations along and perpendicular to the arrow directions:

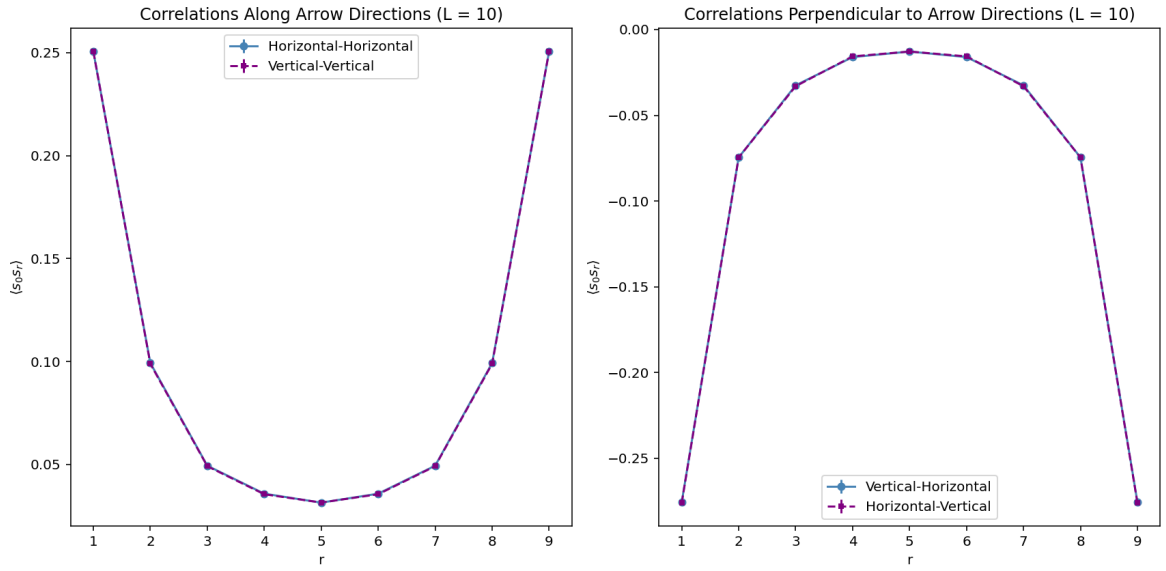

Figure 3: Decay of correlations along and perpendicular to the arrow directions for a system of size $L = 10$ computed in $320,000$ iterations.

Hence, there only remain two physically distinct `corr_type` settings in our implementation. In a next step, we sample the correlation function for different system sizes L:

L	**10**	**20**	**30**	**40**	**50**
$\mathbf{N}_{\text{iter}}$	320,000	192,000	96,000	51,200	28,800
$\mathbf{N}_{\text{i}}$	4,000	3,200	2,400	1,600	1,200
$\mathbf{N}_{\text{bins}}$	80	60	40	32	24

Table 1: System sizes L and parameters for the simulation and binning analysis.

The logarithmically rescaled graphs of figure [4] suggest that the correlations decay indeed with a power law . However, as a result of the periodic boundary conditions, the correlations increase again once the distance exceeds half of the length of the system. Moreover, the decay exponents may be fitted by selecting only sites in the proximity of the reference vertex. It can be expected that the true decay exponent can only be retrieved exactly in the thermodynamic limit where boundary effects vanish.

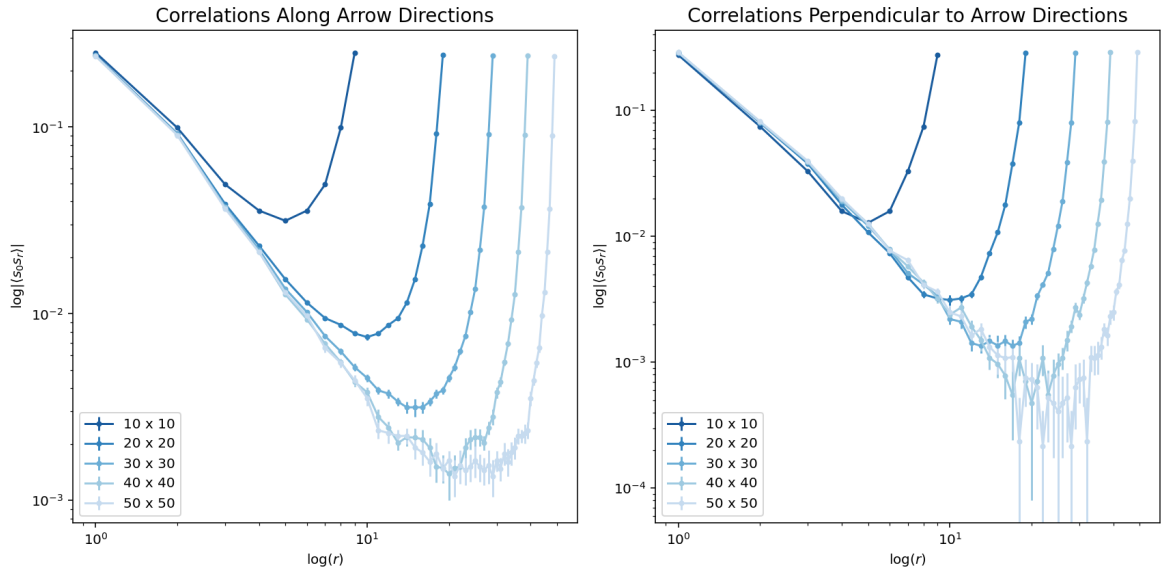

Figure 4: Decay of correlations for systems of different sizes L on a log-log scale.

In figure [5], the correlation curves were fitted with a linear function and yield the power law exponent of $a \approx 2.0$ in the limit of large system sizes L.

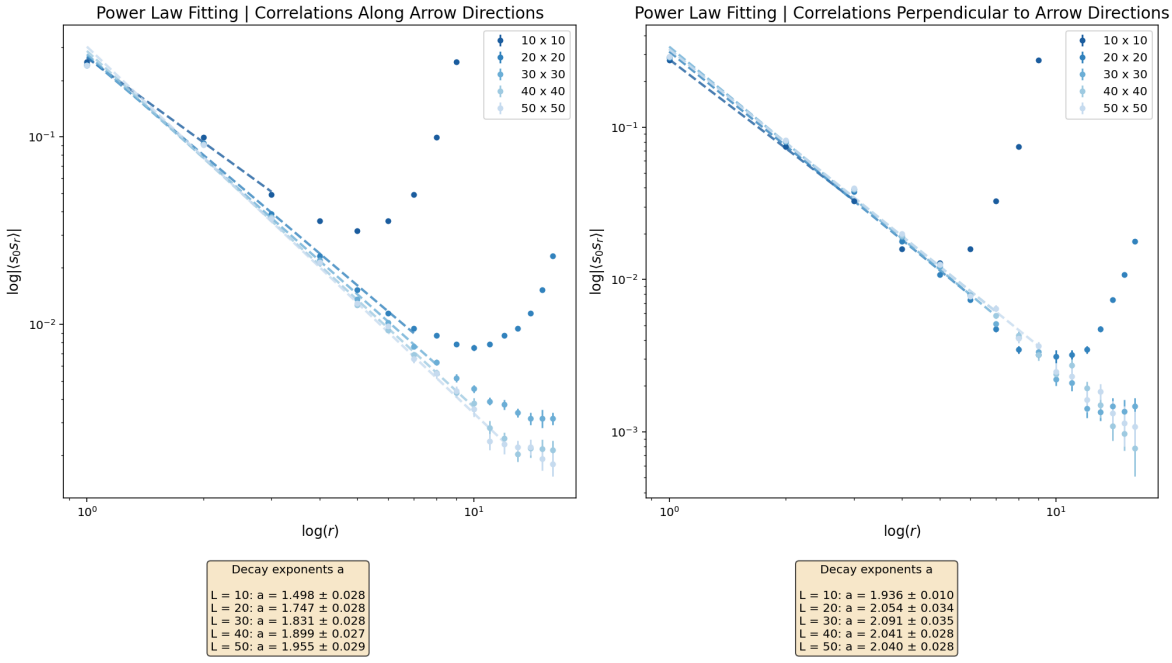

Figure 5: Linear fitting of the decay exponents a on the log-log scale.

Our results support the theoretical prediction that the correlation function in the 6-vertex model at infinite temperature decay $\propto 1/r^2$ in the thermodynamic limit. This value is consistent with known results from exactly solvable models and conformal field theory (CFT) arguments (see reference [1]).

References

[1] R. J. Baxter. *Exactly solved models in statistical mechanics*. 1982. doi:10.1142/9789814415255_0002.

[2] R. J. Baxter. The six and eight-vertex models revisited. *Journal of Statistical Physics*, 116(1–4):43–66, August 2004. URL: http://dx.doi.org/10.1023/B:JOSS.0000037215.07702.93, doi:10.1023/b:joss.0000037215.07702.93.

[3] Nikolay Prokof'ev and Boris Svistunov. Worm algorithms for classical statistical models. *Phys. Rev. Lett.*, 87:160601, Sep 2001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.87.160601, doi:10.1103/PhysRevLett.87.160601.

[4] Nicolai Reshetikhin. Literature overview: 6-vertex model. https://math.berkeley.edu/~reshetik/6-vertex.html. Accessed: 2025-07-01.