2015-Fall

# Monte Carlo Method

## ASSN - 04

Division of liberal arts and sciences
20145071 Junghoon Seo

## 1. Problem cognition

This problem is to compute some physical properties in the 2D Ising Model. They are the followings :

1. Total energy configuration
2. Total magnetization configuration
3. Capacity (or Specific heat)
4. Spontaneous magnetization
5. Susceptibility
6. Critical temperature using Binder formula

Simulated annealing and Metropoil-Hasting algorithm are used to a stable state of 2D Ising system.

## 2. Code (Programming Language : R)

```
Ann = function(A, m, n) {
  if (m == nr) Ann1 = A[1,n] else Ann1 = A[m+1,n] # bottom
  if (n == 1) Ann2 = A[m,nc] else Ann2 = A[m,n-1] # left
  if (m == 1) Ann3 = A[nr,n] else Ann3 = A[m-1,n] # top
  if (n == nc) Ann4 = A[m,1] else Ann4 = A[m,n+1] # right
  return(Ann1 + Ann2 + Ann3 + Ann4)
}

nr = 8; nc = 8 # Number of rows and columns
A = matrix(nrow = nr, ncol = nc)
npass = 2e5 # Number of passes for each temperature
nequil = 1e5 # Number of equilibration steps for each T
T_hi = 3 # Temperature to start scan at
T_lo = 1.5 # Temperature to finish scan at
dT = 0.1 # Temperature scanning interval
nscans = as.integer((T_hi - T_lo)/dT) + 1
# Initialize results table
M8 = matrix(nrow = nscans, ncol = 5, byrow=TRUE,
dimnames=list(rep("",nscans),c("T","E_av","Cv","Mag_av","Mag_sus")))

T_energy_sample = seq(1, nequil)
T_magnetization_sample = seq(1, nequil)

for (sIc in 1:nscans) { # T scan loop
  temp = T_hi - dT*(sIc - 1)
  # Initialize variables
  beta = 1/temp
  oc = 0 # output count
  E_av = 0
  E2_av = 0
  mag_av = 0
  mag2_av = 0

  A[1,1] = 1
  for (i in 1:(nr - 1)) A[i+1,1] = -A[i,1]
```

```
  for (j in 1:(nc - 1)) A[,j+1] = -A[,j]

  for (ipass in 0:npass) { # Monte Carlo passes at T
    if (ipass > nequil) {
      oc = oc + 1 # output count
      mag = sum(A)/(nr*nc)
      mag_av = mag_av + mag
      mag2_av = mag2_av + mag^2
      E = 0
      for (m in 1:nr) {
        for (n in 1:nc) {
          E = E - A[m,n]*Ann(A,m,n)
        }
      }
      E = E/(2*nr*nc)
      E_av = E_av + E
      E2_av = E2_av + E^2
      if (temp == 2.0 && ipass != 0){
        T_energy_sample[ipass-nequil] = E_av/(ipass-nequil)
        T_magnetization_sample[ipass-nequil] = mag_av/(ipass-nequil)
      }
    }
    # Choose a random spin to change
    m = sample(nr,1,replace=TRUE)
    n = sample(nc,1,replace=TRUE)
    ts = -A[m,n] # Flip sign of spin
    dU = -2*ts*Ann(A,m,n)
    eta = runif(1)
    if(exp(-beta*dU) > eta) A[m,n] = ts
  } # end MC passes at T

  M8[sIc,1] = temp
  M8[sIc,2] = E_av/oc
  M8[sIc,3] = beta^2*(E2_av/oc - (E_av/oc)^2)
  M8[sIc,4] = abs(mag_av/oc)
  M8[sIc,5] = beta*(mag2_av/oc - (mag_av/oc)^2)
  cat(c(temp, mag_av,mag2_av,E_av,E2_av),"\n") # not shown
  if (temp == 2.0){
    plot(seq(1, nequil), T_energy_sample, xlab = "N", ylab = "Total energy (E/N)", main =
"Total Energy Configuration @ Tem=2.0 & L=8", type = 'l', col="#228B22")
    dev.new()
    plot(seq(1, nequil), T_magnetization_sample, xlab = "N", ylab = "Total magnetization
(M/N)", main = "Total magnetization @ Tem=2.0 & L=8", type = 'l', col="#228B22")
    dev.new()
  }
} # end T scans

nr = 10; nc = 10 # Number of rows and columns
A = matrix(nrow = nr, ncol = nc)
npass = 2e5 # Number of passes for each temperature
nequil = 1e5 # Number of equilibration steps for each T
```

```
T_hi = 3 # Temperature to start scan at
T_lo = 1.5 # Temperature to finish scan at
dT = 0.1 # Temperature scanning interval
nscans = as.integer((T_hi - T_lo)/dT) + 1
# Initialize results table
M10 = matrix(nrow = nscans, ncol = 5, byrow=TRUE,
dimnames=list(rep("",nscans),c("T","E_av","Cv","Mag_av","Mag_sus")))

for (slc in 1:nscans) { # T scan loop
  temp = T_hi - dT*(slc - 1)
  # Initialize variables
  beta = 1/temp
  oc = 0 # output count
  E_av = 0
  E2_av = 0
  mag_av = 0
  mag2_av = 0
  # Set up initial checkerboard spin configuration
  A[1,1] = 1
  for (i in 1:(nr - 1)) A[i+1,1] = -A[i,1]
  for (j in 1:(nc - 1)) A[,j+1] = -A[,j]
  for (ipass in 0:npass) { # Monte Carlo passes at T
    if (ipass > nequil) {
      oc = oc + 1 # output count
      mag = sum(A)/(nr*nc)
      mag_av = mag_av + mag
      mag2_av = mag2_av + mag^2
      E = 0
      for (m in 1:nr) {
        for (n in 1:nc) {
          E = E - A[m,n]*Ann(A,m,n)
        }
      }
      E = E/(2*nr*nc)
      E_av = E_av + E
      E2_av = E2_av + E^2
    }
    # Choose a random spin to change
    m = sample(nr,1,replace=TRUE)
    n = sample(nc,1,replace=TRUE)
    ts = -A[m,n] # Flip sign of spin
    dU = -2*ts*Ann(A,m,n)
    log_eta = log(runif(1))
    if(-beta*dU > log_eta) A[m,n] = ts
  } # end MC passes at T
  M10[slc,1] = temp
  M10[slc,2] = E_av/oc
  M10[slc,3] = beta^2*(E2_av/oc - (E_av/oc)^2)
  M10[slc,4] = abs(mag_av/oc)
  M10[slc,5] = beta*(mag2_av/oc - (mag_av/oc)^2)
  cat(c(temp, mag_av,mag2_av,E_av,E2_av),"\n") # not shown
```

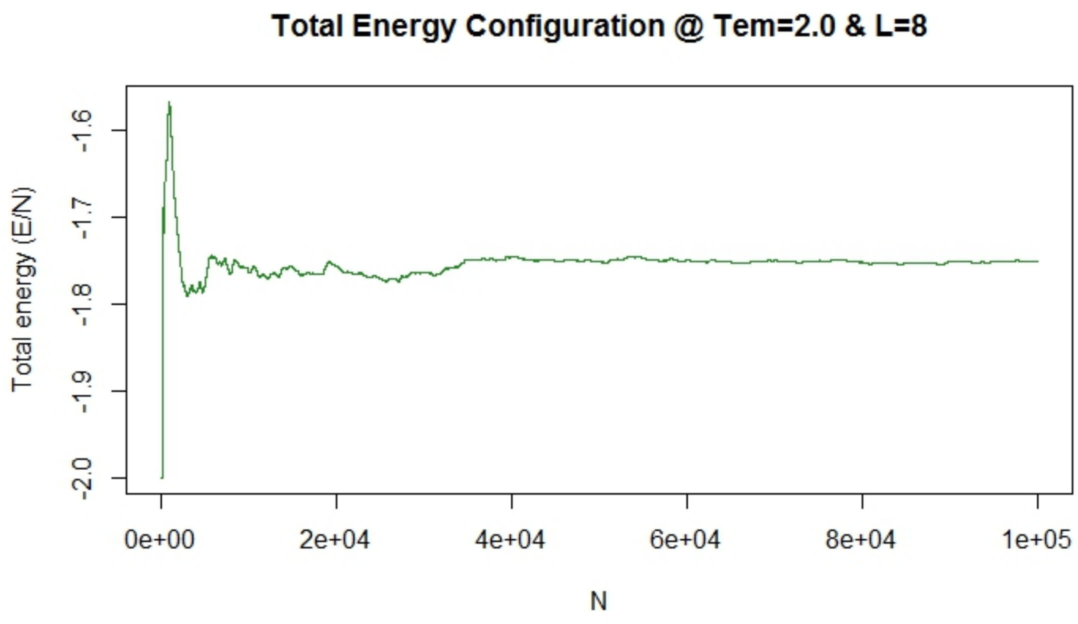
```
} # end T scans

nr = 12; nc = 12 # Number of rows and columns
A = matrix(nrow = nr, ncol = nc)
npass = 2e5 # Number of passes for each temperature
nequil = 1e5 # Number of equilibration steps for each T
T_hi = 3 # Temperature to start scan at
T_lo = 1.5 # Temperature to finish scan at
dT = 0.1 # Temperature scanning interval
nscans = as.integer((T_hi - T_lo)/dT) + 1
# Initialize results table
M12 = matrix(nrow = nscans, ncol = 5, byrow=TRUE,
dimnames=list(rep("",nscans),c("T","E_av","Cv","Mag_av","Mag_sus")))

for (sIc in 1:nscans) { # T scan loop
  temp = T_hi - dT*(sIc - 1)
  # Initialize variables
  beta = 1/temp
  oc = 0 # output count
  E_av = 0
  E2_av = 0
  mag_av = 0
  mag2_av = 0
  # Set up initial checkerboard spin configuration
  A[1,1] = 1
  for (i in 1:(nr - 1)) A[i+1,1] = -A[i,1]
  for (j in 1:(nc - 1)) A[,j+1] = -A[,j]
  for (ipass in 0:npass) { # Monte Carlo passes at T
    if (ipass > nequil) {
      oc = oc + 1 # output count
      mag = sum(A)/(nr*nc)
      mag_av = mag_av + mag
      mag2_av = mag2_av + mag^2
      E = 0
      for (m in 1:nr) {
        for (n in 1:nc) {
          E = E - A[m,n]*Ann(A,m,n)
        }
      }
      E = E/(2*nr*nc)
      E_av = E_av + E
      E2_av = E2_av + E^2
    }
    # Choose a random spin to change
    m = sample(nr,1,replace=TRUE)
    n = sample(nc,1,replace=TRUE)
    ts = -A[m,n] # Flip sign of spin
    dU = -2*ts*Ann(A,m,n)
    log_eta = log(runif(1))
    if(-beta*dU > log_eta) A[m,n] = ts
  } # end MC passes at T
```

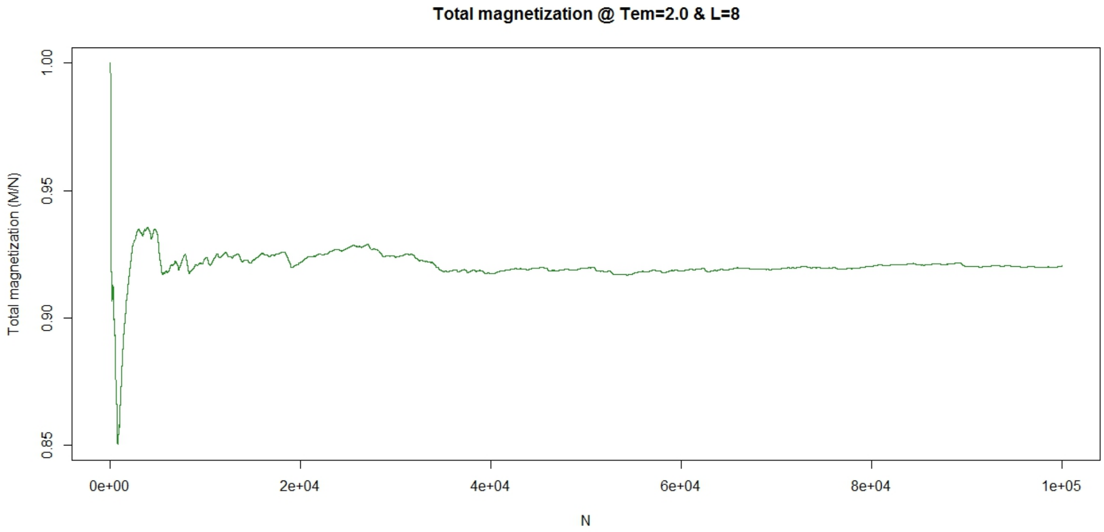
```
  M12[sIc,1] = temp
  M12[sIc,2] = E_av/oc
  M12[sIc,3] = beta^2*(E2_av/oc - (E_av/oc)^2)
  M12[sIc,4] = abs(mag_av/oc)
  M12[sIc,5] = beta*(mag2_av/oc - (mag_av/oc)^2)
  cat(c(temp, mag_av,mag2_av,E_av,E2_av),"\n") # not shown
} # end T scans

plot(M8[,1], M8[,2], xlab = "T", ylab = "<E>/N", main="<E>/N vs. Tem @ L = 8", type = 'o')
dev.new()
plot(M8[,1], M8[,4], xlab = "T", ylab = "<|M|>/N", main="<|M|>/N vs. Tem @ L = 8", type = 'o')
dev.new()
plot(M8[,1], M8[,3], xlab = "T", ylab = "Cv/N", main="Cv/N vs. Tem", ylim=c(0,0.03), pch=4,
type="o")
par(new=T)
plot(M10[,1], M10[,3], ylab="", xlab="", ylim=c(0,0.03), pch=0, type="o")
par(new=T)
plot(M12[,1], M12[,3], ylab="", xlab="", ylim=c(0,0.03), pch=2, type="o")
legend(x=2.7,y=0.015, c("L=8", "L=10","L=12"), cex=0.7, pch=c(2,0,4))
abline(v=2.3,col="red",lty="dotted")
dev.new()
plot(M8[,1], M8[,5], xlab = "T", ylab = "X/N", main="Magnetic Susceptibility vs. Tem",
ylim=c(0,0.40), pch=4, type="o")
par(new=T)
plot(M10[,1], M10[,5], ylab="", xlab="", ylim=c(0,0.40), pch=0, type="o")
par(new=T)
plot(M12[,1], M12[,5], ylab="", xlab="", ylim=c(0,0.40), pch=2, type="o")
legend(x=1.7,y=0.05, c("L=8", "L=10","L=12"), cex=0.7, pch=c(2,0,4))
abline(v=2.3,col="red",lty="dotted")
```

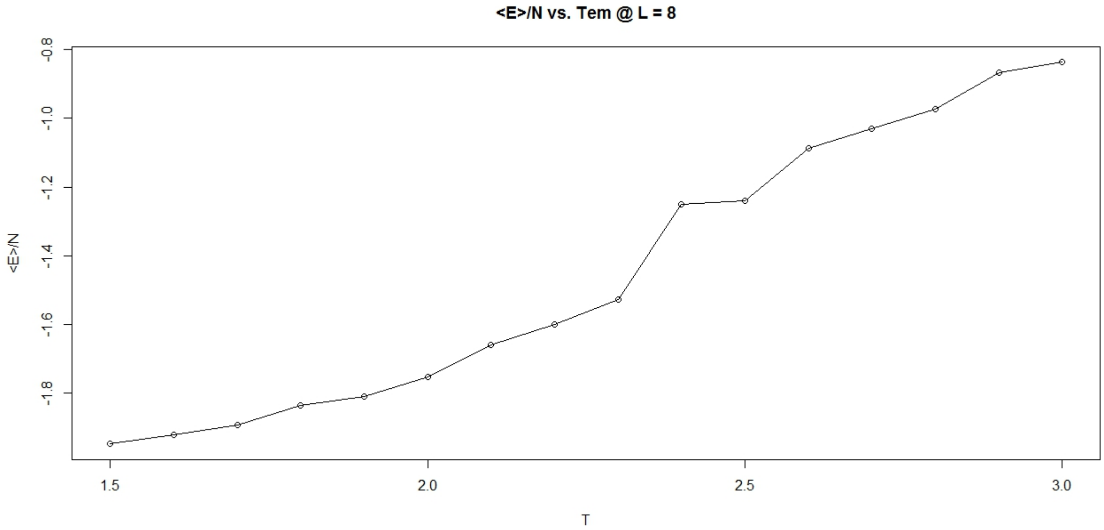
## 3. Result of implementation

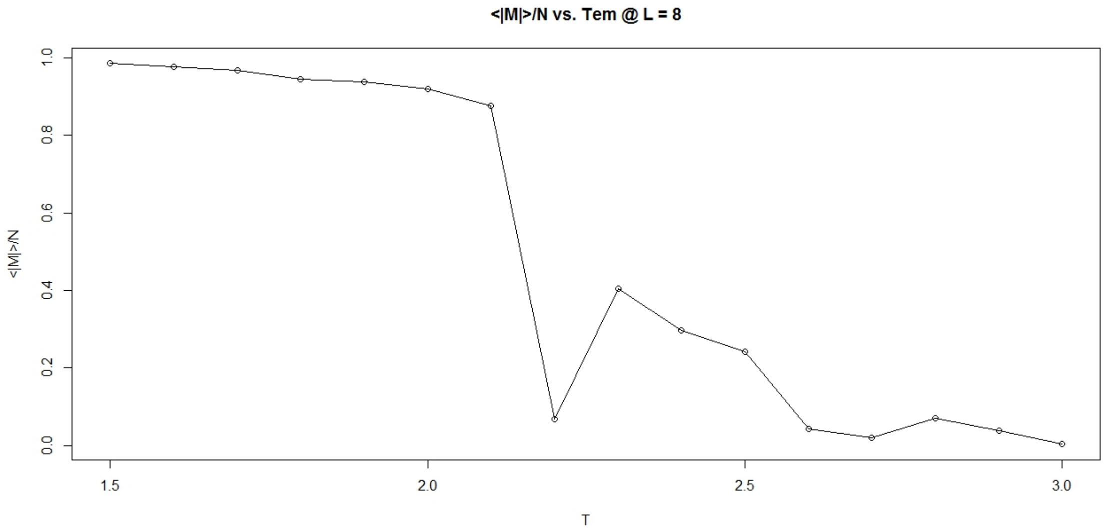
```
> source('~/R/two_dim_ising_model.R')
3 310.875 17903.69 -83724.94 76982.36
2.9 -3833.219 16152.95 -86620.44 81354.11
2.8 7098.844 25080.27 -97287.31 103681.5
2.7 -2068.219 29044.54 -102971.1 115365.1
2.6 -4363.062 31262.71 -108608.2 127644.8
2.5 -24111.69 44481.62 -124063.9 163787.1
2.4 29816.88 44774.34 -125154 167922.6
2.3 -40407.03 68449.97 -152699.4 240277.7
2.2 6866.406 73126.08 -160018.8 263461.8
2.1 -87596.09 78262.26 -165986.5 281407.9
2 92035.19 85246.3 -175179.5 310560.4
1.9 -93754.28 88456.05 -180970 330551.7
1.8 -94589.31 89979.82 -183430.4 339325
1.7 -96707.62 93770.95 -189221.1 359753.5
1.6 -97764.56 95695.82 -192215.1 370583.4
1.5 98521.44 97145.13 -194784.8 380211.2
3 8232.44 13467.29 -83983.6 75172.46
2.9 -3731.22 11642.6 -84731.44 75724.99
2.8 -16362.74 15588.26 -91293.56 88229.76
```

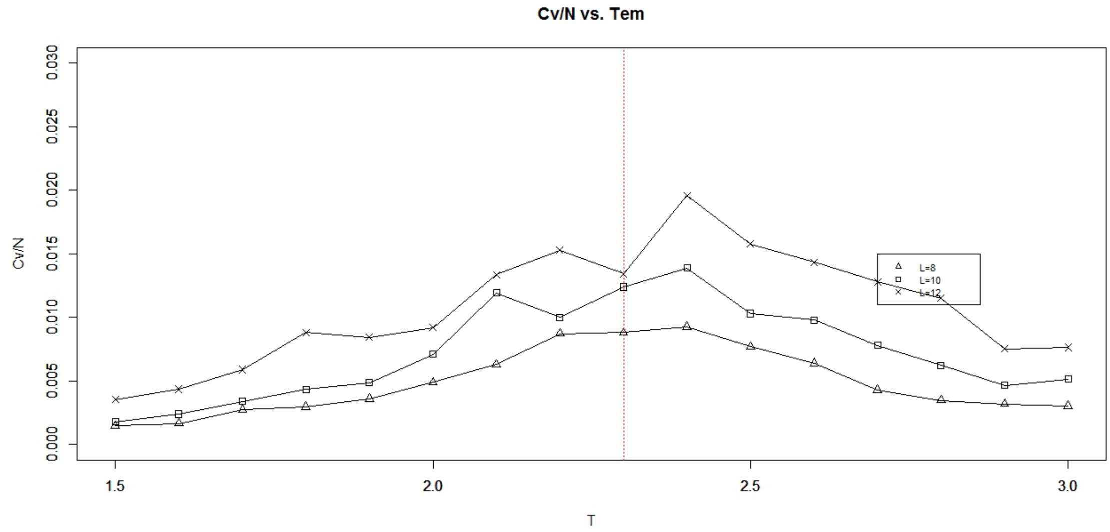
2.7 -6483.78 21841.47 -100486.2 106668.4
2.6 -2925.32 30344.41 -112113.2 132321.8
2.5 -8152.3 28007.56 -112010.3 131909.6
2.4 5016.44 44485.43 -128401.8 172866.6
2.3 30182.46 61397.07 -147351.4 223684.9
2.2 -83625.64 72119.58 -158951.8 257501.9
2.1 -84361.06 73920.17 -163486 272547.2
2 90289.92 82212.22 -172551.6 300583.9
1.9 94314.46 89247.49 -182315.4 334145.1
1.8 95734.28 91815.55 -185792 346590.1
1.7 97073.8 94344.89 -190096 362346.5
1.6 -98059.48 96216.78 -193121.7 373576.8
1.5 98838.52 97723.13 -195671.8 383282
3 2777.625 9095.385 -82515.81 70824.38
2.9 -3494.792 9283.273 -87154.67 78653.7
2.8 9692.583 9731.182 -89361.08 82582.04
2.7 22634.19 13156.26 -96230.03 95728.93
2.6 -27912.21 23530.21 -107338 119539.6
2.5 15193.51 27628.71 -115824.6 138957.5
2.4 13351.28 36899.52 -125493.4 162818.9
2.3 52490.78 51474.39 -141692.1 205424.9
2.2 -77718.58 63587.19 -152527.3 236867.6
2.1 -86610.1 75869.1 -164989.2 274987.4
2 91788.08 84628.88 -175799.2 311013.2
1.9 -94271.53 89072.83 -182050.4 332719.8
1.8 95964.53 92210.65 -186496.9 348771.7
1.7 -96682.61 93592.73 -189106 358406.3
1.6 98193.33 96457.68 -193488.5 374805.2
1.5 -98619.14 97290.33 -195113.1 381022.2

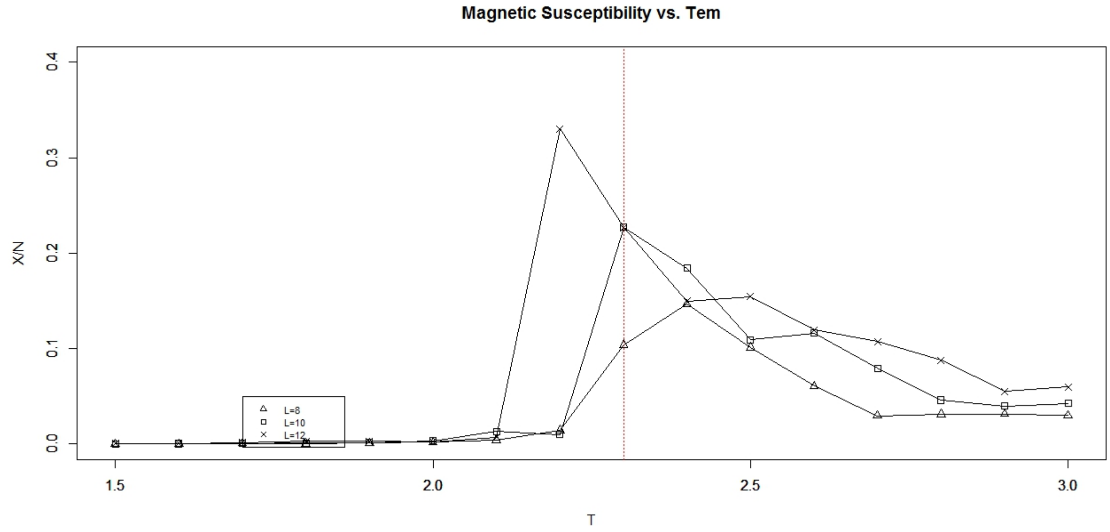



Total magnetization @ Tem=2.0 & L=8
Total magnetization (M/N)
1.00
0.95
0.90
0.85
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
N


<E>/N vs. Tem @ L = 8
<E>/N
-0.8
-1.0
-1.2
-1.4
-1.6
-1.8
1.5
2.0
2.5
3.0
T


<|M|>/N vs. Tem @ L = 8
<|M|>/N
1.0
0.8
0.6
0.4
0.2
0.0
1.5
2.0
2.5
3.0
T




## 4. Conclusion

1) Along each Monte Carlo Step, the total magnetization and total energy converges to the point at which the system has the maximized entropy(the minimized energy).

2) The spontaneous magnetization configuration increases and the energy configuration decreases while the temperature decreases.

3) Been known well, it is certain that there is a critical temperature at which susceptibility and heat capacity of the system jump. Around T = 2.3 ~ 2.4 is it under the simulation condition.

4) The more lattice the system has, the larger height of peak of susceptibility and heat capacity of system is at the critical temperature.