Monte Carlo simulation

Basic idea: random walk and diffusion equivalence

DiffusionL P(x,t) is the probability distribution of a particle for different positions x at a given time t.

- At the initial time t=0, the particle is localized at the origin. $P(x,0) =, \delta(x)$.
- $\frac{\partial P}{\partial t} = D\partial^2 P/\partial^2 x,.$
- $P(x,t) = e^{-\frac{x^2}{4Dt}}/\sqrt{4\pi Dt},$
- $\partial_t P(x,t) = (\frac{x^2}{4Dt^2} - \frac{1}{2t})e^{-\frac{x^2}{4Dt}}/\sqrt{4\pi Dt}$
- $\partial_x P(x,t) = -\frac{x}{2Dt}\, e^{-\frac{x^2}{4Dt}}/\sqrt{4\pi Dt}$
- $\partial_x^2 P(x,t) = [-\frac{1}{2Dt} + (\frac{x}{2Dt})^2]e^{-\frac{x^2}{4Dt}}/\sqrt{4\pi Dt}$
- $\int dx\, P(x,t) = \int du\, e^{-u^2}/\sqrt{\pi}$=1

Diffusion

- P(x,t) is the probability distribution of a particle for different positions x at a given time t.
- $\frac{\partial P}{\partial t} = D\partial^2 P/\partial^2 x$, D is the diffusion coefficient.
- At the initial time t=0, $P(x) = \delta(x)$. The particle is localized at the origin.
- $P(x,t) = e^{-\frac{x^2}{4Dt}}/\sqrt{4\pi Dt}$,
- $< x^2(t) > = \int du\, u^2 P(u,\text{t}) = 4\text{Dt}$
- .

Random Walk: the Langevin equation

- $\frac{dx}{dt} = \xi$, ξ is random, $< \xi > = 0, < \xi(\mathrm{t})\xi(\mathrm{t}') > = \delta(t - t')4D$
- $x(t) = \int_0^t du\, \xi(\mathrm{u})$,
- $< x^2(t) > = \int_0^t du \int_0^t du' < \xi(\mathrm{u}')\, \xi(u) > = 4Dt$

- Random walk and diffusion are equivalent.
- We shall use the random walk approach to calculate averages.

Random walk and diffusion are equivalent.
We shall use the random walk approach to calculate averages.

Instead of the diffusion equation we have the master equation for the time dependence of the probability distribution **P**

- d**P**/dt = **OP**
- Many possible forms for **O**. **P** is specified by its components P_j, the probability of finding configuration $\{x\}_j$. In component form we have
- $\frac{dP_i}{dt} = (OP)_i = \sum_j P_j(-p_{i,j} + p_{j,i})$, $p_{j,i}$ is the conditional probability to go from j to i.
- What is p? For the equilibrium distribution $\frac{d\boldsymbol{P}_0}{dt} = 0$.
- Thus $\boldsymbol{OP_0} = 0$. A sufficient condition is that $\frac{p_{i,j}}{p_{j,i}} = \exp[-\beta(E_j - E_i)]$.

  This is called detailed balance. This also determines the condition on the random force in the corresponding Langevin equation

Efficient importance sampling

- For the equilibrium distribution $\boldsymbol{P}_0$ the average of a physical quantity Q in equilibrium is
- $< Q > = \int dx\, Q(x) P_0(x)$.
- In a Monte Carlo simulation we have a random walk and generate a series of configurations x_i such that the probability of this is given by $P_0(x_i)$ and calculate the average as
- $< Q > = \sum_i Q(x_i)$

Metropolis scheme

- Transition probability between configuration i and j is given by:
- $\frac{p_{i,j}}{p_{j,i}} = \exp[-\beta(E_j - E_i)]$

- From configuration $\{x\}_i$ we generate a configuration $\{x\}_j$ with the probability proportional to $\exp[-\beta(E_j - E_i)]$.

Metropolis scheme in more detail

- Start with some initial configuration and keep generating more configuration.
- For example, for spins, starts with some initial configuriation such as that all spins aligned. Change the spin configuration by some random amount. For example, for spins that are vectors, choose a spin and rotate the spin by some random amount by changing the angles θ, φ by a random amount with a maxmum magnitude calld the step size. The spin components are $S_z = \cos\theta$, $S_x = \sin\theta\cos\varphi$, $S_y = \sin\theta\sin\varphi$
- Call this configuration j’

Metropolis scheme in more detail

- Calculate the Boltzmann factor $\mathrm{p} = \exp[-\beta(E_{j'} - E_i)]$.
- For our example for real systems, the energy is a sum of three terms: The exchange energy $E_{ex} = -J/2 \sum_{i,\delta} \vec{S}_i \cdot \vec{S}_{i+\delta}$ that aligns each spin and its neighbours, the anisotropy anergy $E_a = -\sum_{i,} [K(\vec{S}_i \cdot\cdot \vec{n})^2 \quad + K'(\vec{S}_i \cdot \vec{n})^4]$ that alingns each spin along a direction **n,** and the coupling energy with the external field **B,** $E_{ext} = -g\mu_B \sum_{i,\delta} \vec{S}_i \cdot \boldsymbol{B}$
- If p>1, then accept the move call this the new configuration j. Otherwise generate a random number x. If p>x, then we accept the move, otherwise the moe is rejected.
- If the step size is large, the acceptance ratio (rate of acceptance) is low. Usually one choose a step size so that the acceptance ratio is 30%

Practical matter:

- To access the convergence to equilibrium, we calculate the averages at different instances in time over some small sample size and see how this changes. In your program, we divide the number of MC steps into blocks and look at the average of each block.
- After the system has approached equilibrium, averages can be taken.

Simulations of 2D Ising model

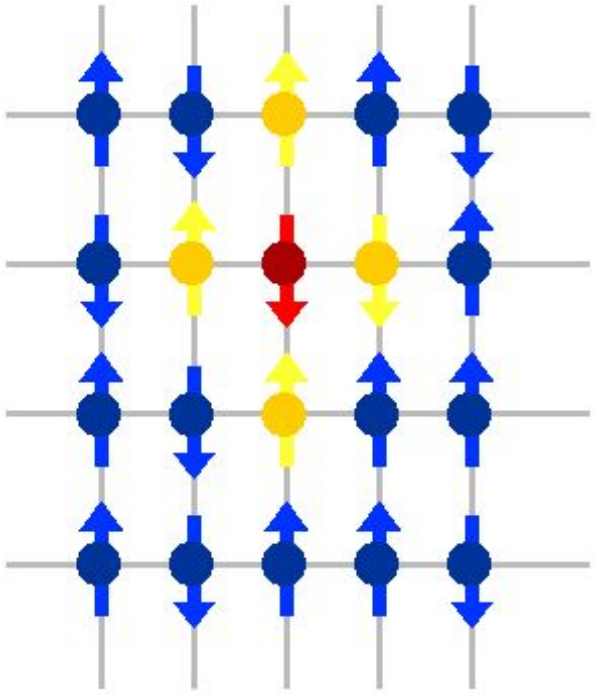


Figure: Lattice spin model with nearest neighbor interaction. The red site interacts only with the 4 adjacent yellow sites.

- We use the Ising model to demonstrate the studies of phase transitions.
- The Ising model considers the interaction of elementary objects called *spins* which are located at sites in a simple, 2-dimensional lattice,

$$\hat{H} = -J \sum_{i,j=nn(i)}^{N} \hat{S}_i \hat{S}_j - \mu_0 B \sum_{i=1}^{N} \hat{S}_i.$$

- Magnetic ordering:
  - $J > 0$: lowest energy state is *ferromagnetic*,
  - $J < 0$: lowest energy state is *antiferromagnetic*.

Equibrium properties

Mean energy $$\langle E \rangle = \mathrm{Tr}\, \hat{H}\hat{\rho},$$

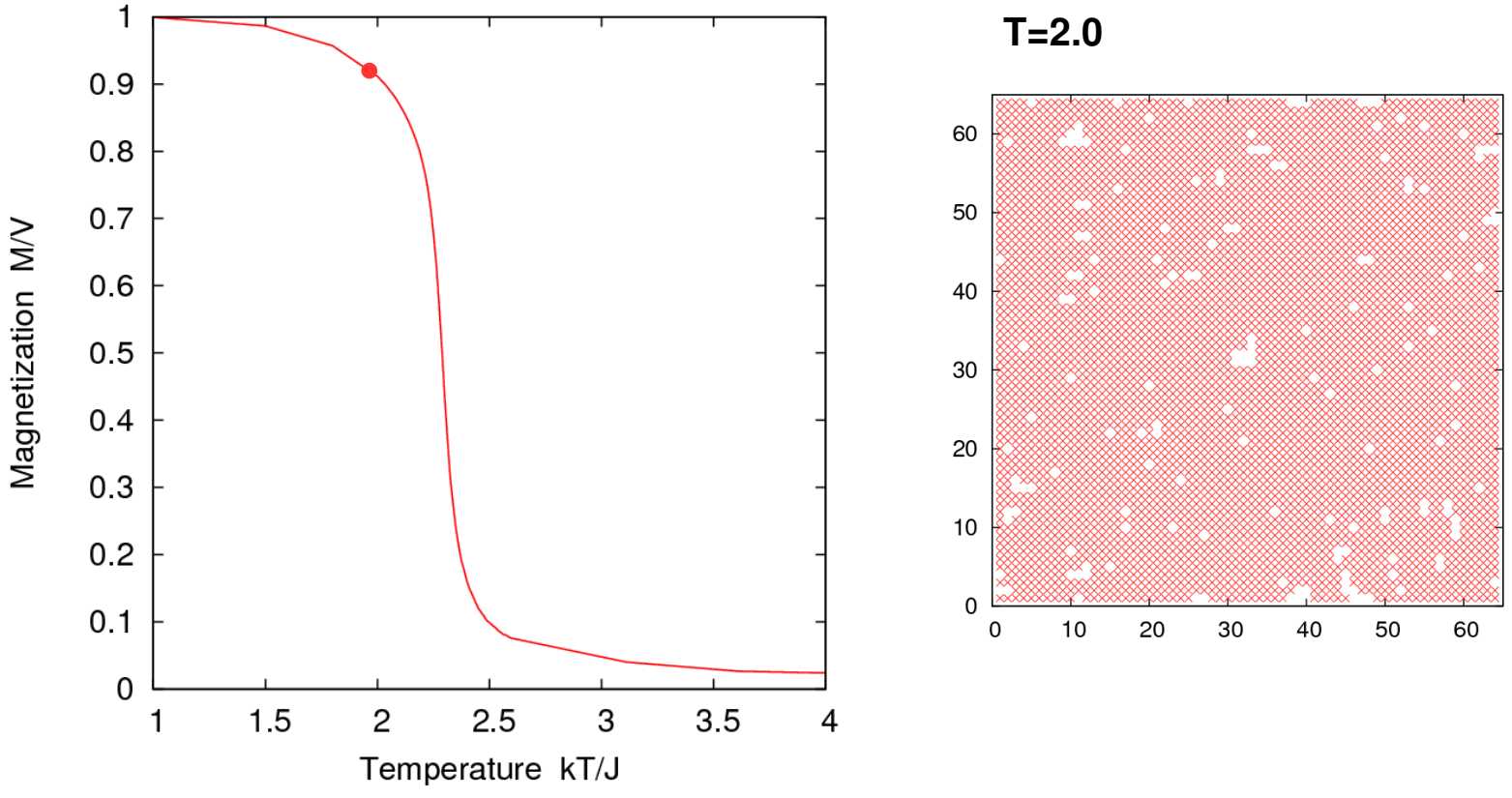
Heat capacity $$C = \frac{\partial \langle E \rangle}{\partial T} = \frac{1}{k_\mathrm{B} T^2} \left(\langle E^2 \rangle - \langle E \rangle^2 \right),$$

Mean magnetization $$\langle M \rangle = \left\langle \left| \sum_{i=1}^{N} S_i \right| \right\rangle,$$

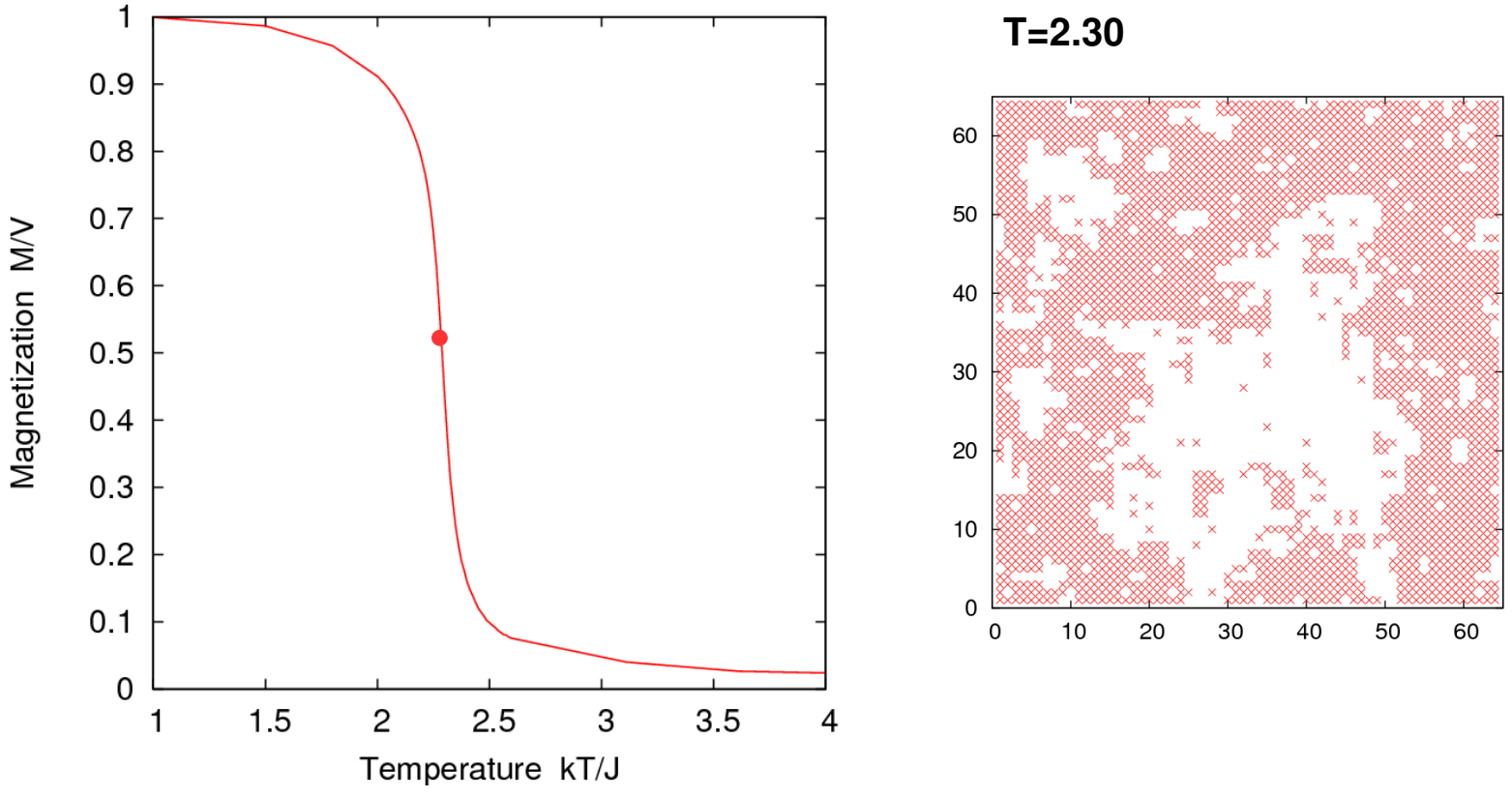
Linear magnetic susceptibility $$\chi = \frac{1}{k_\mathrm{B} T} \left(\langle M^2 \rangle - \langle M \rangle^2 \right),$$

where $\langle M \rangle$ and $\langle M^2 \rangle$ are evaluated at zero magnetic field ($B = 0$).

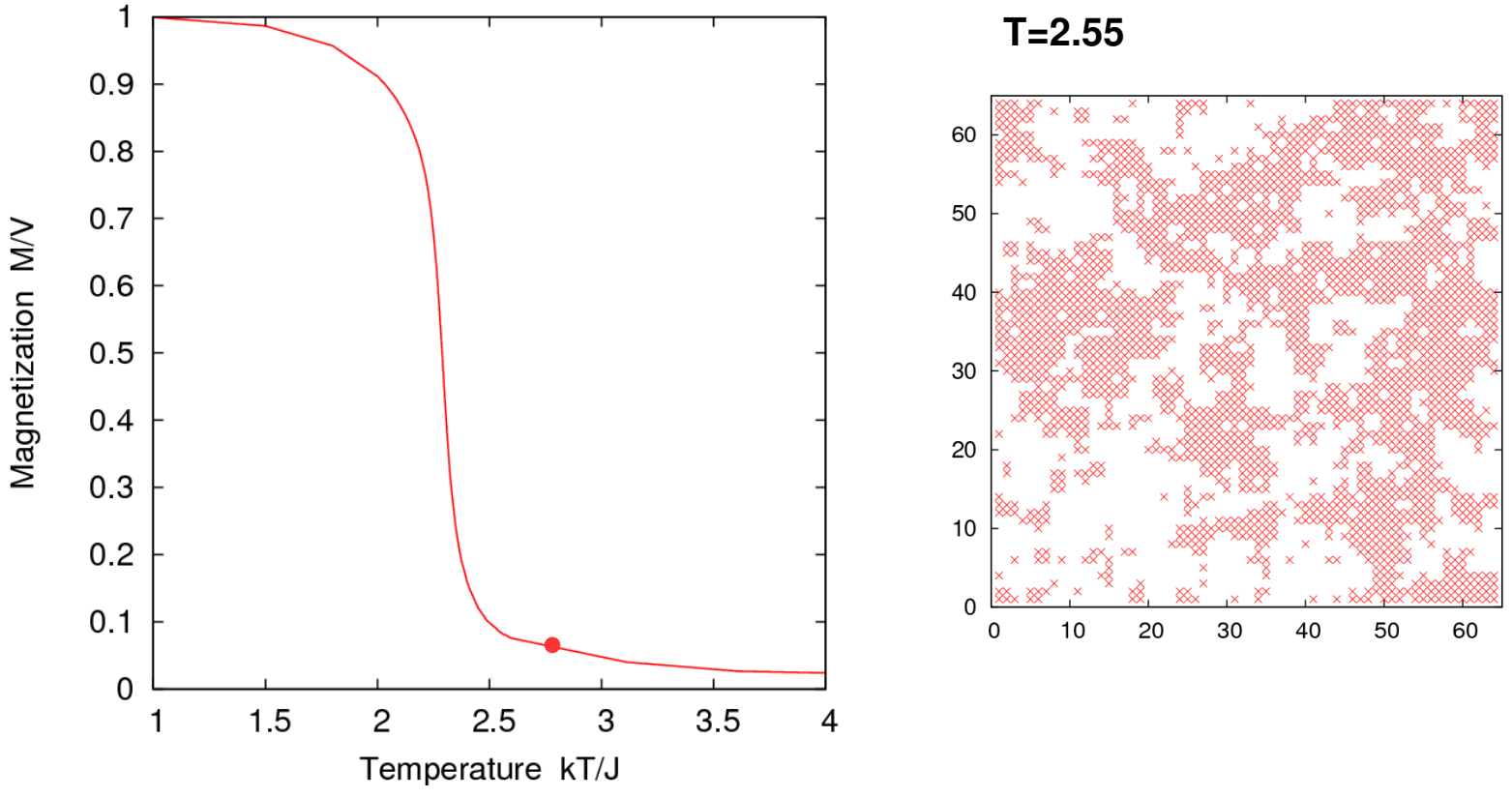
Magnetization in 2D Ising model ($J > 0$, $L^2 = 64^2$)



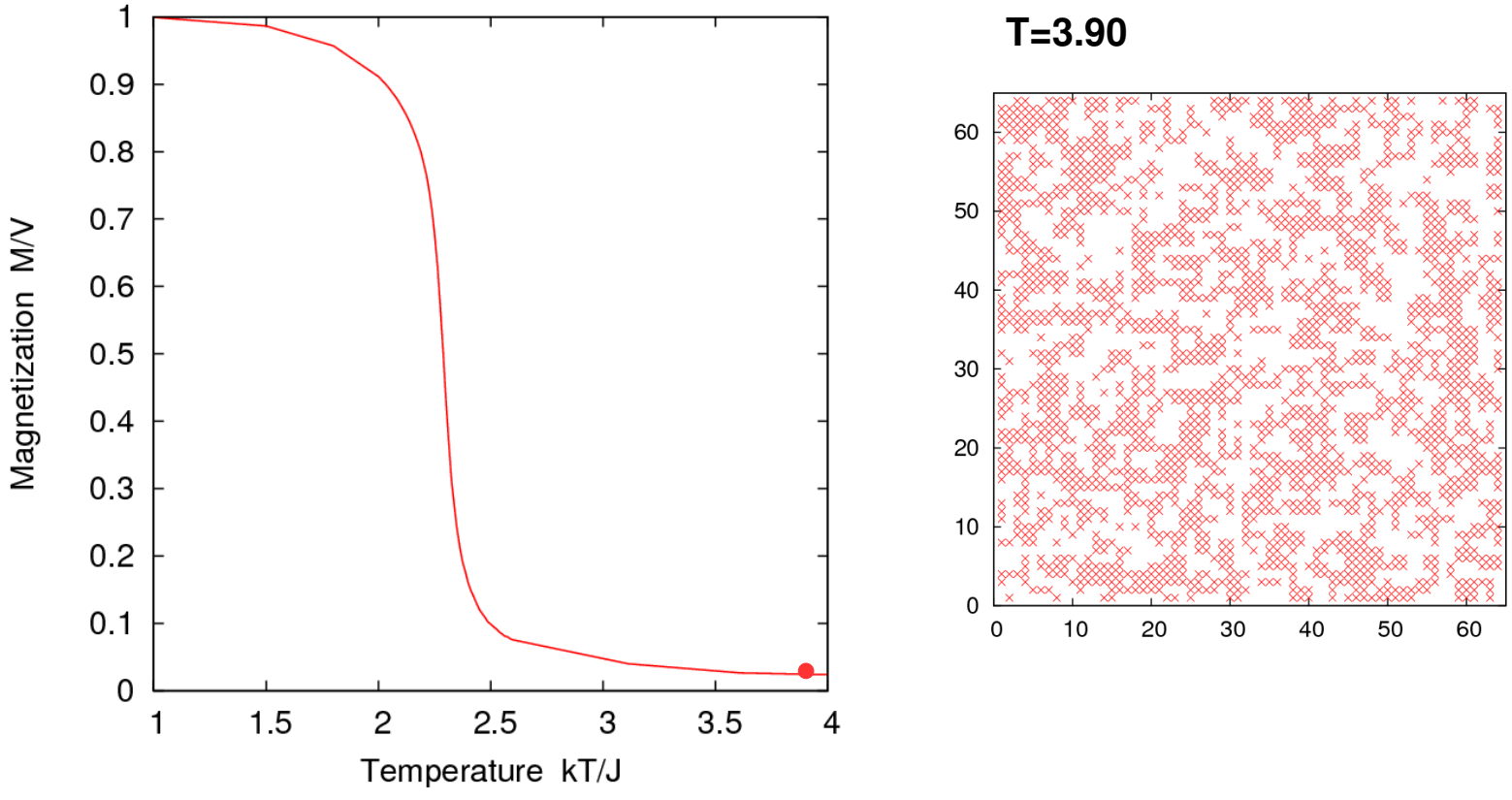
Magnetization in 2D Ising model ($J > 0$, $L^2 = 64^2$)



Magnetization in 2D Ising model ($J > 0$, $L^2 = 64^2$)



Magnetization in 2D Ising model ($J > 0$, $L^2 = 64^2$)



Monte Carlo simulation

Program that you will get

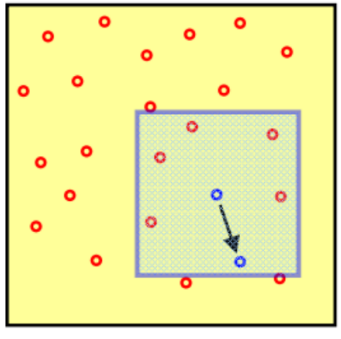
Homework

- Determine <M(T)> and step sizes for a few T

Metropolis sampling method (1953)

1. Start from initial (random) configuration R_0.
2. Randomly displace one (or more) of the particles.
3. Compute energy difference between two states: $\Delta E = V(R_{i+1}) - V(R_i)$.
4. Evaluate the transition probability which satisfies the detailed balance:

$$\upsilon(\mathbf{R}_i, \mathbf{R}_{i+1}) = \frac{p_B(\mathbf{R}_{i+1})}{p_B(\mathbf{R}_i)} = \min\left[1, e^{-\beta\Delta E}\right]$$



- $\Delta E \leq 0$: always accept new configuration.
- $\Delta E > 0$: accept with prob. $p = e^{-\beta\Delta E}$

5. Repeat steps (2)–(4) to obtain a final estimation: $\bar{A} = \langle A \rangle \pm \delta A$, with the error: $\delta A = \sqrt{\tau_A \sigma_A^2 / M}$.

- We reduce a number sampled configurations to $M \sim 10^6 \ldots 10^8$.
- We account only for configurations with non-vanishing weights: $e^{-\beta V(\mathbf{R}_i)}$.

Five minute lesson on Fortran

- Do loop: repeat the tasks indicated by the statements between the beginning and the end of the do loop, For the example, after the do loop x=a+(a+1)+(a+2)+…b

- x=0
- Do 123 i=a,b

  x=x+i

- 123 continue

From a spin configuration, change the orientation of a spin by some random amount, call this configuration j' .P(k): angle φ, T(k): angle θ of spin k

Start with some initial configuration and keep generating more configuration.

do ib=1,nblock0

To access the convergence to equilibrium, we calculate the averages at different instances in time over some small sample size and see how this changes. In your program, we divide the number of MC steps into blocks and look at the average of each block.

- DO 1 I=1,nstep
- *change the orientation of **one** spin by some random amount*
- DO 3 k=1,natoms
- *From a spin configuration, rotate a spin by some random amount, φ' pp=, $\theta' =$* Pt
- df, dt : size of the change, decided by you at the beginning; TP= 2 pi , dmod= modulo, duni() random number between 0 and 1
- PP(2)=dmod(P(k)+(duni()-0.5d0)*df,TP)
- Pt(2)=dmod(t(k)+(duni()-0.5d0)*dt,TP)

- *convert from angle into xyz (123) componets sn (1,2,3)*
- sn(3)=dcos(pt(2))
- sn(1)=dsin(pt(2))
- ens=dabs(sn(1))
- sn(2)=sn(1)*dsin(pp(2))
- sn(1)=sn(1)*dcos(pp(2))

Calculate E=E(anisotropy)+E.H+E (exchage) for the orginal and rotated spins.

- PP(1)=P(k)
- Pt(1)=t(k)
- eos=dabs(dsin(t(k)))
- so(1)=s(k,1)
- so(2)=s(k,2)
- so(3)=s(k,3)
- *c anisotropy energy change along direction lad, xa1, xa2 are constants put in at the beginning of the program*
- eoa=(xa1+xa2*so(lad)*so(lad))*so(lad)*so(lad)
- ena=(xa1+xa2*sn(lad)*sn(lad))*sn(lad)*sn(lad)
- *C coupling energy to the external magnetic field*
- enh=sn(lhd)*yh
- eoh=so(lhd)*yh

calculate the factor $\mathrm{p} = \exp[-\beta(E_j, - E_i)]$.

- *c exchange energy*
- enx=0.d0
- eox=0.d0
- dex=0.
- *C sum over nearest neughbouts, nn labels nearest neighbor*
- do ln=1,nnn
  - ll=nn(k,ln)
- dso=0.
- dsn=0.
- do l=1,3
- dso=dso+s(ll,l)*so(l)
- dsn=dsn+s(ll,l)*sn(l)
- enddo
-

calculate the factor $\mathrm{p} = \exp[-\beta(E_j, - E_i)]$.

dex=dex+dsn-dso

- enx=enx+dsn*xj
- eox=eox+dso*xj
- enddo
- *C* $E_j, - E_i$*=de*
- DE=enp*2.+ena+enh-eoh-(eop*2.+eoa)+dex*xj*2.
- C *p=pdr ,* ens, eos are from the Jacobian of the spkerical coordinate
- pdr=dexp(-de)*ens/eos

If p>1, then accept the move by updating the orginalangles with the new ones and call this the new configuration j.

- IF(pDr.gE.1.d0) THEN
- *c updating angles*
- P(k)=PP(2)
- t(k)=Pt(2)
- C we have looked at E/kT , tm is T
- c updating energies
- ex=ex+2.*(enx-eox)*tm
- ep=ep+2.*(enp-eop)*tm
- ea=ea+(ena-eoa)*tm
- eh=eh+(enh-eoh)*tm

If p>1, then accept the move by updating the orginalangles with the new ones and call this the new configuration j.

- *c updating components of the spins*
- do l=1,3
- s(k,l)=sn(l)
- sm(l)=sm(l)-so(l)+sn(l)
- enddo

Otherwise generate a random number x. If p>x, then we accept the move, otherwise the move is rejected.

- ELSE
- XX=duni()
- IF(XX.le.prob) THEN
- *C accepts the move, updsateing the information*
- P(k)=PP(2)
- t(k)=Pt(2)
- do l=1,3
- s(k,l)=sn(l)
- sm(l)=sm(l)-so(l)+sn(l)
- enddo
- ex=ex+2.*(enx-eox)*tm
- ep=ep+2.*(enp-eop)*tm
- ea=ea+(ena-eoa)*tm
- eh=eh+(enh-eoh)*tm

Typical output from my program

- computing3600 lattice sites dimensionality 2
- number density 0.11547E+01
- hexagonal close packed lattice q displs 0.00000 0.00000 0.50000 q displs 0.50000 test 1.31607401295249 3600 1.15470053837925 1800 0.75983566045761 1.31607396931524 1.24080643766120 60 30 0 npuc 2 nvacancies 0
- box size 0.60000E+02 0.51962E+02
- nearest and next nearest neighbor distance 0.10000E+01 0.17321E+01

Beginning output

- 1 Sx+y 0.67746 E+00 0.72000E+06 0.68645E+00 0.36000E+07 0.48081E-02
- 2 S 0.18253 E-01 0.72000E+06 0.22730E-01 0.36000E+07 0.35834E-02
- 3 Sz 0.16315 E-01 0.72000E+06 0.19596E-01 0.36000E+07 0.24062E-02
- 4 sx+y2 0.45904 E+00 0.72000E+06 0.47198E+00 0.36000E+07 0.71687E-02
- 5 s 2 0.43555 E-03 0.72000E+06 0.73160E-03 0.36000E+07 0.19822E-03
- 6 sz2 0.40368 E-03 0.72000E+06 0.56662E-03 0.36000E+07 0.12823E-03
- 7 edip 0.00000 E+00 0.72000E+06 0.00000E+00 0.36000E+07 0.00000E+00
- 8 exc 0.19394-E+01 0.72000E+06-0.19531E+01 0.36000E+07 0.91277E-02
- 9 eanis 0.13816-E+00 0.72000E+06-0.14037E+00 0.36000E+07 0.12595E-02
- 10 acrat 0.35598 E+00 0.72000E+06 0.35322E+00 0.36000E+07 0.19200E-02
- 11 eh 0.00000 E+00 0.72000E+06 0.00000E+00 0.36000E+07 0.00000E+00

Output at the end:

- 1 Sx+y 0.68181 E+00 0.72000E+06 0.68326E+00 0.14400E+08 0.14625E-02
- 2 S 0.12820 E-01 0.72000E+06 0.22834E-01 0.14400E+08 0.16745E-02
- 3 Sz 0.26563 E-01 0.72000E+06 0.22343E-01 0.14400E+08 0.18442E-02
- 4 sx+y2 0.46494 E+00 0.72000E+06 0.46712E+00 0.14400E+08 0.21214E-02
- 5 s 2 0.23026 E-03 0.72000E+06 0.79088E-03 0.14400E+08 0.11139E-03
- 6 sz2 0.10399 E-02 0.72000E+06 0.75116E-03 0.14400E+08 0.10449E-03
- 7 edip 0.00000 E+00 0.72000E+06 0.00000E+00 0.14400E+08 0.00000E+00
- 8 exc 0.19445-E+01 0.72000E+06-0.19471E+01 0.14400E+08 0.24222E-02
- 9 eanis 0.13935-E+00 0.72000E+06-0.13959E+00 0.14400E+08 0.35645E-03
- 10 acrat 0.35536 E+00 0.72000E+06 0.35447E+00 0.14400E+08 0.50297E-03
- 11 eh 0.00000 E+00 0.72000E+06 0.00000E+00 0.14400E+08 0.00000E+00

Simulations of 2D Ising model

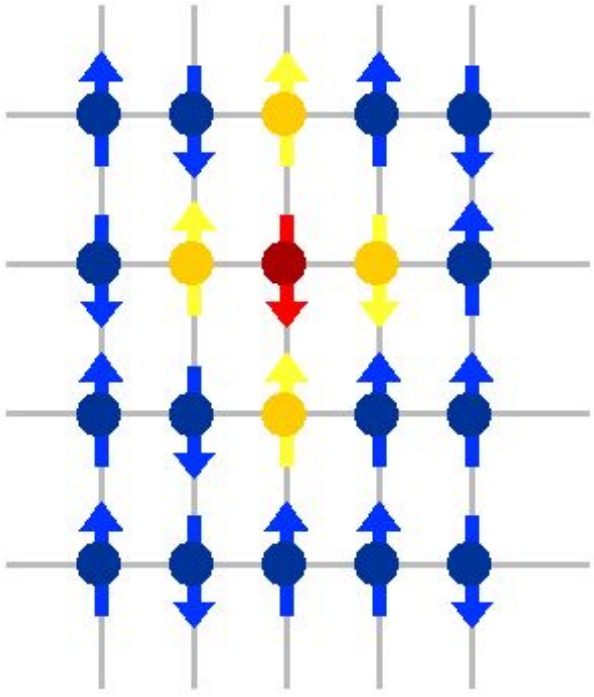


Figure: Lattice spin model with nearest neighbor interaction. The red site interacts only with the 4 adjacent yellow sites.

- We use the Ising model to demonstrate the studies of phase transitions.
- The Ising model considers the interaction of elementary objects called *spins* which are located at sites in a simple, 2-dimensional lattice,

$$\hat{H} = -J \sum_{i,j=nn(i)}^{N} \hat{S}_i \hat{S}_j - \mu_0 B \sum_{i=1}^{N} \hat{S}_i.$$

- Magnetic ordering:
  - $J > 0$: lowest energy state is *ferromagnetic*,
  - $J < 0$: lowest energy state is *antiferromagnetic*.

Equibrium properties

Mean energy $$\langle E \rangle = \mathrm{Tr}\, \hat{H}\hat{\rho},$$

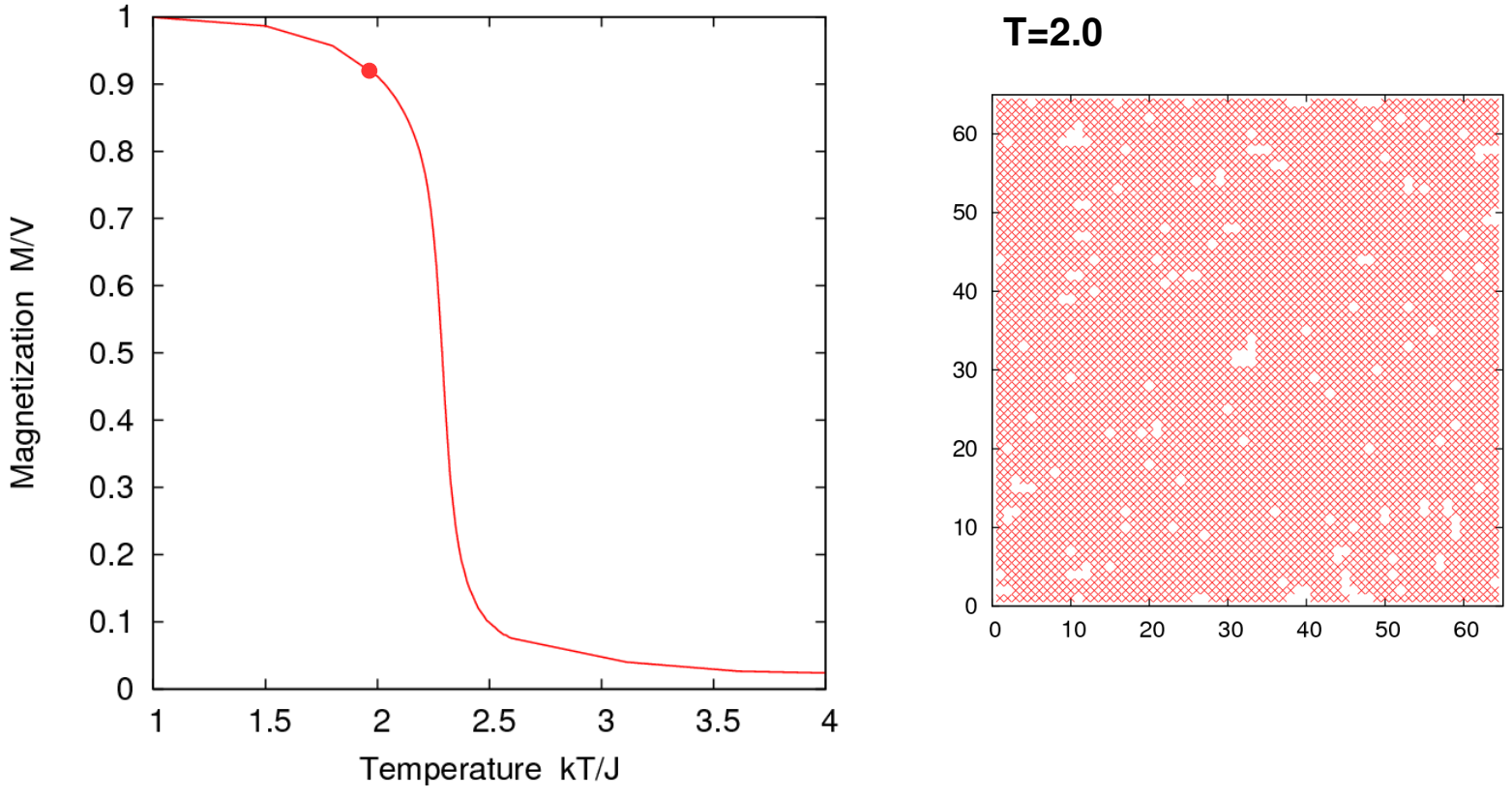
Heat capacity $$C = \frac{\partial \langle E \rangle}{\partial T} = \frac{1}{k_\mathrm{B} T^2} \left(\langle E^2 \rangle - \langle E \rangle^2 \right),$$

Mean magnetization $$\langle M \rangle = \left\langle \left| \sum_{i=1}^{N} S_i \right| \right\rangle,$$

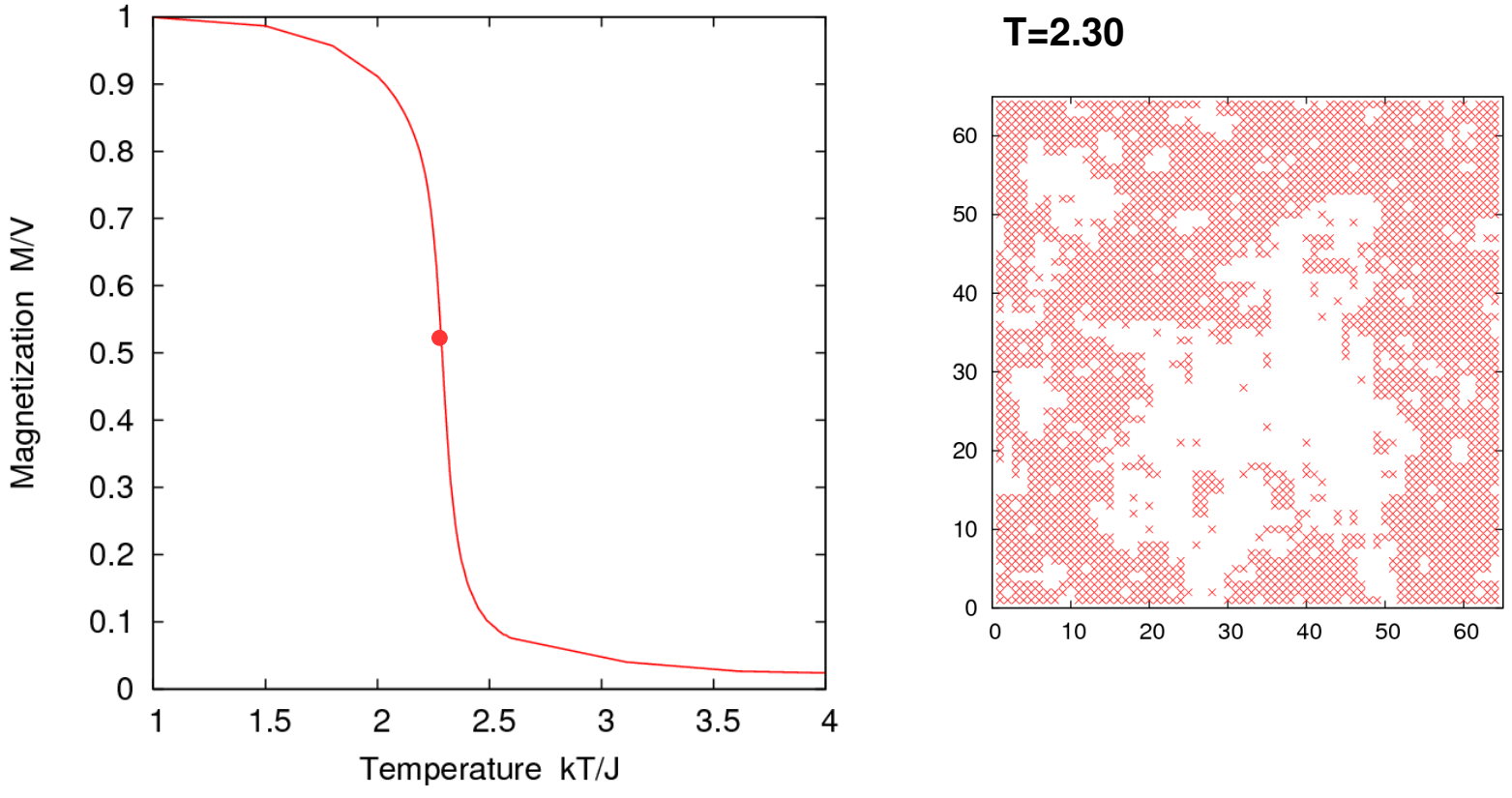
Linear magnetic susceptibility $$\chi = \frac{1}{k_\mathrm{B} T} \left(\langle M^2 \rangle - \langle M \rangle^2 \right),$$

where $\langle M \rangle$ and $\langle M^2 \rangle$ are evaluated at zero magnetic field ($B = 0$).

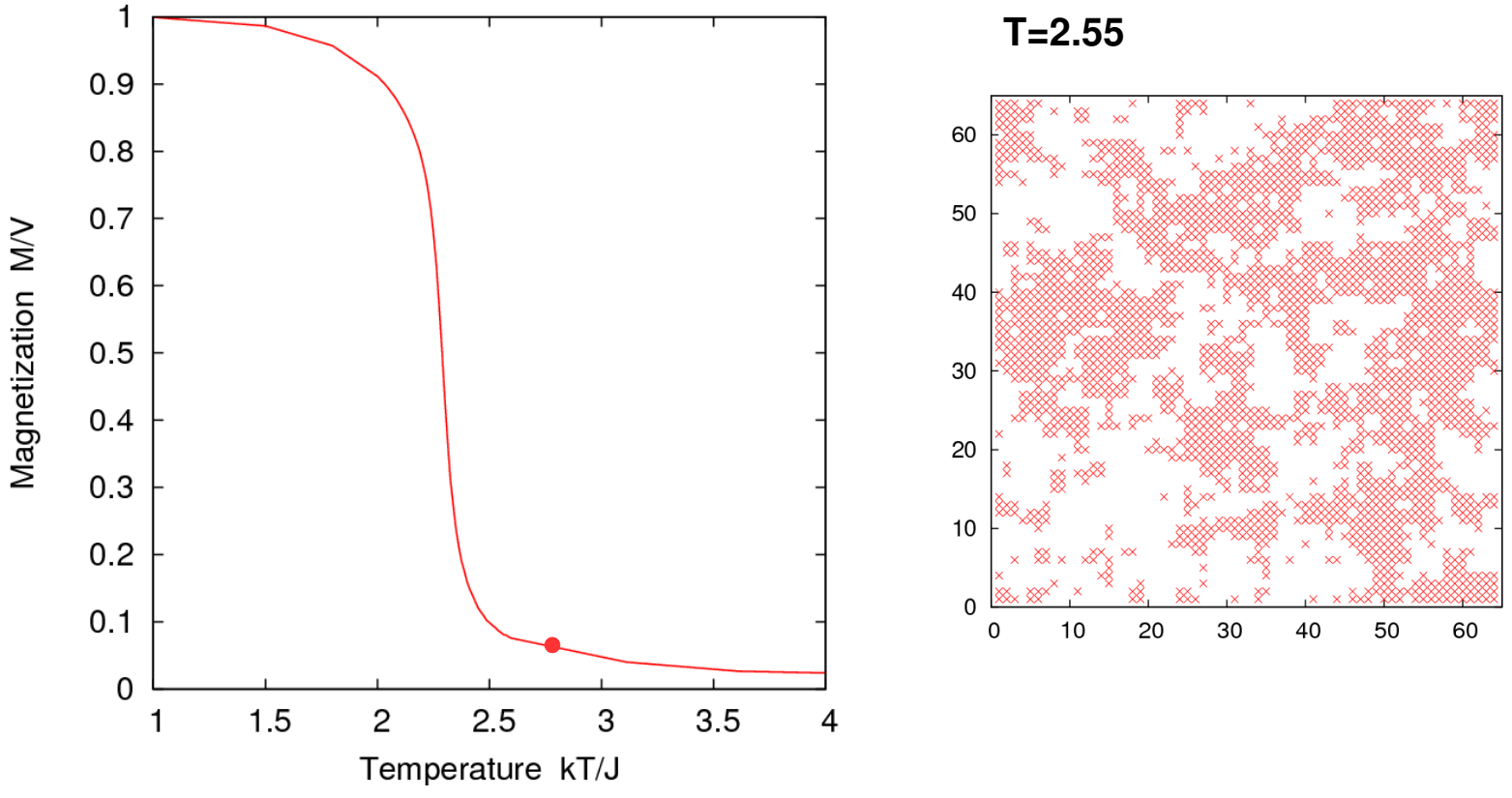
Magnetization in 2D Ising model ($J > 0$, $L^2 = 64^2$)



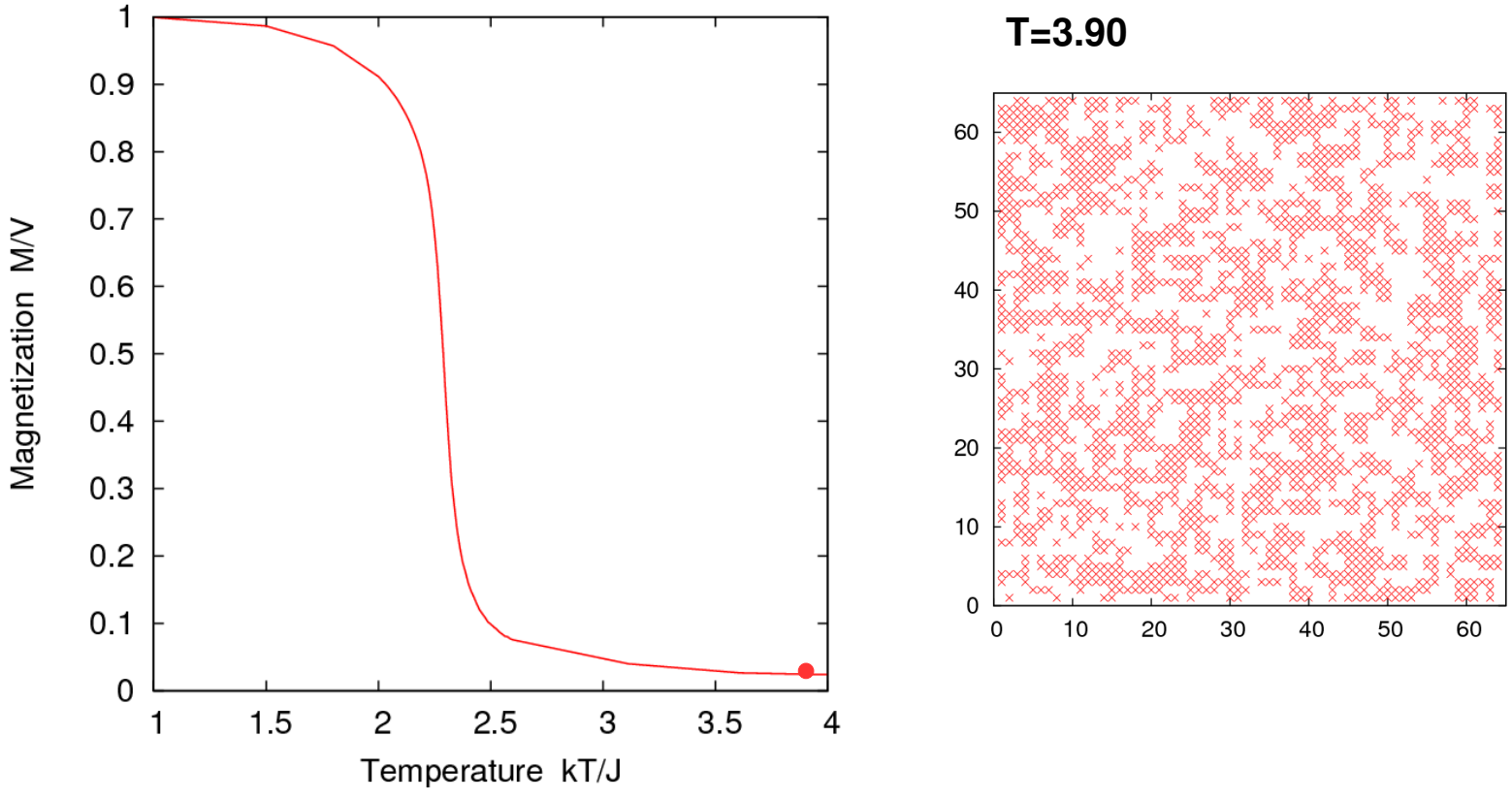
Magnetization in 2D Ising model ($J > 0$, $L^2 = 64^2$)



Magnetization in 2D Ising model ($J > 0$, $L^2 = 64^2$)



Magnetization in 2D Ising model ($J > 0$, $L^2 = 64^2$)



Histogram method

- In the averages of physical quantities $< Q > = \int Q\ exp(-\beta E)$, one may be interested in averages at more than one temperatures.
- For example,
- $< Q > = \int Q\ exp(-\beta' E) = \int Q' \exp[-\beta \mathrm{E}]$ where
- $Q' = Q\ exp[-(\beta' - \beta)E]$
- From the configuration x_i generated, we can calculate the energy $E(x_i)$ and count the number of configuration (histograms) $N_i$with energies within a mesh size ΔE of this energy. We then get
- $< Qf(E) > = \sum_i Q(x_i) N_i f[E(x_i)] / \sum_i N_i$

The master equation also provides a way to look at the approach to equilibrium and not just the equilibrium properties

- Start with an initial distribution P(t=0). As time develops, P approaches the equilibrium distribution.
- This can be seen mathematically as follows.
- We represent P(t=0) in terms of the eigenfunction of O:
- $O|n> = -\lambda_n|n>, \lambda_0 = 0;\ P(t=0) = \sum_n P_n|n>.$
- $;\ P(t) = \sum_n P_n e^{-\lambda_n t}|n>;\ P(t=\infty) = P_0|0>.$
- Close to a phase transition some of the eigenvalues are very small. This is called critical slowing down.