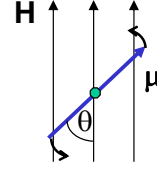# Magnetism, the Ising Model, and Monte Carlo Simulations

Paramagnetism:

Consider a compass needle with magnetic dipole moment μ in a uniform magnetic field **H**. The needle, mounted at its pivot point, experiences a torque that tends to align the dipole with the magnetic field.

The corresponding potential energy is given by $\qquad E = -\vec{\mu} \cdot \vec{H} = -\mu H \cos(\theta)$

The classical theory of paramagnetism considers a system of N non-interacting magnetic dipoles, each with magnetic moment μ, on a lattice.
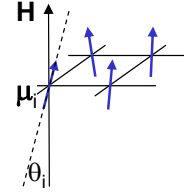
The energy is the sum of the contributions of the individual dipoles.

$$E = -\sum_{i=1}^{N} \vec{\mu} \cdot \vec{H}_i = -\sum_{i=1}^{N} \mu_i H \cos\theta_i$$

In quantum mechanics, the dipole moment μ and its projection μ_z along the z-axis (direction of the magnetic field H) have discrete values.

$$\mu = g\mu_B J \quad \text{with} \quad J = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \ldots \quad \text{or} \quad J = 0,1,2,\ldots$$

$$\mu_z = g\mu_B m' \quad \text{with} \quad m' = -J, -J+1, \ldots, J-1, J$$

$$g = \frac{3}{2} + \frac{S(S+1) - L(L+1)}{2J(J+1)} \quad \text{Lande's g - factor}$$

$$\mu_B = \frac{e\hbar}{2m_0 c} \quad \text{Bohr magneton, } m_0 \text{ is the electron mass}$$

$$\varepsilon_{m'} = -g\mu_B H m' \quad \text{energy eigenvalues}$$

Note: to simplify notation, we will absorb constants like the Lande g-factor, the Bohr magneton, etc. into the definition of our variables.

$$\mu_z = g\mu_B m' = \pm\mu_B \rightarrow s = \pm 1$$
$$m = \bar{\mu}_z \rightarrow m = \bar{s}$$
$$\mu_B H \rightarrow B$$

The Hamiltonian is the function for the energy of the system.

With the new variables, the Hamiltonian for a paramagnetic system of "spin 1/2" particles is

$$H_{\text{paramagnetic}} = -B\sum_{i=1}^{N} s_i$$

The Ising model

Originally a model for <u>ferromagnetism</u>, it has applications to many other systems that undergo phase transitions (remember our discussion of phase transitions in fluids).

A material is <u>ferromagnetic</u> if it has a non-zero magnetization in zero magnetic field (spontaneous magnetization). As the temperature increases, the spontaneous magnetization decreases and vanishes at a critical temperature, the Curie temperature T_c. For temperatures larger than T_c the material is paramagnetic.
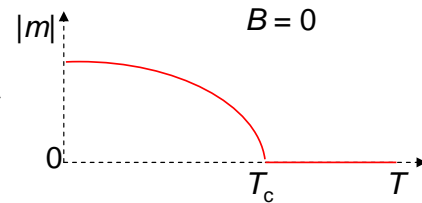
No spontaneous magnetization for ideal (= non-interacting) paramagnets $\Rightarrow$ include <u>interactions</u>

Ferromagnetism is observed only in the solid state $\Rightarrow$ employ a <u>lattice model</u>

Consider a regular lattice of N sites with each site occupied by a particle with spin 1/2. If there is a magnetic field, we will assume that it is parallel to the z-axis. We will also assume that the magnetic moments of the particles are aligned with the z-axis so that they can point either up or down.
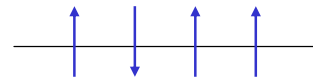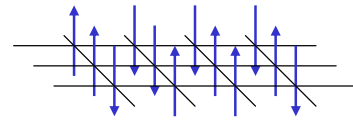
dimension
$d = 1$
coordination number
$z = 2$

$d = 2$
$z = 4$

3

In the Ising model, each spin is assumed to <u>interact only with its nearest neighbors</u> on the lattice. Since spontaneous magnetization implies that the spins are aligned with each other in the absence of a magnetic field, it is natural to include a term that will lower the energy if neighboring spins are parallel and raise the energy if the spins are antiparallel.

The interaction part of the Hamiltonian is written as:
$$H_{\text{int}} = -J \sum_{\langle i,j \rangle} s_i s_j$$

where J is a positive constant (not to be confused with the total angular momentum in quantum mechanics) that indicates the strength of the interaction and the sum runs over nearest neighbor pairs $\langle i,j \rangle$.

For a lattice with coordination number z, there are $Nz/2$ terms in H_{int} (each of the N spins has z nearest neighbors, the factor 1/2 avoids double counting).

The Ising Hamiltonian is the sum of the interacting and paramagnetic terms

$$H_{\text{Ising}} = -J \sum_{\langle i,j \rangle} s_i s_j - B \sum_{i=1}^{N} s_i$$

4

Monte Carlo Simulations

A method to determine thermal averages of observable quantities like the internal energy, the magnetization of an Ising system, the coverage of a surface, etc.

Let O be some observable quantity
O_μ its value in the microstate μ
p_μ the probability to be in microstate μ
$<O>$ its average in the canonical ensemble

$$\langle O \rangle = \frac{\sum_\mu O_\mu e^{-\beta E_\mu}}{\sum_\mu e^{-\beta E_\mu}} = \sum_\mu p_\mu O_\mu$$

If the system is very small, we can generate all microstates, calculate the probabilities, and evaluate the average $<O>$ exactly $\rightarrow$ exact enumeration .

In general, this is impossible.
Example: For the Ising model with N spins the number of microstates is $2^N \sim 10^{77}$ for N=16*16

$\Rightarrow$ Select a subset $\{\mu_1, \mu_2,\ldots, \mu_M\}$ of microstates and approximate $<O>$ by summing only over these states.

Remember: the probability distribution is strongly peaked near the most probable value.

$$\langle O \rangle_{app} = \frac{\sum_{k=1}^{M} O_{\mu_k} e^{-\beta E_{\mu_k}}}{\sum_{k=1}^{M} e^{-\beta E_{\mu_k}}}$$

$\Rightarrow$ A set $\{\mu_1, \mu_2,\ldots, \mu_M\}$ of microstates chosen at random with uniform probability (random sampling) is likely to contain mostly improbable states $\Rightarrow <O>_{app}$ is a very poor approximation to $<O>$.

Challenge: Generating probable microstates (importance sampling)
        $\rightarrow$ Metropolis algorithm

Monte Carlo Simulations of the two-dimensional Ising model on a square lattice

The microstates

Lattice parameters:
linear dimension L (here: L=16)
total number of lattice sites $N = L*L$ (here N=256)
coordination number $z = 4$

Each site of the lattice is described by coordinates (x,y) with $1 \le x \le L$ and $1 \le y \le L$.

Each site of the lattice is occupied by a spin, which can have values +1 or -1 .

$\Rightarrow$ We know the microstate of the system if we know the value of the spin at each lattice site (x,y).

microstate $= \{\text{spin}(x, y) : x, y \in \{1,2,\cdots,L\}, \text{spin}(x, y) = \pm 1\}$

$\Rightarrow$ Describe the microstate by an $L \times L$ matrix:
x = row number
y = column number
spin(x,y) = matrix element

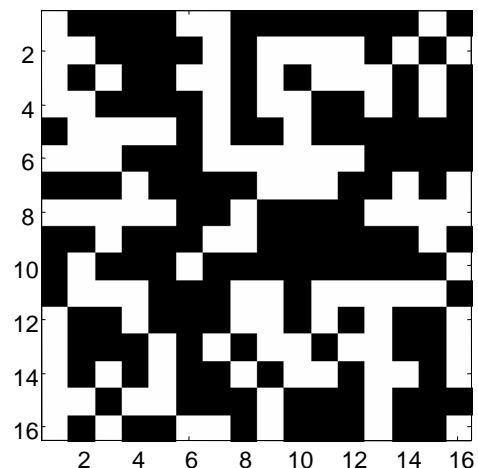Image of a microstate:
black: spin(x,y) = −1
white: spin(x,y) = +1

In Matlab:

<u>Creating an initial microstate</u>

```matlab
% set up the initial configuration
for y=1:L
   for x=1:L
      if rand>0.5    % spins are randomly set to 1 or 0
         spin(x,y)=1;
      else
         spin(x,y)=-1;
      end
      % alternative initial configurations;
      % uniform spin configuration
      % spin(x,y)=1;
      % checker board spin configuration
      % spin(x,y)=2*mod(x+y,2)-1;
   end
end
```

<u>Evaluating a microstate</u>

The <u>magnetization</u> in our units is the sum over the spins

total magnetization : $\quad M = \sum_{y=1}^{L} \sum_{x=1}^{L} \text{spin}(x, y)$

magnetization per spin : $\quad m = M / N \equiv M_{\text{st}}$

note: for small systems, focus on $|m|$ rather than m

The <u>internal energy</u> E is evaluated from the

Hamiltonian : $\quad H = -J \sum_{\langle i,j \rangle} s_i s_j - B \sum_{i=1}^{N} s_i$

under <u>periodic boundary conditions.</u>

magnetization per site -0.10938
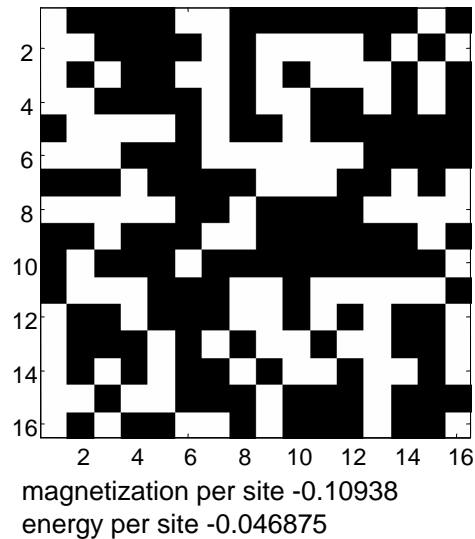energy per site -0.046875

Define:

$$\text{ip}(k) \equiv \begin{cases} k-1 & \text{for } k = 2 \cdots L \\ L & \text{for } k = 1 \end{cases}$$

$$\text{is}(k) \equiv \begin{cases} k+1 & \text{for } k = 1 \cdots L-1 \\ 1 & \text{for } k = L \end{cases}$$

$$\text{spinsum} \equiv \text{spin}(\text{ip}(x), y) + \text{spin}(\text{is}(x), y) + \text{spin}(x, \text{ip}(y)) + \text{spin}(x, \text{is}(y))$$

$\Rightarrow$ total internal energy : $\quad E = \sum_{x=1}^{L} \sum_{y=1}^{L} s(x, y) * \left[-B - \frac{1}{2} J * \text{spinsum} \right]$

$\Rightarrow$ internal energy per site : $\quad E_{\text{st}} = E / N$

<u>Metropolis algorithm for the Ising model</u>

(0)   Prepare an initial state
(1)   Choose a site
(2)   Calculate the energy change ΔE which results if the spin at the site is flipped
(3)   Decide if the spin is to be flipped:
                if $\Delta E < 0$ flip the spin
                if $\Delta E > 0$ flip only with probability $e^{-\beta\Delta E}$
                      $\rightarrow$ generate a random number r with $0 < r < 1$
                              if $r < e^{-\beta\Delta E}$ flip the spin, otherwise don't flip the spin
(4)   Repeat from (1)

Note:
The sequence of steps (1)-(4) constitutes one attempted spin flip, the time is τ_{trial} .
We will "sweep" through the lattice, i.e. visit the sites in turn and attempt to flip the spin.
One attempted flip per spin (one sweep through the lattice) is one Monte Carlo step $\tau_{\text{MC}} = N\tau_{\text{trial}}$

In the program, all quantities are dimensionless:

$$E \rightarrow E/|J| \quad \text{with} \quad J = \begin{cases} +1 & \text{for ferromagnetic systems} \\ -1 & \text{for antiferromagnetic systems} \end{cases}$$

$$B \rightarrow B/|J|$$

$$T \rightarrow T_{\text{red}} = k_{\text{B}}T/|J|$$

From the program ising_visual.m the code for a single Monte Carlo step

```matlab
for x=1:L  % visit the sites in turn
   for y=1:L
      %calculate the energy contribution (n.n. interaction)
      spinsum=spin(ip(x),y)+spin(is(x),y)...
         +spin(x,ip(y))+spin(x,is(y));
      inter=-J*spin(x,y)*spinsum-B*spin(x,y);
      %The change in energy if the spin is flipped is -2*inter, hence
      %if inter>0 then energy is gained by flipping a spin
      %therefore flip the spin.
      %if inter<0 then energy is lost by a spin flip,
      %therefore flip only with Boltzmann probability
      if inter>0
         spin(x,y)=-spin(x,y);          % flip the spin
         mcurrent=mcurrent+2*spin(x,y); % update the magnetization
         ecurrent=ecurrent-2*inter;     % update the internal energy
      else
         boltz=exp(2*inter/Tred);
         if rand<boltz
            spin(x,y)=-spin(x,y);          % flip the spin
            mcurrent=mcurrent+2*spin(x,y);% update the magnetization
            ecurrent=ecurrent-2*inter;    % update the internal energy
         end
      end
   end
end
```

Time scales of Monte Carlo simulations

τ_{trial}	time for an elementary move attempt
$\tau_{MC}=N\tau_{trial}$	time for one attempted elementary move per "particle" = one Monte Carlo step = 1 MC step • <u>basic unit of time</u>
$\tau_{corr}=k_{corr}\tau_{MC}$	time (in MC steps) over which microstates are correlated • estimate from *autocorrelation* functions, see below
$\tau_{equilib}=k_{equilb}\tau_{MC}$	time for the system to reach equilibrium • depends on the circumstances (temperature, initial state, etc.) • must be determined by experience • <u>major source of error</u>: evaluating microstates before the system has reached equilibrium
$\tau_{obs}=k_{obs}\tau_{MC}$	time over which microstates are evaluated • demand $\tau_{obs} \gg \tau_{corr}$
$\tau_{run}=\tau_{equilib}+\tau_{obs}$ $\tau_{run}=k_{run}\tau_{MC}$	total run time
$\tau_{block}=\tau_{obs}/n_{block}$ $\tau_{block}=k_{block}\tau_{MC}$	microstates within each block are averaged separately and the block averages are used for error estimates, see below

Evaluation of simulation results and error estimates from block averaging
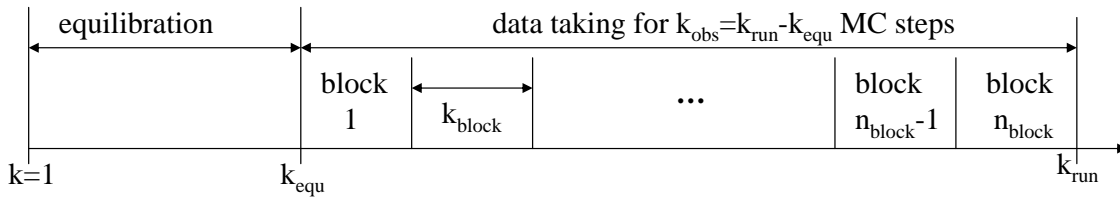
For each MC step after equilibration ($k > k_{equ}$) calculate the quantities of interest, for example the mean squared end-to-end distance of a chain or the internal energy of a fluid. We describe the block-averaging procedure here for a general variable that we call *m*.

Question: Why can't we just average over the whole run and obtain the mean and the standard deviation from those data?

Answer: The standard deviation σ is a measure for the fluctuations of a quantity about its mean. These fluctuations have physical meaning and are called generalized susceptibilities. The fluctuations of the energy, for example, are proportional to the heat capacity. The fluctuations tell us about a physical property of the system, not about the quality of the simulation.

Block averaging: divide the observation time $\tau_{obs}= k_{obs}\tau_{MC}$ into blocks, calculate the average within each block, and then the average over blocks and calculate the standard error.

$$k_{obs} = n_{block}k_{block} \quad \text{for example} \quad k_{obs} =1000, \quad n_{block} =10 \quad \Rightarrow \quad k_{block} =100$$

For each block $b \in \{1, \cdots, n_{\text{block}}\}$ calculate the average within the block

$$m(b) = \frac{1}{k_{\text{block}}} \sum_{k=k_b+1}^{k_b+k_{\text{block}}} m_k \quad \text{with} \quad k_b = (b-1)k_{\text{block}}$$

calculate the average over the blocks

$$\langle m \rangle = \frac{1}{n_{\text{block}}} \sum_{b=1}^{n_{\text{block}}} m(b)$$

and the standard error of the mean

$$\sigma_m = \left(\frac{1}{n_{\text{block}}(n_{\text{block}}-1)} \sum_{b=1}^{n_{\text{block}}} (m(b) - \langle m \rangle)^2 \right)^{1/2}$$

Note: Block averaging is only one method to estimate <u>statistical errors</u> of a simulation, other methods (e.g. "boot-strapping") are available. All have their advantages and disadvantages, but it is important that you have some idea how reliable your results are and that you report your error estimates.

Note: Statistical errors give no information about systematic errors.

<u>Systematic errors:</u>

The most common systematic error is <u>insufficient equilibration</u>. Run your simulations with different equilibration times (at least one very long one) and compare the results to be sure you are not reporting equilibration effects.

Poor <u>random number generators</u> used to be a source of systematic errors. Random number generators have improved, but you should check before you use one for the first time.

<u>Finite-size effects</u> will affect your results. They may be so strong, that your simulation is not representative of the macroscopic system you are trying to simulate (e.g. if your simulation box for a simulation of chain molecules that you expect to crystalize is so small that the chains cannot possibly arrange themselves in a crystalline structure). In the more benign cases, your results will depend only quantitatively on the system size. You should run simulations for different system sizes and discern the trends. You may be able to extrapolate to infinite system size.

<u>Non-ergodic algorithms</u> do not allow the system to explore (in principle) all of its microstates. There is no way of proving that an algorithm is ergodic, so you will have to be careful if you try a new algorithm.

<u>Configurational bias</u> is the preferential sampling of certain microstates. This is not necessarily a problem. The so-called "configurational-bias" Monte Carlo methods employ biased algorithms to speed up the simulation. However, if you are not aware of a bias, your results will not reflect the equilibrium state.

Autocorrelation functions

<u>Example: Two-time correlation function of the internal energy</u>

The <u>two-time correlation function</u> of the energy is defined as

$$\hat{C}(t) = \frac{1}{\tau_{\mathrm{obs}} - t} \int_0^{\tau_{\mathrm{obs}} - t} dt' \left(E(t') - \langle E \rangle \right)\left(E(t + t') - \langle E \rangle \right)$$

$$\Rightarrow \hat{C}(t) = \frac{1}{\tau_{\mathrm{obs}} - t} \int_0^{\tau_{\mathrm{obs}} - t} dt' \left(E(t') E(t + t') - \langle E \rangle^2 \right)$$

For discrete time, measured in MC steps

$$\langle E \rangle = \frac{1}{k_{\mathrm{obs}}} \sum_{k=1}^{k_{\mathrm{obs}}} E_k \qquad \text{where} \quad E_k = \text{internal energy after } k \text{ MC steps}$$

$$\hat{C}(t_j) = \frac{1}{k_{\mathrm{obs}} - j} \sum_{l=1}^{k_{\mathrm{obs}} - j} E_l E_{l+j} - \left(\frac{1}{k_{\mathrm{obs}} - j} \sum_{l=1}^{k_{\mathrm{obs}} - j} E_l \right)\left(\frac{1}{k_{\mathrm{obs}} - j} \sum_{l=1}^{k_{\mathrm{obs}} - j} E_{l+j} \right)$$

Example for autocorrelation functions: magnetization and heat capacity of the *Ising model*

The autocorrelation functions are expected to decay exponentially $\sim e^{-t/\tau}$
and allow us to estimate the correlation times. Note that the correlation times for the
magnetization and the energy differ from each other. The observation time τ_{obs} has to be
much larger than the longest correlation time τ_{corr} .
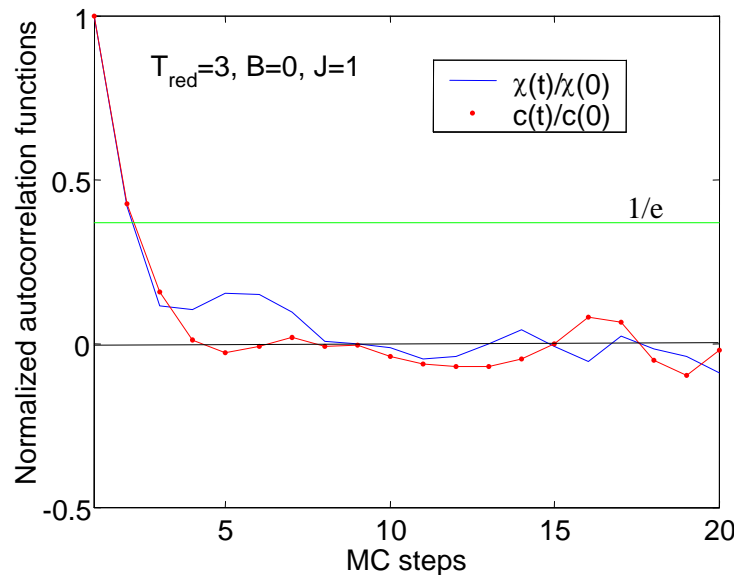
Relation to heat capacity

The constant - volume heat capacity is defined as

$$C = \left(\frac{\partial E}{\partial T}\right)_{N,V}$$

and related to the fluctuations of the internal energy **at fixed temperature**

$$C = \frac{1}{k_B T^2}\left\langle\left(E - \langle E\rangle\right)^2\right\rangle = \frac{1}{k_B T^2}\left(\langle E^2\rangle - \langle E\rangle^2\right)$$

The quasi-ergodic hypothesis relates ensemble averages to time averages.

$$\Rightarrow \quad \langle E\rangle = \frac{1}{\tau_{obs}}\int_0^{\tau_{obs}} dt'\, E(t') \quad \text{and} \quad \langle E^2\rangle = \frac{1}{\tau_{obs}}\int_0^{\tau_{obs}} dt'\, E^2(t')$$

$$\Rightarrow \quad C = \frac{1}{k_B T^2}\frac{1}{\tau_{obs}}\int_0^{\tau_{obs}} dt'\left(E^2(t') - \langle E\rangle^2\right)$$

compare with the definition of the time correlation function

$$\Rightarrow \quad C = \frac{1}{k_B T^2}\hat{C}(0)$$

17