The Worm Algorithm

physics760: Computational Physics
Final Project

Ajay S. Sakthivasan Dongjin Suh

Universität Bonn

March 15, 2023

Presentation Outline

Presentation Outline

Introduction

- The Metropolis algorithm is a widely used Monte Carlo method for the Ising model.
- The problem of critical slowing down
- Prokof'ev and Svistunov proposed an alternative update algorithm called the Worm Algorithm (WA)
- WA preserves the local nature of the update step, but achieves a very small dynamical exponent.

Presentation Outline

- A mathematical model to understand the behaviour of systems phase transitions, like ferromagnetic materials

- 2D Ising model: Magnetic system consisting of interacting spins on a two-dimensional lattice

The Ising model

Hamiltonian:

$$H = J \sum_{\langle i,j \rangle} s_i s_j - h \sum_i s_i$$

Partition function:

$$Z = \sum_s \mathrm{e}^{-\beta H} = \sum_s \mathrm{e}^{-[-\beta J \sum_{\langle i,j \rangle} s_i s_j - \beta h \sum_i s_i]}$$

Physical observables

describe the properties of the system change at phase transition

- Magnetization

$$M = \frac{1}{N}\sum_{i}^{N}\sigma_i \quad (1)$$

  with N being the amount of sites. For a square lattice, one simply replaces N by L^2, where L is the lattice length and σ the spin.
- Energy
- Autocorrelation - Dynamical Exponent z $\tau \approx L^z$ for large J and β
- Susceptibility

$$\chi = (k_B\beta)\cdot(\langle M^2\rangle - \langle M\rangle^2) \quad (2)$$

- Specific heat

$$C = (k_B\beta)^2\cdot(\langle E^2\rangle - \langle E\rangle^2) \quad (3)$$

Presentation Outline

Metropolis-Hastings Algorithm

- Monte Carlo simulation method
- generate samples from a probability distribution
- iterative update of the system with the Accept-Reject method
- Implement Metropolis-Hastings method
  1. Random configuration of an N $\times$ N lattice
  2. Flip the spin at the site
  3. Calculate the energy cost ΔE of the flip
  4. The reject/accept step

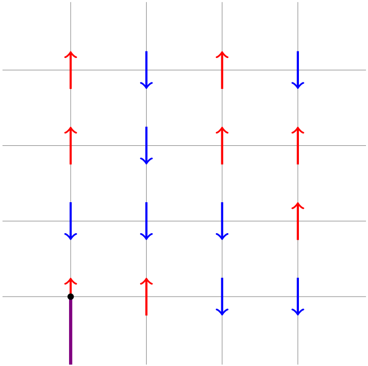
The Worm Algorithm

- The worm algorithm is an alternative to the standard Metropolis algorithm.

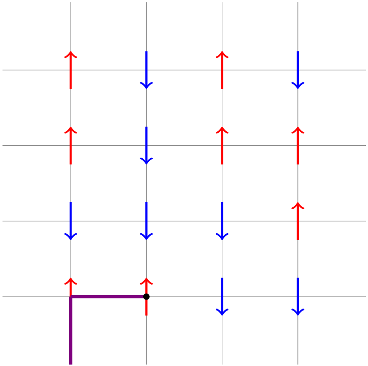
- There are many variations of the worm algorithm in literature.
- Our first implementation of the worm algorithm is as follows:
  1. Start with an arbitrary lattice configuration.[1]
  2. Select an arbitrary point to create the worm.
  3. Grow the worm:
     1. Choose a random direction to move.
     2. If the new point is of the same spin as the old point, add it to the worm with probability 1.
     3. If not, perform a Metropolis-like check. If a flip is favourable, flip and add
     4. This way, we create a worm with equal spins.
     5. Break when the worm head meets its tail or if a flip is no longer favourable.
  4. Measure the observables.
  5. Carry out the desired number of iterations, with a new worm every time.
- There are a few caveats.

[1] We started with an "almost" cold start.

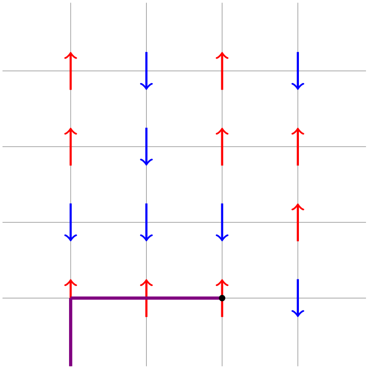
- Given below is a worm at an intermediate step of its growth.
- The lattice below is a small part of the total lattice.



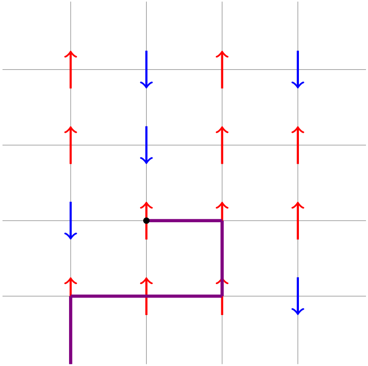
- The worm decides to move right.
- The new site has the same spin as the old site.



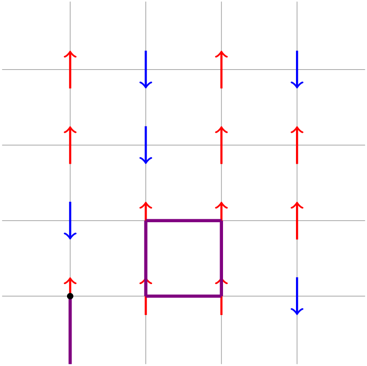
- The worm decides to move right.
- The new site has the opposite spin as the old site.
- Now we perform a metropolis-like check and decide whether to flip or not.

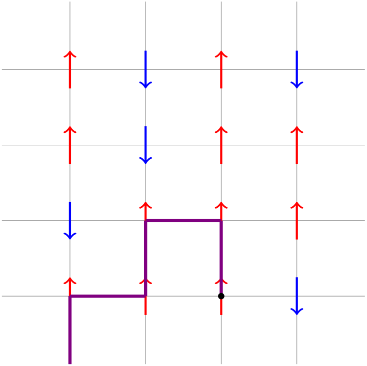


- The worm after a few more steps.



- Caveat: If the worm tries to move to a point that is already a part of the worm, we need to update the head accordingly.
- We want only an even number of bonds between the sites.
- We ensure this by choosing the new head and breaking an old bond appropriately.





- Caveat: If the worm tries to leave the lattice, let it choose a different dirrection.

Presentation Outline

Algorithm behavior - Metropolis

Algorithm behavior - Worm Algorithm

Susceptibility and Heat Capacity - Metropolis

Susceptibility and Heat Capacity - Worm Algorithm

Dynamical Critical Exponent - Metropolis

Dynamical Critical Exponent - Worm Algorithm

Presentation Outline