Analytical and Numerical Solution in The One-Dimensional Ising Model

University of Seoul – Department of Physics

2022. 08. 25.

2021550026

전성배

Concept

1. Analytical Solution in The One-Dimensional Ising Model

1-1. Introduction to Ising Model

A simple but fundamental model that takes interaction only between nearest-neighbor spins into account

In Ising model, there are only up and down spins.

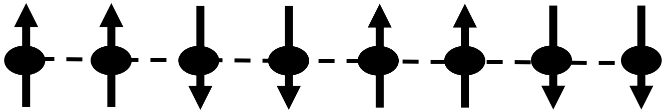

Fig. 1.1. The one-dimensional Ising model

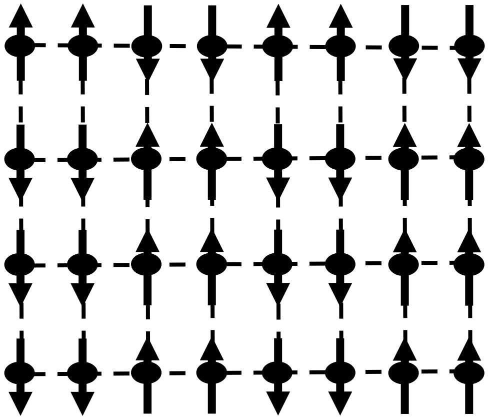

Fig. 1.2. The two-dimensional Ising model

1. Analytical Solution in The One-Dimensional Ising Model

Fig. 1.3. Parallel links

Fig. 1.4. Antiparallel links

The energy of parallel link is $-J$

The energy of antiparallel link is $+J$

$$\boldsymbol{N = N_{+} + N_{-}}$$

$$\boldsymbol{magnetization\ M = \mu\,(N_{+} - N_{-})}$$

$$\mathbf{interaction\ energy\ E_i = -\,J\,(N_{++} + N_{--} - N_{+-} - N_{-+})}$$

1-2. Partition Function

$$E(\{\sigma_i\}) = -J\sum_{i=1}^{N}\sigma_i\sigma_{i+1} - \mu B\sum_{i=1}^{N}\sigma_i$$

μ: magnetic moment
J: coupling constant
B: external magnetic field
$\beta = 1/k_B T$

by periodic boundary condition

$$Z = \sum_{\{\sigma_i\}} e^{-\beta E(\{\sigma_i\})} = \sum_{\sigma_1}\dots\sum_{\sigma_N}\exp[\beta\sum_{i=1}^{N} J\sigma_i\sigma_{i+1} + \frac{\mu}{2}(\sigma_i + \sigma_{i+1})]$$

$$M(\sigma_i, \sigma_{i+1}) = \exp[\beta\sum_{i=1}^{N} J\sigma_i\sigma_{i+1} + \frac{\mu}{2}(\sigma_i + \sigma_{i+1})],$$

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{e^{\beta+h}} & \boldsymbol{e^{-\beta J}} \\ \boldsymbol{e^{-\beta J}} & \boldsymbol{e^{\beta(J+h)}} \end{pmatrix}$$

$$Z = \sum_{\sigma_1}\dots\sum_{\sigma_N} A_{\sigma_{1,},\sigma_{2,}}A_{\sigma_2,\sigma_{3,}}\dots A_{\sigma_{N-1},\sigma_N}A_{\sigma_N,\sigma_1} = tr(A^N) = tr\left(A_{diag}^N\right) = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}^N = \lambda_1^N + \lambda_2^N$$

1. Analytical Solution in The One-Dimensional Ising Model

$$|A - \lambda I| = 0$$

$$\begin{vmatrix} e^{\beta(J+\mu B)} - \lambda & e^{-\beta J} \\ e^{-\beta J} & e^{\beta(J-\mu B)} - \lambda \end{vmatrix} = 0$$

$$\lambda = e^{\beta J}\cosh(\beta\mu B) \pm \sqrt{e^{2\beta J}\cosh^2(\beta\mu B) - 2\sinh(2\beta J)}$$

$$if\ \lambda_1 > \lambda_2\ then\ \ {\lambda_1}^N \gg\ {\lambda_2}^N,$$

$$\boldsymbol{Z} \approx \boldsymbol{\lambda_1}^{\boldsymbol{N}} = \left[e^{\beta J}\cosh(\beta\mu B) + \sqrt{e^{2\beta J}\cosh^2(\beta\mu B) - 2\sinh(2\beta J)}\right]^{\boldsymbol{N}}$$

$$for\ N \to \infty$$

1-3. Energy, Heat Capacity, Magnetization and Susceptibility

Free energy

$$F = -k_B T ln[Z] = -N k_B T ln[e^{\beta J} \cosh(\beta \mu B) + \sqrt{e^{2\beta J} \cosh^2(\beta \mu B) - 2 \sinh(2\beta J))}]$$

Entropy at B = 0

$$F = -k_B T ln[Z] = -N k_B T ln\left[e^{\beta J} + e^{-\beta J}\right] = -N k_B T ln[2 cosh \beta J]$$

$$\boldsymbol{S = -\frac{\partial F}{\partial T} = N k_B \left[\ln\left(2 cosh \frac{J}{k_B T}\right) - \frac{J}{k_B T}\left(tanh \frac{J}{k_B T}\right)\right]}$$

$$\mathbf{E = F + ST = -NJ \tanh \frac{J}{k_B T}}$$

1. Analytical Solution in The One-Dimensional Ising Model

Heat Capacity at B = 0

$$C_V = \frac{dE}{dT} = \frac{NJ^2}{k_B T^2}\,\mathrm{sech}^2\,\frac{J}{k_B T}$$

Magnetization

$$M = -\frac{\partial F}{\partial B} = \beta\mu N k_B T\,\frac{e^{\beta J}\sinh\beta\mu B + \dfrac{e^{2\beta J}\sinh\beta\mu B\cosh\beta\mu B}{\sqrt{e^{2\beta J}\cosh^2\beta\mu B - 2\sinh 2\beta J}}}{e^{\beta J}\cosh\beta\mu B + \sqrt{e^{2\beta J}\cosh^2\beta\mu B - 2 sinh\, 2\beta J}}$$

In a weak magnetic field $\sinh\beta\mu B \approx 1, \cosh\ \beta\mu B \approx 1$

$$M \approx N\mu^2\beta\, e^{2\beta J} B = \frac{N\mu^2}{k_B T}\, e^{\frac{2J}{k_B T}}\, B$$

Susceptibility

$$\chi = \mu_0 \frac{N\mu^2}{k_B T}\, e^{\frac{2J}{k_B T}}\, B$$

Graph

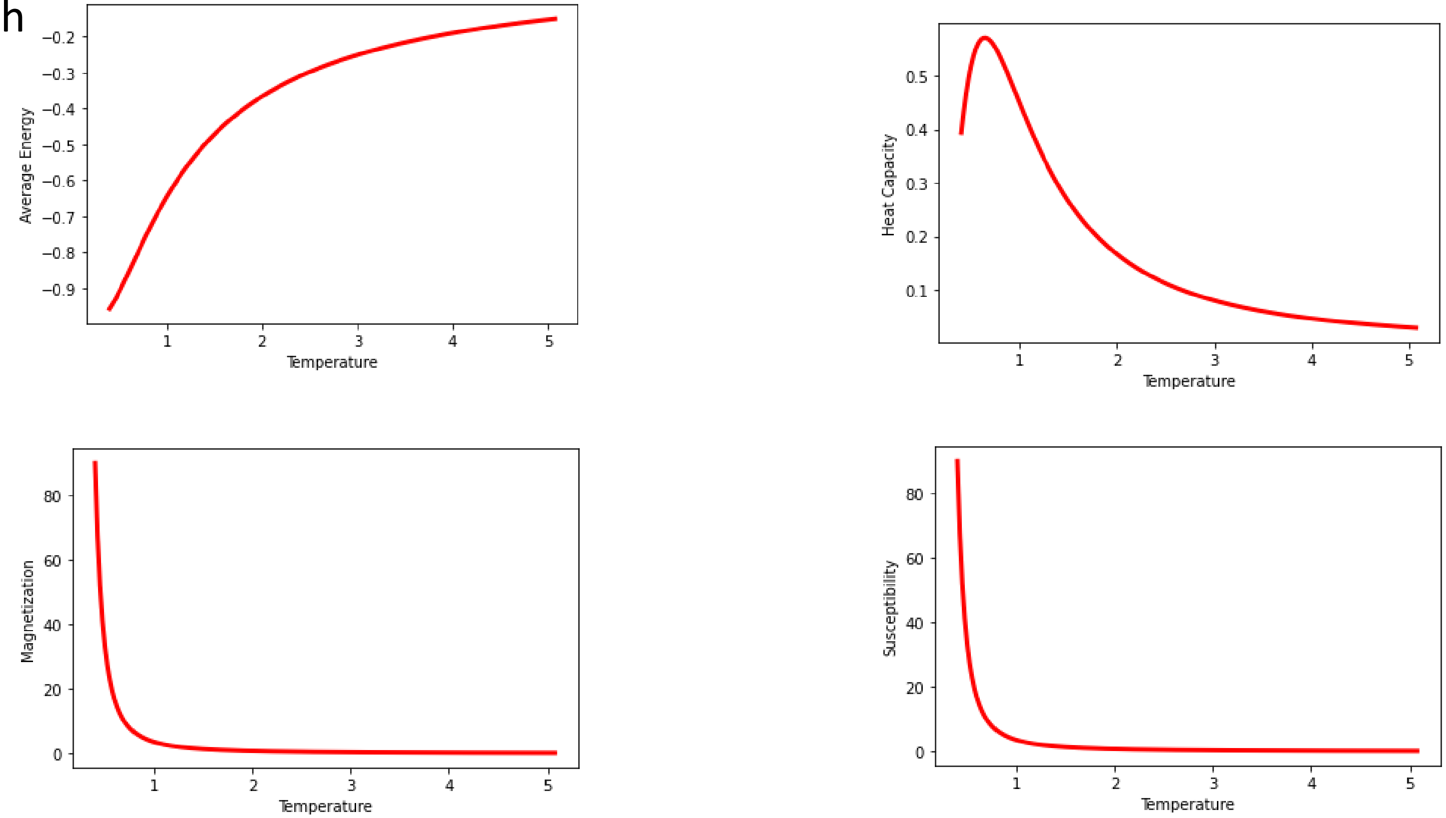

Fig. 1.4. Analytical solution graph in the one-dimensional Ising model

2-1. Monte Carlo Method

Monte Carlo Method is algorithm to get approximation using repeated random sampling.

For example, π approximation is caculated from Monte Carlo Method in picture.
(It is called Direct Sampling)

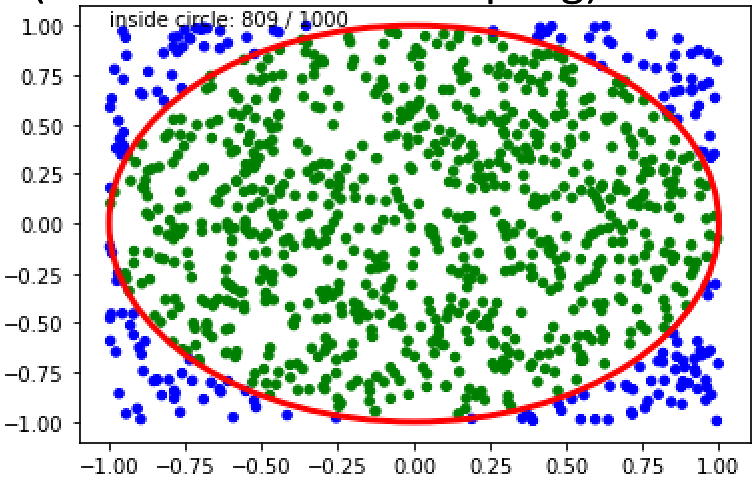

Fig. 2.1. pi approximation by Monte Carlo method

$$\frac{the\ number\ of\ ball\ in\ Circle}{the\ number\ of\ ball\ in\ Square} = \frac{Area\ of\ Circle}{Area\ of\ Square}$$

$$= \frac{\pi \times 1^2}{2 \times 2}$$

Thus, $\pi = 4 \times \frac{the\ number\ of\ ball\ in\ Circle}{the\ number\ of\ ball\ in\ Square}$

2-2. Markov Chain (MC)

Markov Chain Model is probabilistic model that present situation is only depended on last situation. After repeating, it reaches **Stationary Distribution.**

For example, π approximation can be calculated from Markov Chain.

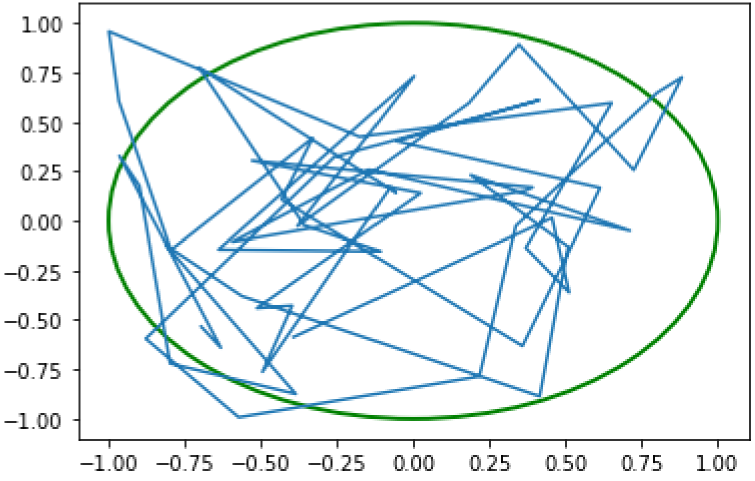

Fig. 2.2. pi approximation by Markov Chain

Start at (-1,-1) and Add random number between -1 and 1 each x and y coordinates.

If new coordinate is inside square (that x, y is from -1 to 1), new coordinate is saved.
If not, last coordinate is maintained.

And count the number of inside the circle.

$$\pi = 4 \times \frac{counted\ number}{total\ trial}$$

2-3. Markov Chain Monte Carlo (MCMC) and Metropolis Algorithm

Markov Chain + Monte Carlo

=

(Repeated random sampling)

+ (State only dependent on last state)

One simple example of Markov Chain Monte Carlo(MCMC) is Metropolis Algorithm.

Metropolis Algorithm

Select initial value: $x_0 \sim \pi(x)$

Repeat: $i = 1,2,\ldots$

- Generate test sample
  $x^* \sim q(x_{i+1} | x_i)$

- $\alpha(x^*, x_i) = \min\{1, \frac{\pi(x^*)}{\pi(x_i)}\}$

- Generate uniform random number:
  $u \sim U(0,1)$

$if\ u < \alpha \qquad x_{i+1} = x^*$

$else \qquad x_{i+1} = x_i$

2-4. Application Metropolis Algorithm to The One-Dimensional Ising Model

Select initial value: $x_0 \sim \pi(a)$

$$\pi(a) = \frac{e^{\beta E_a}}{Z},$$ (Boltzmann distribution)

Flip and get $\pi(b)$

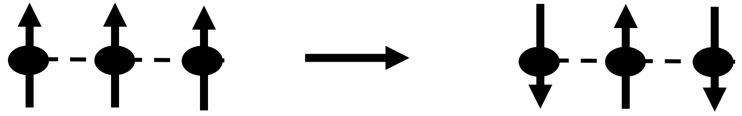

Fig. 2.3. Flip in

$$x^* \sim \pi(b) = \frac{e^{\beta E_b}}{Z}$$

- $\alpha(x^*, x_i) = \min\{1, \frac{\pi(b)}{\pi(a)}\}$

Generate uniform random number:

$$\mathrm{u} \sim U(0,1)$$

$$\begin{array}{ll} if\ u < \alpha & x_{i+1} = x^* \\ else & x_{i+1} = x_i \end{array}$$

2-5. Detailed balance

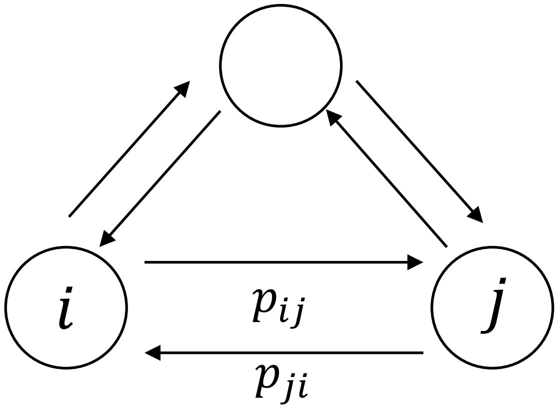

Fig. 2.4. Detailed balance diagram

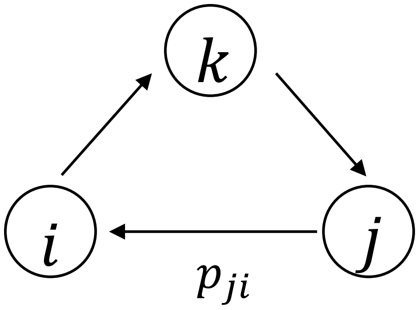

Fig. 2.5. Global balance diagram

$$\pi_i p_{ij} = \pi_j p_{ji} \ \forall i, j$$

The amount of probability flowing from i to j

= The amount of probability flowing from j to i

That represents no net flux of probability

and **Stationary Distribution**

2. Numerical Solution in The One-Dimensional Ising Model

$$case\ 1)\ \pi(a) > \pi(b)$$

$$p(a \to b) = \min\left[1, \frac{\pi(b)}{\pi(a)}\right] = \frac{\pi(b)}{\pi(a)}$$

$$\Rightarrow \pi(a)p(a \to b) = \pi(b)$$

$$p(b \to a) = \min\left[1, \frac{\pi(a)}{\pi(b)}\right] = 1$$

$$\Rightarrow \pi(b)p(b \to a) = \pi(b)$$

$$\therefore \boldsymbol{\pi(a)p(a \to b) = \pi(b)P(b \to a)}$$

$$case\ 2)\ \pi(a) < \pi(b)$$

$$p(a \to b) = \min\left[1, \frac{\pi(b)}{\pi(a)}\right] = 1$$

$$\Rightarrow \pi(a)p(a \to b) = \pi(a)$$

$$p(b \to a) = \min\left[1, \frac{\pi(a)}{\pi(b)}\right] = \frac{\pi(a)}{\pi(b)}$$

$$\Rightarrow \pi(b)p(b \to a) = \pi(a)$$

$$\therefore \boldsymbol{\pi(a)p(a \to b) = \pi(b)P(b \to a)}$$

2. Numerical Solution in The One-Dimensional Ising Model

$$hamiltonian\ H =- \mathrm{J}\,(\mathrm{N}_{++} + \mathrm{N}_{--} - \mathrm{N}_{+-} - \mathrm{N}_{-+})$$

$$< E > = \frac{1}{N}\frac{1}{MC} < H >$$

$$< M >= \frac{1}{N}\frac{1}{MC} |\,(N_{+} - N_{-})\,|$$

$$< C_V >= \frac{\partial < E >}{\partial T} = \frac{\partial \beta}{\partial T}\frac{\partial < E >}{\partial \beta} = -\beta^2 \frac{\partial < E >}{\partial \beta} = \beta^2 \frac{\partial^2}{\partial \beta^2} logZ = \beta^2 (< E^2 \succ \ - < E >^2)$$

$$< S >= \frac{\partial < M >}{\partial T} = \frac{\partial \beta}{\partial T}\frac{\partial < M >}{\partial \beta} = -\beta^2 \frac{\partial < M >}{\partial \beta} = \beta^2 (< M^2 \succ \ - < M >^2)$$

2. Numerical Solution in The One-Dimensional Ising Model

2-6. Python code

```python
import random, math
import matplotlib.pyplot as plt
import numpy as np

global n, J, N, kb

n = 20
J = 1
kb = 1.3
```

Fig. 2.6. python code – part1(import and define)

```python
def mk_matrix_1d():
    data_init = np.zeros([n])
    data_init = np.random.choice([-1,1], size = (n))
    return data_init
```

Fig. 2.7. python code – part1(make random matrix)

```python
def pi_value_1d(data_unit, beta):

    Energy = 0

    x = random.randint(0,(n-1))

    right = data_unit[(x+1)%n]
    left = data_unit[(x-1)%n]
    Energy += data_unit[x] * (right + left)

    pi_a = math.exp((+1)*beta*Energy)
    pi_b = math.exp((-1)*beta*Energy)
    pi_divided = pi_b / pi_a

    if pi_divided <= 1:
        if random.random() < pi_divided:
            data_unit[x] = -data_unit[x]
    elif pi_divided > 1:
        data_unit[x] = -data_unit[x]

    return data_unit
```

Fig. 2.8. python code – part3(flip)

2. Numerical Solution in The One-Dimensional Ising Model

```python
def matrix_repeat_1d(number, matrix_matrix):
    for i in range(number):
        matrix_matrix = pi_value_1d(matrix_matrix, 1/0.1)
    return matrix_matrix
```

Fig. 2.9. python code – part3(repeat)

```python
def cal_temperature(matrix_cal, beta):

    Sum_Energy = 0
    Sum_Energy_square = 0
    Avg_Energy = 0
    Avg_Energy_square = 0
    Sum_M_square = 0
    Sum_M = 0
    Avg_M=0
    Avg_M_square = 0

    Sum_Hc = 0
    Sum_Sus = 0

    #Repeat N times

    for j in range(N):

        H = 0
        H_square = 0


        #Calculate Hamiltonian

        for x in range(n):
            right = matrix_cal[(x+1)%n]
            left = matrix_cal[(x-1)%n]

            h_temp =  -J * (matrix_cal[x] * (right + left)) / 2

            H += h_temp / (n)
            H_square += h_temp**2 / (n)

        Sum_Hc += beta * beta * (H_square - H*H)

        M = np.sum(matrix_cal)

        Sum_Energy += H
        Sum_Energy_square += H_square
        Sum_M += abs(M)
        Sum_M_square += M*M
        matrix_cal = pi_value_1d(matrix_cal, beta)

```

Fig. 2.10. python code – part3(caculate)

2. Numerical Solution in The One-Dimensional Ising Model

2-7. Result

T = 0.4

0.4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0.5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0.6	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1
0.7	-1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0.8	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	-1	-1	-1	-1
0.9	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	1	1	1	1	1	1	1
1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	1	1	1	-1
1.1	-1	-1	-1	-1	1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	1	1
1.2	-1	-1	1	1	-1	-1	-1	-1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1
1.3	-1	-1	-1	-1	1	-1	1	1	1	-1	1	1	1	-1	-1	-1	-1	-1	-1	-1
1.4	-1	1	1	1	1	1	1	-1	-1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1
1.5	-1	1	1	1	1	1	-1	1	1	1	1	-1	-1	-1	-1	-1	-1	1	-1	-1
1.6	1	1	1	1	1	1	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1
1.7	-1	1	1	-1	1	-1	-1	-1	1	1	-1	1	-1	1	1	1	1	1	-1	1
1.8	1	1	-1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1	1	1	1	1	1
1.9	1	-1	-1	1	1	1	1	1	1	-1	-1	1	-1	1	-1	1	1	1	1	1
2	1	1	1	1	1	1	1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	1	1
2.1	1	1	1	-1	1	-1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
2.2	1	1	1	-1	-1	1	-1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
2.3	1	1	-1	1	1	1	1	1	1	1	1	-1	1	1	1	1	1	1	1	-1
2.4	1	1	1	1	1	1	1	1	1	1	-1	1	-1	-1	1	1	-1	-1	1	1
2.5	1	1	1	1	1	-1	-1	1	1	1	1	1	1	1	-1	-1	1	1	1	1
2.6	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	-1	-1
2.7	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	1
2.8	1	-1	1	1	1	1	1	-1	-1	-1	1	-1	-1	-1	-1	1	1	-1	-1	1

T = 2.8

Fig. 2.11.(a) distribution diagram

T = 2.9

2.9	1	1	1	1	-1	-1	-1	1	1	1	1	1	1	-1	-1	1	1	-1	-1	1
3	1	1	1	-1	-1	1	1	1	1	1	1	-1	1	1	1	-1	-1	-1	1	1
3.1	1	1	-1	1	-1	1	-1	-1	1	1	-1	-1	-1	1	1	1	1	1	-1	-1
3.2	-1	-1	1	1	-1	-1	-1	-1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	-1
3.3	1	-1	-1	1	1	-1	1	-1	-1	-1	-1	1	-1	1	1	1	-1	-1	-1	1
3.4	-1	-1	-1	-1	-1	1	-1	1	1	1	-1	-1	1	1	1	-1	-1	1	1	1
3.5	-1	-1	1	1	1	-1	1	1	1	1	1	-1	1	-1	1	-1	-1	-1	-1	-1
3.6	1	-1	-1	1	1	1	1	1	1	-1	-1	1	-1	1	-1	-1	-1	1	-1	-1
3.7	-1	-1	1	1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	-1	1	-1	-1	-1
3.8	1	-1	1	-1	-1	-1	-1	1	1	1	-1	-1	1	-1	-1	-1	1	1	1	1
3.9	1	1	1	-1	1	1	-1	-1	1	-1	-1	1	-1	-1	1	1	-1	1	-1	1
4	-1	1	-1	-1	-1	1	-1	1	1	1	-1	1	1	-1	1	-1	1	-1	-1	-1
4.1	-1	1	-1	-1	-1	-1	-1	1	1	1	-1	1	1	1	1	-1	-1	-1	1	-1
4.2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	-1	1	-1	-1	-1	1	-1
4.3	1	-1	-1	-1	-1	-1	1	-1	-1	-1	1	1	1	1	1	1	1	-1	1	-1
4.4	-1	-1	-1	-1	-1	1	1	1	-1	-1	-1	1	-1	1	-1	1	1	1	1	1
4.5	-1	1	1	-1	1	1	-1	-1	1	-1	-1	-1	1	-1	-1	1	-1	-1	1	1
4.6	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	-1	-1	1	1	1	1	1
4.7	-1	-1	1	1	1	-1	-1	1	-1	-1	-1	1	-1	-1	1	-1	-1	1	1	-1
4.8	-1	-1	-1	-1	-1	-1	-1	1	1	-1	-1	1	1	-1	1	-1	-1	1	1	-1
4.9	-1	-1	1	-1	1	-1	-1	1	-1	1	-1	-1	1	1	-1	1	-1	-1	1	-1
5	1	-1	1	-1	-1	-1	1	-1	1	-1	1	1	1	1	1	1	1	1	-1	-1

T = 5

Fig. 2.11.(b) distribution diagram

2. Numerical Solution in The One-Dimensional Ising Model

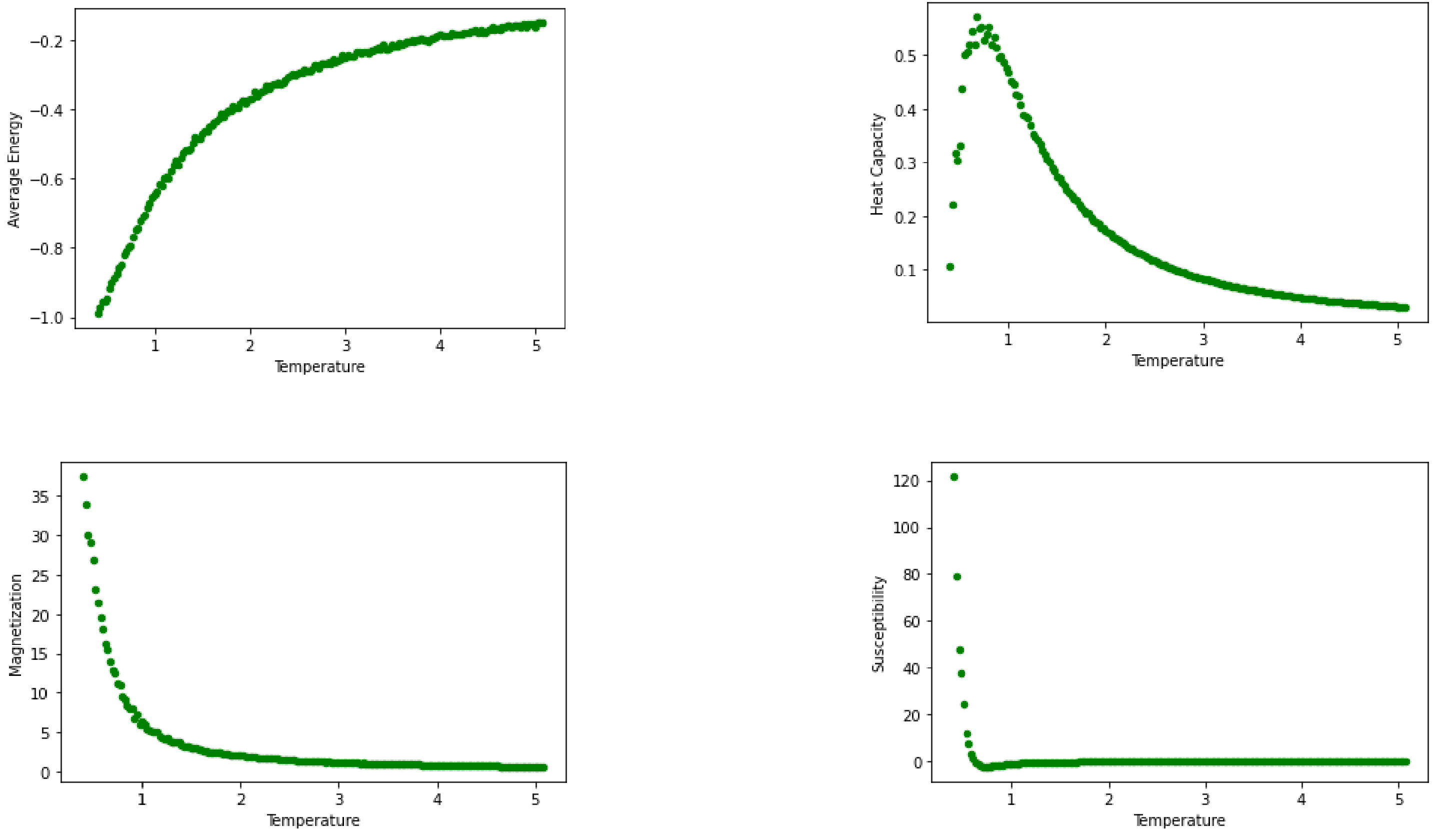

Fig. 2.12. Analytical solution graph in the one-dimensional Ising model

2-8. Comparison of Analytical and Numerical Solution

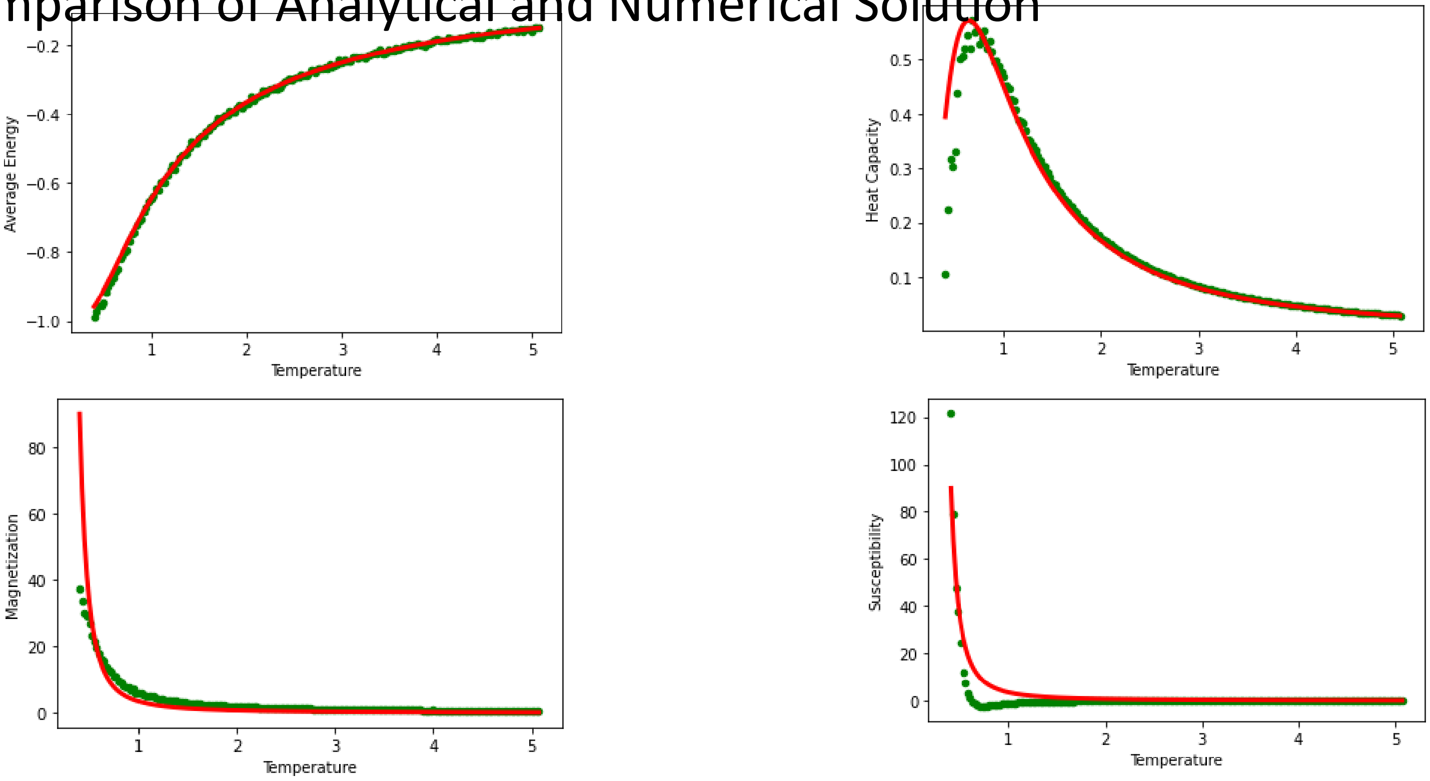

Fig. 2.13. Comparison of analytical and numerical solution graph in the one-dimensional Ising model

3. Reference

1. Daijiro Yoshioka: Statistical Physics (Springer, 2007)
2. David P.Landau & Kurt Binder: A guide to Monte Carlo Simulation in Statistical Physics Second Eddition (Cambridge, 2005)
3. Giuseppe Grosso, Giuseppe Pastori Parravicini:Solid State Physics (Second Edition 2014)
4. Werner Krauth: Statistical Mechanics: Algorithms and Computations (Oxford master serie s in physics, 2006)
5. Siddhartha Chib, Edward Greenburg: Understanding the Metropolis_Hastings Algorithm (The American Statistician, 1995)