PHY453: Computational Physics | Assignment 4

Anantha Rao | Reg no 20181044

27 Sept 2021

1 Results of the 3D Ising Model Simulation

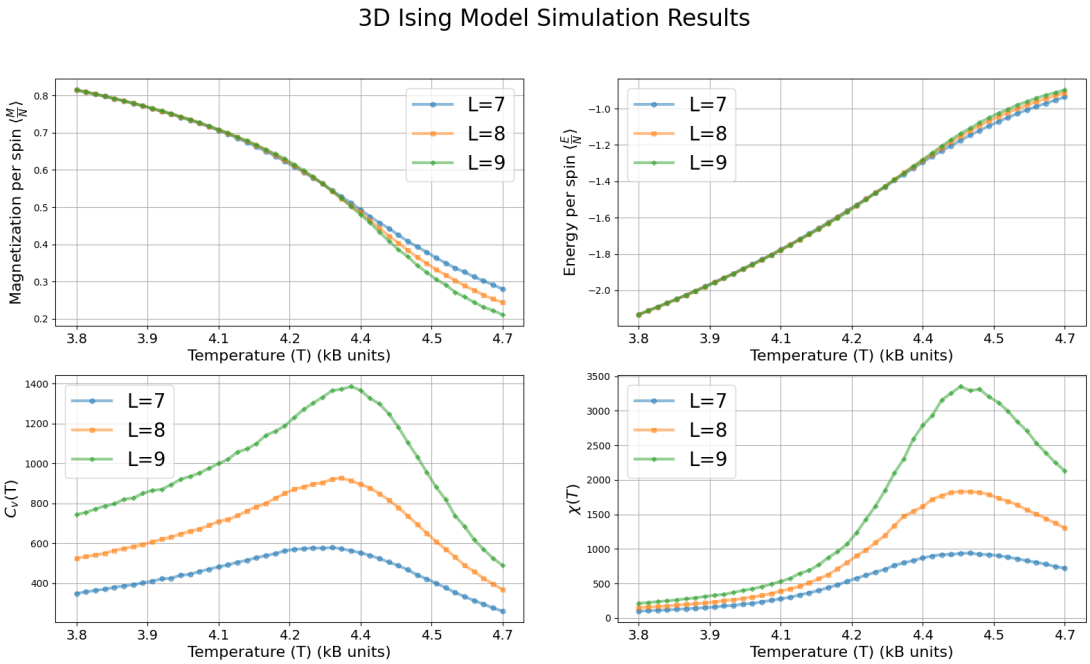

Figure 1: Results obtained after averaging over 10^6 iterations at each temperature

Q1: The value of $\chi(L)$ at T=4.5d0 for different values of L are approximately:

L (lattice size)	$\chi(L)$ value
7	924.25
8	1839.48
9	3248.93

Q2: The value of C_v at the peak position for L=8 is $\approx$ 926.57

Q3: The value of C_v at the peak position for L=9 is $\approx$ 1423.36

Q4: At temperature 3.8, the value for magnetization per spin for L=7 is $\approx$ 0.814506 units.

Q5: Using the Principe of detailed balance we find that the number of particles jumping per second from E_{10} to E_5 is 10 per second.

Q6: Please check figure 3

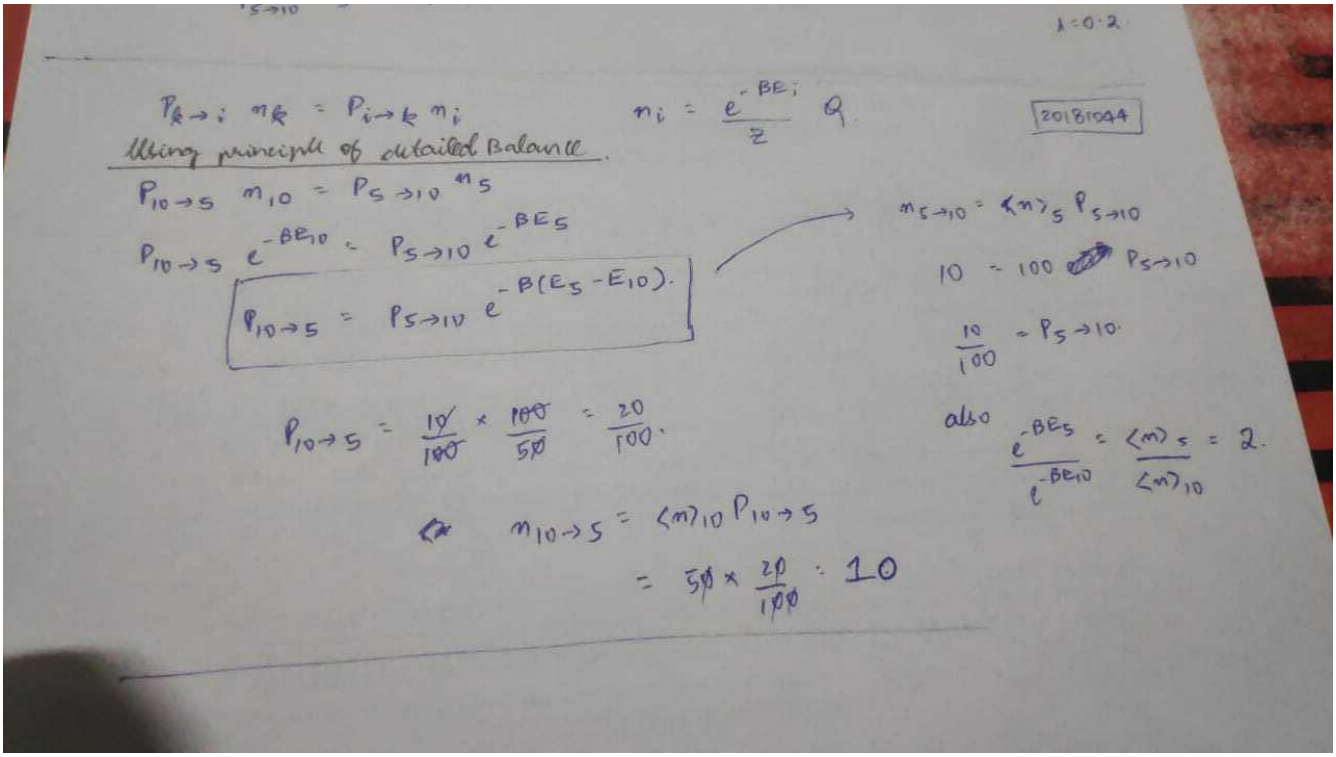

Figure 2: Working out the Transition probability using Principle of Detailed Balance

2 Binder's Cumulant

From the numerical data, the three Binder parameter curves intersect around $T \approx 4.50$. This implies that the actual transition temperature is around 4.5 $k_B T$ units which is very close to the value of 4.51 discussed in the lecture.

3 Fortran Code

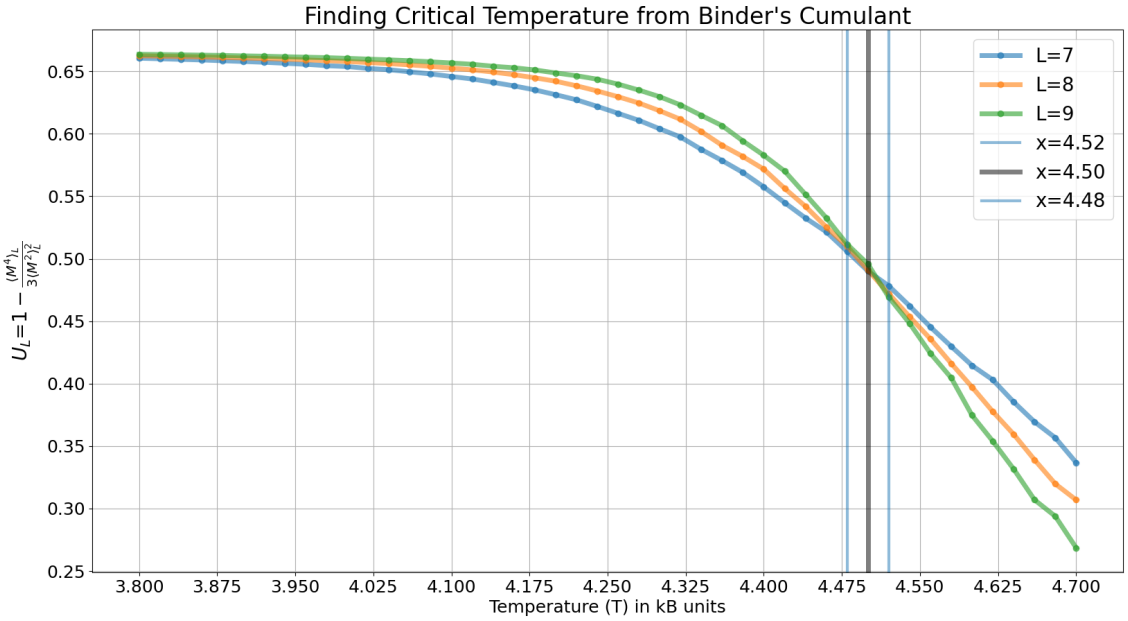

Figure 3: Determination of the Critical Temperature using Binders Cumulant (Here $T_c \approx 4.5$)

```
! Program to generate a lattice of spin-1/2 particles and study its
! ferromagnetic properties
! Author: Anantha Rao (Reg no: 20181044)

program ising_3d
    implicit none

    integer:: i,j,k,L,p,a,b,c,d,f,g,niter,time,mm,nn,oo,N, T_temp, n_equil, n_stat
    real*8:: r,E,M,mag,E1,Ef,dE,u,h, av_M, av_E, av_e_n, av_m_n, av_E2, av_M2, chi, cv, av_m4

    real*8:: T, binder_cum, J_ising=1.0 !assigning value to relevant parameters: k_B, J_ising

    integer, dimension(:,:,:), allocatable :: spin
    integer:: seed

    seed=44859

    print*, "Welcome to this program! This program simulates a 3D ising model with all spins
initially in the random state"
    print*, "Enter the number of lattice points in one dimension for (LxLxL)"
    read*,L
    print*, "Enter the number of iterations"
    read*, niter

    allocate(spin(L,L,L))
    E=0.0d0 ! Instantaneous Energy of the lattice
    M=0.0d0 ! Instantaneous magnetization of the lattice
    N = L*L*L ! Total number of spins in lattice

    n_equil = 10000 ! At each step, we collect data after n_equil steps ie the equilibration time
    n_stat = 10 ! Collect statistical data every n_stat steps

    ! Initialize your lattice
    open(unit=71, file="initial_ising_3D.dat")
    p=0
    do i=1,L
        do j=1,L
            do k=1,L
             call RANDOM_NUMBER(r)
                ! spin(i,j,k) = 1
                if (r < 0.5) then
                    spin(i,j,k) = -1
                else
                    spin(i,j,k)=1
                end if
            ! Write down this configuration
            write(71,*) float(i), float(j), float(k), float(p), float(spin(i,j,k))
            end do
        end do
    end do

    close(71)

    ! Calculate Initial magnetization and energy
    do i=1,L
        do j=1,L
            do k=1,L
            a=i+1; b=i-1; c=j+1; d=j-1; f=k+1; g=k-1 !identifying the 6 neighbors of spin(i,j)

            ! Setting PBCs (Periodic Boundary conditions)
            if (i==L)a=1
            if (i==1)b=L
            if (j==L)c=1
            if (j==1)d=L
            if (k==L)f=1
            if (k==1)g=L

            E = E - J_ising*float((spin(i,j,k))*
(spin(a,j,k)+spin(b,j,k)+spin(i,c,k)+spin(i,d,k)+spin(i,j,f)+spin(i,j,g)))
            M = M + spin(i,j,k)
            end do
        end do
    end do

    ! Calculate instantaneous magnetization per spin
    mag = M/float(N)
    E=E*0.5d0
    print*, "Initial energy E, E per spin is: ",E, E/float(N)
    print*, "Initial magentization M, M per spin is: ", M,M/float(N)

    ! INITIALIZATION COMPLETE
    ! ------------------------------------
    ! EVOLVE TO REACH EQUILIBRIUM

    open(unit=10, file="ising_L7_3d_data_new.dat")

    do T_temp=470,380,-2 !TEMPERATURE LOOP
        T = dfloat(T_temp)/100.0d0  !FIX T

        av_M = 0.0d0 ; av_e_N = 0.0d0 ! AVG. E, M, of ENTIRE LATTICE
        av_m2 = 0.0d0 ; av_e2 = 0.0d0 ! <E2>, <M2> of ENTIRE LATTICE
        av_m4 = 0.0d0 ! <M4> for calculating Binder's cumulant

    do time=1,niter ! Loop over number of MCS
        do mm=1,L
            do nn=1,L
                do oo=1,L
                ! Choosing a lattice site (i,j,k)
                call random_number(r); i=int(r*float(L))+1
                call random_number(r); j=int(r*float(L))+1
                call random_number(r); k=int(r*float(L))+1

                a=i+1;b=i-1;c=j+1;d=j-1;f=k+1;g=k-1 !identify the neighbors of spin(i,j)

                if(i==L)a=1; if(i==1)b=L; if(j==L)c=1; if(j==1)d=L; if(k==L)f=1; if(k==1)g=L !PBC

                ! BEFORE TRIAL FLIP
                E1 = -J_ising*float((spin(i,j,k))*
(spin(a,j,k)+spin(b,j,k)+spin(i,c,k)+spin(i,d,k)+spin(i,j,f)+spin(i,j,g)))

                !TRIAL FLIP
                spin(i,j,k) =-spin(i,j,k)

                ! AFTER TRIAL FLIP
                Ef = -J_ising*float((spin(i,j,k))*
(spin(a,j,k)+spin(b,j,k)+spin(i,c,k)+spin(i,d,k)+spin(i,j,f)+spin(i,j,g)))
                dE = Ef-E1

                ! METROPOLIS ALGORITHM
                if(dE .<=0.0) then
                    E=E+dE
                    M=M+(2.0*float(spin(i,j,k)))
                else
                    u=exp(-dE/(T))
                    call random_number(h)
                    if (h<u) then
                        E=E+dE
                        M=M+(2.0*float(spin(i,j,k))) ! Instantaneous mag. Of entire lattice
                    else
                        spin(i,j,k) = -spin(i,j,k) ! Trial flip not accepted; E & M not updated
                    end if
                end if
                end do
            end do
            end do

        ! ------------------------------------------------------------
        ! AFTER REACHING EQUILIBRIUM, COLLECT STATISTICAL DATA

        if (time .gt. n_equil) then
        !    if (mod(time, n_stat) .eq. 0) then

            mag = abs(M/dfloat(N)))  ! M=1*1*1 : instantaneous magnetization per spin
            av_M = av_M+mag          ;   av_e = av_e+E/dfloat(N) ! PER SPIN
            av_M_n = av_M_n+ abs(M)  ;   av_e_n = av_e_n+E ! Av. E, M of ENTIRE LATTICE
            av_M2 = av_M2+(M*M)      ;   av_e2 = av_e2+(E*E) ! AVG. E2, M2 of ENTIRE LATTICE
            av_m4 = av_m4+(M*M*M*M)

        ! end if
        end if
        end do  ! do time = 1:niter ! loop over no of MCS

        av_M  = av_M/dfloat(niter - n_equil)  ; av_e = av_e/dfloat(niter - n_equil)
        av_e2 = av_e2/dfloat(niter - n_equil) ; av_e_n = av_e_n/dfloat(niter- n_equil)
        av_M2 = av_M2/dfloat(niter - n_equil) ; av_M_n = av_M_n/dfloat(niter - n_equil)
        av_m4 = av_m4/dfloat(niter - n_equil)


        binder_cum = 1 - (av_m4/(3*(av_m2*av_m2)))

        cv = (av_e2 - av_e_n*av_e_n)/(T*T)
        chi = (av_M2 - av_M_n*av_M_n)/(T)

        write(10,*) T, av_M, av_E, cv, chi, binder_cum ! Writing down E and M with number of
iterations
        end do ! do T_temp = 47:38
    close(10)
    deallocate(spin)


end program ising_3d
```

Figure 4: Fortran Code for the 3D Ising model simulation