Chapter 15

Thermodynamical Properties of Spin Models

We now turn to one of the most important applications of Monte Carlo methods to physics, namely the estimation of thermodynamic properties of physical systems. We will specialize our discussion to the case of idealized magnets, that can be described in terms of a set of N discrete spin variables $\mathbf{s} \equiv s_1, \ldots, s_N$ that can be in one of two states: up $(+1)$ or down (-1). In this case, the task is to compute thermal expected values of some observable that, generally speaking, also depend on the spin variables, say $O(\mathbf{s})$, over the Boltzmann probability distribution:

$$\langle O \rangle = \frac{\sum_{\mathbf{s}} e^{-\beta E(\mathbf{s})} O(\mathbf{s})}{\sum_{\mathbf{s}'} e^{-\beta E(\mathbf{s}')}} \,, \tag{15.0.1}$$

this expression can be immediately recast as a Monte Carlo–friendly object, since it is just an expected value

$$\langle O \rangle = \mathbb{E}_P[O] \,, \tag{15.0.2}$$

over the probability density

$$P(\mathbf{s}) = \frac{e^{-\beta E(\mathbf{s})}}{\sum_{\mathbf{s}'} e^{-\beta E(\mathbf{s}')}} \,. \tag{15.0.3}$$

Notice that the summation over $\mathbf{s}$ means summing over all possible values of the N spins, which entails a summation over 2^N possible configurations. This is an astronomically large number of possible states, and performing this summation by brute force, by enumerating all possible states, is unfeasible already starting at $N \simeq 30$. On the other hand, as we have seen from the previous discussion Monte Carlo methods can bypass the curse of dimensionality.

15.1 The 2D Ising Model Hamiltonian

The Ising model is a simplified mathematical model of ferromagnetism in statistical mechanics. In two dimensions, it consists of a square lattice of $N = L \times L$ spins $s_1 \ldots s_N$.

The Hamiltonian (energy function) for the 2D Ising model is given by:

$$E(\mathbf{s}) = -J \sum_{\langle l,m \rangle} s_l s_m - h \sum_{l=1}^{N} s_l \tag{15.1.1}$$

where:

- $s_l = \pm 1$ is the spin at site l
- J is the coupling constant (J > 0 for ferromagnetic interactions)
- h is the external magnetic field
- $\langle l, m \rangle$ denotes summation over nearest neighbors

15.2 Critical Behavior and Correlation Length

The 2D Ising model exhibits a second-order phase transition at the critical temperature T_c. Near this temperature, several quantities show power-law behavior:

1. Magnetization (for $T < T_c$):
$$m \sim (T_c - T)^{\beta}, \quad \beta = \frac{1}{8} \tag{15.2.1}$$

2. Susceptibility:
$$\chi \sim |T - T_c|^{-\gamma}, \quad \gamma = \frac{7}{4} \tag{15.2.2}$$

3. Specific heat:
$$c \sim |T - T_c|^{-\alpha}, \quad \alpha = 0 \text{ (logarithmic divergence)} \tag{15.2.3}$$

4. Correlation length:
$$\xi \sim |T - T_c|^{-\nu}, \quad \nu = 1 \tag{15.2.4}$$

The correlation length ξ is defined through the spin-spin correlation function:

$$G(r) = \langle s_l s_m \rangle - \langle s_l \rangle \langle s_m \rangle \sim e^{-r/\xi} \tag{15.2.5}$$

where r is the distance between spins l and m.

15.3 Critical Temperature

In the absence of an external field, the critical temperature for the 2D Ising model is known exactly:

$$T_c = \frac{2J}{k_B \ln(1+\sqrt{2})} \approx 2.269185 J/k_B \tag{15.3.1}$$

This result was first derived by Lars Onsager in 1944.

15.4 The Metropolis Algorithm for the Ising Model

The Metropolis algorithm is used to sample configurations from the Boltzmann distribution:

$$P(\mathbf{s}) \propto e^{-\beta E(\mathbf{s})} \tag{15.4.1}$$

where $\beta = 1/(k_B T)$, k_B is the Boltzmann constant, and T is the temperature.

Using the notation developed in the previous Chapter, we identify $i \equiv \mathbf{s}(i)$, with $i \in [1, 2^N]$ labelling all possible spin configurations, e.g. with the following ordering

$$\begin{aligned} \mathbf{s}(1) &= (1, 1, \ldots 1) \\ \mathbf{s}(2) &= (-1, 1, \ldots 1) \\ \ldots &= \ldots \\ \mathbf{s}(2^N) &= (-1, -1, \cdots -1) \end{aligned}$$

$P_i \equiv P(\mathbf{s}(i))$ and the transition probability $Q_{ij} \equiv Q(\mathbf{s}(i)|\mathbf{s}(j))$. In this case, we will consider a simple transition probability that involves picking a spin at random and flipping its sign. This transition probability is clearly symmetric ($Q_{ij} = Q_{ji}$) and simplifies the acceptance probability $\alpha_{ij} \equiv A(\mathbf{s}(j)|\mathbf{s}(i))$, which reads

$$A(\mathbf{s}(j)|\mathbf{s}(i)) = \min\left[1, \frac{P(\mathbf{s}(j))}{P(\mathbf{s}(i))}\right] \tag{15.4.2}$$

$$= \min\left[1, \exp -\beta\left[\mathrm{E}(\mathbf{s}(\mathrm{j})) - \mathrm{E}(\mathbf{s}(\mathrm{i}))\right]\right] \tag{15.4.3}$$

$$= \min\left[1, \exp -\beta\left[\Delta \mathrm{E}\right]\right]. \tag{15.4.4}$$

Importantly, this acceptance probability can be compute without the knowledge of the normalization of the Boltzmann probability (which is hard to compute), and involves only computing an energy difference between two spin configurations. Moreover, the energy change ΔE for flipping a spin can be calculated efficiently, making use of the locality of the interactions. Say for example we pick a random spin r to be flipped, then after the flip we have

$$\mathbf{s}(j) = (s_1 \cdots - s_r \ldots s_N), \tag{15.4.5}$$

thus

$$E(\mathbf{s}(j)) = -J \sum_{\langle l,m\rangle \neq r} s_l s_m - h \sum_{l \neq r}^{N} s_l + J \sum_{\langle r,m\rangle} s_r s_m + h s_r \tag{15.4.6}$$

$$= -J \sum_{\langle l,m\rangle} s_l s_m - h \sum_{l}^{N} s_l + 2J \sum_{\langle r,m\rangle} s_r s_m + 2h s_r \tag{15.4.7}$$

$$= E(\mathbf{s}(i)) + 2s_r \left[J \sum_{\langle r,m\rangle} s_m + h \right], \tag{15.4.8}$$

thus the energy difference is simply

$$\Delta E = 2s_r \left(J \sum_{\langle r,m\rangle} s_m + h \right) \tag{15.4.9}$$

where $\langle r, m\rangle$ denotes a summation over the 4 nearest neighbors of site r.

Algorithmically we have the following steps:

Algorithm 15.1 Ising Model Monte Carlo Simulation

```
Initialize spin configuration {s_l} (e.g., randomly or all spins up)
for many Monte Carlo steps do
    Choose a spin s_l randomly
    Calculate the energy change ΔE if this spin were flipped
    if ΔE ≤ 0 then
        Accept the flip
    else
        Accept the flip with probability e^{-βΔE}
    end if
    if flip is accepted then
        Flip the spin s_i
    end if
end for
After equilibration, calculate observables of interest
```

15.5 Observables

Key observables in the Ising model simulation include:

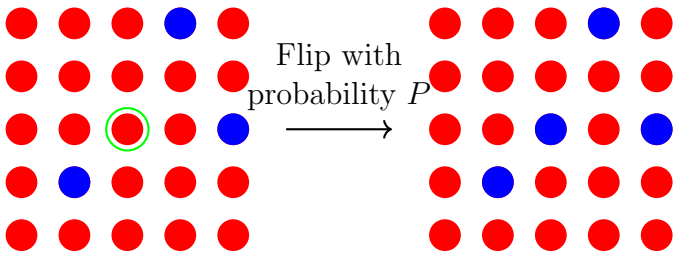

Figure 15.4.1: Illustration of the Metropolis algorithm with local spin flip. Left: Initial state with chosen spin (green circle). Red circles represent up spins, blue circles represent down spins. Right: Possible state after attempting to flip the chosen spin.

1. Magnetization per spin:
$$m = \frac{1}{N}\sum_{l} s_l \tag{15.5.1}$$

2. Energy per spin:
$$e = \frac{1}{N}E \tag{15.5.2}$$

3. Specific heat:
$$c = \frac{\beta^2}{N}(\langle E^2\rangle - \langle E\rangle^2) \tag{15.5.3}$$

4. Magnetic susceptibility:
$$\chi = \beta N(\langle m^2\rangle - \langle m\rangle^2) \tag{15.5.4}$$

where $\langle\cdot\rangle$ denotes thermal average, and N is the total number of spins.

15.6 Implementation Considerations

1. Periodic boundary conditions are typically used to minimize finite-size effects.

2. The system should be equilibrated before collecting data for observables.

3. Multiple independent runs or long runs with appropriate error analysis are necessary for accurate results.

4. The critical temperature for the 2D Ising model is known exactly:
$$T_c = \frac{2J}{k_B \ln(1+\sqrt{2})} \approx 2.269 J/k_B \tag{15.6.1}$$

15.7 Finite-Size Scaling

For finite systems, observables near the critical point follow scaling relations. For a system of linear size L:

1. Magnetization: $m \sim L^{-\beta/\nu}\tilde{m}(tL^{1/\nu})$
2. Susceptibility: $\chi \sim L^{\gamma/\nu}\tilde{\chi}(tL^{1/\nu})$
3. Specific heat: $c \sim L^{\alpha/\nu}\tilde{c}(tL^{1/\nu})$

where $t = (T - T_c)/T_c$ is the reduced temperature, and $\tilde{m}$, $\tilde{\chi}$, and $\tilde{c}$ are scaling functions. These scaling relations allow for the precise extraction of critical exponents from simulations of finite systems.

Chapter 16

Advanced Sampling Methods

16.1 Cluster Algorithms for the Ising Model

While the Metropolis algorithm is effective for simulating the Ising model, it can suffer from critical slowing down near the phase transition. Cluster algorithms, such as the Wolff algorithm, can significantly reduce this problem by updating large clusters of spins simultaneously. The Wolff algorithm is summarized below.

Algorithm 16.1 Wolff Algorithm for Ising Model Simulation

```
Choose a random starting spin
Initialize an empty cluster
Add the starting spin to the cluster
while there are unchecked neighbors of the cluster do
    for each unchecked neighboring spin of the same sign do
        Calculate p = 1 − e^{−2βJ}
        Generate a uniform random number r ∈ [0, 1]
        if r < p then
            Add the neighboring spin to the cluster
        end if
    end for
end while
Flip all spins in the cluster
```

16.1.1 Detailed Balance

To understand the algorithm, let's consider two configurations a and b that differ by a flipped cluster, as shown in Figure 16.1.1.

Let's examine the transition probabilities:

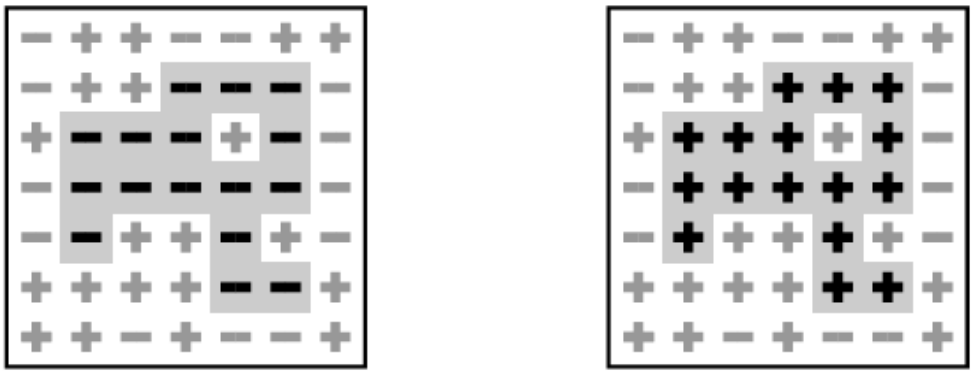

Figure 16.1.1: The Wolff cluster algorithm for the Ising model adds, with probability p, a link connecting a site outside the cluster to a site already in the cluster (thereby adding the site). In the configuration a, construction of the cluster (as shown) stopped with 9 links "$--$", corresponding to an a priori probability $Q(b|a) = A_{\text{interior}} \times (1-p)^9$. The return move stops with probability $Q(a|b) = Q_{\text{interior}} \times (1-p)^{19}$, as there are 19 links "$++$" across the boundary in configuration b. Figure reproduced from W. Krauth, arXiv:0311623 (2003).

$$Q(b|a) = Q_{\text{interior}} \times (1-p)^{n_{\text{same}}} \tag{16.1.1}$$
$$Q(a|b) = Q_{\text{interior}} \times (1-p)^{n_{\text{diff}}} \tag{16.1.2}$$

where:

- Q_{interior} is the probability of growing the interior of the cluster (same for both directions)
- n_{same} is the number of bonds between same-sign spins across the cluster boundary in configuration a
- n_{diff} is the number of bonds between different-sign spins across the cluster boundary in configuration b

The energy difference between configurations a and b is crucial for understanding how the algorithm satisfies detailed balance. This difference arises solely from the bonds crossing the cluster boundary, as all internal bonds of the cluster contribute the same energy before and after the flip.

Let's define:

- n_{same}: number of bonds between same-sign spins across the cluster boundary in configuration a
- n_{diff}: number of bonds between different-sign spins across the cluster boundary in configuration a

For each same-sign bond in configuration a:

- It contributes $-J$ to the energy in configuration a
- After the cluster flip, it becomes a different-sign bond in b, contributing $+J$
- The energy difference for each such bond is $+2J$

Conversely, for each different-sign bond in configuration a:

- It contributes $+J$ to the energy in configuration a
- After the cluster flip, it becomes a same-sign bond in b, contributing $-J$
- The energy difference for each such bond is $-2J$

Therefore, the total energy difference is:

$$E(b) - E(a) = (2J \times n_{\text{same}}) + (-2J \times n_{\text{diff}}) = 2J(n_{\text{same}} - n_{\text{diff}}) \tag{16.1.3}$$

The acceptance probability for the cluster flip is:

$$A(b|a) = \min\left\{1, \frac{P(b)Q(a|b)}{P(a)Q(b|a)}\right\} \tag{16.1.4}$$

$$= \min\left\{1, e^{-2\beta J(n_{\text{same}} - n_{\text{diff}})} \frac{(1-p)^{n_{\text{diff}}}}{(1-p)^{n_{\text{same}}}}\right\} \tag{16.1.5}$$

$$= \min\left\{1, \left[\frac{e^{-2\beta J}}{1-p}\right]^{n_{\text{same}}} \left[\frac{1-p}{e^{-2\beta J}}\right]^{n_{\text{diff}}}\right\} \tag{16.1.6}$$

We clearly see that if we choose $p = 1 - e^{-2\beta J}$, this acceptance always evaluates to 1, meaning every proposed cluster flip is accepted.

This algorithm is particularly efficient because:

- It flips large clusters of spins in a single step, allowing for rapid changes in the system's state.
- The cluster size naturally adapts to the system's correlation length, becoming especially effective near the critical point.
- Every proposed move is accepted, eliminating wasted computation on rejected moves.

16.2 Parallel Tempering

Parallel tempering, also known as replica exchange, is an advanced Monte Carlo method designed to improve sampling efficiency, especially for systems with rugged energy landscapes. This technique is particularly useful for simulations of complex systems that may get trapped in local energy minima.

The key idea of parallel tempering is to simulate multiple replicas of the system at different temperatures simultaneously. Periodically, the method attempts to exchange configurations between neighboring temperatures. This allows configurations to move between different temperatures, helping to overcome energy barriers and explore the phase space more efficiently.

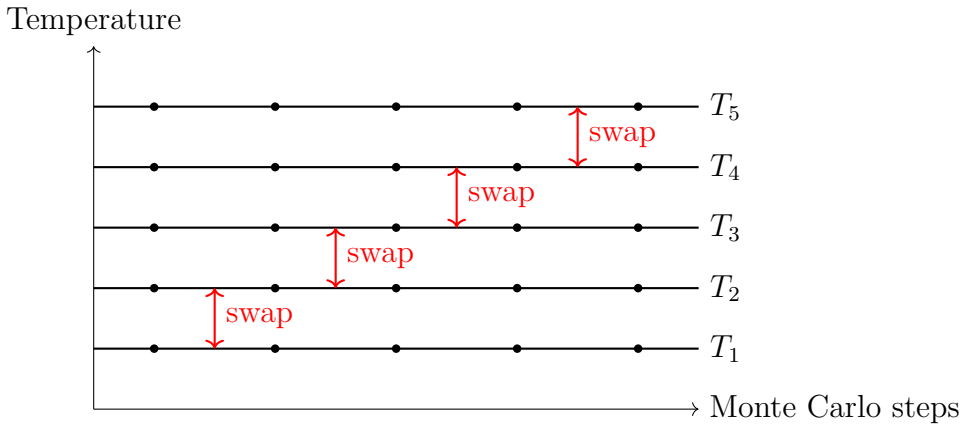

Figure 16.2.1: Schematic representation of parallel tempering. Each horizontal line represents a replica at a different temperature. Black dots represent Monte Carlo steps, and red arrows indicate attempted configuration swaps between adjacent temperatures.

The parallel tempering algorithm can be described as follows:

Algorithm 16.2 Parallel Tempering

```
Initialize M replicas at temperatures T_1 < T_2 < ··· < T_M
for each Monte Carlo step do
    for each replica k do
        Perform a Monte Carlo move (e.g., Metropolis) at temperature T_k
    end for
    for k = 1 to M − 1 with probability p_swap do
        Calculate Δ = (β_k − β_{k+1})(E_k − E_{k+1})
        where β_k = 1/(k_B T_k) and E_k is the energy of replica k
        Accept swap with probability min(1, e^Δ)
        if swap accepted then
            Exchange configurations of replicas k and k + 1
        end if
    end for
    Collect data for analysis
end for
```

16.2.1 Proof of Acceptance Probability

The acceptance probability for swapping configurations between two adjacent temperatures is derived from the Metropolis-Hastings acceptance rule, by considering an enlarged configuration space comprising all the replicas at the different temperatures. Specifically, call the state of the system

$$\mathbf{X} \;=\; (X_1 \dots X_M),$$

and the energies of each replica:

$$E(X_l) \;\equiv\; E_l,$$

then the joint probability of all replicas having a certain energy is just the product of the individual probabilities (since there are no interactions among replicas):

$$\begin{aligned} P(\mathbf{X}) &= \Pi_{i=1}^{M} P(X_i) \\ &= \Pi_{i=1}^{M} \frac{e^{-\beta_i E_i}}{Z_i} \\ &\propto e^{-\sum_i \beta_i E_i}. \end{aligned}$$

When we consider swapping two replicas, say k and l with inverse temperatures β_k and β_l, and energies E_k and E_l, respectively, we have that the new configuration is just

$$\mathbf{X}_{\text{swap}} \;=\; (X_1 \dots X_l \dots X_k \dots X_M),$$

where without loss of generality we have taken $k < l$. Then,

$$\frac{P(\mathbf{X}_{\text{swap}})}{P(\mathbf{X})} = \frac{e^{-\beta_l E_k - \beta_k E_l}}{e^{-\beta_l E_l - \beta_k E_k}} \tag{16.2.1}$$

$$= e^{-\beta_l(E_k - E_l) - \beta_k(E_l - E_k)}. \tag{16.2.2}$$

$$= e^{(\beta_l - \beta_k)(E_l - E_k)}. \tag{16.2.3}$$

Since the swap move is clearly symmetric, the Metropolis-Hastings acceptance is just for this ratio is:

$$A(\mathbf{X}_{\text{swap}}|\mathbf{X}) = \min\left[1, e^{(\beta_l - \beta_k)(E_l - E_k)}\right]. \tag{16.2.4}$$

This proves the acceptance probability used in the algorithm.

16.2.2 Key Considerations

- Temperature selection: The range and spacing of temperatures are crucial for efficient sampling. Temperatures should be close enough to allow frequent exchanges but far enough apart to span the desired range.
- Swap frequency: The frequency of attempted swaps (p_{swap}) should be balanced to allow sufficient sampling at each temperature while promoting exploration across temperatures.
- Number of replicas: More replicas can improve sampling but increase computational cost. The optimal number depends on the system and available resources.

Parallel tempering is particularly effective for systems with multiple metastable states or those exhibiting phase transitions. It helps in overcoming energy barriers that might trap standard Monte Carlo methods, leading to more efficient exploration of the configuration space and improved convergence of thermodynamic averages.