Fys3150/4150 - Computational Physics

4. Ising model - Monte Carlo and Metropolis algorithms

Due date **Nov** *overdue*, **2016**

Magnus Christopher Bareid
un: magnucb

November 22, 2016

Abstract

Empty void

Contents

1 Studies of phase transitions in magnetic systems

1.1 Introduction

The aim of this project is to study a widely popular model to simulate phase transitions, the so-called Ising model in two dimensions. At a given critical temperature, this model exhbits a phase transition from a magnetic phase (a system with a finite magnetic moment) to a phase with zero magnetization.

As with all other projects in this course, the important thing is to make the algorithm work. The basic energy calculation of any two-dimensional lattice boils down to this form:

$$E = -J \sum_{<kl>}^{N} s_k s_l \tag{1}$$

where s_k and s_l are ± 1 (up or down spin), N is the total spins in the lattice, J expresses the strength of interaction between neighbouring spins, which are referred to in the summation by $< kl >$ as it sums up interaction only between lattice-neighbours near index k and l.

Through the course of this report, we shall investigate different variations on differently-sized lattices and their modelled physical properties.

1.2 Analytical model of 2×2 lattice

For our first trick, we shall simply investigate a 2×2 lattice of spins and its properties: its expectation value for energy and for absolute magnetic moment (mean magnetization), its specific heat and magnetic susceptibility.

1.2.1 The underlying math

Firsly, we'll bee needing the partition function

$$Z = \sum_i e^{-\beta E_i} \tag{2}$$

where E_i is a spin configuration's energy level, which i counts over, and $\beta = 1/(k_B T)$ as a physical variable depending on temperature - a parameter which will become important in later sections.

The lattice's energy is given from the same form as equation (1). The magnetic moment of such a lattice's spin combination is also important, and will yield values in a similar, but different pattern to the energy variations. For this we use:

$$M = \sum_{i=1}^{N} s_i \tag{3}$$

where M_i signifies a lattice location's spin orientation. This reveals to us a set of energies and magnetizations that the 2×2 lattice and its possible spin configurations may yield.

# $s_i = 1$	# states	E	M
4	1	-8J	4
3	4	0	2
2	4	0	0
2	2	8J	0
1	4	0	-2
0	1	-8J	-4

Table 1: Overview over how number of spins in a direction determines energy and magnetization of a lattice, and its multiplicity.

For the related expectation values, these are formed by multiplying by the probability of each state, which is determined with a natural exponent and the partition function that we elaborated earlier:

$$< E^n > = \frac{1}{Z} \sum_i E_i^n e^{-\beta E_i} \tag{4}$$
$$< |M|^n > = \frac{1}{Z} \sum_i M_i^n e^{-\beta E_i}$$

where i iterates over all possible combinations - not over all locations in a lattice.

Furthermore, the formulation for specific heat and magnetic susceptibility reads thus:

$$C_V = \frac{< E^2 > - < E >^2}{k_B T^2} \tag{5}$$
$$\chi = \frac{< M^2 > - < M >^2}{k_B T}$$

1.2.2 Results

When all of these energy and magnetic combinations are input to a simple python script, the analytical values which we were supposed to investigate, as mentioned in the beginning of the section, may be evaluated:

$< E >$	-7.98392834375 ≈ -7.9839
$< M >$	3.99464293099 ≈ 3.9946
C_V	0.128329327457 ≈ 0.12833
χ	0.0160429580649 ≈ 0.016043

Table 2: Analytical values as determined by a python script.

1.3 Making a working numerical model

Now comes the fun part, modelling the previous system numerically, using Monte Carlo style in C++.

1.3.1 How the script works

The script begins by taking in several values of the lattice system, such as lattice length, how many Monte Carlo cycles which we will cycle the system over, temperature for the lattice to hold. The temperature in this case is set to 1 for simplicity's sake. Physical units and confusions apply.

Then the inital lattice matrix is set up with corresponding energy and magnetization quantities; for this problem's purposes we simply start with all spins up.

At every iteration of the Monte Carlo cycle, the lattice is adjusted according to principles of physics: A random spin is chosen to turn its value. If the new energy state has less energy than the previous one, the new energy state is then accepted its new properties for energy and magnetization are applied to the system.

Expectation values are yielded from dividing by number of Monte Carlo cycles which the system has had to go through in order to achieve its current properties. The iterated values are stored to data files for later use, expectation values are printed to screen along with specific heat and magnetic susceptibility.

1.3.2 Results and discussion

A run for several Monte Carlo simulation lengths yielded these values, which are compared with analytical results as shown in table 2.

Quant.	Result	≈ Δ Ana.
$< E >$	-7.2	0.7839
$< M >$	3.63	0.3646
C_V	5.76	5.632
χ	1.44	1.424

Table 3: Results after 10^1 Monte Carlo cycles.

Quant.	Result	≈ Δ Ana.
$< E >$	-7.92	0.0639
$< M >$	3.96	0.0346
C_V	0.6336	0.5053
χ	0.1584	0.1423

Table 4: Results after 10^2 Monte Carlo cycles.

Quant.	Result	≈ Δ Ana.
$< E >$	-7.984	0.0001
$< M >$	3.992	0.0026
C_V	0.127744	0.0006
χ	0.031936	0.0159

Table 5: Results after 10^3 Monte Carlo cycles.

Quant.	Result	≈ Δ Ana.
$< E >$	-7.9856	0.0017
$< M >$	3.995	0.0004
C_V	0.114993	0.0133
χ	0.015575	0.0005

Table 6: Results after 10^4 Monte Carlo cycles.

Quant.	Result	≈ Δ Ana
$< E >$	-7.982	0.0019
$< M >$	3.99396	0.0007
C_V	0.143676	0.0153
χ	0.0182035	0.0022

Table 7: Results after 10^5 Monte Carlo cycles.

Quant.	Result	≈ Δ Ana.
$< E >$	-7.98436	0.0004
$< M >$	3.99479	0.0001
C_V	0.124875	0.0035
χ	0.0156048	0.0004

Table 8: Results after 10^6 Monte Carlo cycles.

It would seem that already after having passed 10^3 Monte Carlo cycles, the accuracy starts to become surprisingly impressive, and varies less. However, the clear trend is of course that the more one increases the number of cycles for the program to run through, the more accurate results. My own verdict would claim that any run of the script with more than 10^6 Monte Carlo cycles would be accurate enough for most purposes.

2 Appendix

2.1 Code and GitHub

All my code is located at this address:
`https://github.com/magnucb/p4`

References

[1] Jensen, M 2016, `ParaIsingModel.cpp`,

viewed 22^{th} of November 2016, `https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Programs/ParallelizationMPI/ParaIsingModel.cpp`.

[2] Jensen, M 2016, Computational Physics Lecture Notes Fall 2015:

13.2.2: Canonical Ensemble &

13.3.1: Ising Model and Phase Transitions in Magnetic Systems,

viewed 22^{th} of November 2016, `http://compphysics.github.io/ComputationalPhysics/doc/Lectures/lectures2015.pdf`