King Fahd University of Petroleum & Minerals

Thermal and Statistical Physics

Physics 430

Project

Two Dimensional Ising Model

By : Abdullah alkathiry

ID: 201765290

Table of Contents

Introduction

The Ising model is a mathematical model of ferromagnetic materials in statistical mechanics. The model consists of discrete variables that represent magnetic dipole moments of atomic "spins". Spins are arranged in a lattice, each spin is allowed to interact with its nearest neighbors. Neighboring spins that agree have lower energy than those that disagree; the system tends to have the lowest energy, but heat resists this alignment of spins. When the dipole is spin up the discrete variable is 1 when it spins down it is -1

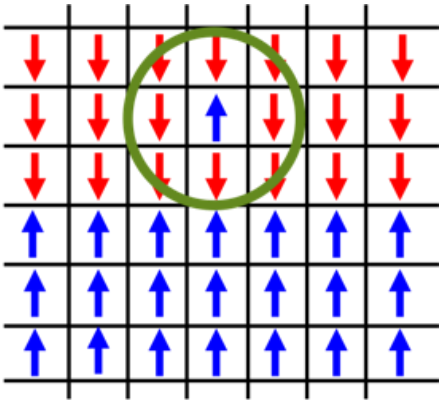
Historical review

The Ising model, was named after the physicist Ernst Ising. The Ising model was invented by the physicist Wilhelm Lenz in1920, and he gave it as a problem to his student Ernst Ising. The one-dimensional Ising model was solved analytically by Ising alone in his thesis in 1925. it has no phase transition, and he concludes from it that the higher dimension will not have a phase transition [5].

The two-dimensional square-lattice Ising model is much harder and was only given an analytic description much later, by Lars Onsager in 1944. It is usually solved by a transfer-matrix method. The two-dimensional square-lattice Ising model is one of the simplest statistical models that have a phase transition. However, The three-dimensional model does not have an analytical solution yet.

The model

In Ising model the only interaction considered is the nearest neighbors interaction. When two neighbors aligned the will have energy -j if they opposite they will have energy j.



The total energy of the lattice is given by

$$U = -j \sum_{\text{neighbor}} s_i s_j + \mu B \sum_i s_i$$

Where we sum across different neighbors, j is the constant of energy units, s_i is the spin of site i can be up "1" or down "-1". B is the external magnetic field μ is the dipole magnetic moment.

To solve the system analytically we need to find the partition function:

$$Z = \sum_{all\ states} e^{-\beta U}$$

Summing over all possible states

For lattice of size 10×10 , the number of possible states is $2^{10^2} = 2^{100}$ which is very larg number that is very hard to compute analytically and is very hard to simulate numerically using ordinary Monte Carlo method thus we will use the metropolis algorithm for the calculation

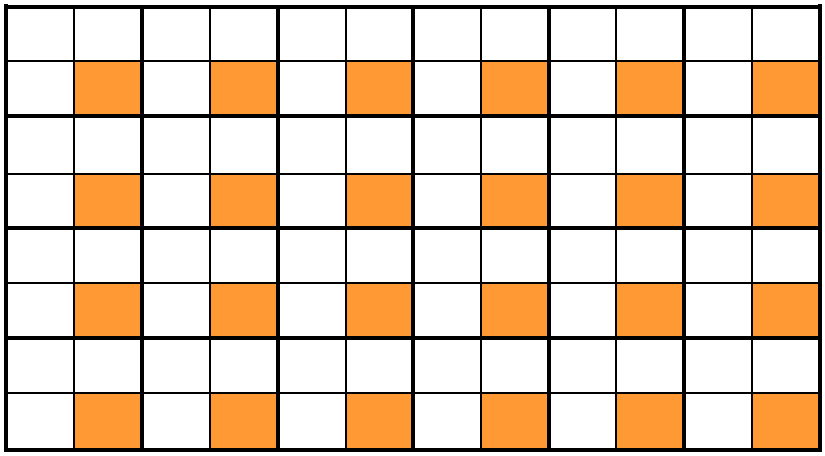
The Metropolis Algorithm

Metropolis Algorithm named after Nicholas Metropolis a Greek American physicist. The algorithm is actually a modified Monte Carlo method, where, instead of choosing configurations randomly, then weighting them by factor of $e^{-\frac{E}{kT}}$, we choose configurations with a probability $e^{-\frac{E}{kT}}$ and weight them equally. [4]

We start with some configuration and then select a random site in the lattice then calculate the energy difference of the system if the spin is flipped. If the energy is less than the original configuration, then we flip if the energy is greater. then we generate a random number between 0 and 1 and compare it with $e^{-\frac{E}{kT}}$ if it is less we flip otherwise we do not flip and choose another random site and apply the same procedure.

To utilize the vectorization operation in MATLAB to speed up the code we modify the way of choosing the random sites for the flips. we compute the energy difference for each site in the

lattice, but we cannot flip two neighboring sites at the same time. So, to avoid that I divide the lattice into smaller lattices each with size, 2×2 and apply the algorithm to the same element in all sites like the following



So, we choose a random site from these four, say we choose the orange same as the figure above. we apply the Metropolis algorithm at all orange sites simultaneously. This way speeds up the code tremendously.

Simulation

To do simulation it is better to work with dimensionless quantities. Therefore we define the following:

Temperature: $T_{sim} = \frac{k_b T}{j}$ energy per site: $E = \frac{U}{N\, j}$

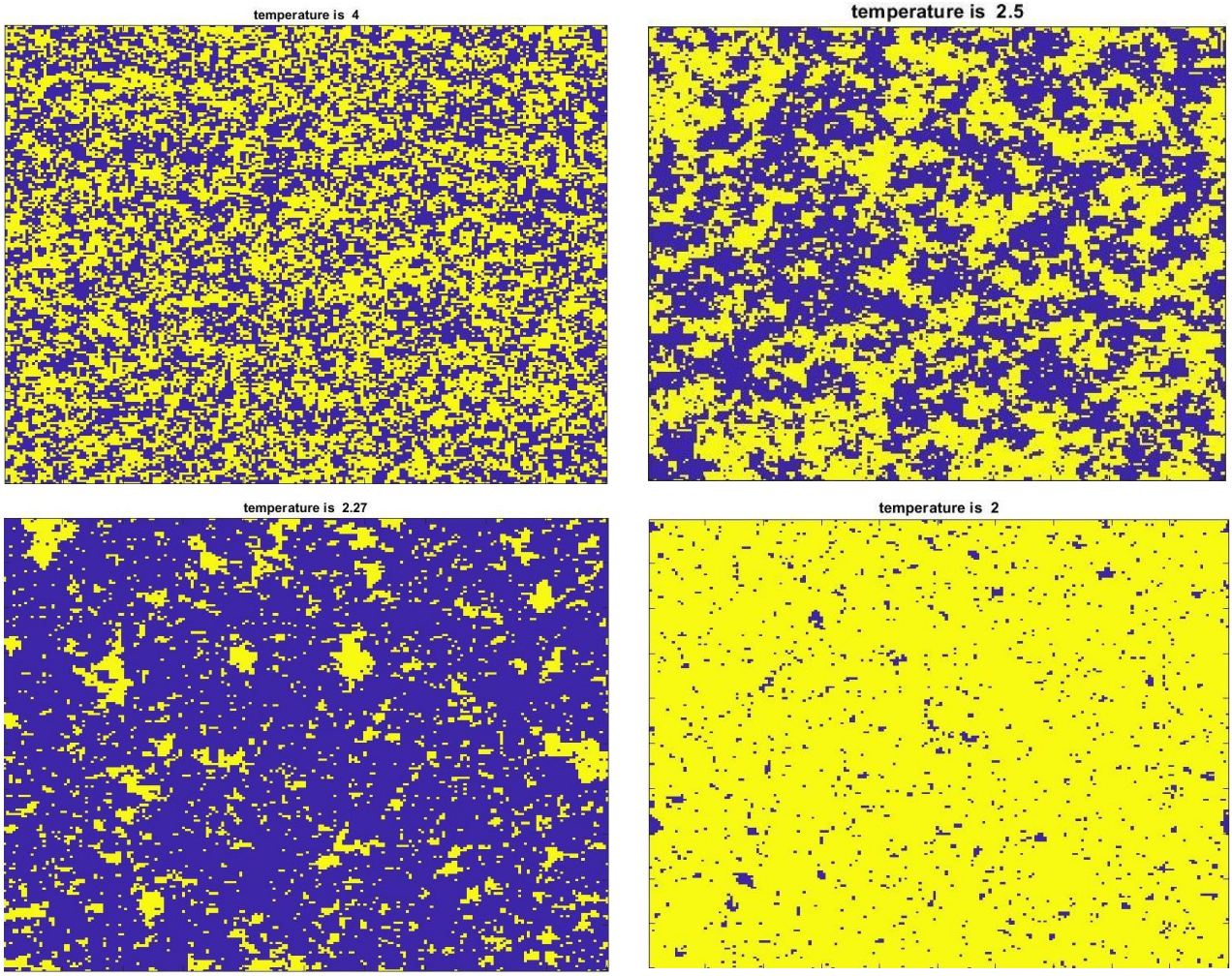
magnetization per site $M_{sim} = \frac{M}{\mu\, N}$ magnetic susceptibility per site $\chi_{sim} = \frac{\chi}{\mu^2 N}$

magnetic field $B_{sim} = \frac{\mu B}{j}$

where k_b is the Boltzmann constant. the subscript sim donates that is the dimensionless quantities used in the simulation (subscript is written here for illustration it will be omitted in the following parts)

Without magnetic field

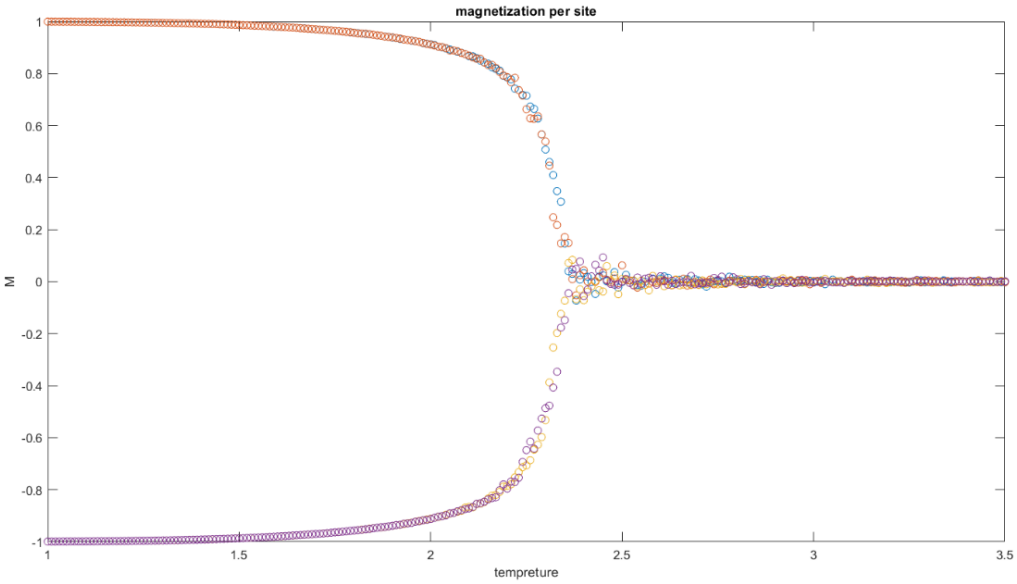
The lattice



We start with some random configuration as discussed above then after running the code for 2000 times we show the images of the lattice at some selected temperature. (Yellow color for spin up and blue for spin down)

To find the Energy, magnetization. Heat capacity, susceptibility, and correlation length then running the code 2000 times for each temperature then taking the average of all the iterations except the first 200 iterations to allow the system to reach the equilibrium. The temperature range of the simulation is from 1 to 3.5 in steps of 0.01. MATLAB is used for the simulation with the parallel computing toolbox to increase the speed of the code.

Magnetization

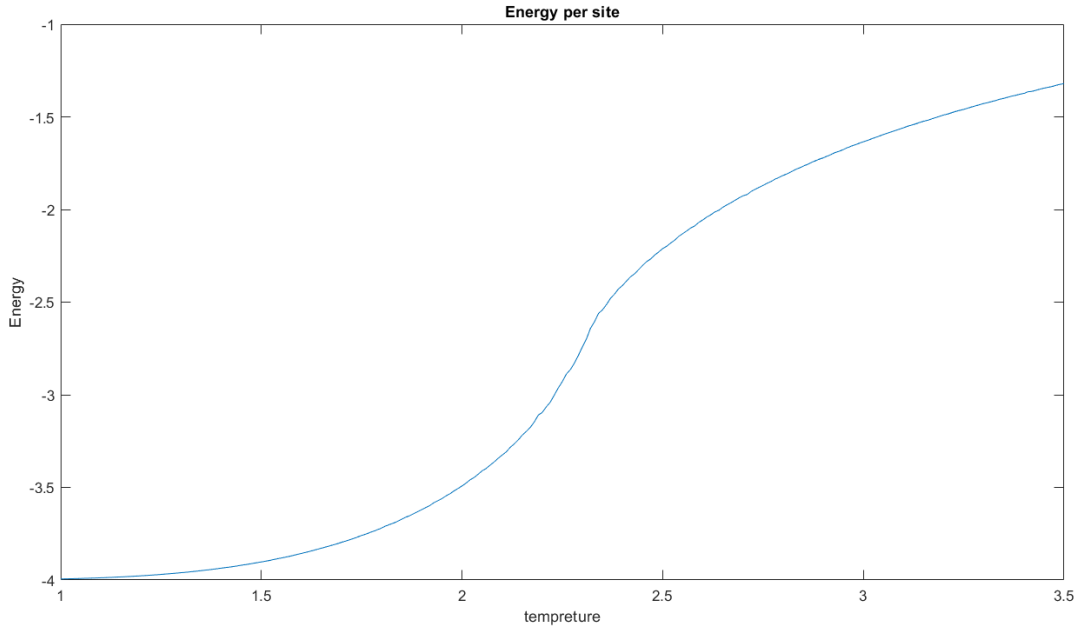


We notice that at low temperature the spins are aligned either up or down. Near the critical temperature, the spins start to have a random distribution. At a high temperature far from the critical temperature. The spins are completely random and there as many up as many downs.

Note: we don't have metastable states because we start with a random distribution of spins with 75% up and repeat with 75% down. This way is used to avoid metastable states is used in [1]. in addition, when we iterate over temperatures, we will use configuration from the previous temperature as the initial configuration to speed up the code and avoid the metastable states (as suggested by problem 8.27 in the course textbook).

Energy

to have results that are easy to compare and understand we divide the total energy by the number of sites to calculate the energy per site.



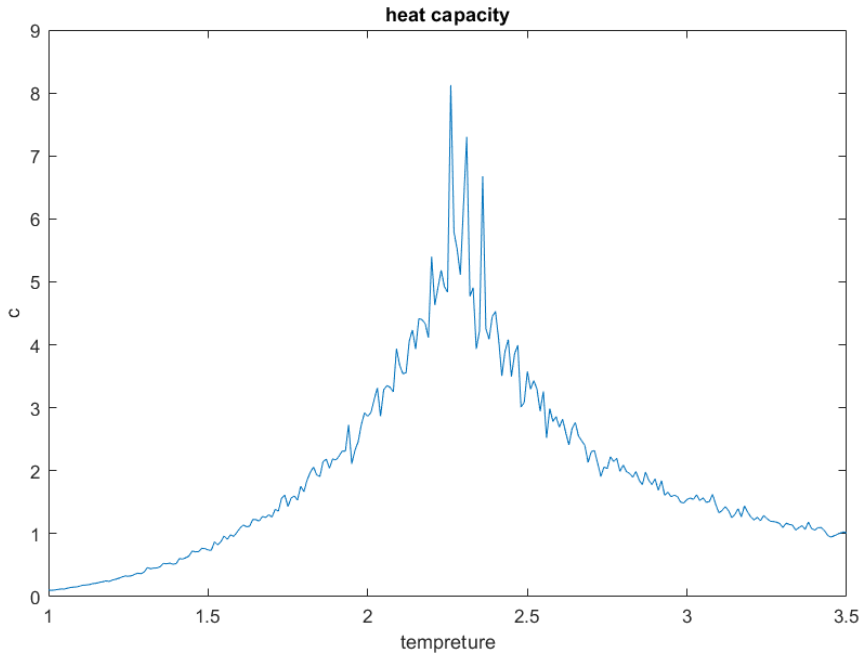
At low temperature, all the four neighbors will be aligned therefore energy is -4 (in 2D the are 4 neighbors so the lowest energy is - 4) at high temperature there will be as many aligned neighbors as anti-aligned neighbors, therefore, energy will approach 0.

Heat capacity

Due to the statistical fluctuation in the energy result we cannot use the definition $C = \frac{\partial U}{\partial T}$ to find the heat capacity instead we will use the following relation which utilizes the statistical variation in the calculation of the heat capacity $C = \frac{\sigma_U^2}{k\,T^2}$. the plotted is dimensionless specific heat

$$c = \frac{C}{Nk}$$

Where σ_U is the stander deviation of the variation in the energy.



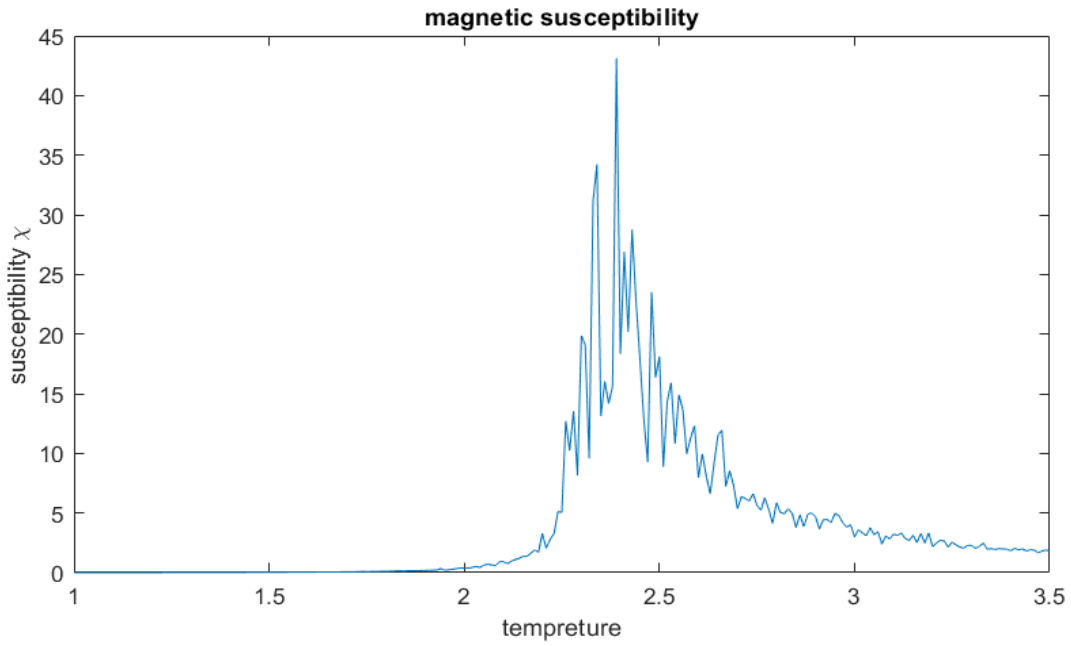
The heat capacity shows a change in behavior at temperature near 2.27 (the peak in the figure is at T= 2.26) which indicates a phase transition. The metropolis algorithm performs poorly near the critical temperature [2]. Therefore, the graph is not very smooth.

The magnetic susceptibility

The magnetic susceptibility for ferromagnetic material is defined as $\chi = \left(\frac{\partial M}{\partial B}\right)_T$ but similar to heat capacity. Due to the fluctuation, we cannot take the derivative. However, we can use statistical mechanics to find the susceptibility from the fluctuation [3].

$$\chi = \frac{\sigma_M^2}{kT}$$

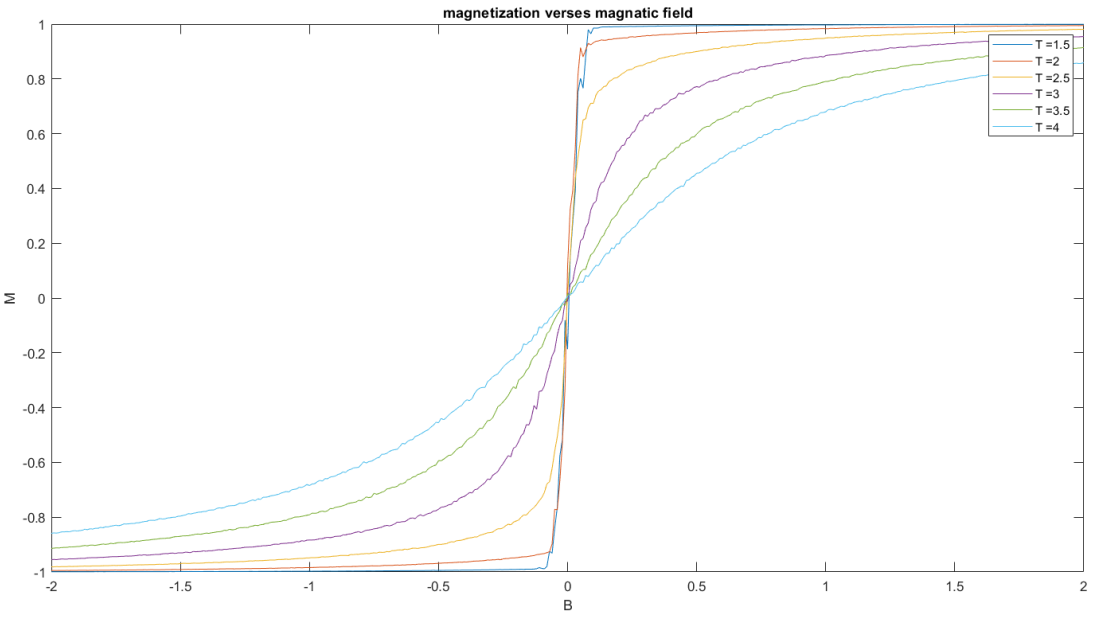
Where σ_M is the variance of the fluctuation in the magnetization. Furthermore, in the code, we work with dimensionless susceptibility per site defined above.



Similar to heat capacity the magnetic susceptibility shows a phase transformation near the critical temperature.

With magnetic field

Now we will examine the effect of the magnetic field in the magnetization. The external magnetic field effect in the magnetization is shown below

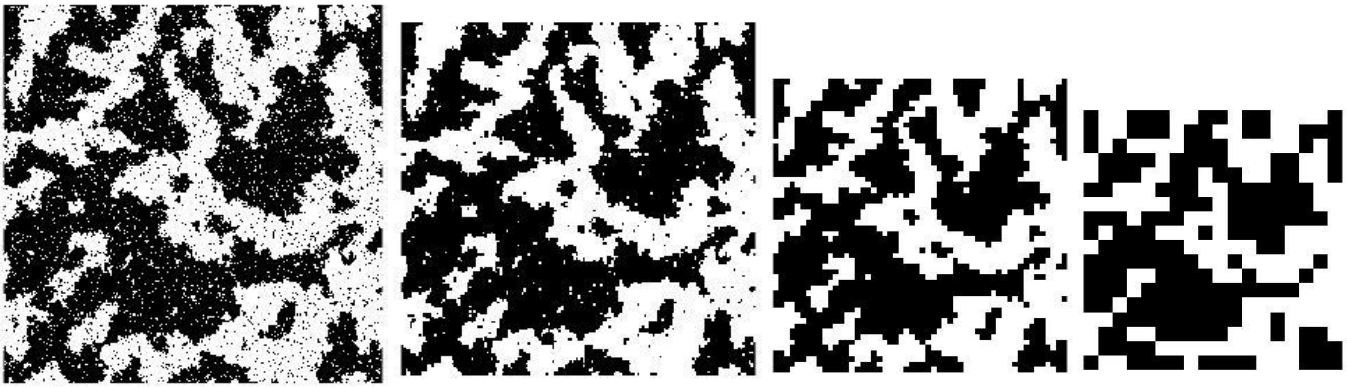


At low temperature, all the spins will align with the magnetic field even if the field is weak. At higher temperatures, it will need a stronger magnetic field to make all the spins aligned with it.

Block spin transformation

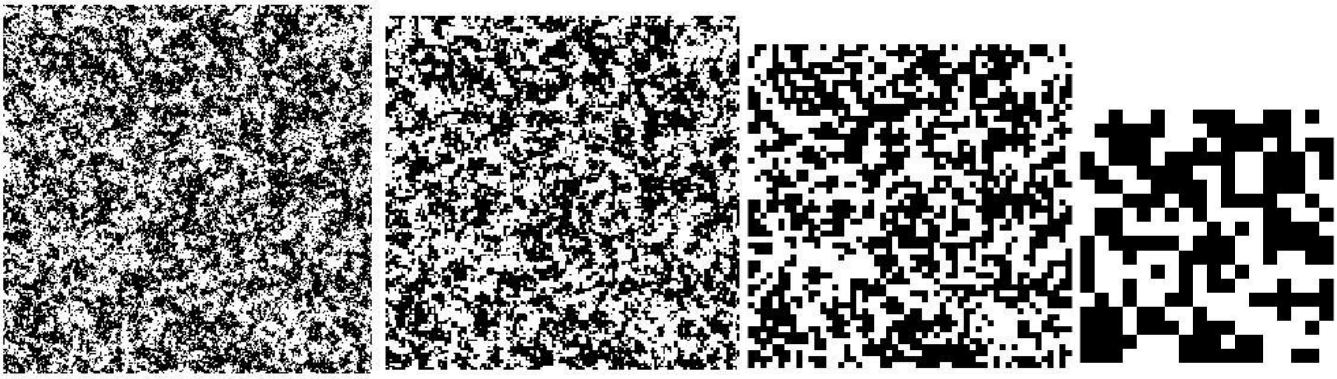
In this part each 3×3 spins will be replaced by the majority rule so it will be up if 5 or more spins are up, and down if 5 or more are down (black represents spin up and white is spin down)

For temperature less than the critical temperature



Here we can see that if we apply the block spin transformation the lattice behaves as it has a lower temperature with each transformation.

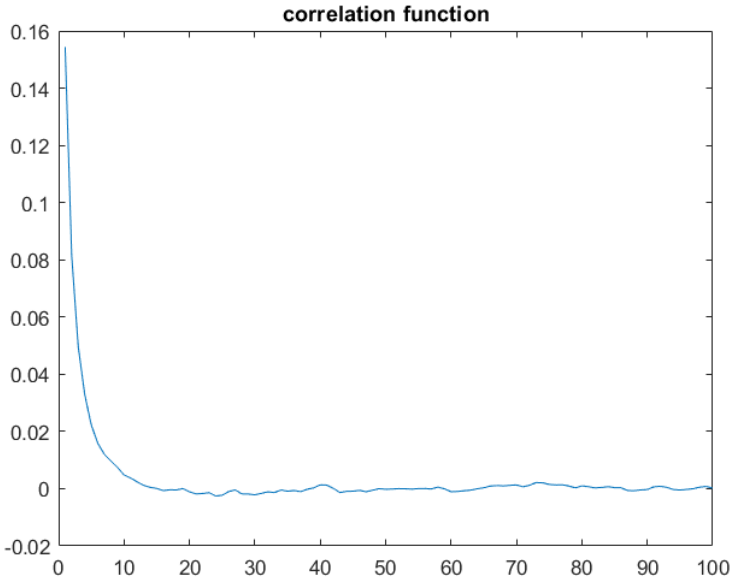
For temperature higher than the critical temperature



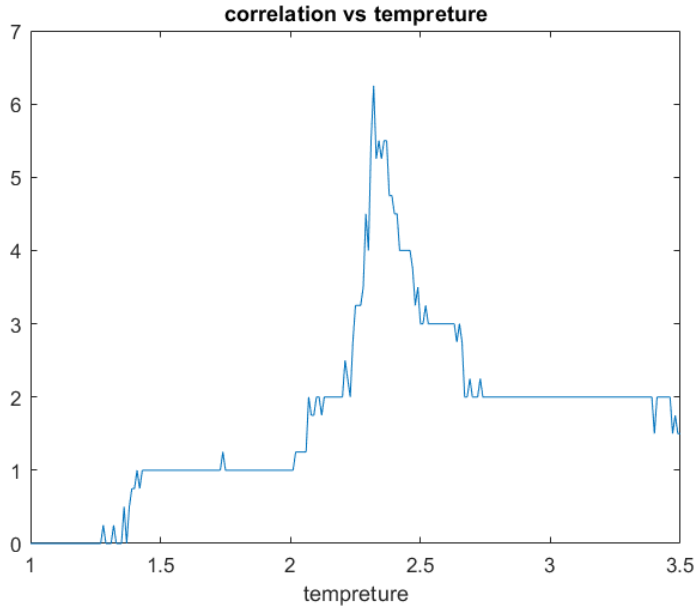
For temperature higher than the critical temperature. Block spin transformation should result in a higher temperature. However, it is difficult to say that one block has a temperature higher than the others by looking at it by eyes because all the images look random.

Correlation

Then we calculate the correlation function defined by $c(r) = \overline{s_i s_j} - \overline{s_i}^2$. here is a plot of the correlation function at temperature T=2.2. the correlation function is a function of distance where the distance here is measured as an integer step of lattice constant.



the correlation length is defined as the distance at which the correlation function is decreased by factor e. Iterating over temperatures recording the correlation length for each temperature then plotting the correlation length versus temperature gives the following result



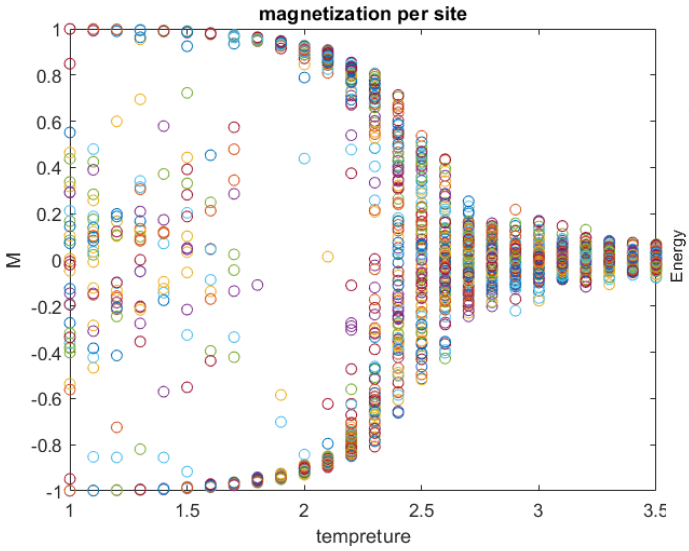
From the graph clearly the correlation length increases near the critical temperature. This tells us that near the critical temperature the long-range interactions are important.

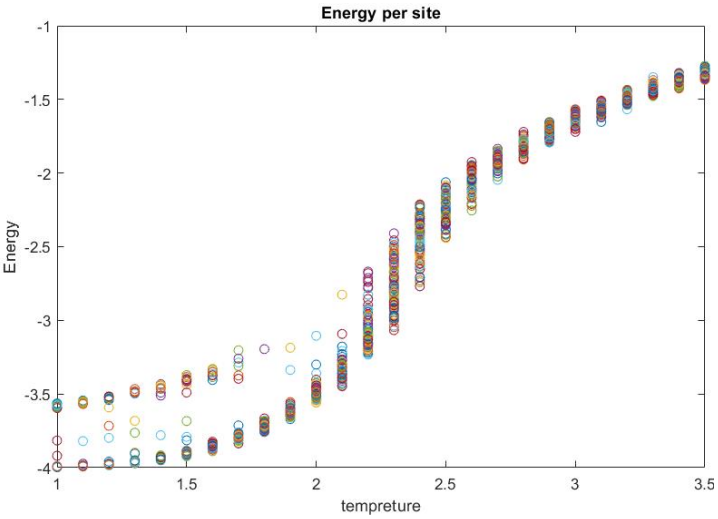
The correlation length should diverge at the critical temperature. However, it did not diverge this is due to many reasons, for example, the lattice is finite, and the number of iterations was not enough to calculate the correlation function (calculating the average correlation function over thousands of iterations cannot be done in personal laptop).

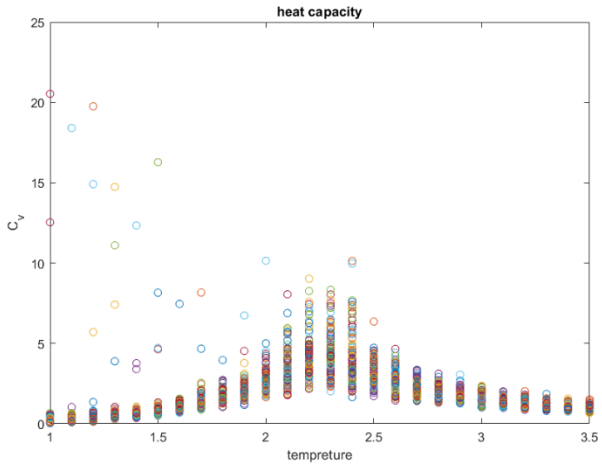
Simulation with random initial condition

So far, we use the initial condition where 75% of the spins are either up or down, then we use the configuration of the previous temperature as an initial condition for the next temperature. Here will examine the effect of random initial condition (50% up and 50% down) the system may not reach equilibrium for a finite number of iterations and stay in metastable states, therefore, it needs a large number of iterations and to run the code many times. So will use a smaller lattice of size 24*24 to be able to run the code on a personal laptop, the result:

Magnetization, Energy, and specific heat







We see a lot of variation in the heat capacity at low temperatures. And a lot of data points in energy and magnetization that were not present in the previous parts. This is because some of the configurations are in metastable states.

Conclusion

In this project we simulated the 2D square lattice Ising model for ferromagnets using the Metropolis algorithm. below a certain temperature (the critical temperature) domain of spins starts to form, making a clusters in which the spins are aligned together in the same direction. In addition, the heat capacity shows phase transition at that temperature. Furthermore, the magnetic susceptibility was calculated for different temperatures which show a phase transition similar to heat capacity. Therefore, we conclude that the 2D Ising model undergoes a phase transition at temperature near 2.27. In addition, the correlation length grows to a maximum near the critical temperature. Telling us that near critical temperature the long-range interactions are crucial.

References

[1] - Mohammed, J. and Mahapatra, S., 2018. A comparative study of 2d Ising model at different boundary conditions using Cellular Automata. *International Journal of Modern Physics C*, 29(08), p.1850066.

[2] - Walter, Jean-Charles & Barkema, Gerard. (2014). An introduction to Monte Carlo methods. Physica A: Statistical Mechanics and its Applications. 418. 10.1016/j.physa.2014.06.014.

[3] - Gould, H., & Tobochnik, J. (2010). *Statistical and Thermal Physics: With Computer Applications*. Princeton University Press.

[4] - Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward
Teller. Equation of state

[5]- Ising, Thomas & Folk, Reinhard & Kenna, Ralph & Berche, Bertrand & Holovatch, Yu. (2017). The Fate of Ernst Ising and the Fate of his Model. Journal of Physical Studies. 21. 10.30970/jps.21.3002.

Appendix: code

```
close;clear
T=1:0.01:3.5;
a = length(T);
ET = zeros(a,4);
MT = zeros(a,4);
CT = zeros(a,4);
Xi = zeros(a,4);
N=100;% the code generate lattice of size 2N*2N
cr_length =zeros(a,4);
w = zeros(4,N);
tic

% we use parfor instead of for loop to utlize the parallel computing
% feature and use all the cores in the CPU
parfor i = 1:4
    % initial condtion here is not completly random 75% of the spins are
    % either up or down to avoid metstable states
    s = ((i<3)*(rand(2*N)<0.25)+ (i>2)*(rand(2*N)>0.25) )* -2 + 1;
    for j = 1:a
        [s,E,M,C,x]=ising_sim_B(s,N,T(j),0);
        ET(j,i) = E;% energy
        MT(j,i) = M;% magnatization
        CT(j,i) = C;% heat capacity
        Xi(j,i) = x;
        % calculate corrlation length from correlation function
        cor = corrlation(s,N);
        m = max(max(cor));
        r = find(diff(cor <= m*0.368));%correlation length
        if isempty(r)||(m <(10^-3)) r = 0; end %for small values result
        cr_length(j,i)= r(1) ;                  % not accurate
        [T(j) i] % printed  just to show code is running
    end
end
toc

subplot(221)
plot(T,mean(ET,2))
subplot(222)
plot(T,mean(CT,2))
subplot(223)
plot(T,MT,'o')
subplot(224)
plot(T,mean(Xi,2),'o')
figure
plot(T,mean(cr_length,2))
```

```
%% ising model with magnatic field
N =100;
B = -1.5:0.02:1.5;
T = 1.5:0.5:4.5;
a = length(B);
b = length(T);
MT = zeros(a,b);
parfor j =1:a
    for i =1:b
        s = (rand(2*N)<0.5)*-2 +1;
        [~,~,M,~,~] = ising_sim_B(s,N,T(i),B(j));
        MT(j,i) = M;
        [T(i) B(j)]
    end
end
figure
plot(B,MT)
legend("T =1.5 ","T =2 ","T =2.5 ","T =3 ","T =3.5 ","T =4 ")

%% show the lattice
T=[1 1.5 2 2.27 2.5 3 4];
a = length(T);
N=100;
for i = 1:2
    % initial condtion here is not completly random 75% of the spins are
    % either up or down to avoid metstable states
    s = ((i<2)*(rand(2*N)<0.25)+ (i>1)*(rand(2*N)>0.25) )* -2 + 1;
    for j = 1:a
        [s,~,~,~,~]=ising_sim_B(s,N,T(j),0);
        figure; imagesc(s)
        title("temperature is  " + T(j))
        [T(j) i] % printed  just to show code is running
    end
end

%% block spin transformation
N = 243;
s =(rand(2*N)<0.5)* -2 + 1;
[s,~,~,~,~]=ising_sim_B(s,N,2.5,0);
s1 = block_s_tr(s,2*N);
s2 = block_s_tr(s1,2*N/3);
s3 = block_s_tr(s2,2*N/9);
subplot(221);title("first")
imshow(s,'InitialMagnification',3000)
subplot(222);title("second")
imshow(s1,'InitialMagnification',3000)
subplot(223);title("third")
imshow(s2,'InitialMagnification',3000)
subplot(224);title("third")
imshow(s3,'InitialMagnification',3000)
```

```
%%  ising model simulation function
function [s,E, M, C,x] = ising_sim_B(system,N,T,B)
    l = 2000;% number of iteration
    s =system;
    M = zeros(l,1);
    E = zeros(l,1);
    chosed(:,:,1) = repmat([1 0;0 0],N);
    chosed(:,:,2) = repmat([0 1;0 0],N);
    chosed(:,:,3) = repmat([0 0;1 0],N);
    chosed(:,:,4) = repmat([0 0;0 1],N);
    for iter = 1:l
        DeltaE = 2 * (s.* (circshift(s,[ 0 1])+ circshift(s,[ 0 -1])+...
                circshift(s, [ 1 0])+ circshift(s, [-1 0])+B)) ;
        i = randi([1,4]);
        transitions = ((rand(2*N) < exp(-DeltaE/T)) & chosed(:,:,i))* -2 +1;
        s = s.* transitions;
        M(iter) = sum(sum(s));
        E(iter) = -sum(sum(DeltaE))/2;
    end
    C = var(E(200:end)).*(2*T*N)^-2; %we do not count the first 200
    E = mean(E(200:end))/(4*N^2);  % iteration because the system might not
    x = var(M(200:end))./(4*T*N^2);% be able to reach equiliprium
    M = mean(M(200:end))/(4*N^2);
end
```

```
%% block spin transformation
function s = block_s_tr(system,N)
    s =zeros(N/3);
    for i = 1:N/3
        for j = 1:N/3
            s(i,j) = (sum(sum(system(3*i-2:3*i,3*j-2:3*j)))<0)*-2 + 1;
        end
    end
end
```

```
%% correlation
function cr = corrlation(s,N)
    cr = zeros(1,N);
    for i = 1:N
        A = 0.25*s.* (circshift(s,[ 0 i])+ circshift(s,[ 0 -i])+...
                circshift(s, [ i 0])+ circshift(s, [-i 0]));
            cr(i) = mean(mean(A))-mean(mean(s)).^2;
    end
end
```