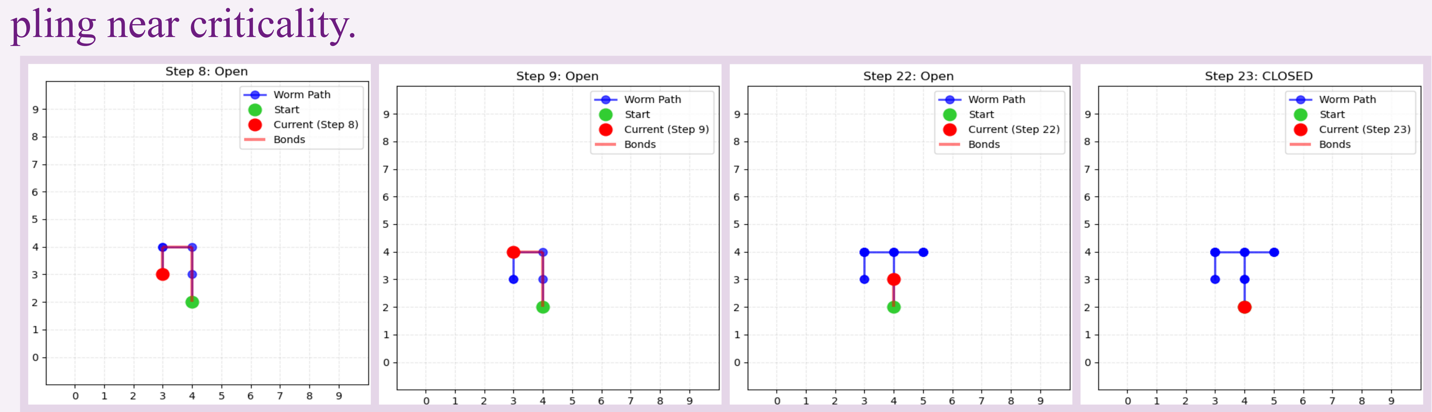
Ising Model: Worm Algorithm

Ellis Benitez[1], Jakia Chowdhury[1], Owen Gemignani[1], Carter Rauch[1]

[1]Department of Physics and Astronomy, Amherst College

Background

We used the Worm Algorithm, like the Metropolis algorithm, to sweep through 2D Ising Model lattices and measure observables such as the 2-point correlation, susceptibility, energy, and specific heat. While Metropolis performs well at low and high temperatures, it suffers from critical slowing down near the critical temperature—its autocorrelation time grows rapidly with lattice size. The worm algorithm addresses this by reducing the growth of autocorrelation time, allowing for more efficient sampling near criticality.



Method

A worm has 2 ends ***i*** and ***m***, which stand for "Ira" and "Masha". The stationary end, ***i***, starts at a random point on the lattice, with initial length 0, while the other end, ***m***, moves through the lattice by creating or erasing bonds between the sites. Bonds are formed between 2 neighboring sites when they have aligning spins.

Bonds are represented by n_b, which can take 2 values: $n_b = 0$ means no bond is present and $n_b = 1$ means a bond is present. A worm propagates by flipping bonds, which is dictated by the following probability.

$$P = min[1, tanh(\beta)^b] \qquad b(m_1 \rightarrow m_2) = \begin{cases} +1, & n_b = n_b + 1 \\ -1, & n_b = n_b - 1 \end{cases}$$

The worm continues to increment following equations above. Once ***m*** reaches ***i***, the worm closes and a new worm is initiated. We implemented the theory in a function where an L × L × 4 3D array is used to keep track of all the bonds of an L × L lattice. A direction is randomly picked from where ***m*** is, and if flipping that bond is probabilistically preferable, the position of ***m*** and the bond from both sites are updated.

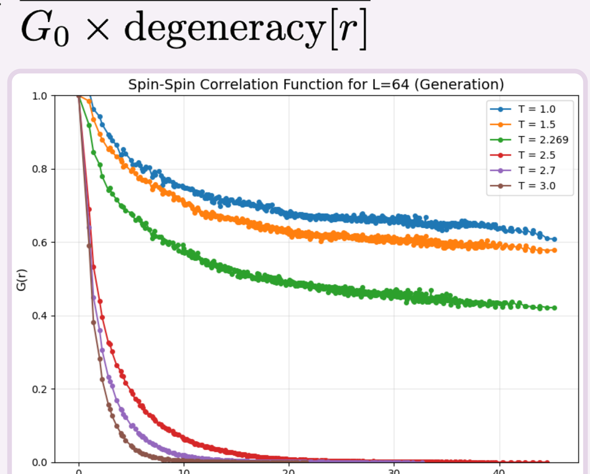
Observables

Two-Point Correlation

The worm algorithm allows us to compute the two-point correlation function:

$$G(r) = \frac{\text{Weight of open loops}}{\text{Weight of closed loops}} = \frac{G_r[r]}{G_0 \times \text{degeneracy}[r]}$$

- For T > Tc, the correlation decays exponentially with distance: G(r) → 0. C(r) is proportional to $e^{-r/\xi}$.
- For T < Tc, the correlation function plateaus to a non-zero number, indicating long-range order: C(r) →1 - m² → 0.
- At T = Tc, C(r) decays as a power law rather than an exponential and is proportional to $r^{-1/4}$.

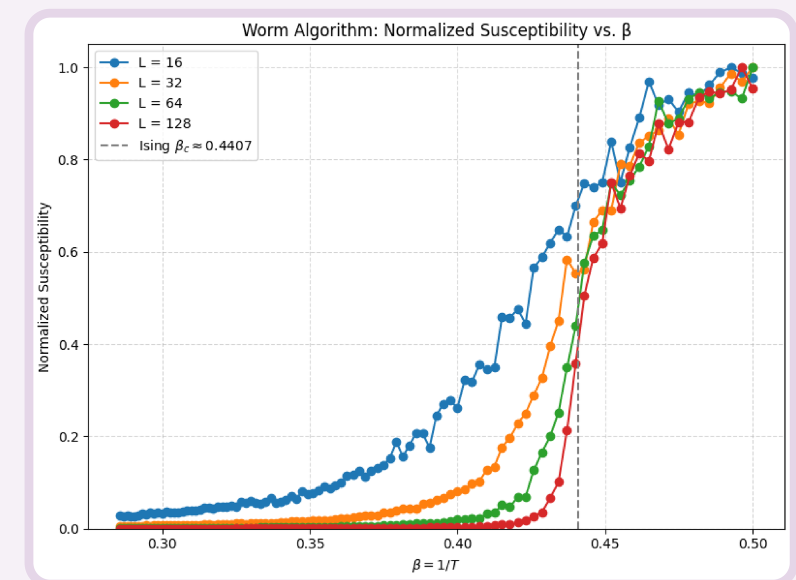


Observables (cont.)

Susceptibility

- The susceptibility program for Worm Algorithm is similar to the program for Metropolis. They both rely on a main, core algorithm and a function that computes susceptibility.
- The main difference differences between the two is that the Worm Algorithm samples the two-point correlation function and uses "degeneracy" function.
- Shell degeneracy counts how many lattice pairs correspond to each squared distance r².
- In the Worm Algorithm susceptibility program, we multiplied the shell degeneracy by the normalized correlation function to compute magnetic susceptibility χ:

$$\chi = \sum_r G(r) \cdot D(r)$$



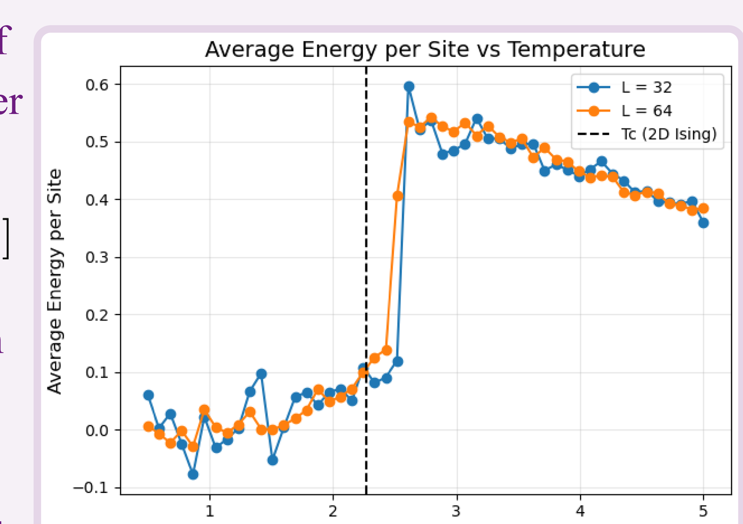
- Comparing to the Metropolis Algorithm, it can be seen that there is an inflection point, instead of a prominent peak, which Cleary's thesis[4] explains is caused by high temperature expansion and thus at low temperatures the recorded susceptibility value is not the physical susceptibility value.
- We also found $\beta_c(\infty) = 0.44022 \pm 0.00649$, (where $\beta = 1/T_c$), by plotting susceptibility vs. temperature for various lattice sizes and measuring the peak of each curve as the critical temperature. Plotting these peak temperatures against 1/L, we used the y-intercept of the linear fit to estimate the critical temperature in the thermodynamic limit. This is consistent with Cleary's results.

Energy

- Recognizing that <Nb> is the number of occupied bonds and 2N is the total number of bonds, we can express the energy as:

$$\langle E \rangle = -J\left[N_{\text{occupied}} - N_{\text{empty}} \cdot \tanh(\beta)\right]$$

- For this figure, a significant transition in energy is observed near the critical temperature, indicating the phase transition. This is similar to the results from the Me-

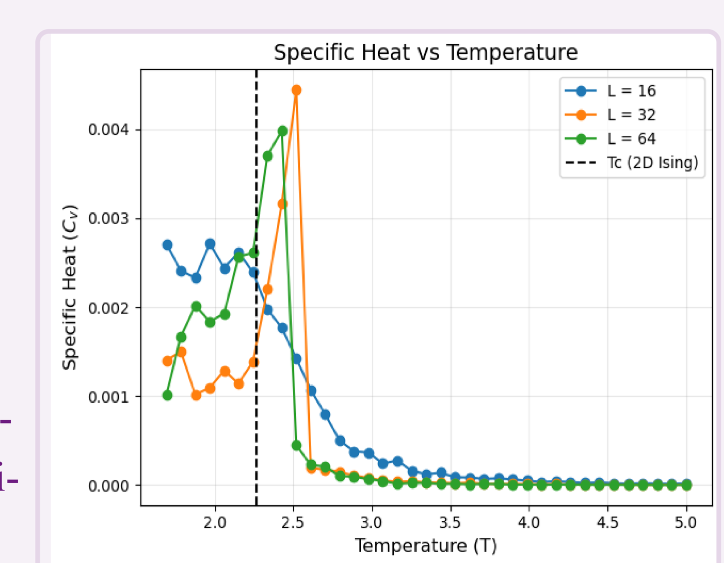


Specific Heat

- The specific heat quantifies the fluctuation in energy with respect to temperature. It is defined as:

$$C_V = \frac{\langle E^2 \rangle - \langle E \rangle^2}{T^2}$$

- A clear peak is observed near the critical temperature, consistent with the expected behavior of specific heat regarding phase transition.

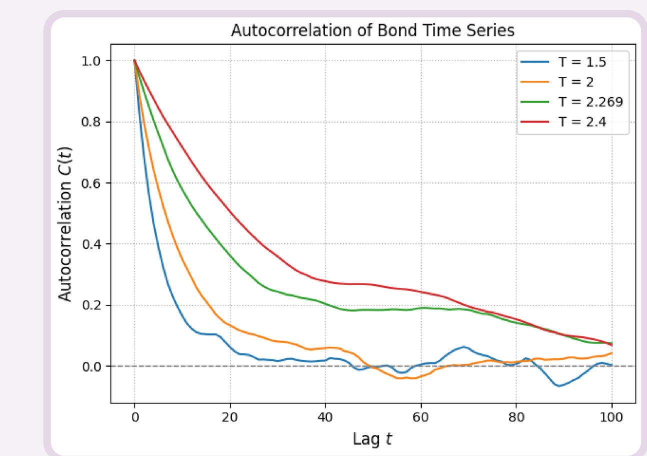


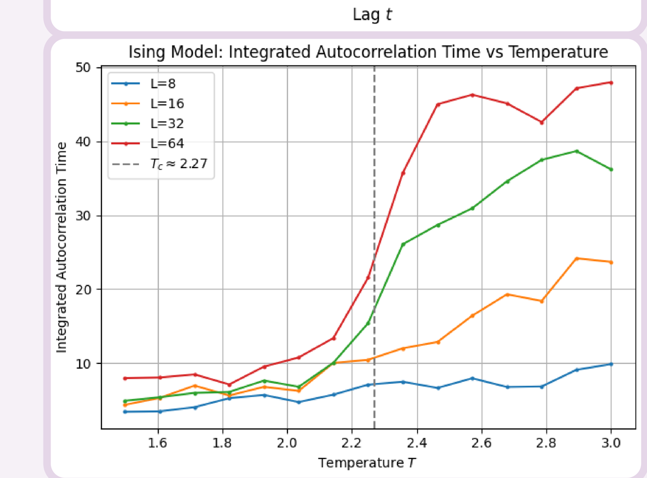
Results

Autocorrelation

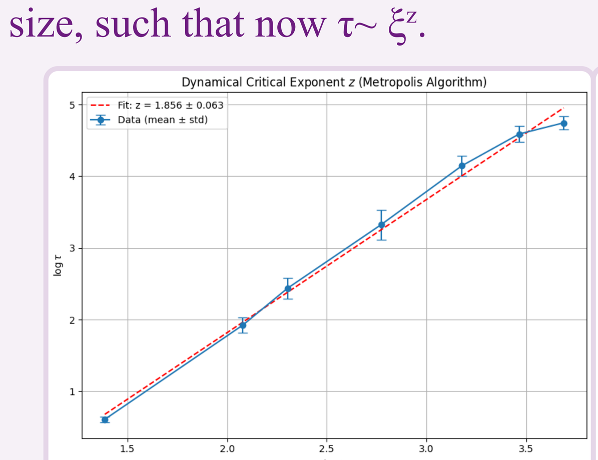
- The autocorrelation function quantifies the time it takes for an observable of our system to become uncorrelated with a past iteration of the system
- For the Worm algorithm we measured time in number of worm; ie a worm closing a loop corresponds to one "step" in time
- Integrated autocorrelation time, denoted by, is extracted from the autocorrelation function and helps create a "recording interval"
- For example a recorded = n means that we must run a n number of worms before we can sample in order to have statistically independent data.

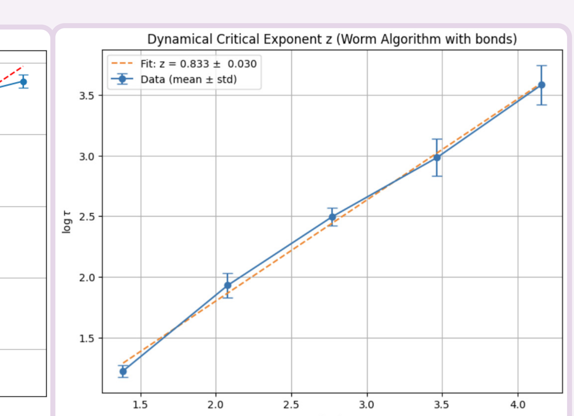
$$C(t) = \frac{\langle O(t') \cdot O(t+t') \rangle - \langle O \rangle^2}{\langle O^2 \rangle - \langle O \rangle^2}$$





- The dynamical critical exponent is our quantitative measurement of critical slowing down, as it measures how quickly the system decorrelates at critical temperature
- The critical exponent can be observed by how integrated autocorrelation time scales with correlation length in the form: τ~ ξ^z, however for our measurements it is notable that for finite lattices at critical temperature, correlation length is approximately lattice size, such that now τ~ ξ^z.





- Metropolis: z =1.856 ±0.063
- Worm: z =0.833 ± 0.030

Conclusion

- By using the Worm Algorithm we found a critical temperature, measured in beta where ($\beta = 1/T_c$) of β = 0.44022±0.00649. The theoretical value of β = 0.4407[2] falls within the uncertainty limit of our measurement.
- We measured a dynamical critical exponent of 1.856 ±0.063 for the Metropolis Algorithm and 0.833 ±0.030 for the Worm Algorithm while the theoretical result for the Metropolis Algorithm is z=2.13 and for the Worm Algorithm z=0.25[6]

Acknowledgments

We would first like to extend our greatest gratitude to Professor Loinaz for not only giving us this opportunity, but for his continued guidance, support, and patience in this endeavor. We would also like to thank Andrew Leaker ('26) for all of the meaningful conversations and providing us with knowledge of his expertise.