

ISING MODEL AND SIMULATION

Computational Physics Course Project

The Ising model is a famous and applicable model in the statistical mechanics. In this project, I propose to consider the Ising model and its computational simulation. In this text, I try to review the main idea of the Ising model and Metropolis algorithm, the computational approach to this problem. Finally, I report the result of my own simulation and compare its results with analytical solution as a conclusion.

Sayed Mohammad Mahdi Sadrnezhaad

Guided by Dr. Nima Ghaleh

Fall 2008



ISING MODEL AND SIMULATION

INTRODUCTION

Statistical mechanics is a very active area of research and there are many open problems. In fact, the general phenomenon of phase transitions is still actively being researched. An example of these phenomena is ferromagnetic transition in the Curie temperature. In this phenomenon, Ferromagnetic materials, when heated, eventually lose their magnetic properties. This loss becomes complete above the Curie temperature, named after the French physicist Pierre Curie, who discovered it in 1895 [1-2].

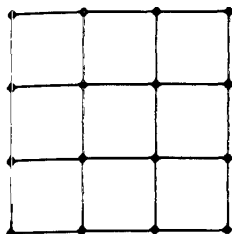
Paramagnetism is the normal induced magnetization of a material when it is put into a magnetic field. Paramagnetism could be explained by assuming that all spins are independent. Ferromagnetism is the spontaneous magnetization occurring in certain metals such as iron and nickel. To explain this phenomenon we must take into account the interaction between spins [2].

The Ising model is an attempt to simulate the structure of this phase transition. Originally, it was invented for the phase transition of ferromagnets at the Curie temperature; however, in the course of time it was realized that with only slight changes the model can also be applied to other phase transitions, like order-disorder transitions in binary alloys. Furthermore, the model may be applied to several modern problems of many particle physics, for instance for the description of so-called spin glasses. These are metals having amorphous instead of crystalline structures, which have the interesting property of no vanishing entropy at $T = 0$. Recently, it has been realized that Ising's idea (in modified form) could also explain pattern recognition in schematic neural networks. Thus, this model gains more and more importance for the development of models for the human brain [3-8].

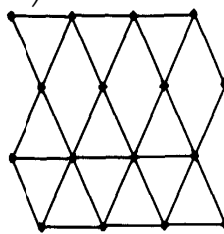
The main idea of Ising model is one orientation spin-interaction of fixed point on the lattice. This simplification is based on quantum mechanics concepts 'exchange force' [1]. It's meaning that the only outer electrons (conduction electrons) on the matter have interaction and their interaction is only in z-direction. On the other hand, this means that the characters of lattice like dimension, size, morphology and structure are variables of this model. We have to resume these quantities to have a specific sample of Ising model.

The other consideration of Ising model is that the only nearest neighbors interactions are necessary and others can be neglect [5]. The finally assumption is that the lattice in statistical limit are periodic [6].

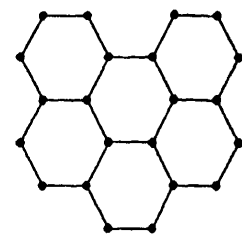
$$H = \left(J \sum_{\{i,j\}_{n.n.}} \sigma_{iz} \sigma_{zj} \right) \text{ where } \sigma_i \text{ are Pauli's matrix}$$



(a)



(b)



(c)

Figure 1. The 2-dimension lattice;

(a) Square (4 nearest neighbor); (b) Triangular (6 nearest neighbor) (c) Hexagonal (3 nearest neighbor). [7]

There are exact analytical solutions for 1 and 2 dimensions Ising models. In this project I have to simulate the 2-dimension Ising model with Square lattice. To see exact solution of this structure with addition theoretical detail, refer to Riechl [7].

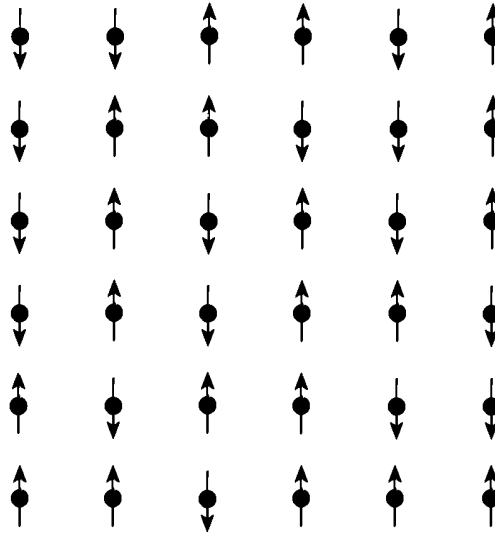


Figure 2. The 2-dimension Ising with square lattice.

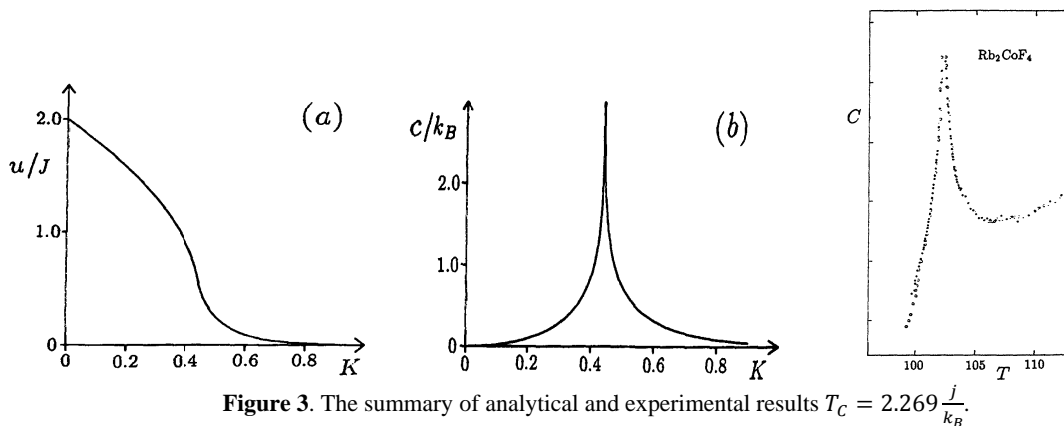


Figure 3. The summary of analytical and experimental results $T_C = 2.269 \frac{j}{k_B}$.

COMPUTATIONAL APPROACH

The computational simulation of Ising model is based on the two facts: The statistical nature of events and the Ising Hamiltonian [9]. From statistical mechanics, we know the probability of every event A is proportional to Boltzmann factor:

$$P(A) \propto \exp\left(-\frac{\epsilon_A}{k_B T}\right)$$

so the transition probability of $A \rightarrow B$ phase transition is:

$$P(A \rightarrow B) \propto \exp\left(-\frac{\Delta\epsilon}{k_B T}\right)$$

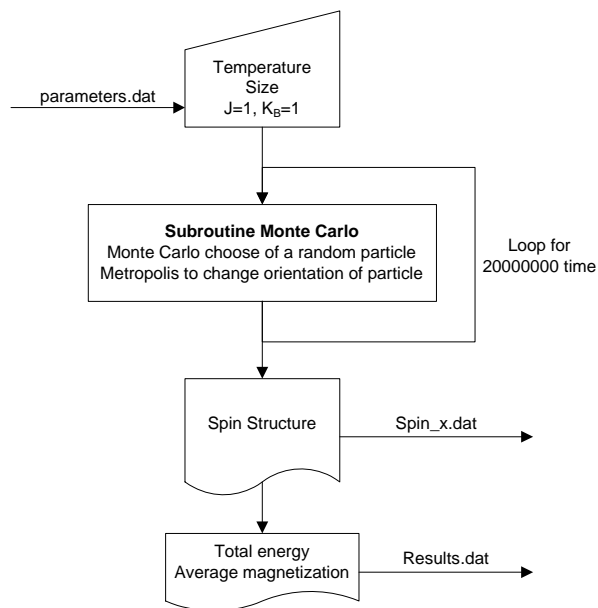
The computational algorithm which applies this factor at a problem is Metropolis algorithm [9], which I needed to simulate the Ising model. In this section, I will introduce the Metropolis algorithm and then present my own simulation algorithm.

(a) Metropolis Algorithm Steps [9]:

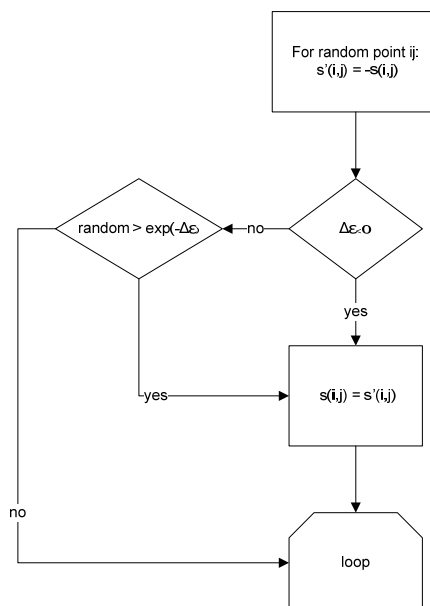
1. Start with an arbitrary spin configuration (initial conditions): $\{s_{ij}\}$

1. Disorder condition (hot start): In this state we start our program with complete disordered state, which means the entropy is maximal.
 2. Order condition (cold start): In this state we start our program with complete ordered state, which means the entropy is minimal and the temperature is 0.
2. Generate a new state:
1. Choose i_j^{th} element randomly (Monte Carlo)
 2. Reverse i_j^{th} spin direction to create a trial configuration
 3. Calculate the energy of the trial configuration
 4. If $E_{\text{new}} < E_{\text{(sij)}}$ then accept the change
 5. Else with random process with P probability accept the change

(b) Simulation Algorithm [Appendix: Project.f90]:



(c) Metropolis Algorithm [Appendix: Subroutine MonteCarlo]:



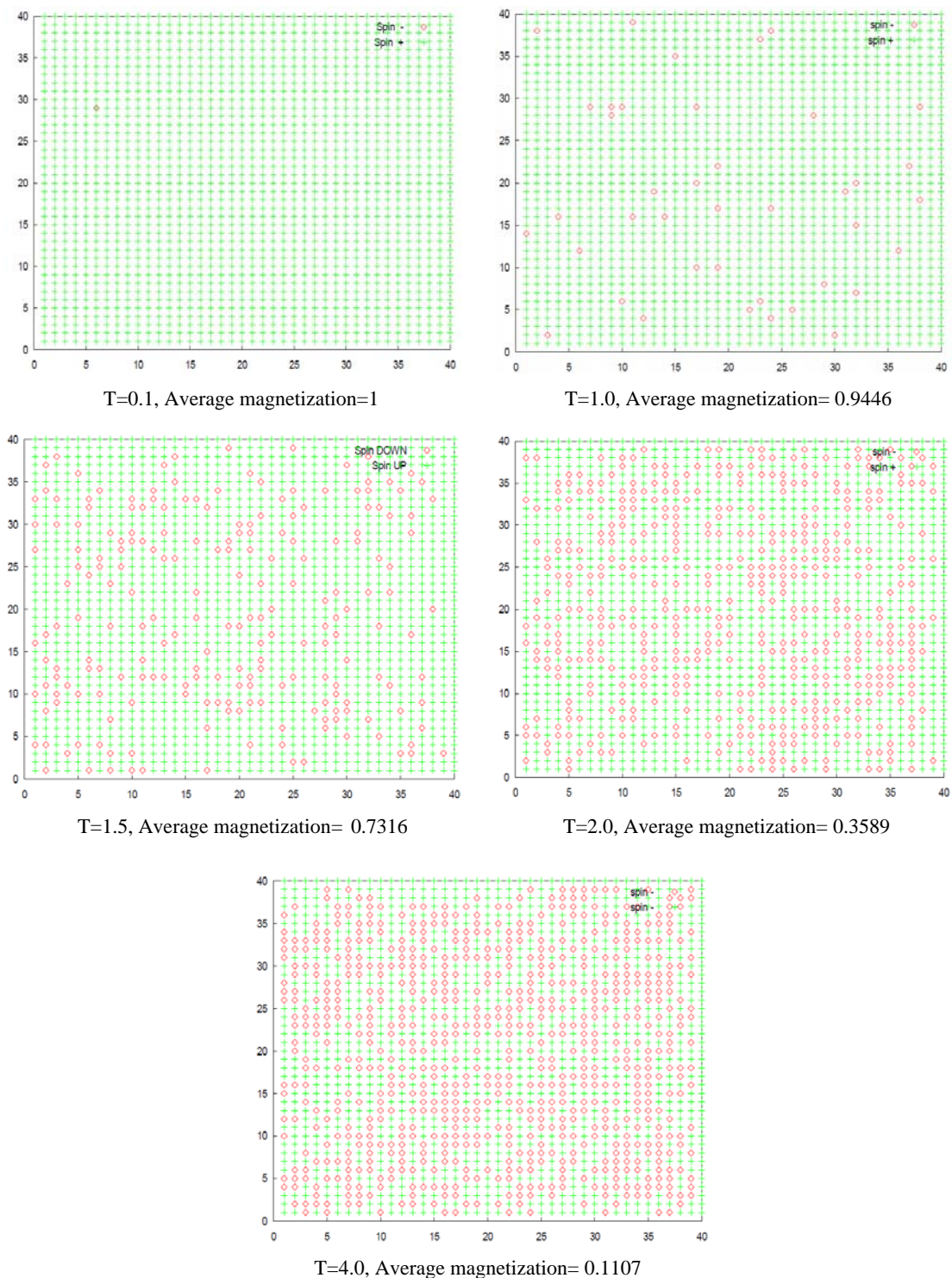


Figure 4. The result structures in 40×40 lattice.

CONCLUSION

The result structures of simulation (in 40×40 lattice) are shown in Figure 4. As we expect, the entropy of system increasing by the temperature of system.

Figure 5 are the charts of results for 20×20, 40×40, 60×60 and 100×100 in compare of analytical graph. As we expect, the value of the average magnetization in high temperature are decrease by increasing of system. It’s mean that the system becomes nearest to statistical limit and the simulation work well as a statistical system. The other point is that the qualitative behavior of system and the simulation’s results are same. On the other hand, we

know from renormalization group method in the critical points we have a self similarity property for the system. This figure shows that the critical point is $T_C \cong 2.1 j/k_B$, which is near to Curie temperature ($T_C = 2.269 j/k_B$) validly.

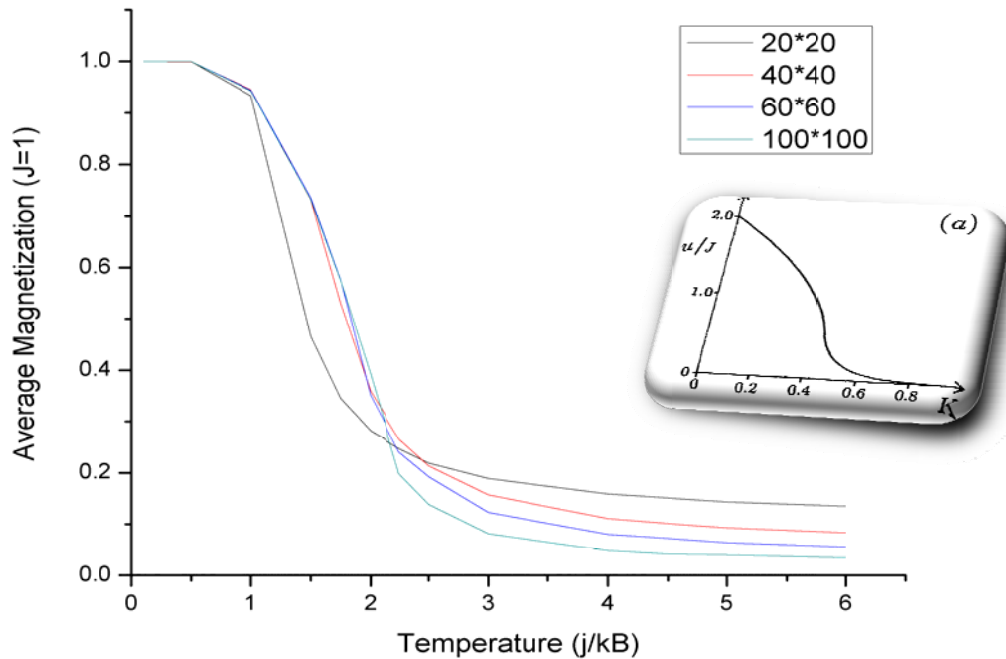


Figure 5. The result average magnetization simulation results.

ACKNOWLEDGMENT

It's a pleasure to thank Dr. Esfarjani and Dr. Ghaleh, my professors of computational physics and Dr. Langari, Dr. Sadat and Dr. Ketabi, my professors of statistical physics and thermodynamics.

REFERENCES

1. Microsoft® Student with Encarta Premium 2009 DVD.
2. Dorlas, T.C. "Statistical Mechanics: Fundamentals and Model Solutions", IOP Publishing Ltd, (1999), Ch. 28, 160-173.
3. Greiner, W., Neise, L. and Stoecker, H. "Thermodynamics and Statistical Mechanics", Springer, (1995), Ch. 18, 436-456.
4. Nishimori, H., "Statistical Physics of Spin Glasses and Information Processing: An Introduction", Oxford University Press, (2001), Ch. 1, 1-10.
5. Pathria, K. S., "Statistical Mechanic", 2nd ed., Butterworth-Heinemann, (1996), Ch. 11, 314-319.
6. Huang, K., "Statistical Mechanics", 2nd ed., John Wiley & Sons, (1987), Ch. 14, 341-363.
7. Reichl, L.E., "A Modem Course in Statistical Physics", 2nd ed., John Wiley & Sons, (1998), Ch. 8, 462-485.
8. Dalvit, D. A. R., Frastai, J. and Lawrie, I. D., "Problems on Statistical Mechanics", IOP Publishing Ltd, (1999), Ch. 5, Prob. 14-28.
9. Landau, R. and Paez, M. J. "Computation Physics: Problem Solving with Computers", John Wiley & Sons, (1976), Ch. 22, 297-305.

APPENDIX: Project.f90

```

!_____
!Computational Physics Project (Fall 2008)
!By Sayed Mohammad Mahdi Sadrnezhaad - 8711105
!smmsadr@gmail.com - smmsadr@merc.ac.ir
!_____
!This program written to calculate the equilibrium states of the Ising Hamiltonian
!with Monte-Carlo methods.
!_____
module parameters
!In this module we define the value of our parameters.
  integer N,Total
  real(8) J,Total_energy,kB,T,Average_magnetization
  integer(4) P
  integer s(100,100)
end module parameters
!_____
Program Monte_Carlo_for_Ising_Hamiltonian
  use parameters
  implicit none
  !integer i
  open (10,file='p.dat',status='old')
    read (10,*) P
  close (10)
  Average_magnetization=0
  Total=0
  open (20,file='Results.dat',status='unknown')
  call parameters_reader
  call Ising_Hamiltonian
  close (20)
  open (10,file='p.dat',status='unknown')
    write (10,*) P
  close (10)
end Program Monte_Carlo_for_Ising_Hamiltonian
!_____
subroutine parameters_reader
!This subroutine reads parameters from input file 2.
  use parameters
  implicit none
  integer b,c
  s=0
  open (10,file='parameters.dat',status='old')
    read (10,*) J
    read (10,*) N
    read (10,*) kB
    read (10,*) T
  close (10)
  !open (10,file='First_spin.dat',status='old')
    do b=1,N
      do c=1,N
        !read (10,*) s(b,c) !For when we want read first spin from file.
        s(b,c)=1
      end do
    end do
  !close(10)
end subroutine
!_____
subroutine Montecarlo(counter)
  use parameters
  implicit none
  integer r_x,r_y,counter !,reject
  real r,Probability,ran
  r_x=int(ran(P)*(N-1)+1)
  r_y=int(ran(P)*(N-1)+1)
  r=ran(P)
  !print*,N,r_x,r_y
  if (Probability(r_x,r_y)>1) then
    s(r_x,r_y)=-1*s(r_x,r_y)
  else
    if (Probability(r_x,r_y)>r) then
      if (Probability(r_x,r_y)<1) s(r_x,r_y)=-1*s(r_x,r_y)
    end if
  end if
  counter=counter+1
  !print*,counter

```

```

end subroutine
!-----
function Probability(x,y)

    use parameters
    implicit none
    integer x,y,x1,y1,x2,y2
    real Probability,delta_E,beta
    beta=-1/(kB*T)
    x1=mod(x+N-2,N)+1
    x2=mod(x,N)+1
    y1=mod(y+N-2,N)+1
    y2=mod(y,N)+1
    delta_E=-J*s(x,y)*(s(x1,y1)+s(x1,y2)+s(x2,y1)+s(x2,y2))
    Probability=exp(beta*delta_E)
end function
!-----
subroutine Ising_Hamitunian
    use parameters
    implicit none
    integer counter,b_b_b !,Tc,r_x,r_y,i
    !real r
    b_b_b=100
    Total=0
    Average_magnetization=0
    Total_energy=0
    counter=0
    do while(counter<=2000000)
        call Montecarlo(counter)
        if (counter>b_b_b*1000) then
            b_b_b=b_b_b+1
            call Energy_calculator
            call Magnetization_Calculator !(counter)
            !if (Number_of_Zero>300000) then
            call write_results(counter)
            !Print*,Counter,Average_magnetization>Total_energy
        end if
    end do
end subroutine
!-----
subroutine Energy_calculator
    use parameters
    implicit none
    integer x,y,x1,y1,x2,y2
    !real E
    Total_energy=0
    do x=1,N
        do y=1,N
            x1=mod(x+18,N)+1
            x2=mod(x,N)+1
            y1=mod(y+18,N)+1
            y2=mod(y,N)+1
            Total_energy=Total_energy-J*s(x,y)*(s(x1,y1)+s(x1,y2)+s(x2,y1)+s(x2,y2))/4
        end do
    end do
    Total_energy=Total_energy/(N**2)
end subroutine
!-----
subroutine Magnetization_Calculator !(c)
    use parameters
    implicit none
    integer f,g,sum2 !,c
    sum2=0
    do f=1,N
        do g=1,N
            sum2=sum2+s(f,g)
        end do
    end do
    Average_magnetization=sum2+Average_magnetization
    total=total+1
end subroutine
!-----
subroutine write_results(c)
!This subroutine writes results in output file.
    use parameters

```

```
implicit none
integer i,k,c
open (30,file='spin_1.dat',status='unknown')
open (31,file='spin_2.dat',status='unknown')
  do i=1,N
    do k=1,N
      if (s(i,k)==1) then
        write (30,4) i,k,s(i,k)
      else
        write (31,4) i,k,s(i,k)
      end if
    end do
  end do
close (30)
close (31)
write (20,4) c,Total_energy,Average_magnetization/(total*(N**2))
4 format(9(2x,g17.11))
end subroutine
!_____
```