

Statistical Physics: Concepts and Applications

Werner Krauth
Laboratoire de Physique Statistique
Ecole Normale Supérieure
werner.krauth@ens.fr

Maurizio Fagotti
Département de Physique
Ecole Normale Supérieure
mfagotti@phys.ens.fr

October 7, 2015

Chapter 6

Presence / Absence of
transition in 1d systems. 1D
Ising model. (TD2)

Chapter 7

2D Ising model - Solution of Kac and Ward (CM3) Preliminary version 7 October 2015

In this lecture, we approach the statistical mechanics of the Ising model, which has inspired generations of physicists. This archetypal physical system undergoes an order–disorder phase transition and many properties can be computed exactly, in two dimensions. properties with more complicated models which cannot be analyzed so well.

The first focus is on enumeration, which applies to the Ising model because of its finite number of configurations, even though this number grows exponentially with the lattice size. We shall enumerate the spin configurations, and also the loop configurations of the Ising model’s high-temperature expansion, which can be summed for very large and even infinite lattices, leading to Onsager’s analytic solution in two dimensions.

Theoretical approaches to the Ising model have met with outstanding success. However, it suffices to modify a few parameters in the model, for example to let the sign of the interaction be sometimes positive and sometimes negative, to cause all combined approaches to get into trouble. Remarkably, the above-mentioned enumeration of loop configurations still works. Onsager’s analytic solution of the Ising model thus turns into a powerful algorithm for solving two-dimensional spin glasses.

The Ising model describes spins $\sigma_k \pm 1$, $k = 1, \dots, N$, on a lattice, for example the two-dimensional square lattice shown in Fig. 7.1. In the simplest case, the ferromagnetic Ising model, neighboring spins prefer to align. This means that pairs $\{+, +\}$ and $\{-, -\}$ of neighboring spins direction have a lower energy than

antiparallel spins (pairs $\{+, -\}$ and $\{-, +\}$), as expressed by the energy

$$E = -J \sum_{\langle k, l \rangle} \sigma_k \sigma_l. \quad (7.1)$$

The sum is over all pairs of neighbors. The parameter J is positive, and we shall take it equal to one. In a two-dimensional square lattice, the sites k and l then differ by either a lattice spacing in x or a lattice spacing in y . In a sum over pairs of neighbors, as in Eq. 7.1, we consider each pair only once, that is, we pick either $\langle k, l \rangle$ or $\langle l, k \rangle$. Alg. energy-ising.py implements Eq. 7.1 with the help of a neighbor scheme that we have encountered already earlier. The sum n runs over half the neighbors, so that each pair $\langle l, k \rangle$ is indeed counted only once. We also note that the two-dimensional lattice we consider may either have periodic boundary conditions or be planar.

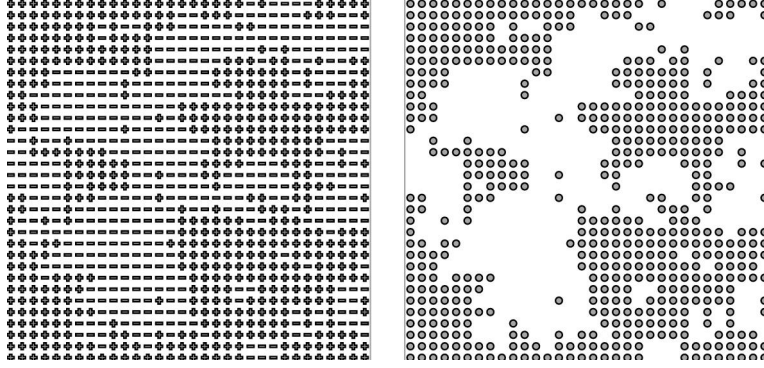


Figure 7.1: Ising model, in its use for magnets (left) and for lattice gases (right)

The Ising model's prime use is for magnets. Fig. 7.1, however, illustrates that it can also serve to describe particles on a lattice. Now, a variable $\tilde{\sigma}_k = 1, 0$ signals the presence or absence of a particle on site k . Let us suppose that particles prefer to aggregate: two particles next to each other have a lower energy than two isolated particles. The simplest configurational energy is

$$E = -4\tilde{J} \sum_{\langle k, l \rangle} \tilde{\sigma}_k \tilde{\sigma}_l.$$

However, the transformation $\tilde{\sigma}_k = \frac{1}{2}(\sigma_k + 1)$ brings us back to the original Ising model.

The main difference between the Ising model considered as a magnet and as a lattice gas is in the space of configurations: for a magnet, the spins can be up or down, more or less independently of the others, so that all of the 2^N configurations $\{\sigma_1, \dots, \sigma_N\} = \{\pm 1, \dots, \pm 1\}$ contribute to the partition function. For the lattice gas, the number of particles, equivalent to the proportions of up and

down spins, must be kept constant, and the partition function is made up of all configurations with a fixed $M = \sum_k \sigma_k$. For large N , the two versions of the Ising model become more or less equivalent: it is sufficient to include a constant external magnetic field, in other words, a chemical potential.

7.0.1 Listing spin configurations

Let us quickly enumerate all the spin configurations of the Ising model; in fact, we list them one after another. Most simply, each configuration $i = 1, \dots, 2^N$ of N Ising spins is related to the binary representation of the number $i - 1$. For the 2×2 Ising model, we can enumerate the numbers from 0 through 15, obtain the binary representation, then change each zero into a down spin and each one into an up spin. is 1010, which yields a spin More efficiently, we may enumerate all 2^N spin configurations through a sequence of 2^N spin-flips, one at a time. (Equivalently, one may enumerate all numbers $\{0, \dots, 2^N - 1\}$ by changing a single digit at a time during the enumeration.) Algorithms that perform such enumerations are called Gray codes, and an application of a Gray code for four spins is shown in Fig. 7.2. We may couple the Gray code enumeration to an update of the energy. This is implemented in Alg. `enumerate-ising.py`). Of course, the Gray code still has exponential running time and is practical only for testing purposes.

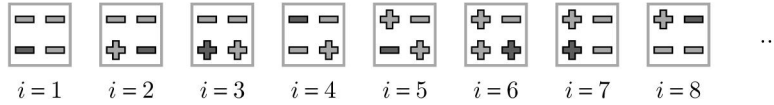


Figure 7.2: List of Ising-model configurations on a 2×2 square lattice, generated by the Gray code (only the dark spins flip).

Note that Alg. `enumerate-ising.py` does not directly compute the partition function at inverse temperature β , but rather the temperature-independent number of configurations with energy E , in other words, the density of states $\mathcal{N}E$. From the density of states, we get all the thermodynamics at any temperature, without any effort.

7.1 Listing loop configurations

The word “enumeration” has two meanings: it refers to listing items (configurations), but it also applies to simply counting them. The difference between the two is of more than semantic interest: in the list generated by Alg. `enumerate-ising.py`, we were able to pick out any information we wanted, for example the number of configurations of energy E and magnetization M , that is, the density of states $\mathcal{N}(E, M)$. In this subsection we discuss an alternative enumeration for the two-dimensional Ising model. It does not list the spin configurations,

but rather all the loop configurations which appear in the high-temperature expansion of the Ising model. This program will then turn, in Section ??, into an enumeration of the second kind (the counting), as pioneered by Kac and Ward[8]. It counts configurations and obtains $Z(\beta)$ for a two-dimensional Ising system of any size (Kaufman, 1949)[9], and even for the infinite system (Onsager, 1944)[10]. However, it then counts without listing. For example, it finds the number $\mathcal{N}(E)$ of configurations with energy E but does not tell us how many of them have a magnetization M .

Van der Waerden, in 1941 [11], noticed that the Ising-model partition function,

$$\begin{aligned} Z &= \sum_{\sigma} \exp\left(J\beta \sum_{\langle k,l \rangle} \sigma_k \sigma_l\right) \\ &= \sum_{\sigma} \prod_{\langle k,l \rangle} e^{J\beta \sigma_k \sigma_l}, \end{aligned} \quad (7.2)$$

allows each term $e^{J\beta \sigma_k \sigma_l}$ to be expanded and rearranged into just two terms, one independent of the spins and the other proportional to $\sigma_k \sigma_l$:

$$\begin{aligned} e^{\beta \sigma_k \sigma_l} &= 1 + \beta \sigma_k \sigma_l + \frac{\beta^2}{2!} \underbrace{(\sigma_k \sigma_l)^2}_{=1} + \frac{\beta^3}{3!} \underbrace{(\sigma_k \sigma_l)^3}_{=\sigma_k \sigma_l} + \cdots - \cdots \\ &= \underbrace{\left(1 + \frac{\beta^2}{2!} + \frac{\beta^4}{4!} + \cdots\right)}_{\cosh \beta} + \sigma_k \sigma_l \underbrace{\left(\beta + \frac{\beta^3}{3!} + \frac{\beta^5}{5!} + \cdots\right)}_{\sinh \beta} \\ &= (\cosh \beta) (1 + \sigma_k \sigma_l \tanh \beta) \end{aligned}$$

Inserted into Eq. 7.2, with $J = +1$, this yields

$$Z(\beta) = \sum_s \prod_{\langle k,l \rangle} ((\cosh \beta) (1 + \sigma_k \sigma_l \tanh \beta)). \quad (7.3)$$

For concreteness, we continue with a 4×4 square lattice without periodic boundary conditions (with $J = 1$). This lattice has 24 edges and 16 sites, so that, by virtue of Eq. 7.3, its partition function $Z_{4 \times 4}(\beta)$ is the product of 24 parentheses, one for each edge:

$$\begin{aligned} Z_{4 \times 4}(\beta) &= \sum_{\{\sigma_1, \dots, \sigma_{16}\}} \cosh^{24} \beta \overbrace{(1 + \sigma_1 \sigma_2 \tanh \beta)}^{\text{edge 1}} \overbrace{(1 + \sigma_1 \sigma_5 \tanh \beta)}^{\text{edge 2}} \\ &\quad \times \dots (1 + \sigma_{14} \sigma_{15} \tanh \beta) \underbrace{(1 + \sigma_{15} \sigma_{16} \tanh \beta)}_{\text{edge 24}}. \end{aligned} \quad (7.4)$$

We multiply out this product: for each edge (parenthesis) k , we have a choice between a “one” and a “tanh” term. This is much like the option of a spin-up

or a spin-down in the original Ising-model enumeration, and can likewise be expressed through a binary variable n_k :

$$n_k = \begin{cases} 0 & (\equiv \text{edge } k \text{ in Eq. 7.4 contributes 1}) \\ 1 & (\equiv \text{edge } k \text{ contributes } (\sigma_{s_k} \sigma_{s'_k} \tanh(\beta))) \end{cases},$$

where s_k and s'_k indicate the sites at the two ends of edge k . Edge $k = 1$ has $\{s_1, s'_1\} = \{1, 2\}$, and edge $k = 24$ has, from Eq. 7.4, $\{s_{24}, s'_{24}\} = \{15, 16\}$. Each factored term can be identified by variables

$$\{n_1, \dots, n_{24}\} = \{\{0, 1\}, \dots, \{0, 1\}\}.$$

For $\{n_1, \dots, n_{24}\} = \{0, \dots, 0\}$, each parenthesis picks a “one”. Summed over all spin configurations, this gives 2^{16} . Most choices of $\{n_1, \dots, n_{24}\}$ average to zero when summed over spin configurations because the same term is generated with $\sigma_k = +1$ and $\sigma_k = -1$. Only choices leading to spin products $\sigma_s^0, \sigma_s^2, \sigma_s^4$ at each lattice site s remain finite after summing over all spin configurations. The edges of these terms form loop configurations, such as those shown for the 4×4 lattice in Fig. 7.3.

The list of all loop configurations may be generated by Alg. `edge-ising.py`, a recycled version of the Gray code for 24 digits, coupled to an incremental calculation of the number of spins on each site. The $\{o_1, \dots, o_{16}\}$ count the number of times the sites $\{1, \dots, 16\}$ are present. The numbers in this vector must all be even for a loop configuration, and for a nonzero contribution to the sum in Eq. 7.4.

Table 7.1: Numbers of loop configurations in Fig. 7.3 with given numbers of edges (the figure contains one configuration with 0 edges, 9 with 4 edges, etc). (From Alg. `edge-ising.py`).

# Edges	# Confs
0	1
4	9
6	12
8	50
10	92
12	158
14	116
16	69
18	4
20	1

For the thermodynamics of the 4×4 Ising model, we only need to keep track of the number of edges in each configuration, not the configurations themselves.

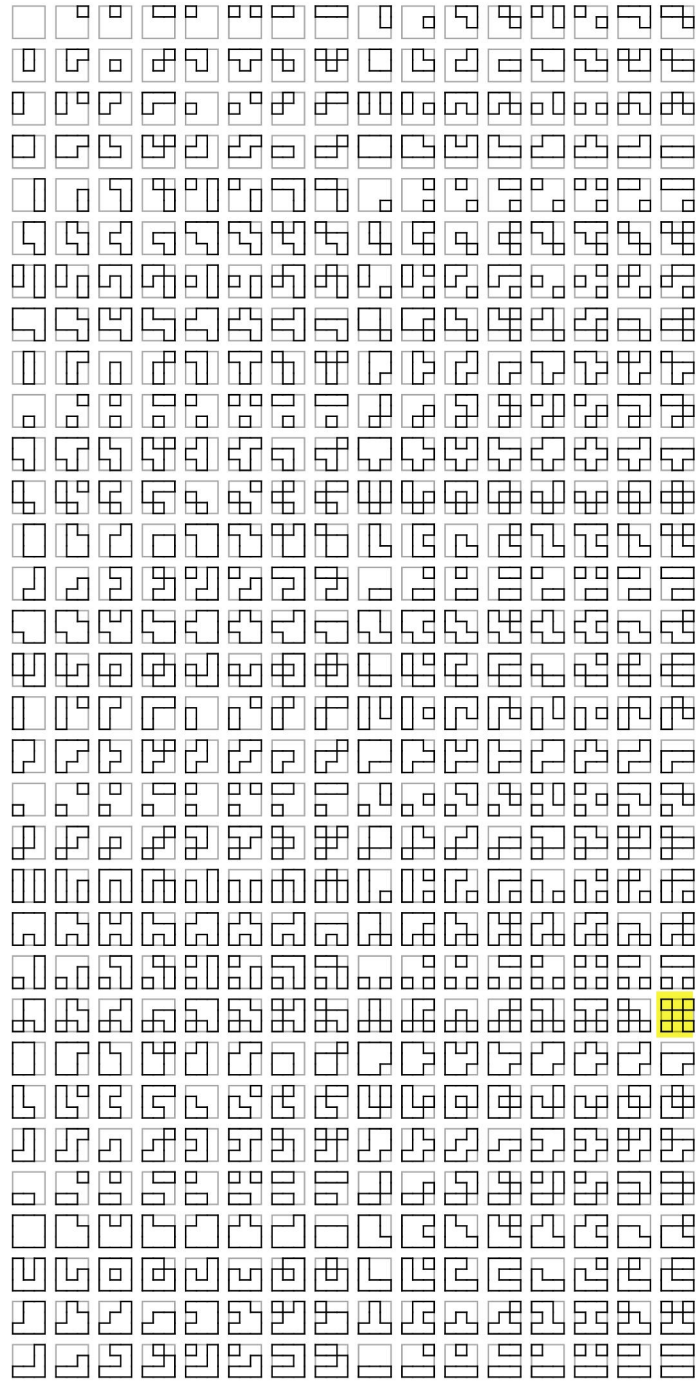


Figure 7.3: The list of all 512 loop configurations for the 4×4 Ising model without periodic boundary conditions. There is one “golden” configuration.

Tab. 7.1, which shows the number of loop configurations for any given number of edges, thus yields the exact partition function for the 4×4 lattice without periodic boundary conditions:

$$Z_{4 \times 4}(\beta) = (2^{16} \cosh^{24}(\beta))(1 + 9 \tanh^4 \beta + 12 \tanh^6 \beta + \dots + 4 \tanh^{18} \beta + 1 \tanh^{20} \beta). \quad (7.5)$$

Partition functions obtained from this expression are easily checked against the Gray-code enumeration that we had before.

7.2 Counting (not listing) loops in two dimensions

Following Kac and Ward[8], we now construct a matrix whose determinant counts the number of loop configurations in Fig. 7.3. This is possible because the determinant of a matrix $U = (u_{kl})$ is defined by a sum of permutations P (with signs and weights). Each permutation can be written as a collection of cycles, a “cycle configuration”. Our task will consist in choosing the elements u_{kl} of the matrix U in such a way that the signs and weights of each cycle configurations correspond to the loop configurations in the two-dimensional Ising model. We shall finally arrive at a computer program which implements the correspondence, and effectively solves the enumeration problem for large two-dimensional lattices. For simplicity, we restrict ourselves to square lattices without periodic boundary conditions, and consider the definition of the determinant of a matrix U ,

$$\det U = \sum_{\text{permutations}} (\text{sign } P) u_{1P_1} u_{2P_2} \dots u_{NP_N}.$$

We now represent P in terms of cycles. The sign of a permutation P of N elements with n cycles is $\text{sign } P = (-1)^{N+n}$ (an example may be found in the SMAC 1.2.2). In the following, we shall consider only matrices with even N , for which $\text{sign } P = (-1)^{\# \text{ of cycles}}$. The determinant is thus

$$\begin{aligned} \det U &= \sum_{\substack{\text{cycle} \\ \text{configs}}} (-1)^{\# \text{ of cycles}} \underbrace{u_{P_1 P_2} u_{P_2 P_3} \dots u_{P_M P_1}}_{\text{weight of first cycle}} \underbrace{u_{P'_1 P'_2} \dots}_{\text{other cycles}} \\ &= \sum_{\substack{\text{cycle} \\ \text{configs}}} \left(\left\{ \begin{array}{c} (-1) \cdot \text{weight of} \\ \text{first cycle} \end{array} \right\} \right) \times \dots \times \left(\left\{ \begin{array}{c} (-1) \cdot \text{weight of} \\ \text{last cycle} \end{array} \right\} \right). \end{aligned}$$

It follows from this representation of a determinant in terms of cycle configurations that we should choose the matrix elements u_{kl} such that each cycle corresponding to a loop on the lattice (for example $(P_1, \dots, P_M)$) gets a negative sign (this means that the sign of $u_{P_1 P_2} u_{P_2 P_3} \dots u_{P_M P_1}$ should be negative). All cycles not corresponding to loops should get zero weight.

We must also address the problem that cycles in the representation of the determinant are directed. The cycle $(P_1, P_2, \dots, P_{M-1}, P_M)$ is different from the cycle $(P_M, P_{M-1}, \dots, P_2, P_1)$, whereas the loop configurations in Fig. 7.3 have no sense of direction.

7.2.1 2x2 lattice, naive 4×4 matrix

For concreteness, we start with a 2×2 lattice without periodic boundary conditions, for which the partition function is

$$Z_{2 \times 2} = (2^4 \cosh^4 \beta)(1 + \tanh^4 \beta). \quad (7.6)$$

The prefactor in this expression (2^N multiplied by one factor of $\cosh \beta$ per edge) was already encountered in Eq. 7.5. We can find naively a 4×4 matrix $\hat{U}_{2 \times 2}$ whose determinant generates cycle configurations which agree with the loop configurations. Although this matrix cannot be generalized to larger lattices, it illustrates the problems which must be overcome. This matrix is given by

$$\hat{U}_{2 \times 2} = \begin{bmatrix} 1 & \gamma \tanh(\beta) & \cdot & \cdot \\ \cdot & 1 & \cdot & \gamma \tanh \beta \\ \gamma \tanh(\beta) & \cdot & 1 & \cdot \\ \cdot & \cdot & \gamma \tanh(\beta) & 1 \end{bmatrix}.$$

(In the following, zero entries in matrices are represented by dots.) The matrix must satisfy

$$Z_{2 \times 2} = (2^4 \cosh^4 \beta) \det \hat{U}_{2 \times 2},$$

and because of

$$\det \hat{U}_{2 \times 2} = 1 - \gamma^4 \tanh^4 \beta,$$

we have to choose $\gamma = \exp i\pi/4 = \sqrt[4]{-1}$. The value of the determinant is easily verified by expanding with respect to the first row, or by naively going through all the 24 permutations of 4 elements. Only two permutations have nonzero contributions: the unit permutation $(\begin{smallmatrix} 1234 \\ 1234 \end{smallmatrix})$, which has weight 1 and sign 1 (it has four cycles), and the permutation, $(\begin{smallmatrix} 2431 \\ 1234 \end{smallmatrix}) = (1, 2, 4, 3)$, which has weight $\gamma^4 \tanh^4 \beta = -\tanh^4 \beta$. The sign of this permutation is -1 , because it consists of a single cycle.

The matrix $\hat{U}_{2 \times 2}$ cannot be generalized directly to larger lattices. This is because it sets u_{21} equal to zero because $u_{12} \neq 0$, and sets $u_{13} = 0$ because $u_{31} \neq 0$; in short it sets $u_{kl} = 0$ if u_{lk} is nonzero (for $k \neq l$). In this way, no cycles with hairpin turns are retained (which go from site k to site l and immediately back to site k). It is also guaranteed that between a permutation and its inverse (in our case, between the permutation $(\begin{smallmatrix} 1234 \\ 1234 \end{smallmatrix})$ and $(\begin{smallmatrix} 2431 \\ 1234 \end{smallmatrix})$), at most one has nonzero weight.

For larger lattices, this strategy is too restrictive. We cannot generate all loop configurations from directed cycle configurations if the direction in which

Table 7.2: Correspondence between lattice sites and directions, and the indices of the Kac–Ward matrix U

Site	Direction	Index
1	$\rightarrow$	1
	$\uparrow$	2
	$\leftarrow$	3
	$\downarrow$	4
2	$\rightarrow$	5
	$\uparrow$	6
	$\leftarrow$	7
	$\downarrow$	8
$\vdots$	$\vdots$	$\vdots$
k	$\rightarrow$	$4k - 3$
	$\uparrow$	$4k - 2$
	$\leftarrow$	$4k - 1$
	$\downarrow$	$4k$

the edges are gone through is fixed. We would thus have to allow both weights u_{kl} and u_{lk} different from zero, but this would reintroduce the hairpin problem. For larger N , there is no $N \times N$ matrix whose determinant yields all the loop configurations.

Kac and Ward’s solution to this problem associates a matrix index, not with each lattice site, but with each of the four directions on each lattice site (see Tab. 7.2), and a matrix element with each pair of directions and lattice sites. Matrix elements are nonzero only for neighboring sites, and only for special pairs of directions (see Fig. 7.4), and hairpin turns can be suppressed.

For concreteness, we continue with the 2×2 lattice, and its 16×16 matrix $U_{2 \times 2}$. We retain from the preliminary matrix $\hat{U}_{2 \times 2}$ that the nonzero matrix element must essentially correspond to terms $\tanh \beta$, but that there are phase factors. This phase factor is 1 for a straight move (case a in Fig. 7.4); it is $\exp(i\pi/4)$ for a left turn, and $\exp(-i\pi/4)$ for a right turn.

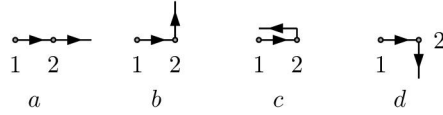


Figure 7.4: Graphical representation of the matrix elements in the first row of the Kac–Ward matrix $U_{2 \times 2}$

Table 7.3: The matrix elements of Fig. 7.4 that make up the first row of the Kac–Ward matrix $U_{2 \times 2}$ (see Eq. 7.7).

Case	Matrix element	value	type
a	$u_{1,5}$	$\nu = \tanh \beta$	(straight move)
b	$u_{1,6}$	$\alpha = e^{i\pi/4} \tanh \beta$	(left turn)
c	$u_{1,7}$	0	(hairpin turn)
d	$u_{1,8}$	$\bar{\alpha} = e^{-i\pi/4} \tanh \beta$	(right turn)

The nonzero elements in the first row of $U_{2 \times 2}$ are shown in Fig. 7.4, and taken up in Tab. 7.3. We arrive at the matrix

$$U_{2 \times 2} = \begin{bmatrix} 1 & \cdot & \cdot & \cdot & \nu & \alpha & \cdot & \bar{\alpha} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \bar{\alpha} & \nu & \alpha & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \bar{\alpha} & \nu & \alpha & \cdot \\ \cdot & \bar{\alpha} & \nu & \alpha & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \nu & \alpha & \cdot & \bar{\alpha} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \alpha & \cdot & \bar{\alpha} & \nu & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \bar{\alpha} & \nu & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \bar{\alpha} & \nu & \alpha & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \alpha & \cdot & \bar{\alpha} & \nu & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{bmatrix}. \quad (7.7)$$

The matrix $U_{2 \times 2}$ contains four nonzero permutations, which we can generate with a naive program (in each row of the matrix, we pick one term out of $\{1, \nu, \alpha, \bar{\alpha}\}$, and then check that each column index appears exactly once). We concentrate in the following on the nontrivial cycles in each permutation (that are not part of the identity). The identity permutation, $P^1 = (\begin{smallmatrix} 1 & \cdots & 16 \\ 1 & & 16 \end{smallmatrix})$, one of the four nonzero permutations, has only trivial cycles. It is characterized by an empty nontrivial cycle configuration c_1 . Other permutations with nonzero weights are

$$c_2 \equiv \begin{pmatrix} \text{site} & 1 & 2 & 4 & 3 \\ \text{dir.} & \rightarrow & \uparrow & \leftarrow & \downarrow \\ \text{index} & 1 & 6 & 15 & 12 \end{pmatrix}$$

and

$$c_3 \equiv \begin{pmatrix} \text{site} & 1 & 3 & 4 & 2 \\ \text{dir.} & \uparrow & \rightarrow & \downarrow & \leftarrow \\ \text{index} & 2 & 9 & 16 & 7 \end{pmatrix}.$$

Finally, the permutation c_4 is put together from the permutations c_2 and c_3 , so that we obtain

$$\begin{aligned} c_1 &\equiv 1, \\ c_2 &\equiv u_{1,6} u_{6,15} u_{15,12} u_{12,1} = \alpha^4 = -\tanh^4(\beta), \\ c_3 &\equiv u_{2,9} u_{9,16} u_{16,7} u_{7,2} = \bar{\alpha}^4 = -\tanh^4(\beta), \\ c_4 &\equiv c_2 c_3 = \alpha^4 \bar{\alpha}^4 = \tanh^8(\beta). \end{aligned}$$

We thus arrive at

$$\det U_{2 \times 2} = 1 + 2 \tanh^4 \beta + \underbrace{\tanh^8 \beta}_{\text{see Eq. 7.6}} = \left(1 + \tanh^4 \beta\right)^2, \quad (7.8)$$

and this is proportional to the square of the partition function in the 2×2 lattice (rather than the partition function itself).

The cycles in the expansion of the determinant are oriented: c_2 runs anticlockwise around the pad, and c_3 clockwise. However, both types of cycles may appear simultaneously, in the cycle c_4 . This is handled by drawing two lattices, one for the clockwise, and one for the anticlockwise cycles (see Fig. 7.5). The cycles $\{c_1, \dots, c_4\}$ correspond to all the loop configurations that can be drawn simultaneously in both lattices. It is thus natural that the determinant in Eq. 7.8 is related to the partition function in two independent lattices, the square of the partition function of the individual systems.

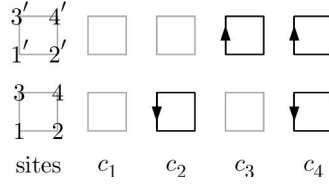


Figure 7.5: Neighbor scheme and cycle configurations in two independent 2×2 Ising models.

Before moving to larger lattices, we note that the matrix $U_{2 \times 2}$ can be written in more compact form, as a matrix of matrices:

$$U_{2 \times 2} = \begin{bmatrix} 1 & u_{\rightarrow} & u_{\uparrow} & \cdot \\ u_{\leftarrow} & 1 & \cdot & u_{\uparrow} \\ u_{\downarrow} & \cdot & 1 & u_{\rightarrow} \\ \cdot & u_{\downarrow} & u_{\leftarrow} & 1 \end{bmatrix} \quad \left(\begin{array}{l} \text{a } 16 \times 16 \text{ matrix,} \\ \text{see Eq. 7.10} \end{array} \right), \quad (7.9)$$

where 1 is the 4×4 unit matrix, and furthermore, the 4×4 matrices $u_{\rightarrow}$, $u_{\uparrow}$, $u_{\leftarrow}$, and $u_{\downarrow}$ are given by

$$\begin{aligned} u_{\rightarrow} &= \begin{bmatrix} \nu & \alpha & \cdot & \bar{\alpha} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}, & u_{\uparrow} &= \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \bar{\alpha} & \nu & \alpha & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}, \\ u_{\leftarrow} &= \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \bar{\alpha} & \nu & \alpha \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}, & u_{\downarrow} &= \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \alpha & \cdot & \bar{\alpha} & \nu \end{bmatrix}. \end{aligned} \quad (7.10)$$

The difference between Eq. 7.7 and Eq. 7.9 is purely notational.

The 2×2 lattice is less complex than larger lattices. For example, one cannot draw loops in this lattice which sometimes turn left, and sometimes right. (On the level of the 2×2 lattice it is unclear why left turns come with a factor α and right turns with a factor $\bar{\alpha}$.) This is what we shall study now, in a larger matrix. Cycle configurations will come up that do not correspond to loop configurations. We shall see that they sum up to zero.

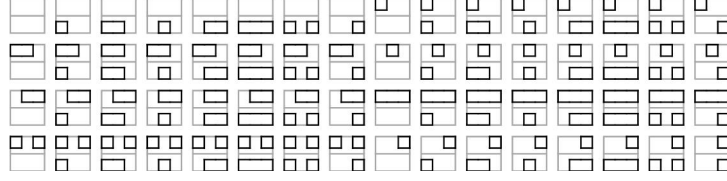


Figure 7.6: All 64 loop configurations for two uncoupled 4×2 Ising models without periodic boundary conditions (a subset of Fig. 7.3).

For concreteness, we consider the 4×2 lattice (without periodic boundary conditions), for which the Kac–Ward matrix can still be written down conveniently. We understand by now that the matrix and the determinant describe pairs of lattices, one for each sense of orientation, so that the pair of 4×2 lattices corresponds to a single 4×4 lattice with a central row of links eliminated. The 64 loop configurations for this case are shown in Fig. 7.6. We obtain

$$U_{4 \times 2} = \begin{bmatrix} 1 & u_{\rightarrow} & \cdot & \cdot & u_{\uparrow} & \cdot & \cdot & \cdot \\ u_{\leftarrow} & 1 & u_{\rightarrow} & \cdot & \cdot & u_{\uparrow} & \cdot & \cdot \\ \cdot & u_{\leftarrow} & 1 & u_{\rightarrow} & \cdot & \cdot & u_{\uparrow} & \cdot \\ \cdot & \cdot & u_{\leftarrow} & 1 & \cdot & \cdot & \cdot & u_{\uparrow} \\ u_{\downarrow} & \cdot & \cdot & \cdot & 1 & u_{\rightarrow} & \cdot & \cdot \\ \cdot & u_{\downarrow} & \cdot & \cdot & u_{\leftarrow} & 1 & u_{\rightarrow} & \cdot \\ \cdot & \cdot & u_{\downarrow} & \cdot & \cdot & u_{\leftarrow} & 1 & u_{\rightarrow} \\ \cdot & \cdot & \cdot & u_{\downarrow} & \cdot & \cdot & u_{\leftarrow} & 1 \end{bmatrix}. \quad (7.11)$$

Written out explicitly, this gives a 32×32 complex matrix $U_{4 \times 2} = (u_{k,l})$ with

elements

[illegible]

This matrix is constructed according to the same rules as $U_{2 \times 2}$, earlier.

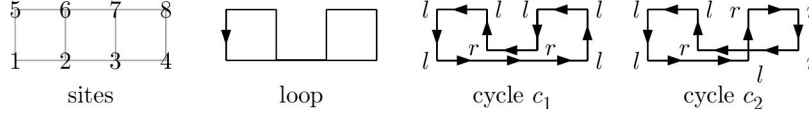


Figure 7.7: A loop in the 4×2 system, not present in Fig. 7.6. Weights of c_1 and c_2 cancel.

The cycle c_2 in Fig. 7.7 can be described by the following trajectory:

$$\text{cycle } c_2 \equiv \begin{bmatrix} \text{site} & 1 & 2 & 3 & 7 & 8 & 4 & 3 & 2 & 6 & 5 \\ \text{dir.} & \rightarrow & \rightarrow & \uparrow & \rightarrow & \downarrow & \leftarrow & \leftarrow & \uparrow & \leftarrow & \downarrow \\ \text{index} & 1 & 5 & 10 & 25 & 32 & 15 & 11 & 6 & 23 & 20 \end{bmatrix}.$$

This cycle thus corresponds to the following product of matrix elements:

$$\{\text{weight of } c_2\} : u_{1,5}u_{5,10} \dots u_{23,20}u_{20,1}.$$

The cycle c_2 makes four left and four right turns (so that the weight is proportional to $\bar{\alpha}^4 \alpha^4 \propto +1$) whereas the cycle c_1 turns six times to the left and twice to the right, with weight $\bar{\alpha}^6 \alpha^2 \propto -1$, canceling c_2 .

A naive program easily generates all of the nontrivial cycles in $U_{4 \times 2}$ (in each row of the matrix, we pick one term out of $\{1, \nu, \alpha, \bar{\alpha}\}$, and then check that each column index appears exactly once). This reproduces the loop list, with 64 contributions, shown in Fig. 7.6. There are in addition 80 more cycle configurations, which are either not present in the figure, or are equivalent to

cycle configurations already taken into account. Some examples are the cycles c_1 and c_2 in Fig. 7.7. It was the good fortune of Kac and Ward that they all add up to zero.

On larger than 4×2 lattices, there are more elaborate loops. They can, for example, have crossings (see, for example, the loop in Fig. 7.8). There, the cycle configurations c_1 and c_2 correspond to loops in the generalization of Fig. 7.6 to larger lattices, whereas the cycles c_3 and c_4 are superfluous. However, c_3 makes six left turns and two right turns, so that the overall weight is $\alpha^4 = -1$, whereas the cycle c_4 makes three left turns and three right turns, so that the weight is $+1$, the opposite of that of c_3 . The weights of c_3 and c_4 thus cancel.

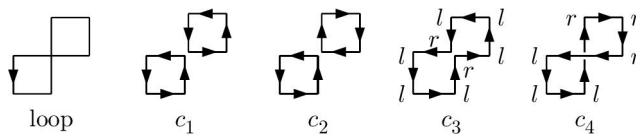


Figure 7.8: Loop and cycle configurations. The weights of c_3 and c_4 cancel.

For larger lattices, it becomes difficult to establish that the sum of cycle configurations in the determinant indeed agrees with the sum of loop configurations of the high-temperature expansion, although rigorous proofs exist to that effect. However, at our introductory level, it is more rewarding to proceed heuristically. We can, for example, write down the 144×144 matrix $U_{6 \times 6}$ of the 6×6 lattice for various temperatures (using Alg. combinatorial-ising.py), and evaluate the determinant $\det U_{6 \times 6}$ with a standard linear-algebra routine. Partition functions thus obtained are equivalent to those resulting from Gray-code enumeration, even though the determinant is evaluated in on the order of $144^3 \simeq 3 \times 10^6$ operations, while the Gray code goes over $2^{35} \simeq 3 \times 10^{10}$ configurations. The point is that the determinant can be evaluated for lattices that are much too large to go through the list of all configurations.

The matrix $U_{L \times L}$ for the $L \times L$ lattice contains the key to the analytic solution of the two-dimensional Ising model first obtained, in the thermodynamic limit, by Onsager (1944). To recover Onsager's solution, we would have to compute the determinant of U , not numerically as we did, but analytically, as a product over all the eigenvalues. Analytic expressions for the partition functions for Ising models can also be obtained for finite lattices with periodic boundary conditions. To adapt for the changed boundary conditions, one needs four matrices, generalizing the matrix U (compare with the analogous situation for dimers in chapter xx. Remarkably, evaluating $Z(\beta)$ on a finite lattice reduces to evaluating an explicit function (see the classical papers by Kaufman (1949) [9] and Ferdinand and Fisher (1969) [12].

The analytic solutions of the Ising model have not been generalized to higher dimensions, where only Monte Carlo simulations, high-temperature expansions, and renormalization-group calculations allow to compute to high precision the properties of the phase transition. These properties, as mentioned, are universal,

that is, they are the same for a wide class of systems, called the Ising universality class.

7.3 Density of states from thermodynamics

The direct and indirect enumeration algorithms in this chapter differ in the role played by the density of states. In Alg. 0Denumerate-ising, it was appropriate to first compute $\mathcal{N}(E)$, and later determine partition functions, internal energies, and specific heat capacities at any temperature, in $\propto N$ operations. In contrast, the indirect enumerations in Section ?? determine the partition function $Z(\beta)$, not the density of states. Computing $Z(\beta)$ from $\mathcal{N}(E)$ is straightforward, but how to recover $\mathcal{N}(E)$ from $Z(\beta)$ requires some thought: The mathematical problem of the present section is common to many basic problems in statistical and solid state physics, and appears also in the interpretation of experimental or Monte Carlo data. In the presence of statistical uncertainties, it is very difficult to solve, and may often be ill-defined. This means, in our case, that algorithms exist for computing $\mathcal{N}E$ if the partition functions were computed exactly. If, however, $Z(\beta)$ is known only to limited precision, the output generated by the slightly perturbed input can be drastically different from the exact output.

Bibliography

- [1] W. Krauth, *Statistical Mechanics: Algorithms and Computations*. Oxford Master Series in Physics, Oxford University Press, UK, 2006.
- [2] B. Efron, “Maximum likelihood and decision theory,” *Ann. Statist.*, vol. 10, pp. 340–356, 06 1982.
- [3] B. Efron, “Bootstrap methods: another look at the jackknife,” *The Annals of Statistics*, pp. 1–26, 1979.
- [4] P. Diaconis and B. Efron, “Computer intensive methods in statistics,” *Scientific American*, vol. 248, no. 5, pp. 116–130, 1983.
- [5] B. Gnedenko and A. Kolmogorov, *Limit distributions for sums of independent random variables*. Addison-Wesley series in statistics, Addison-Wesley, 1954.
- [6] R. H. Swendsen, “Statistical mechanics of colloids and Boltzmann’s definition of the entropy,” *American Journal of Physics*, vol. 74, pp. 187–190, Mar. 2006.
- [7] H. S. Leff and M. H. Coopersmith, “Translational Invariance Properties of a Finite One-Dimensional Hard-Core Fluid,” *Journal of Mathematical Physics*, vol. 8, pp. 306–314, Feb. 1967.
- [8] M. Kac and J. C. Ward, “A combinatorial solution of the two-dimensional ising model,” *Phys. Rev.*, vol. 88, pp. 1332–1337, Dec 1952.
- [9] B. Kaufman, “Crystal statistics. ii. partition function evaluated by spinor analysis,” *Phys. Rev.*, vol. 76, pp. 1232–1243, Oct 1949.
- [10] L. Onsager, “Crystal statistics. i. a two-dimensional model with an order-disorder transition,” *Phys. Rev.*, vol. 65, pp. 117–149, Feb 1944.
- [11] B. L. van der Waerden, “Die lange Reichweite der regelmäßigen Atomanordnung in Mischkristallen,” *Zeitschrift für Physik*, vol. 118, pp. 473–488, July 1941.
- [12] A. E. Ferdinand and M. E. Fisher, “Bounded and inhomogeneous ising models. i. specific-heat anomaly of a finite lattice,” *Phys. Rev.*, vol. 185, pp. 832–846, Sep 1969.