Computation of the Ising partition function for grid graphs*

Per Håkan Lundow

Abstract

The Ising partition function for a graph counts the number of bipartitions of the vertices into sets of given sizes, with a given size of the induced edge cut. This is expressed as a 2 variable generating function which is easily translatable into the partition function studied in statistical physics. The author has exactly computed this generating function for the square-, triangular- and union jack grid with open boundary, the largest having 256 vertices. We describe the computing process, which is based upon transfer matrices and a simple use of symmetry, and do a rudimentary analysis of the phase transition which occurs when the edge cut attains a certain critical size.

1 Introduction

The Ising model of ferromagnetism has been thoroughly investigated since its formulation in the 1920's. It was solved for the 1-dimensional case by Ising himself and the 2-dimensional case without external field was solved by Onsager in the 1940's. However, not much is known for higher dimensions or indeed for the 2-dimensional case with external field, which is the object of study in this article. For a well-written introduction to the subject we refer the reader to Cipra [Cip87]. The standard Ising partition function Z, for a graph on N vertices, is defined as

$$Z(\beta, J, H, N) = \sum_{\sigma} e^{\beta \mathcal{H}(\sigma)}$$

where the sum is taken over all 2^N assignments of ± 1 to the N variables σ_i. Here $\beta = 1/kT$ with k being Boltzmann's constant and T is the temperature. The Hamiltonian $\mathcal{H}$ is defined as

$$\mathcal{H}(\sigma) = -J \sum_{\{i,j\}} \sigma_i \sigma_j - H \sum_i \sigma_i$$

where the first sum is taken over the edges (nearest-neighbours) in the graph and the second sum over the vertices. The constants J and H describe the interaction between nearest-neighbours and an external field, respectively. We will modify the definition of this partition function into a purely combinatorial

*Research reports, No 14, 1999, Department of Mathematics, Umeå University

function defined for any graph. More specific, for a graph G on n vertices and m edges it is defined as

$$Z(G;\, x, y) = \sum_{i,j} a_{i,j}\, x^i\, y^j$$

where $a_{i,j}$ is the number of bipartitions of the vertices into parts of order $(n-j)/2$ and $(n+j)/2$ respectively, with $(m-i)/2$ edges between them. It is then a matter of evaluating the partition function in the right point to obtain the standard function, i.e.

$$Z(G;\, e^{-\beta J},\, e^{-\beta H}) = Z(\beta, J, H, N).$$

This article is devoted to the subject of exact computation of $Z(G;\, x, y)$. This is of some interest not least since one may extract e.g. the matching polynomial from this partition function. There is a (practical) closed formula for the number of perfect matchings in $P_m \times P_n$, constructed by Kasteleyn [Kas67] using the method of Pfaffians, but the matching polynomial, and indeed the number of matchings, remains difficult to compute. From $Z(G;\, x, 1)$ one can extract the Euler polynomial which counts the number of factors in G where each vertex has even degree (or, each component is an Eulerian graph). Though this polynomial is easily computed for $C_m \times C_n$ with $m, n \leq 32$ using Mathematica, see Beale [Bea96][1], the full partition function in both variables is obtainable only for considerably smaller grids. The largest computed so far are $C_{10} \times C_{10}$ by Baker [Bak94] (though the coefficients aren't stated) and $C_{12} \times C_{12}$ by Häggkvist and Lundow (manuscript in preparation).

We will describe a computing process which gives the partition function for $P_n \times P_n$, where $n \leq 16$. As more computational power gets available the method will render the next two or three grids as well. The method relies on the use of transfer matrices. Having been in use since the beginning of the Ising model it has proven to be a powerful ally. In fact, computing $Z(C_{10} \times C_{10};\, x, y)$ with the standard transfer matrix is quite manageable on todays computers. In combination with a technique for compression of transfer matrices, see Lundow [Lun99], the author has computed e.g. $Z(C_4 \times C_4 \times P_n;\, x, y)$ for $n = 1, \ldots, 10$ and $Z(C_4 \times C_4 \times P_n;\, x, 1)$ for $n = 1, \ldots, 32$. Without the compression, the matrices would have had order $2^{16} = 65536$, but with compression the order reduces to 402. Note that the entries of the compressed matrix contains polynomials, whereas the original matrix contains only monomials. For graphs of type $C_n \times C_n \times C_n$ nothing has been computed exactly when $n > 4$ and we have to resort to Monte Carlo simulations to obtain information of the coefficients, see [HR$^+$99].

2 Notation and definitions

A simple graph $G = (V, E)$ is an ordered pair where V is a set of vertices and E, the edges, is a set of 2-subsets of V. If W is a subset of $V(G)$ then $G[W]$ is the subgraph of G induced by W. $G - W$ is the graph obtained by deleting the vertices in W, i.e. $G[\overline{W}]$, where $\overline{W} = V(G) \setminus W$. If $W_1, W_2 \subseteq V(G)$ then

[1] Unfortunately the Mathematica program contains a bug, easily fixed though, resulting in only a few correct digits.

$[W_1, W_2]$ is the set of edges having one end in W_1 and the other in W_2. If $F \subseteq E(G)$ then $G - F$ is the graph obtained by deleting the edges F from G. If F is a set of edges having one end in graph G then $G \cup F$ is the graph G together with the edges F and the vertices at the other end of F. We let P_n and C_n denote the path and cycle respectively on n vertices. If G and H are graphs then $G \times H$ denotes the cartesian product of G and H. The vertices of $G \times H$ are $\{(v, w) : v \in V(G), w \in V(H)\}$. Two vertices (v_1, w_1) and (v_2, w_2) are adjacent iff $v_1 = v_2$ and $\{w_1, w_2\} \in V(H)$ or $w_1 = w_2$ and $\{v_1, v_2\} \in V(G)$. For example, $P_m \times P_n$ is the $m \times n$ - grid with open boundary, and $C_m \times C_n$ is the same but with periodic boundary. The function $\sigma : V \to \{-1, +1\}$ is the state of the graph and σ_v is the spin of vertex v. The restriction of σ to W is denoted σW.

Definition 2.1. Given a state σ we define the energy $\nu(G)$ and magnetisation $\mu(G)$ as

$$\nu(G) = \sum_{\{u,v\} \in E(G)} \sigma_u \sigma_v \quad \text{and} \quad \mu(G) = \sum_{v \in V(G)} \sigma_v$$

If $W \subseteq V(G)$ and $F \subseteq E(G)$ then

$$\nu(G, W) = \sum_{\{u,v\} \in [W,W]} \sigma_u \sigma_v, \quad \nu(G, F) = \sum_{\{u,v\} \in F} \sigma_u \sigma_v, \quad \mu(G, W) = \sum_{v \in W} \sigma_v$$

We suppress the G from ν and μ when the context leaves no room for ambiguity.

Definition 2.2. The Ising partition function $Z(G; x, y)$ for a graph $G = (V, E)$ in variables x and y is defined as

$$Z(G;\, x, y) = \sum_{\sigma \in \{-1,+1\}^V} x^{\nu(G)}\, y^{\mu(G)}$$

Since we will use only the variables x and y we henceforth write $Z(G)$. Let $Z(G, \sigma W)$ be the partition function when the vertices in W keeps their spins fixed to σW,

$$Z(G, \sigma W) = \sum_{\sigma \overline{W}} x^{\nu(G)}\, y^{\mu(G)}$$

3 Basic theorems

If we just use Definition 2.2 we need to run through 2^n states to compute $Z(G)$, here some tricks are indicated to cut down the required work.

Lemma 3.1. *Let $W \subseteq V(G)$. Then*

$$\nu(G) = \nu([W, W]) + \nu([W, \overline{W}]) + \nu([\overline{W}, \overline{W}])$$
$$\mu(G) = \mu(W) + \mu(\overline{W})$$

Proof. See Definition 2.1. □

Lemma 3.2. *If $W \subseteq V(G)$ then*

$$Z(G) = \sum_{\sigma W} Z(G, \sigma W)$$

Proof. See Definition 2.2. □

The following theorem is useful since it tells us how to divide up the computation process into smaller chunks.

Theorem 3.3. *If G and H are two graphs with $W = V(G) \cap V(H)$ and $F = E(G) \cap E(H)$ then*

$$Z(G \cup H) = \sum_{\sigma W} Z(G, \sigma W)\, Z(H, \sigma W)\, x^{-\nu(F)}\, y^{-\mu(W)}$$

Proof. Let $A = V(G) \setminus W$, $B = V(H) \setminus W$ and $C = (V(G) \cup V(H)) \setminus W$. By the definition of Z we have

$$Z(G, \sigma W)\, Z(H, \sigma W) = \left(\sum_{\sigma A} x^{\nu(G)} y^{\mu(G)} \right) \left(\sum_{\sigma B} x^{\nu(H)} y^{\mu(H)} \right) = \sum_{\sigma A} \sum_{\sigma B} x^{\nu(G)+\nu(H)} y^{\mu(G)+\mu(H)}$$

By Lemma 3.1 and 3.2 this is

$$\sum_{\sigma A} \sum_{\sigma B} x^{\nu(G \cup H)+\nu(F)} y^{\mu(G \cup H)+\mu(W)} = \sum_{\sigma C} x^{\nu(G \cup H)} y^{\mu(G \cup H)} x^{\nu(F)} y^{\mu(W)}$$

Since F contribute twice to the energy and W contribute twice to the magnetisation this must be compensated for as in the formulation of the theorem. □

Corollary 3.4. *If G and H are vertex disjoint graphs then $Z(G \cup H) = Z(G)\, Z(H)$.*

As the next theorem states, we never need to run through all 2^n states of a graph, only half of them are required by keeping one spin fixed.

Theorem 3.5. *Select a vertex v from the graph G. Then*

$$Z(G;\, x, y) = Z(G, \sigma_v{=}{+}1;\, x, y) \,+\, Z(G, \sigma_v{=}{+}1;\, x, y^{-1})$$

Proof. Say that $Z(G, \sigma_v{=}{+}1) = \sum a_{i,j}\, x^i y^j$. Replacing each state σ with $-\sigma$ while computing this gives us $Z(G, \sigma_v{=}{-}1) = \sum b_{i,j}\, x^i y^j$. Note that switching the sign of each spin leaves the energy unchanged but switches the sign of the magnetisation, i.e. $a_{i,j} = b_{i,-j}$. Thus

$$\begin{aligned} Z(G;\, x, y) &= \sum a_{i,j}\, x^i y^j + \sum b_{i,j}\, x^i y^j = \\ &\sum a_{i,j}\, x^i y^j + \sum a_{i,-j}\, x^i y^j = \\ &\sum a_{i,j}\, x^i y^j + \sum a_{i,j}\, x^i y^{-j} = \\ &Z(G, \sigma_v{=}{+}1;\, x, y) \,+\, Z(G, \sigma_v{=}{+}1;\, x, y^{-1}) \end{aligned}$$

□

4 Polygraphs and transfer matrices

Before we describe how to use the transfer matrix method we introduce the polygraphs, see Babic et al. [B$^+$86]. A polygraph consists of a set of disjoint graphs $G_1, \ldots, G_p$ and a set of binary relations $X_1, \ldots, X_p$ where $X_i \subseteq V(G_{i-1}) \times V(G_i)$, for $i = 1, \ldots, p$, and the binary relations will be thought of as sets of edges. We count the indices modulo p so that e.g. G_0 means G_p and G_{p+1} means G_1. Let $\Omega_p = \Omega(\{G_i\}, \{X_i\}, i = 1, \ldots, p)$ denote the polygraph having vertices $V(G_1) \cup \cdots \cup V(G_p)$ and edges $X_1 \cup E(G_1) \cup \cdots \cup X_p \cup E(G_p)$. Let Γ_p be the polygraph Ω_p without the edges X_1, i.e. with $X_1 = \emptyset$. Also, let $\tilde{\Omega}_p$ be the polygraph different from Ω_p only in that the domain of X_1 is disjoint from $V(G_p)$ and is just some set of vertices. A particularly nice polygraph is when $G_i = G$ and $X_i = X$ for $i = 1, \ldots, p$; denote this polygraph by $\Omega(G, X, p)$ or $\Gamma(G, X, p)$ when $X_1 = \emptyset$. For example, if $G = (\{v\}, \emptyset)$ and $X = \{(v, v)\}$, then $\Omega(G, X, p)$ and $\Gamma(G, X, p)$ is the cycle and path respectively on p vertices. Let D_i be the domain of X_i and let H_i be the graph $X_i \cup G_i$ on vertices $D_i \cup V(G_i)$ and edges $X_i \cup E(G_i)$. We refer to D_i and D_{i+1} as the in-vertices and out-vertices respectively of H_i. Note by the way that D_i and D_{i+1} are disjoint sets.

Having stated this we are now prepared to formulate the transfer matrix method, see also Biggs [Big77]. Define the matrices $\mathbf{T}_i$, for $i = 1, \ldots, p$, having entries

$$\mathbf{T}_i(\zeta, \xi) = Z(H_i,\, \sigma D_i{=}\zeta,\, \sigma D_{i+1}{=}\xi)\, y^{-\mu(D_i)},$$

Note that $\mathbf{T}_i$ has shape $2^{|D_i|} \times 2^{|D_{i+1}|}$.

Theorem 4.1. *For all ζ, ξ,*

$$\mathbf{T}_1 \cdots \mathbf{T}_p(\zeta, \xi) = Z(\tilde{\Omega}_p, \sigma D_1{=}\zeta, \sigma D_{p+1}{=}\xi)\, y^{-\mu(D_1)}$$

Proof. By induction on p. The theorem is true for $p = 1$ due to the definition of the transfer matrix. We assume the theorem to be true for $p-1$ and show it for p.

$$\mathbf{T}_1 \cdots \mathbf{T}_p(\zeta, \xi) = \sum_{\eta} \mathbf{T}_1 \cdots \mathbf{T}_{p-1}(\zeta, \eta)\, \mathbf{T}_p(\eta, \xi)$$

By the induction hypothesis this is

$$\sum_{\eta} Z(\tilde{\Omega}_{p-1}, \sigma D_1{=}\zeta, \sigma D_p{=}\eta)\, y^{-\mu(D_1)}\, \mathbf{T}_p(\eta, \xi) =$$

$$\sum_{\sigma D_p} Z(\tilde{\Omega}_{p-1}, \sigma D_1{=}\zeta, \sigma D_p)\, y^{-\mu(D_1)}\, Z(H_p,\, \sigma D_p, \sigma D_{p+1}{=}\xi)\, y^{-\mu(D_p)}$$

and by Theorem 3.3 we obtain

$$Z(\tilde{\Omega}_p, \sigma D_1{=}\zeta, \sigma D_{p+1}{=}\xi)\, y^{-\mu(D_1)}$$

□

Corollary 4.2.

$$Z(\Omega_p) = \mathrm{trace}(\mathbf{T}_1\, \mathbf{T}_2 \cdots \mathbf{T}_p)$$

Proof. Identify D_1 with D_p so that $\tilde{\Omega}_p$ becomes Ω_p.

$$Z(\Omega_p) = \sum_{\sigma D_1} Z(\Omega_p, \sigma D_1) =$$

$$\sum_{\zeta} Z(\tilde{\Omega}_p, \sigma D_1{=}\zeta, \sigma D_{p+1}{=}\zeta) = \operatorname{trace}(\mathbf{T}_1 \cdots \mathbf{T}_p)$$

□

Corollary 4.3. *Define the vector* $\mathbf{v}$ *with entries* $\mathbf{v}(\zeta) = Z(G_1, \sigma D_2{=}\zeta)$ *and let* $\mathbf{1}$ *be a vector of 1's. Then*

$$Z(\Gamma_p) = \mathbf{v}^{\mathrm{t}}\, \mathbf{T}_2 \cdots \mathbf{T}_p\, \mathbf{1}.$$

Proof. Let $X_1 = \emptyset$ in the previous corollary. □

Corollary 4.4. *If* $G_1 = \cdots = G_p = G$ *and* $X_1 = \cdots = X_p = X$ *then* $T_1 = \cdots = T_p = T$ *and*

$$Z(\Omega_p) = \operatorname{trace}(\mathbf{T}^p), \qquad Z(\Gamma_p) = \mathbf{v}^{\mathrm{t}}\, \mathbf{T}^{p-1}\, \mathbf{1}$$

Example 4.5. We set up the transfer matrix for computing $Z(P_2 \times C_n)$ and $Z(P_2 \times P_n)$, i.e. for the $2 \times n$ grid with and without periodic boundary conditions respectively (in the n direction). Let $G = P_2$ and $X = \{(1,1),(2,2)\}$. The transfer matrix becomes

$$\mathbf{T} = \left(\begin{array}{c|cccc} \sigma & -,- & +,- & -,+ & +,+ \\ \hline -,- & x^3y^{-2} & x^{-1} & x^{-1} & x^{-1}y^2 \\ +,- & xy^{-2} & x & x^{-3} & xy^2 \\ -,+ & xy^{-2} & x^{-3} & x & xy^2 \\ +,+ & x^{-1}y^{-2} & x^{-1} & x^{-1} & x^3y^2 \end{array}\right)$$

and the vector $\mathbf{v}$ becomes

$$\mathbf{v}^{\mathrm{t}} = \left(xy^{-2},\, x^{-1},\, x^{-1},\, xy^2\right)$$

The state of G indicated in Figure 1 correspond to position 2 of $\mathbf{v}$ and the state of $X \cup G$ correspond to row 2 and column 3 of $\mathbf{T}$. Corollary 4.4 tells us how to compute the desired partition functions, i.e. $Z(P_2 \times C_n) = \operatorname{trace}(\mathbf{T}^n)$ and $Z(P_2 \times P_n) = \mathbf{v}^{\mathrm{t}}\, \mathbf{T}^{n-1}\, \mathbf{1}$.

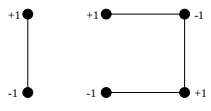

Figure 1: The graphs G and $X \cup G$ of Example 4.5.

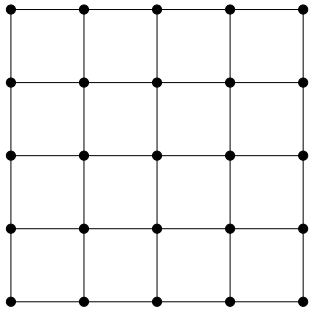

Figure 2: The graph $P_5 \times P_5$.

5 Application to the grid

In this section we describe how to use the theorems in the previous sections to compute the partition function for the open grid $P_m \times P_n$, see Figure 2. Let the vertices of this graph be the set $\{(i,j) : i = 1,\ldots,m,\ j = 1,\ldots,n\}$. We split up the graph into two subgraphs $G = P_k \times P_n$ and $H = P_\ell \times P_n$ where $m = k+\ell-1$. Let G be the graph induced by the set $\{(i,j) : i = 1,\ldots,k,\ j = 1,\ldots,n\}$ and H be the graph induced by the set $\{(i,j) : i = k,\ldots,m,\ j = 1,\ldots,n\}$, see Figure 3. With $W = V(G \cap H)$ and $F = E(G \cap H)$ the graph (W,F) is isomorphic to the path P_n induced by the vertices $\{(k,j) : j = 1,\ldots,n\}$. Now we are ready to apply Theorem 3.3 to G and H. We have

$$Z(P_m \times P_n) = \sum_{\sigma W} Z(G,\sigma W)\, Z(H,\sigma W)\, x^{-\nu(F)}\, y^{-\mu(W)} \tag{1}$$

It remains to compute each $Z(G,\sigma W)$ and $Z(H,\sigma W)$ with the transfer matrix method. We focus on the $Z(G,\sigma W)$, the other case being analogous. Take a path P_k and the relation $X = \{(i,i) : i = 1,\ldots,k\}$ so that $\Gamma(P_k, X, n)$ is isomorphic to G. As in Example 4.5 we set up the transfer matrix $\mathbf{T}$ and the vector $\mathbf{v}$ so that $Z(G) = \mathbf{v}^{\mathrm{t}}\, \mathbf{T}^{n-1}\, \mathbf{1}$. However, we wanted to keep the spins in σW fixed. Note that the matrix $\mathbf{T}$ consists of the four submatrices $\mathbf{T}_{(\pm,\pm)}$ and the vector $\mathbf{v}$ of the two subvectors $\mathbf{v}_\pm$, each of order 2^{k-1}, as in

$$\mathbf{T} = \begin{pmatrix} \mathbf{T}_{(-,-)} & \mathbf{T}_{(-,+)} \\ \mathbf{T}_{(+,-)} & \mathbf{T}_{(+,+)} \end{pmatrix} \qquad \mathbf{v} = \begin{pmatrix} \mathbf{v}_- \\ \mathbf{v}_+ \end{pmatrix}$$

Indeed, the canonical ordering of the rows and columns induces these blocks. Now let the vertices of P_k and $X \cup P_k$ be labelled as in the right picture in Figure 3. The index of $\mathbf{v}$ correspond to the spin of vertex k in P_k and the indices of $\mathbf{T}$ correspond to the spins of vertices k and $2k$ in $X \cup P_k$. Given a σW we just compute the correct vector-matrix product. For $\sigma W = (\sigma_1,\ldots,\sigma_n)$ we have

$$Z(G,\sigma W) = \mathbf{v}_{\sigma_1}^{\mathrm{t}}\ \mathbf{T}_{(\sigma_1,\sigma_2)}\ \mathbf{T}_{(\sigma_2,\sigma_3)}\ \ldots\ \mathbf{T}_{(\sigma_{n-1},\sigma_n)}\ \mathbf{1}$$

For example, with $n = 5$ and $\sigma W = (-1,+1,+1,-1,-1)$, to obtain $Z(G,\sigma W)$ we compute

$$\mathbf{v}_-^{\mathrm{t}}\ \mathbf{T}_{(-,+)}\ \mathbf{T}_{(+,+)}\ \mathbf{T}_{(+,-)}\ \mathbf{T}_{(-,-)}\ \mathbf{1}$$

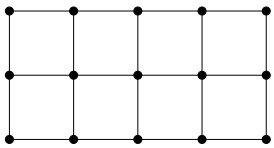
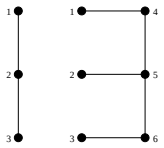

Figure 3: Left: the graph G for $k = 3$ and $n = 5$. Right: the graphs P_k and $X \cup P_k$ for $k = 3$.

Observe that we need not restrict ourselves to grids of type $P_m \times P_n$ when applying this method. It will obviously work for other types of grid graphs such as the triangular grid $P_n \triangleleft P_n$ and the strong product grid $P_m * P_n$ (the union jack grid), see Figure 4.

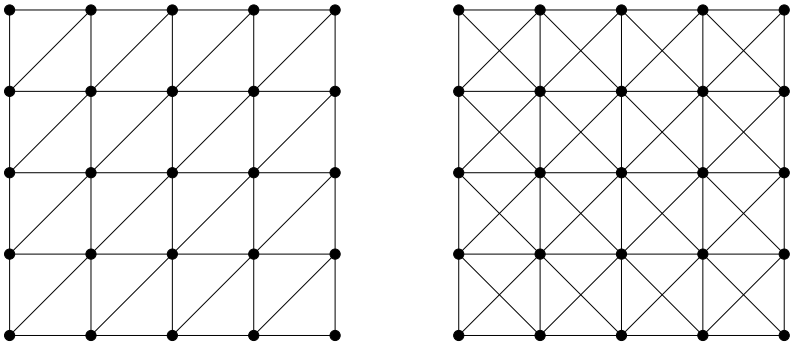

Figure 4: $P_5 \triangleleft P_5$ and $P_5 * P_5$.

We end this section with a few notes on how to cut down still further on the required amount of work. The sum of Equation (1) has 2^n terms and this can be reduced by roughly a factor four. Keeping one of the spins in σW fixed is the first step, this reduces by a factor two. When n is odd we can gain another factor 2 (approximately) by exploiting the symmetry of P_n. In this case we keep the spin of the middle vertex of P_n fixed. Unfortunately this symmetry doesn't exist in the case of triangular grids.

6 A special case

In the special case $P_n \times P_n$ there is an even better way to organise the computations. Let G be the lower triangular part of $P_n \times P_n$, see Figure 5, with W being the vertices on the diagonal. As before we apply Equation (1) and as in the previous section each $Z(G, \sigma W)$ is computed with the transfer matrix method. We split up G into subgraphs $H_1, H_2, \ldots, H_{2n-2}$ as in Figure 6, where each H_i are separated by dashed lines, which also indicate how the H_i are joined. Going from, say left to right, let A_i be the in-vertices and B_i the out-vertices of H_i. We define the transfer matrices $\mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \ldots, \mathbf{T}^{(2n-2)}$ with entries

$$\begin{aligned}
\mathbf{T}^{(1)}(\zeta, \xi) &= Z(H_1,\, \sigma A_1{=}\zeta,\, \sigma B_1{=}\xi), \\
\mathbf{T}^{(i)}(\zeta, \xi) &= Z(H_i,\, \sigma A_i{=}\zeta,\, \sigma B_i{=}\xi)\, y^{-\mu(A_i)}, \quad i = 2, \ldots, 2n-2
\end{aligned}$$

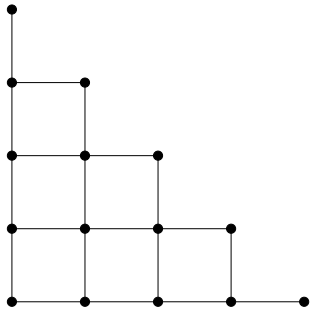

Figure 5: Lower triangular part of $P_5 \times P_5$.

We still want to keep the spins σW fixed so we note that each H_i contains a vertex of W and let $\mathbf{T}_\sigma^{(i)}$ be the submatrix of $\mathbf{T}^{(i)}$ where that particular vertex is fixed to σ. As in the previous section we then have

$$Z(G,\, \sigma W) = \mathbf{T}_{\sigma_1}^{(1)}\ \mathbf{T}_{\sigma_2}^{(2)}\ \mathbf{T}_{\sigma_2}^{(3)} \cdots \mathbf{T}_{\sigma_{n-1}}^{(2n-4)}\ \mathbf{T}_{\sigma_{n-1}}^{(2n-3)}\ \mathbf{T}_{\sigma_n}^{(2n-2)}$$

and then remains to plug this in to Equation (1). For example, with $n = 5$ and $\sigma W = (-1, +1, +1, -1, -1)$, to obtain $Z(G, \sigma W)$ we compute

$$\mathbf{T}_-^{(1)}\ \mathbf{T}_+^{(2)}\ \mathbf{T}_+^{(3)}\ \mathbf{T}_+^{(4)}\ \mathbf{T}_+^{(5)}\ \mathbf{T}_-^{(6)}\ \mathbf{T}_-^{(7)}\ \mathbf{T}_-^{(8)}$$

Observe by the way that $\mathbf{T}_\sigma^{(1)}$ is a 1×2 matrix and $\mathbf{T}_\sigma^{(2n-2)}$ is a 2×1 matrix, so the result is a 1×1 matrix.

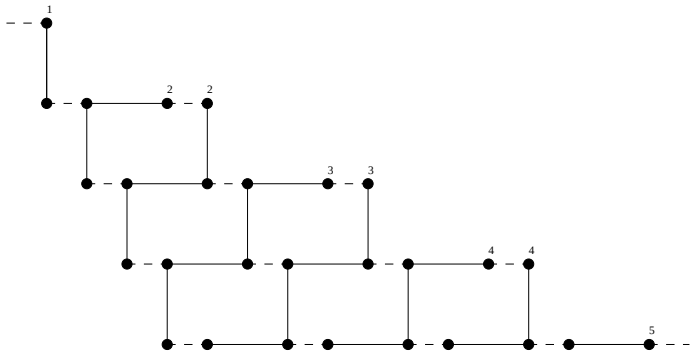

Figure 6: The building blocks.

7 Implementation

The methods described were implemented in Fortran 90 and the source code can be obtained by contacting the author. The programs were run on IBM machines located at the Center for Parallel Computers in Stockholm, and at the High Performance Computing Center North in Umeå, Sweden, using a total of

more than three years of CPU-time. Since the nature of the problem is embarrassingly parallel (map each fixed spin-configuration σW to its own processor), parallelisation is very simple using e.g. MPI.

The author has written a portable module for operations on multi-variate polynomials with coefficients of various types, e.g. REALs and multi-precision integers, see [Lun00a, Lun00b]. The polynomials are stored as linked lists of monomials allowing the polynomials to grow dynamically in size. The resulting files of data are easily read and manipulated using e.g. Mathematica. These files can be downloaded from `<http://www.math.umu.se>`.

Since the transfer matrices never contain anything more complicated than monomials with coefficient 1 there is no need to actually store them, it is more practical to simply generate the (i, j):th position of the matrix with a special function. This was done using the bit-operations available in Fortran 90.

A drawback with doing the computations with fully stored polynomials is of course the memory requirements. The vast majority of the processors available at the time were equipped with a 256 MByte RAM memory, which was quite sufficient. After all, the major obstacle is time. For example, $P_{16} \times P_{16}$ required about two years (sic) of CPU-time.

8 The phase transition

Let us now take a look at the coefficients of the polynomials. We delete first the empty energies and magnetisations and normalise the coefficients so that at each energy the maximum coefficient is 1. The result is shown in the density plots in Figures 7, where the magnetisation is on the x-axis and the energy on the y-axis. We see here a phenomenon which is found in almost all graphs, namely that above a certain critical energy the distribution of the magnetisations undergoes a phase transition; we go from a unimodal distribution to a bimodal distribution, see Figures 8 to 11. The natural question is then, at what critical energy does this take place and what is the asymptotic behaviour for a certain family of graphs? An early study of this phenomenon can be found in [HR97]. Though we will not be able to give an analytic solution to these questions here, we can at least indicate some tendencies. To begin with, we need a definition of critical energy. Though this definition may not be suitable for all graphs, it goes well with the grid graphs.

Definition 8.1. The critical energy ν_c of a graph G is the maximum energy where the distribution of magnetisations is unimodal.

In Table 1 we see where this phase transition occurs for the various grid graphs.

At this point we shall try to guess an asymptotic formula for the value of the relative energy ν_c/m for our three families of grids. From classical Ising model theory, see Lavis and Bell [LB99] volume 1 chapter 8.3, we learn that ν/m at the critical temperature is $1/\sqrt{2}$ for an infinite square grid. For an infinite triangular grid this value is $2/3$. We will simply assume that our critical energies ν_c/m coincide with these values in the limit. For the strong product grid no such value is known (or at least it is a well hidden fact) but we will attempt to guess its corresponding value.

n	$\nu_c(P_n \times P_n)$	$\nu_c(P_n \triangleleft P_n)$	$\nu_c(P_n * P_n)$
2	0	-1	-2
3	4	-2	0
4	2	3	2
5	10	14	8
6	20	23	18
7	32	38	32
8	46	59	52
9	62	78	74
10	82	107	102
11	104	134	132
12	130	167	168
13	156	206	210
14	186		
15	220		
16	256		

Table 1: Critical energy for the computed grids.

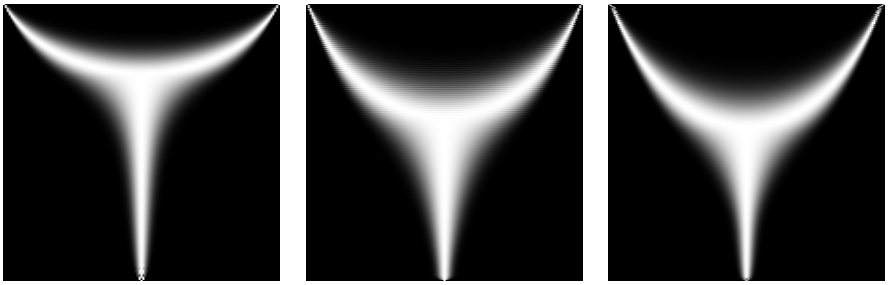

Figure 7: Density plot of polynomial for $P_{12} \times P_{12}$, $P_{12} \triangleleft P_{12}$ and $P_{12} * P_{12}$.

First the square grids, for these we have $m = 2n^2 - 2n$. The plot to the left of Figure 12 shows $1/\sqrt{2} - \nu_c/m$ vs $n^{-3/4}$ (this seems to be a good exponent if we keep to the simple rational ones). The points fit very well to a straight line for $n \geq 7$. The fitted line has equation $\sqrt{2}x$; a nice formula.

Then the triangular grids for which we have $m = 3n^2 - 4n + 1$. We stick to the exponent $3/4$ since it seems to be optimal. The middle plot of Figure 12 shows $2/3 - \nu_c/m$ vs $n^{-3/4}$. The fitted line has equation $3x/2$, another nice formula.

Finally the strong product grids for which $m = 4n^2 - 6n + 2$. Here we are in the dark when it comes to the asymptotic critical energy. If we however stick to the exponent $3/4$, which seems like a good choice, it is not very far-fetched to suggest that $\nu_c/m \approx 0.6$ in the limit (though we could choose from a whole interval between, say, 0.57 and 0.62). Actually we will take a bold approach and use $2 - \sqrt{2} \approx 0.5858$ as limit. The line is then chosen as $1/(2 - \sqrt{2})x \approx 1.7071x$ giving a remarkably good fit, see the right plot in Figure 12, while preserving the patterns of the two previous expressions. Alternatively, we could choose $3/5$ as limit value and $5/3$ as the coefficient for the straight line and still have

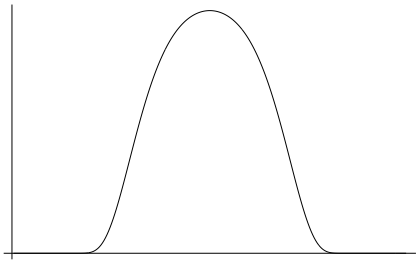

Figure 8: $P_{16} \times P_{16}$ at energy 236

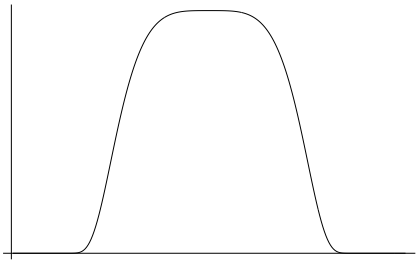

Figure 9: $P_{16} \times P_{16}$ at energy 256

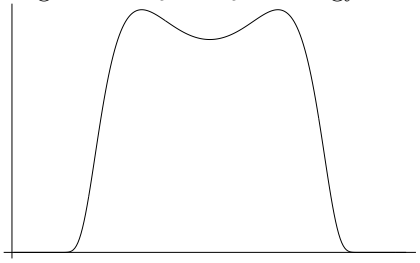

Figure 10: $P_{16} \times P_{16}$ at energy 276

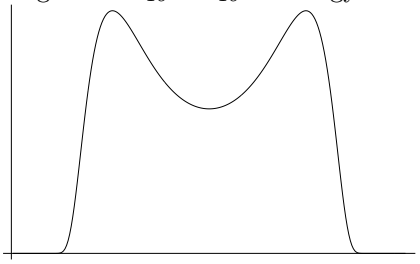

Figure 11: $P_{16} \times P_{16}$ at energy 296

a nicely fitted line, though not quite as good.

To conclude then, we suggest the following formulae describing the asymptotic behaviour of ν_c/m:

$$\begin{aligned} \frac{1}{\sqrt{2}} - \frac{\nu_c}{m} &\sim \frac{\sqrt{2}}{n^{3/4}} && \text{for } P_n \times P_n, \\ \frac{2}{3} - \frac{\nu_c}{m} &\sim \frac{3}{2n^{3/4}} && \text{for } P_n \triangleleft P_n, \\ 2 - \sqrt{2} - \frac{\nu_c}{m} &\sim \frac{1}{(2-\sqrt{2})n^{3/4}} && \text{for } P_n * P_n \end{aligned}$$

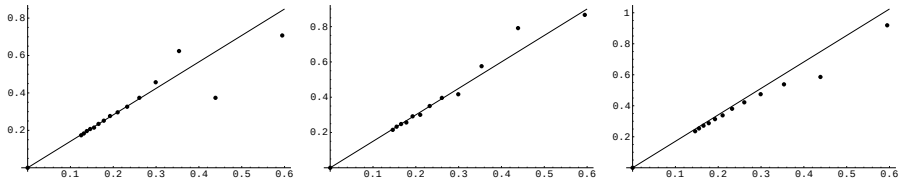

Figure 12: Difference between limit and ν_c/m vs $n^{-3/4}$ for $P_n \times P_n$, $P_n \triangleleft P_n$ and $P_n * P_n$

9 The largest coefficient

The polynomials $Z(G; x, y)$ are much too large to fit in this article, for $P_{16} \times P_{16}$ the polynomial has 61296 terms with a largest coefficient on 75 decimal digits. We turn our eyes to exactly these coefficients for the respective grids and state some conjectures about their location. Let $Z(G; x, y) = \sum_{i,j} a_{i,j}\, x^i\, y^j$ and

$Z(G;x,1) = \sum_i a_i\, x^i$. In Table 2 to 4 we show $\max a_{i,j}$ together with their normalised logarithm in base 2. Note by the way that this is asymptotically 1. We claim the following,

Conjecture 9.1. *For $P_n \times P_n$ we have*

$$\max a_{i,j} = \begin{cases} a_{-2,1} & \text{if } n \text{ is odd} \\ a_{-2,0} & \text{if } n \text{ is even} \end{cases} \qquad \text{for } n \geq 5$$
$$\max a_i = a_0, \quad \text{for } n \geq 2$$

Conjecture 9.2. *For $P_n \triangleleft P_n$ we have*

$$\max a_{i,j} = \begin{cases} a_{-6,1} & \text{if } n \text{ is odd} \\ a_{-5,0} & \text{if } n \text{ is even} \end{cases} \qquad \text{for } n \geq 6$$
$$\max a_i = \begin{cases} a_{-2} & \text{if } n \text{ is odd} \\ a_{-1} & \text{if } n \text{ is even} \end{cases} \qquad \text{for } n \geq 5$$

Conjecture 9.3. *For $P_n * P_n$ we have*

$$\max a_{i,j} = \begin{cases} a_{-6,1} & \text{if } n \text{ is odd} \\ a_{-6,0} & \text{if } n \text{ is even} \end{cases} \qquad \text{for } n \geq 5$$
$$\max a_i = a_{-2}, \quad \text{for } n \geq 3$$

To conclude, the maximum coefficient $a_{i,j}$ does not seem to be located at the energy having the biggest mass.

n	$\max a_{i,j}$	$\frac{\log \max a_{i,j}}{n^2}$
2	4	0.5
3	36	0.574436
4	2320	0.698744
5	695640	0.776319
6	973834528	0.829419
7	5673741811548	0.864642
8	141460632490772352	0.890206
9	1439075980932552 9848172	0.908735
10	609251459832403450801621 4752	0.922991
11	1044151547762547340724445589341 0332	0.933949
12	7347100551219071743055840070535 2992061504	0.942738
13	2084491309826665170718150803756677204192216177596	0.949777
14	2409733602176656071484869610572737554845048184520299591 68	0.955596
15	1120685211229412539966936593770495599384965101147147767531 05770300	0.960399
16	2113701632337937835777788628823007258034310881837158929147348676143354132 16	0.964463

Table 2: Maximum coefficients for $P_n \times P_n$.

n	$\max a_{i,j}$	$\frac{\log \max a_{i,j}}{n^2}$
2	4	0.5
3	33	0.560488
4	2196	0.693791
5	615548	0.76926
6	854319200	0.824172
7	4845797273700	0.859997
8	121168825318878960	0.886716
9	1215767959070393331 0814	0.905732
10	5148986176319146744322186888	0.920563
11	8754829383356913957235691401458166	0.931849
12	6159591604093920040219892881711672376 5264	0.940972
13	1738780812314585269331291365085312353501653534080	0.948229

Table 3: Maximum coefficients for $P_n \triangleleft P_n$.

n	$\max a_{i,j}$	$\frac{\log \max a_{i,j}}{n^2}$
2	6	0.646241
3	38	0.583103
4	2320	0.698744
5	585380	0.766361
6	781838700	0.820619
7	4419808185964	0.857288
8	108087171073133240	0.88414
9	1084820071501505823 1932	0.903702
10	4547619894022801720048409280	0.918772
11	7735977455878602754688545850442768	0.930374
12	5411740978430516100396063210324190305 2286	0.939675
13	1528323994171524803264414984644676340178019401444	0.947127

Table 4: Maximum coefficients for $P_n * P_n$.

References

[B+86] D. Babic et al. The matching polynomial of a polygraph. *Discrete Appl. Math.*, 15:11–24, 1986.

[Bak94] G.A. Baker. The Markov property method applied to Ising model calculations. *J. Stat. Phys.*, 77:955–976, 1994.

[Bea96] P. Beale. Exact distribution of energies in the two-dimensional Ising model. *Phys. Rev.*, 76:78–81, 1996.

[Big77] N. Biggs. *Interaction models.* Cambridge University Press, 1977.

[Cip87] B.A. Cipra. An introduction to the Ising model. *Amer. Math. Monthly*, 94:937–959, 1987.

[HR97] R. Häggkvist and A. Rosengren. Some conjectures concerning the coefficients in the Ising polynomial. Manuscript, 1997.

[HR+99] R. Häggkvist, A. Rosengren, et al. Microcanonical Monte Carlo studies of the coefficients in the Ising polynomial. Manuscript, 1999.

[Kas67] P.W. Kasteleyn. Graph theory and crystal physics. In *Graph theory and theoretical physics.* Academic press, 1967.

[LB99] D.A. Lavis and G.M. Bell. *Statistical mechanics of lattice systems.* Springer, 1999.

[Lun99] P.H. Lundow. Compression of transfer matrices. Technical Report 5, Department of mathematics, Umeå university, 1999.

[Lun00a] P.H. Lundow. Multiprecision integers: a Fortran 90 module. Manuscript, 2000.

[Lun00b] P.H. Lundow. Polynomials: a Fortran 90 module. Manuscript, 2000.