HW10 Finite Temperature 2-d Ising Model

Fei Ding

December 21, 2009

I set a $60\times$ 60 checkerboard and let the sweep go 4000 times. Figure 1 shows autocorrelation, one action means one sweep. I pike up $\tau = 12$, which has the smallest autocorrelation. So I would pick up data points every 12 sweeps while calculating average magnetization in part (ii).

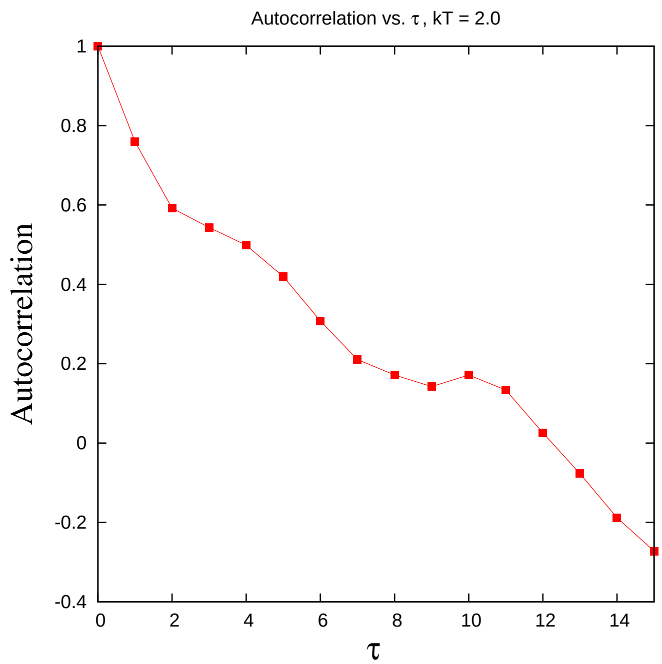

Figure 1: autocorrelation vs. τ

(i) The magnetization for each sweep is calculated as,

$$M = \frac{1}{L^2} \sum_{\texttt{all spin on array i}} \sigma_i \propto (n_\uparrow - n_\downarrow)$$

I plot data for every 12 sweep. The magnetization vs. sweep for $kT = 2.0$ and $kT = 2.5$ were shown on figure 2 with different initial conditions(random start and all spins up start for both cases). It shows that the initial start didn't affect the result. Under the same temperature, the magnetization would converge to the same value. For $kT = 2.0$, the particles tend to spin up, and it would finally get a magnetization around 0.9, while for $kT = 2.5$, the particles tend to have random spins, so the magnetization oscillates around zero.

(ii) The average magnetization and the error were calculated as,

$$\langle M \rangle = \frac{1}{\texttt{numberofsweep}} \sum_{\texttt{sweep } i} M_i$$

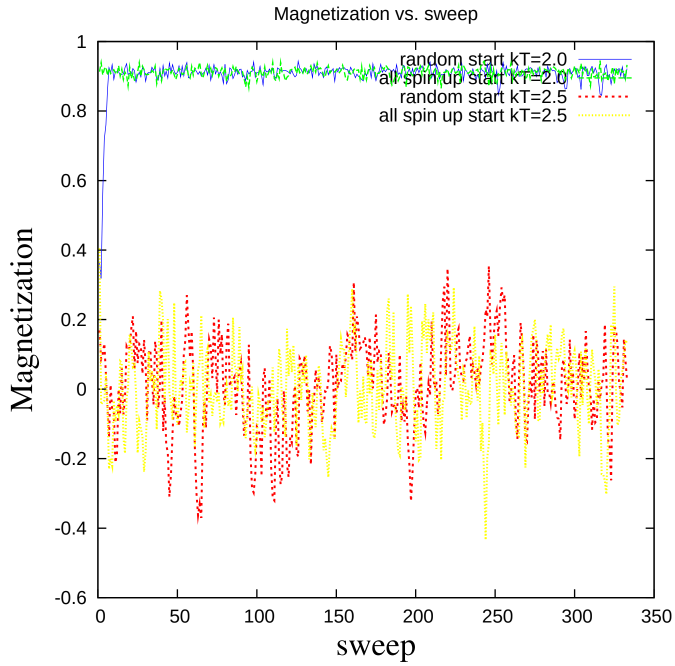

Figure 2: Magnetization vs. sweep

Every 12 sweeps were counted as one data point, correspondingly, the number of sweep should also reduce to sweep/12 to do the average. The curve and fitting was shown on figure 3. The fit result is $kT_c = 2.33$

(iii) For the spin-spin correlation, I only calculated the up, down, left, right neighbour correlation. Divide the region into four parts to get error estimation. For $kT = 2.0$ case, I used straight line to fit the functions, but the fit is a little bit odd.

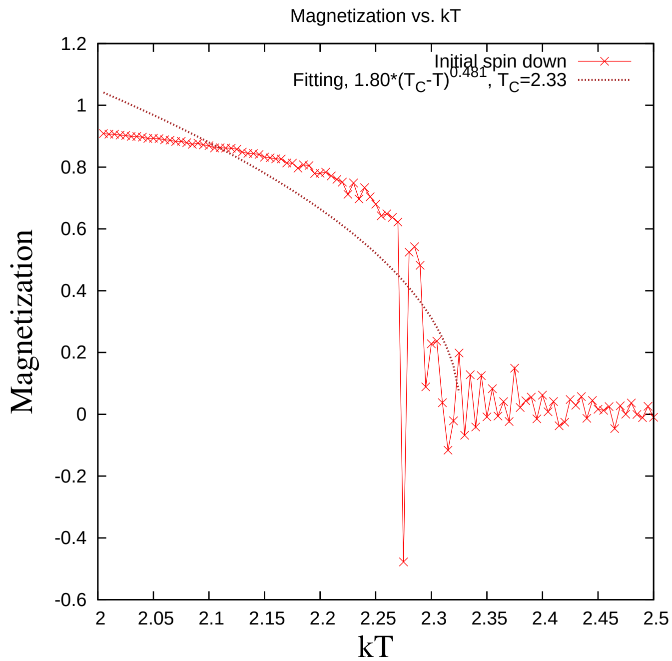

Figure 3: Average Magnetization vs. kT

A Codes

```
PROGRAM main

IMPLICIT NONE
DOUBLEPRECISION :: pi
INTEGER :: seed
DOUBLEPRECISION :: RAND
INTEGER, PARAMETER :: sweepMax = 4000
DOUBLEPRECISION, DIMENSION(sweepMax) :: netM  !netM at the ith sweep
INTEGER, PARAMETER :: n = 60   !50*50 spins total
INTEGER, DIMENSION(0:n+1,0:n+1) :: spin
INTEGER :: Nup, Ndown
DOUBLEPRECISION :: kT
DOUBLEPRECISION :: sum_netM,avgM
DOUBLEPRECISION :: deltaE  !change in energy
DOUBLEPRECISION :: a,random  ! the random number to determine hopping or not
INTEGER :: i,j,sweep

!autocorrelation
INTEGER :: t !tao for autocorrelation
DOUBLEPRECISION :: MiMit,Mi2,Mi,ct !cross term avg,square avg,avg and autocorrelation

!pin-spin correlation
INTEGER :: r !distance
DOUBLEPRECISION, DIMENSION(4) :: spincrossavg, spinsquareavg, spinavg !I'm dividing the area to 4 pieces,
DOUBLEPRECISION, DIMENSION(4) :: spincor
DOUBLEPRECISION :: spincorsum, spincorsquaresum,correlation,correlation2


seed = 918172
CALL SRAND(seed)
```

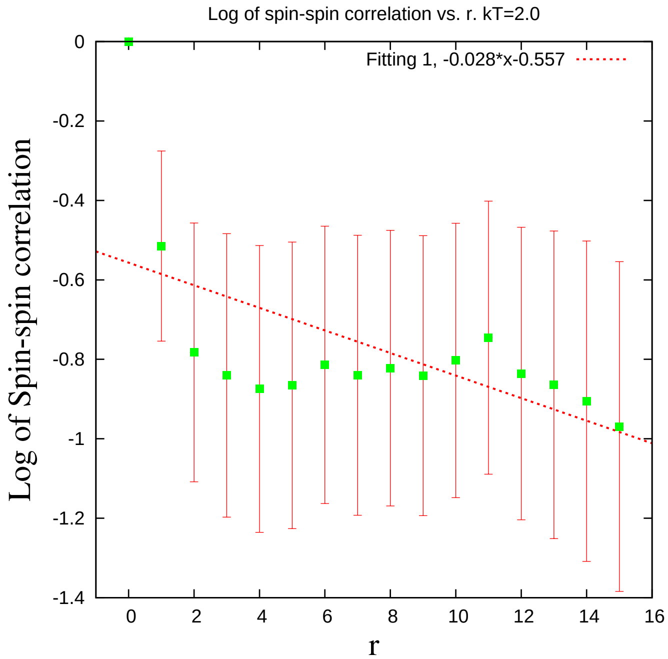

Figure 4: log Spin-Spin correlation function for kT=2.0

```
kT = 2.33D0
!Initializin
DO i = 1, n
DO j = 1, n
    a=RAND()
IF ( a > 0.5D0 ) THEN
    spin(i,j)=1
Nup = Nup + 1
ELSE if (a <=0.5d0) then
    spin(i,j)=-1
Ndown = Ndown + 1
ENDIF
WRITE(11,*) i, j, spin(i,j)
ENDDO
ENDDO
```

```
DO i = 1, n
DO j = 1, n
IF (i==1)  THEN   !PBC
spin(i-1,j) = spin(n,j)
ELSEIF (i==n)  THEN
spin(i+1,j) = spin(1,j)
ELSE
CONTINUE
ENDIF
IF (j==1)  THEN
spin(i,j-1) = spin(i,n)
ELSEIF (j==n)  THEN
```

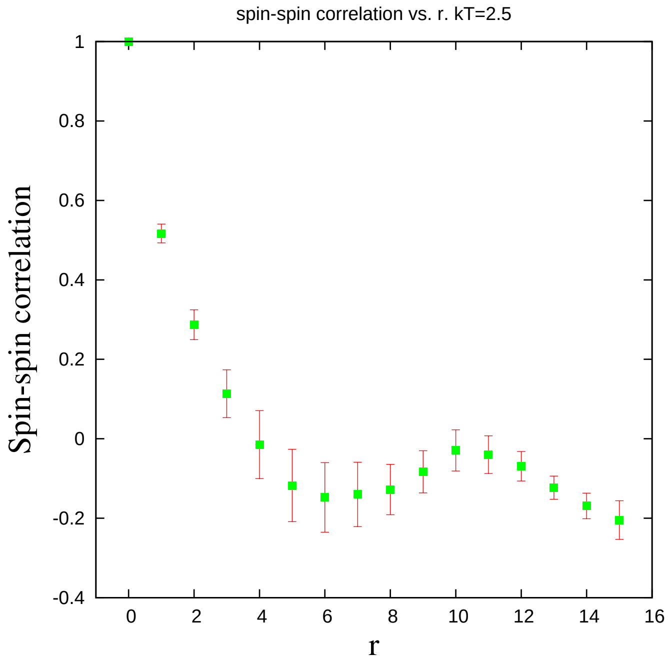

Figure 5: Spin-Spin correlation function for kT=2.5

```
spin(i,j+1) = spin(i,1)
ELSE
CONTINUE
ENDIF
! hamiltonian = hamiltonian + spin(i,j)*( spin(i+1,j) + spin(i-1,j) + spin(i,j-1) + spin(i,j+1) )
ENDDO
ENDDO
!hamiltonian = -0.5D0 * hamiltonian !count pair for once

!all up start
!spin=1
!ndown = 0
!nup = n**2
print *,'nup=',nup,'ndown=',ndown

sum_netM = 0D0
DO sweep = 1, sweepMax
!change configureation, flipping one by one
DO i = 1, n
DO j = 1, n
IF (i==1)  THEN   !reset PBC for each sweep
spin(i-1,j) = spin(n,j)
ELSEIF (i==n)  THEN
spin(i+1,j) = spin(1,j)
ELSE
CONTINUE
ENDIF
IF (j==1)  THEN
spin(i,j-1) = spin(i,n)
ELSEIF (j==n)  THEN
spin(i,j+1) = spin(i,1)
ELSE
```

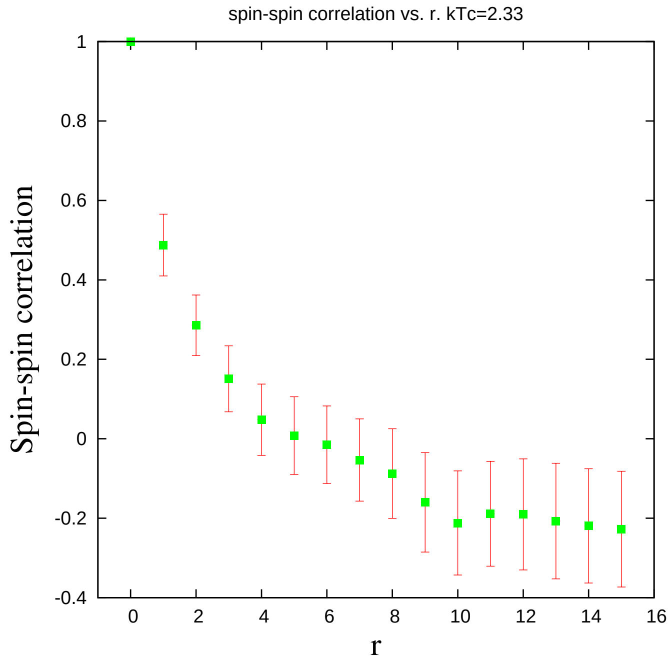

Figure 6: Spin-Spin correlation function for kTc=2.33

```
CONTINUE
ENDIF
deltaE = 2D0 * spin(i,j) * ( spin(i+1,j) + spin(i-1,j) + spin(i,j-1) + spin(i,j+1) )    !check
        !spin:1 to -1 or -1 to 1
IF (deltaE < 0D0) THEN   !accept the flip
Nup = Nup - spin(i,j)
Ndown = Ndown + spin(i,j)
spin(i,j) = -spin(i,j)
ELSE
random = RAND()
IF (random < DEXP(-deltaE/kt)) THEN  !accept with some probability
Nup = Nup - spin(i,j)
Ndown = Ndown + spin(i,j)
spin(i,j) = -spin(i,j)
ELSE
spin(i,j) = spin(i,j)   !do nothing
ENDIF
ENDIF
ENDDO
ENDDO
netM(sweep) = DBLE(Nup - Ndown)/DBLE(n**2)
IF (MOD(sweep,12)==0) THEN
sum_netM = sum_netM + netM(sweep)
avgM = sum_netM/dble(sweep/12)
WRITE(23,*) sweep/12, netM(sweep)
ENDIF

IF (mod(sweep, 100)==0)   THEN
!WRITE(*,*) hamil, netM(sweep), avgM
PRINT*, nup, ndown, nup+ndown
ENDIF
```

```
ENDDO !sweep


!=====autocorrelation================================
DO t = 0, 50
MiMit = 0D0
Mi2 = 0D0
Mi = 0D0

DO sweep = 2001,sweepMax-50
MiMit = MiMit + netM(sweep) * netM(sweep+t)
Mi2 = Mi2 + netM(sweep)**2
Mi = Mi + netM(sweep)
ENDDO
MiMit = MiMit/DBLE(sweepMax-50-2000)
Mi2 = Mi2/DBLE(sweepMax-50-2000)
Mi = Mi/DBLE(sweepMax-50-2000)

ct = ( MiMit - Mi**2 )/( Mi2 - Mi**2 )

WRITE(12,*) t, ct

ENDDO


!=====spin-spin correlation======================================
DO r = 0, n/4
spincrossavg = 0D0
spinsquareavg = 0D0
spinavg = 0D0

DO i = n/4+1, n/2  !upperleft coner
DO j = n/2+1, n*3/4
spincrossavg(1) = spincrossavg(1) + DBLE(spin(i,j)* ( spin(i+r,j) + spin(i-r,j) + spin(i,j+r) + spin(i,j-r
spinsquareavg(1) = spinsquareavg(1) + DBLE(spin(i,j)**2)
spinavg(1) = spinavg(1) + DBLE(spin(i,j))
ENDDO
ENDDO
spincrossavg(1) = spincrossavg(1)/DBLE((n/4)**2)
spinsquareavg(1) = spinsquareavg(1)/DBLE((n/4)**2)
spinavg(1) = spinavg(1)/DBLE((n/4)**2)

DO i = n/4+1, n/2    !lowerleft corner
DO j = n/4+1, n/2
spincrossavg(2) = spincrossavg(2) + DBLE(spin(i,j)* ( spin(i+r,j) + spin(i-r,j) + spin(i,j+r) + spin(i,j-r
spinsquareavg(2) = spinsquareavg(2) + DBLE(spin(i,j)**2)
spinavg(2) = spinavg(2) + DBLE(spin(i,j))
ENDDO
ENDDO
spincrossavg(2) = spincrossavg(2)/DBLE((n/4)**2)
spinsquareavg(2) = spinsquareavg(2)/DBLE((n/4)**2)
spinavg(2) = spinavg(2)/DBLE((n/4)**2)

DO i = n/2+1, n*3/4    !upperright corner
DO j = n/2+1, n*3/4
spincrossavg(3) = spincrossavg(3) + DBLE(spin(i,j)* ( spin(i+r,j) + spin(i-r,j) + spin(i,j+r) + spin(i,j-r
spinsquareavg(3) = spinsquareavg(3) + DBLE(spin(i,j)**2)
spinavg(3) = spinavg(3) + DBLE(spin(i,j))
```

```
ENDDO
ENDDO
spincrossavg(3) = spincrossavg(3)/DBLE((n/4)**2)
spinsquareavg(3) = spinsquareavg(3)/DBLE((n/4)**2)
spinavg(3) = spinavg(3)/DBLE((n/4)**2)

DO i = n/2+1, n*3/4    !lowerright coner
DO j = n/4+1, n*3/4
spincrossavg(4) = spincrossavg(4) + DBLE(spin(i,j)* ( spin(i+r,j) + spin(i-r,j) + spin(i,j+r) + spin(i,j-r
spinsquareavg(4) = spinsquareavg(4) + DBLE(spin(i,j)**2)
spinavg(4) = spinavg(4) + DBLE(spin(i,j))
ENDDO
ENDDO
spincrossavg(4) = spincrossavg(4)/DBLE((n/4)**2)
spinsquareavg(4) = spinsquareavg(4)/DBLE((n/4)**2)
spinavg(4) = spinavg(4)/DBLE((n/4)**2)

spincorsum = 0D0
spincorsquaresum = 0D0
    DO i = 1,4
spincor(i) = (spincrossavg(i) - spinavg(i)**2)/(spinsquareavg(i) - spinavg(i)**2)
spincorsum = spincorsum + spincor(i)
spincorsquaresum = spincorsquaresum + spincor(i)**2
ENDDO
    correlation=spincorsum/4d0
correlation2=spincorsquaresum/4d0

WRITE(34,'(I2, 4ES20.10)') r, (correlation), DSQRT( correlation2 - correlation**2 )/DSQRT(3d0)

ENDDO

DO i = 1, n
DO j = 1, n
WRITE(11,*) i, j, spin(i,j)
ENDDO
ENDDO
END PROGRAM
```