The Ising model

Originally a model for ferromagnetism, it has applications to many other systems that undergo phase transitions (see e.g. Huang).

A material is ferromagnetic 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 ⇒ include interactions
Ferromagnetism is observed only in the solid state ⇒ employ a lattice model

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 $z=2$
- dimension $d=2$
- coordination number $z=4$
Note: to simplify notation, we will absorb constants like the Lande $g$-factor, the Bohr magneton, etc. into the definition of our variables.

\[ \mu_z = g\mu_B m = \pm \mu_B \rightarrow s = \pm 1 \]

\[ m = \mu_z \rightarrow m = \bar{s} \]

\[ \mu_B H \rightarrow B \]

The Hamiltonian for the paramagnetic system becomes

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

In the Ising model, each spin is assumed to interact only with its nearest neighbors 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:

\[ H_{\text{int}} = -J \sum_{\langle i,j \rangle} s_i s_j \]

The sum runs over nearest neighbor pairs \( \langle i,j \rangle \). For a lattice with coordination number \( z \), there are \( Nz/2 \) terms in \( H_{\text{int}} \) (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

\[ H_{\text{Ising}} = -J \sum_{\langle i,j \rangle} s_i s_j - B \sum_{i=1}^{N} s_i \]
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
\[
O_{\mu} \quad \text{its value in the microstate } \mu
\]
\( p_{\mu} \) the probability to be in microstate \( \mu \)
\( \langle O \rangle \) 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 \( \langle O \rangle \) 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 \( N = 16 \times 16 \)

\[ \Rightarrow \]
Select a subset \( \{ \mu_1, \mu_2, \ldots, \mu_M \} \) of microstates and approximate \( \langle O \rangle \) by summing only over these states.

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

\[ \Rightarrow \]
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 \( \Rightarrow \langle O \rangle_{\text{app}} \) is a very poor approximation to \( \langle O \rangle \).

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 \( N = L^2 \) (here \( N = 256 \))
- coordination number \( z = 4 \)

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

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

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

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

⇒ Describe the microstate by an \( L \times L \) matrix:
- \( x \) = row number
- \( y \) = column number
- \( \text{spin}(x, y) \) = matrix element

Image of a microstate:
black: \( \text{spin}(x, y) = -1 \)
white: \( \text{spin}(x, y) = +1 \)
Creating an initial microstate

```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
    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
```

» help rand
RAND   Uniformly distributed random numbers. ... This generator can generate all the floating point numbers in the closed interval \([2^{-53}, 1-2^{-53}]\). Theoretically, it can generate over \(2^{1492}\) values before repeating itself.
Evaluating a microstate

The magnetization in our units is the sum over the spins

total magnetization: \[ M = \sum_{y=1}^{L} \sum_{x=1}^{L} \text{spin}(x, y) \]
magnetization per spin: \[ m = M / N \equiv M_{st} \]

note: for small systems, focus on \(|m|\) rather than \(m\)

The internal energy \(E\) is evaluated from the Hamiltonian:
\[ H = -J \sum_{\langle i,j \rangle} s_i s_j - B \sum_{i=1}^{N} s_i \]
under periodic boundary conditions.

Define:
\[ \text{ip}(k) = \begin{cases} k - 1 & \text{for } k = 2 \cdots L \\ L & \text{for } k = 1 \end{cases} \]
\[ \text{is}(k) = \begin{cases} k + 1 & \text{for } k = 1 \cdots L - 1 \\ 1 & \text{for } k = L \end{cases} \]

\[ \text{spinsum} = \text{spin}(\text{ip}(x), y) + \text{spin}(\text{is}(x), y) + \text{spin}(x, \text{ip}(y)) + \text{spin}(x, \text{is}(y)) \]

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

\[ \Rightarrow \text{internal energy per site: } E_{st} = E / N \]
Metropolis algorithm for the Ising model

(0) Prepare an initial state
(1) Choose a site
(2) Calculate the energy change $\Delta E$ 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 $\tau_{\text{flip}}$.
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 $\tau_{\text{MC}} = N \tau_{\text{flip}}$

In the program, all quantities are dimensionless:

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

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

$$T \rightarrow T_{\text{red}} = k_B T / |J|$$
From the program ising_visual.m the code for a single Monte Carlo step

```plaintext
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

<table>
<thead>
<tr>
<th>( \tau_{\text{flip}} )</th>
<th>time for a single spin flip attempt</th>
</tr>
</thead>
</table>
| \( \tau_{\text{MC}} = N \tau_{\text{flip}} \) | time for one attempted flip per spin = one Monte Carlo step = 1 MC step  
  * basic unit of time |
| \( \tau_{\text{corr}} = k_{\text{corr}} \tau_{\text{MC}} \) | time (in MC steps) over which microstates are correlated  
  * estimate from autocorrelation functions  
  ➢ estimate from plots of ising_corr.m |
| \( \tau_{\text{equilib}} = k_{\text{equilib}} \tau_{\text{MC}} \) | time for the system to reach equilibrium  
  * depends on the circumstances (temperature, initial state, etc.)  
  * must be determined by experience  
  ➢ estimate from plots of ising_visual.m  
  * major source of error: evaluating microstates before the system has reached equilibrium |
| \( \tau_{\text{obs}} = k_{\text{obs}} \tau_{\text{MC}} \) | time over which microstates are evaluated  
  * demand \( \tau_{\text{obs}} >> \tau_{\text{corr}} \) |
| \( \tau_{\text{run}} = \tau_{\text{equilib}} + \tau_{\text{obs}} \) | total run time |
| \( \tau_{\text{run}} = k_{\text{run}} \tau_{\text{MC}} \) | |
| \( \tau_{\text{block}} = \tau_{\text{obs}} / n_{\text{block}} \) | microstates within each block are averaged separately and the block averages are used for error estimates, see below |
| \( \tau_{\text{block}} = k_{\text{block}} \tau_{\text{MC}} \) | |
Autocorrelation functions

The susceptibility \( \chi = \left( \frac{\partial m}{\partial B} \right)_T \) is related to the fluctuations in the magnetization

\[
\chi = \beta N \langle (m - \langle m \rangle)^2 \rangle = \beta N \left( \langle m^2 \rangle - \langle m \rangle^2 \right)
\]

The quasi-ergodic hypothesis relates ensemble averages to time averages. We treat time as continuous for the moment and convert to discrete time steps later

\[
\Rightarrow \quad \langle m \rangle = \frac{1}{\tau_{\text{obs}}} \int_0^{\tau_{\text{obs}}} dt' m(t') \quad \text{and} \quad \langle m^2 \rangle = \frac{1}{\tau_{\text{obs}}} \int_0^{\tau_{\text{obs}}} dt' m^2(t')
\]

The two-time correlation function of the magnetization is defined as

\[
\hat{\chi}(t) = \frac{1}{\tau_{\text{obs}} - t} \int_0^{\tau_{\text{obs}} - t} dt' \left( m(t') - \langle m \rangle \right) \left( m(t + t') - \langle m \rangle \right)
\]

\[
\Rightarrow \quad \hat{\chi}(t) = \frac{1}{\tau_{\text{obs}} - t} \int_0^{\tau_{\text{obs}} - t} dt' \left( m(t')m(t + t') - \langle m \rangle^2 \right)
\]

note: \( \hat{\chi}(0) = \frac{1}{\tau_{\text{obs}}} \int_0^{\tau_{\text{obs}}} dt' \left( m^2(t') - \langle m \rangle^2 \right) = \frac{1}{\beta N} \chi \)
For discrete time, measured in MC steps

\[
\langle m \rangle = \frac{1}{k_{\text{obs}}} \sum_{k=1}^{k_{\text{obs}}} m_k \quad \text{where}
\]

\[
m_k = \frac{1}{N} \sum_{y=1}^{L} \sum_{x=1}^{L} \text{spin}(x, y) = \text{magnetization per spin after } k \text{ MC steps}
\]

Note: For small systems, there can be a spontaneous change from mostly spin up to mostly spin down (or vice versa). This is a “finite-size effect” that does not reflect the physics of the magnetic systems that we want to simulate. Hence, focus on absolute values

\[
\Rightarrow \hat{\chi}(t_j) = \frac{1}{k_{\text{obs}} - j} \sum_{l=1}^{k_{\text{obs}} - j} |m_l| |m_{l+j}| - \left( \frac{1}{k_{\text{obs}} - j} \sum_{l=1}^{k_{\text{obs}} - j} |m_l| \right) \left( \frac{1}{k_{\text{obs}} - j} \sum_{l=1}^{k_{\text{obs}} - j} |m_{l+j}| \right)
\]

Similarly, we can calculate a two-time correlation function for the energy, which is related to the heat capacity.

\[
\hat{c}(t_j) = \frac{1}{k_{\text{obs}} - j} \sum_{l=1}^{k_{\text{obs}} - j} \frac{E_l}{N} \frac{E_{l+j}}{N} - \left( \frac{1}{k_{\text{obs}} - j} \sum_{l=1}^{k_{\text{obs}} - j} \frac{E_l}{N} \right) \left( \frac{1}{k_{\text{obs}} - j} \sum_{l=1}^{k_{\text{obs}} - j} \frac{E_{l+j}}{N} \right)
\]

\[
\hat{c}(0) = \frac{T_{\text{red}}^2}{N} c
\]
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 $\tau_{\text{obs}}$ has to be much larger than the longest correlation time $\tau_{\text{corr}}$.

\[ T_{\text{red}} = 3, \; B = 0, \; J = 1 \]

Normalized autocorrelation functions

- $\chi(t)/\chi(0)$
- $c(t)/c(0)$
Evaluation of simulation results and error estimates from block averaging

For each MC step after equilibration ($k > k_{\text{equ}}$) calculate the quantities of interest, in our case the magnetization per site and the energy per site. We describe the block-averaging procedure here for the magnetization per site, it works the same for the energy. For the susceptibility and the heat capacity, the fluctuations of the magnetization and the energy, respectively, are calculated for each block. The results for the blocks are then averaged and the error is determined.

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

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

$$k_{\text{obs}} = n_{\text{block}} k_{\text{block}} \quad \text{for example} \quad k_{\text{obs}} = 1000, \quad n_{\text{block}} = 10 \implies k_{\text{block}} = 100$$
For each block \( b \in \{1, \ldots, n_{\text{block}}\} \) 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 statistical errors 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 on systematic errors.
Systematic errors:

The most common systematic error is insufficient equilibration. 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 random number generators 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.

Finite-size effects 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 crystallize 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.

Non-ergodic algorithms 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.

Configurational bias 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.