The Ising model
Computational Physics

Tobias de Jong
s0881260

March 30, 2016

1 Introduction

In this paper we will investigate the different properties of the Ising model in 2D and 3D using Monte Carlo simulation methods. The canonical Metropolis single spin flip method as well as the Wolff cluster flip algorithm are used to obtain values for the critical exponents χ, γ and ν in both 2D and 3D.

2 Theory

2.1 Ising model

2.2 Monte Carlo approximation

3 Algorithms

3.1 Metropolis

The Metropolis algorithm [1] utilizes single spin flips to transition between states. By selecting a random spin at each step we make sure that the transition chance between any two neighbouring states μ, ν is equal:

$$g(\mu, \nu) = g(\mu, \nu) = \frac{1}{N}$$

3.2 Wolff

4 Implementation

Both the Metropolis and the Wolff algorithm were implemented using Python and NumPy. While this choice of language will not result in the fastest implementation it yields easily readable and adaptable code. Moreover, simple libraries exist to parallelise the different runs to keep the total wall clock computation time reasonable.

5 Results

5.1 Thermalization

To visualize the thermalisation we do several runs for each value of β, calculate the magnetisation for each and take the mean across different runs. This way, the plot will be a flat line when thermalized. Results for a 40×40 grid and 100 runs per value of beta is given in figure 1. The obvious conclusion here is that the thermalisation time depends quite strongly on β, especially around the critical point, which corresponds with the interesting part of the simulations. An algorithm which would bypass this

behaviour would thus be beneficial. The Wolff algorithm is such an algorithm, as although it still needs to thermalize, the thermalization time does not diverge around the critical point.

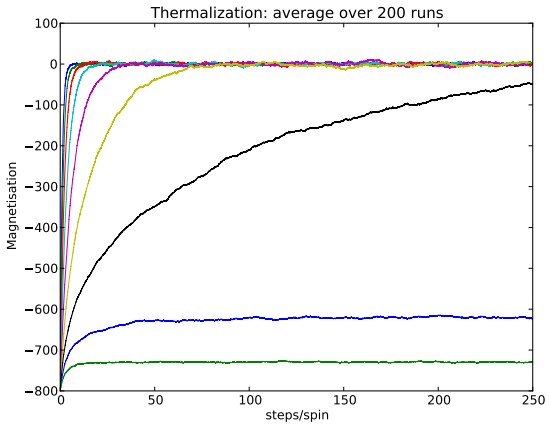

Figure 1: Thermalization

5.2 Auto-correlation

5.3 Finite size scaling

6 Data collapse

References

[1] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *jcp*, 21:1087–1092, June 1953.