Generating functions for enumerating self-avoiding rings on the square lattice

J. Phys. A: Math. Gen. 13 (1980) 3713–3722. Printed in Great Britain

Generating functions for enumerating self-avoiding rings on the square lattice

I G Enting†

Department of Physics, Northeastern University, Boston, Massachusetts 02115, USA

Received 28 April 1980

Abstract. It is shown that generating function techniques provide an efficient means of enumerating the number of self-avoiding rings (polygons) on the square lattice. The techniques can be applied to a number of related problems in lattice statistics and statistical mechanics. The enumeration has been extended to polygons of up to 38 steps.

1. Introduction

The enumeration of polygons on various regular lattices is not only an interesting problem in its own right but is also of considerable importance in the statistical mechanics of lattice models. The results of polygon counts are an important part of many derivations of exact series expansions and, less directly, Domb (1969, 1972, 1974) has used the combinatorial properties of polygons as a basis for estimating the properties of lattice models.

The lattice statistics of polygons (or self-avoiding rings) is related to that of self-avoiding walks (Hammersley 1957, Sykes 1961) and thus to lattice models of polymers (Hiley and Sykes 1961) and to the $n = 0$ limit of the n-vector model (de Gennes 1972).

The main object of the present paper is to use the polygon enumeration problem as an example of the use of generating function techniques in problems which have 'connectivity' constraints when they are treated using transfer matrix techniques. Connectivity constraints do not occur in the Ising model but do occur in systems such as the quantum XY model (because of the time-ordering operator).

The present work grew out of the author's work on the finite lattice method of series expansions (de Neef and Enting 1977, Kim and Enting 1979, Enting and Guttmann 1980). The polygon enumeration differs from these earlier calculations in several important ways.

The finite lattice method involves three stages, two formal and one computational.

(i) The series expansion has to be formally expressed as a linked-graph expansion.

(ii) Formally, the linked-graph expansion has to be re-expressed as a sum of contributions from finite rectangles.

(iii) The contributions for finite rectangles must be computed and then combined in the appropriate way.

Of these steps: (i) the linked-graph formulation was known for many models long

† Present address: CSIRO, Atmospheric Phys., PO Box 77 Mordialloc Vic. 3195, Australia.

before the finite lattice method was first described; (ii) the resummation is given once and for all by Enting (1978); (iii) the contributions from rectangles can be computed efficiently by using techniques based on a transfer matrix formalism.

In the enumeration of polygons we are only concerned with connected graphs, and so the formal aspects of the combinatorics involve ensuring that the generating functions exclude all contributions from two or more co-existing polygons. This change modifies the type of resummation that is required, as indicated in § 3. Section 2 shows how simple classes of generating function can be used to enumerate small polygons, and § 4 extends these concepts to provide rules for handling the general case. Section 5 shows how these rules can be used as the basis of an explicit construction of the generating functions.

2. Examples of polygon generating functions

In graph-theoretical terms the phrase 'n-step polygon' specifies a unique graph. The numbers in which we are interested are u_n, which are defined by taking the limit, for large N, of N^{-1} times the number of ways that an n-step polygon graph can be embedded on a square lattice of N sites. An alternative definition is to take an origin near the centre of an arbitrarily large square lattice, and define $2nu_n$ as the number of walks from the origin which return after n steps without any other self-intersections (self-avoiding rings in the terminology of Sykes *et al* 1972).

In what follows, each of the u_n types of embedding will be regarded as a distinct polygon. The term 'polygon' is being used in a geometrical sense or as a shorthand for 'realisation of an embedding of the polygon graph'. Two polygons are regarded as the same if they can be mapped onto one another by translations. Rotations and reflections of asymmetric polygons are regarded as distinct.

The generating function for polygons on the square lattice is

$$U(x)=\sum_n u_n x^n. \tag{2.1}$$

This section will work with generating functions for particular classes of polygon occurring on rectangular subgraphs of the square lattice, the rectangles having dimensions 3 sites $\times\, n$ sites. Figure 1 shows the 3×6 rectangle together with some of the polygons.

If we draw a vertical line such as the broken line 'p' in figure 1(a) and then specify the intersection of this line with a polygon then, conditional on this fixed intersection, the self-avoidance constraint acts independently on the left and right of p. The number of polygons with a given intersection is simply the number of allowed left-hand sides (a_n with n steps) multiplied by the number of allowed right-hand sides (b_m with m steps). We can construct generating functions $A(x)=\Sigma a_n x^n$ and $B(x)=\Sigma b_n x^n$ for each side. The number of polygons with the fixed cross-section and r steps is just $\Sigma_j\, a_r b_{j-2-r}$, which is the coefficient of x^r in the combined generating function $x^2 A(x)B(x)$. (The -2 and the x^2 are to include the two steps that intersect line p.)

The transfer matrix formalism is based on the same idea that independence of constraints enables us to obtain a combined count by multiplying independent subcounts, or a combined generating function by multiplying partial generating functions.

If we draw two lines such as 'p' and 'q' in figure 1(a), then the constraints act independently in each of the three regions. If we have partial generating functions for

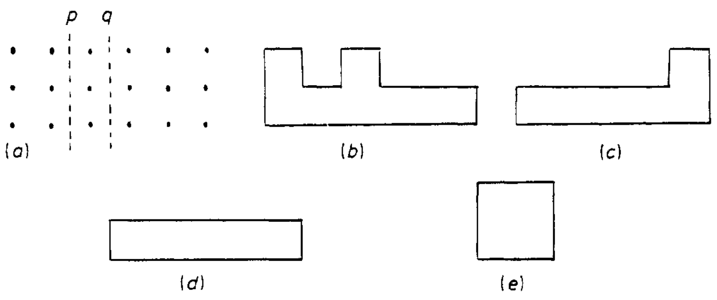

Figure 1. (a) $3\times n$ rectangle. (b), (c), (d), (e) are four polygons that can be embedded in the rectangle in 1, 1, 2 and 4 ways respectively. In (a) the lines p and q divide the rectangle into three disjoint regions so that, given the specification of how a polygon crosses these lines, the self-avoidance constraint acts independently in each region.

all polygon segments to the left of 'p', we can construct corresponding partial generating functions for all polygon segments to the left of 'q', simply by running through all allowed cross sections on p together with all allowed arrangements of steps between p and q. The important points are that if the cross section q is fixed, then all the manipulations are independent of what happens to the right of q and the arrangements to the left of p contribute only through partial generating functions for each cross section on p.

We will now construct the generating function for all ways of putting polygons on the $3\times n$ rectangle, counting only polygons that span the full (n-site) length of the rectangle.

For each vertical line there are three possible cross sections—we call these α, β and β'—which are shown in figure 2. If we draw the vertical line through the leftmost squares of figure 1(a), then the three partial generating functions for the left-hand sides are $A_\alpha^{(1)}=x^2$ and $A_\beta^{(1)}=A_{\beta'}^{(1)}=x$. (The superscript denotes the position of the cross section line.)

In general $A_\beta^{(n)}=A_{\beta'}^{(n)}$ by symmetry. We wish to obtain the $A^{(n+1)}$ from the $A^{(n)}$. The various combinations of bonds are shown in figure 3, leading to the results

$$\begin{aligned}A_\alpha^{(n+1)}&=x^2A_\alpha^{(n)}+x^3(A_\beta^{(n)}+A_{\beta'}^{(n)})\\&=x^2A_\alpha^{(n)}+2x^3A_\beta^{(n)},\end{aligned}\tag{2.2}$$

$$A_\beta^{(n+1)}=A_{\beta'}^{(n+1)}=x^3A_\alpha^{(n)}+x^2A_\beta^{(n)}.\tag{2.3}$$

The generating function for all ways of having a polygon span the length of a $3\times n$ rectangle is obtained by considering the ways of closing each loop, and leads to

$$G_{3,n}=x^4A_\alpha^{(n-1)}+2x^3A_\beta^{(n-1)}.\tag{2.4}$$

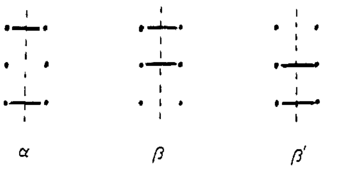

Figure 2. The three distinct ways in which a vertical section line can intersect a polygon on the $3\times n$ rectangle.

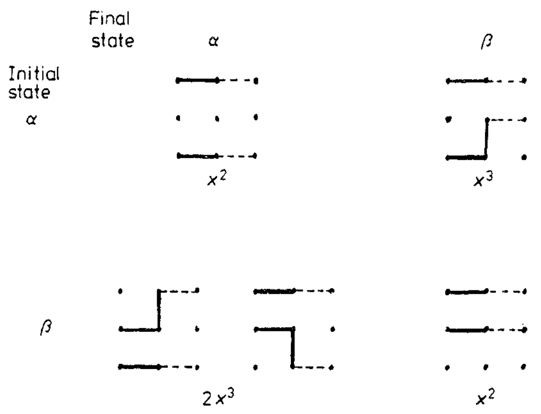

Figure 3. The various combinations of steps that can occur between consecutive section lines. The heavy shaded steps are regarded as being added to the segments to the left. This corresponds to multiplying the corresponding partial generating functions by the algebraic factors indicated.

Table 1 iterates these equations to obtain the generating functions $G_{3,n}$ for $n = 2, 3, 4$.

Table 1. The partial generating functions used in constructing generating functions for $3 \times n$ rectangles. The entries are obtained by successive applications of equations (2.2), (2.3) and (2.4).

n	$A_\alpha^{(n)}$	$A_\beta^{(n)}$	$G_{3,n+1}$
1	x^2	x	$2x^4+x^6$
2	$3x^4$	x^3+x^5	$2x^6+5x^8$
3	$5x^6+2x^8$	x^5+4x^7	$2x^8+13x^{10}+2x^{12}$

These generating functions include contributions from polygons that can fit within the $2\times n$ rectangles, each of which can occur in the $3\times n$ rectangle in two ways. If we define $g_{m,n}$ as the generating function for polygons occurring in the $m\times n$ rectangle but not in any smaller rectangle, then

$$\text{(i)}\quad g_{2,n} = x^{2n}, \tag{2.5}$$

$$\text{(ii)}\quad g_{3,n} = G_{3,n} - 2x^{2n}, \tag{2.6}$$

$$\text{(iii)}\quad g_{n,m} = g_{m,n}, \quad . \tag{2.7}$$

and since each polygon will contribute to one and only one of the $g_{m,n}$ (and in only one way) we can put

$$\begin{aligned} U(x) &\approx g_{2,2}+2g_{2,3}+2g_{2,4}+2g_{2,5}+g_{3,3}+2g_{3,4} \\ &= 2G_{3,4}+G_{3,3}-2x^8+2x^{10}+2x^4 \\ &= x^4+2x^6+7x^8+28x^{10}+4x^{12}. \end{aligned} \tag{2.8}$$

Since any polygon graph of n steps can be embedded in some rectangle of perimeter n, the approximation (2.8) correctly enumerates all polygons of ten or fewer steps.

3. General rules for combining rectangles

In the previous section we introduced g_{mn}, the generating function for polygons that fitted into an $n \times m$ rectangle but which did not fit into any smaller rectangle. Since the smallest rectangle is uniquely defined for each polygon, and since each polygon fits into its smallest rectangle in precisely one way (since rotations are regarded as giving a different polygon), we have

$$U(x) = \sum_{m,n} g_{m,n} \tag{3.1}$$

or, approximately,

$$U(x) \approx U(x)^{(k)} = \sum_{\substack{m,n \\ m+n \leqslant k}} g_{m,n}. \tag{3.2}$$

The approximation $U(x)^{(k)}$ correctly enumerates all polygons with $2k-4$ or fewer steps.

We can use the symmetry $g_{m,n} = g_{n,m}$ to express (3.2) as

$$U(x)^{(k)} = \sum a_{mn} g_{m,n}, \tag{3.3a}$$

$$a_{mm} = 1, \qquad 2m \leqslant k, \tag{3.3b}$$

$$a_{m,n} = 2, \qquad m < n, \qquad m+n \leqslant k, \tag{3.3c}$$

$$a_{m,n} = 0, \qquad \text{otherwise.} \tag{3.3d}$$

In practice we do not calculate $g_{m,n}$ but use $G_{m,n}$, the generating function enumerating all ways of having polygons which fit in an $m \times n$ rectangle but not in any rectangle of length less than n. (The reasons for this choice are given in the following section.)

A polygon that contributes to $g_{m,n}$ will contribute to $G_{p,n}$ in $p-m+1$ ways (each different vertical location contributes):

$$G_{p,n} = \sum_{m \leqslant p} (p-m+1) g_{m,n}. \tag{3.4}$$

This relation can be inverted to give

$$g_{m,n} = G_{m,n} - 2G_{m-1,n} + G_{m-2,n}. \tag{3.5}$$

Inserting (3.5) into (3.3a) with $k = 2p+1$,

$$U(x)^{(2p+1)} = \sum b_{mn} G_{m,n}, \tag{3.6a}$$

$$b_{mm} = 1, \qquad m \leqslant p, \tag{3.6b}$$

$$b_{m-2,m} = -1, \qquad m \leqslant p, \tag{3.6c}$$

$$b_{m,n} = 2, \qquad m \leqslant n, \qquad m+n = 2p+1, \tag{3.6d}$$

$$b_{m,n} = -2, \qquad m \leqslant n, \qquad m+n = 2p, \tag{3.6e}$$

$$b_{m,n} = 0, \qquad \text{otherwise.} \tag{3.6f}$$

The summations in (3.5), (3.6) were obtained using results given by Enting (1978), but they can be verified by explicit summation.

4. General formulation

The generating functions $G_{m,n}$ are constructed by successively moving section lines across finite rectangles, constructing partial generating functions for the polygon segments characterised by particular cross sections. By running through all combinations of possible steps occurring between two successive positions of the section line, partial generating functions for various classes of graph can be constructed.

When running through the possible arrangements of steps, a number of constraints need to be observed:

(i) a weight of x must be associated with each step;

(ii) each vertex must be of order 2 or 0;

(iii) each graph must span the rectangle from end to end;

(iv) the graphs constructed must all consist of only one connected component.

Constraint (iv) is the one that causes most difficulty. One class of two-component graph is illustrated in figure 4. We can exclude these by ensuring that all graphs span the length of the rectangle (hence constraint (iii)) and that we never have two distinct components side by side. As we move the section line, the segments of graphs will consist of a number of disjoint walks that must, in the end, be connected to each other if a polygon is to be produced.

There are two distinct ways in which two loops can be placed relative to one another—side by side or nested; these are shown in figure 5(a). Figure 5(b) shows how these loops can be connected to make a single loop. Only connections equivalent to these are allowed. Connections equivalent to those shown in 5(c) produce two separate components and so must be excluded.

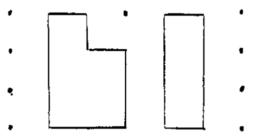

Figure 4. One way in which two separate components could occur. This is prevented by forcing all polygons to run the full length of the rectangle.

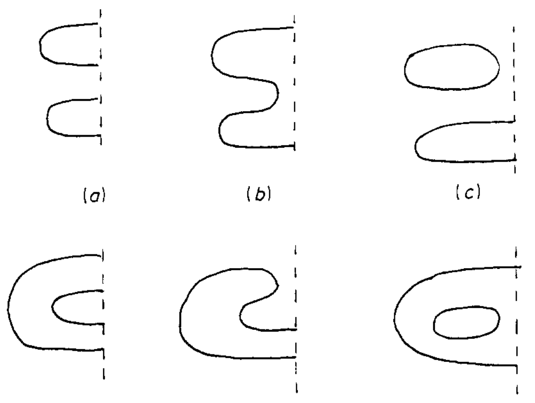

Figure 5. (a) The possible ways in which a pair of loops can be positioned relative to one another. (b) The ways in which each combination can be connected. (c) The types of connection that lead to forbidden, two-component graphs.

The necessary constraint is that no loop can be closed on itself so long as the section line cuts any other path. The constraint (iii) is introduced to ensure that such other paths as may occur will cut the section line. It is the introduction of constraint (iii) that leads to the use of asymmetric generating functions $G_{m,n}$.

To exclude loops which close on themselves we have to label the cross sections. One possible scheme would be to have a unique label associated with each loop but, because of the two-dimensional nature of the system, there is an alternative scheme that is more convenient when used in a computer program. The ends of loops are assigned the labels '1' and '2', depending on whether the end is the lower or the upper end for that loop. (The label '0' is used in the program to denote an edge not occupied by part of any loop.) Figure 6 expresses the connections shown in figure 5 in terms of these labels.

In class (*a*) a '2' is joined to a '1' immediately above it and the intersection and labels need no longer be considered. In (*b*) two '1's are joined, and in (*c*) two '2's. In each case the far end of the inner loop must be relabelled. Because of the possibility of having a third loop (shown as a broken curve) (or even more such loops) nested, the label to be changed is the one that is reached after passing an equal number of '1's and '2's. In 6(*b*) we change the first unmatched '2' (working upwards from the join). In 6(*c*) we change the first unmatched '1' (working downwards).

Figure 6(*d*) shows the case of a '1' being joined to a '2' immediately above it. This is forbidden if any other loops are present (see figure 5(*c*)), but if there are no other loops then the contribution is added to a running total for $G_{m,n}$, with n equal to the current length. The contribution does not contribute to building $G_{m,n+1}$ since we are applying constraint (iii) so as to exclude the type of graph shown in figure 4.

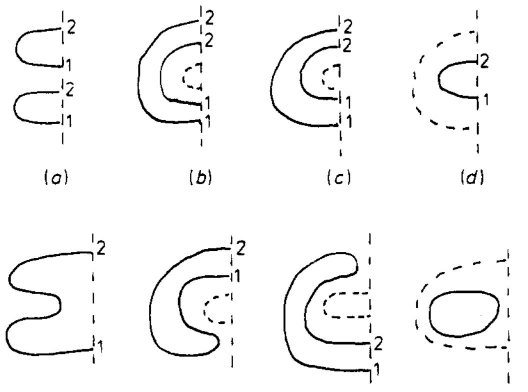

Figure 6. The description of the allowed connections from figure 5(*b*) in terms of the '1', '2' labelling. Case (*d*) does not contribute to subsequent partial generating functions but, if no other loops are present, it does contribute to the $G_{m,n}$ of the appropriate length.

5. Square lattice generating functions

The calculations can be shortened by a considerable amount by noting that the section lines used in § 2 (see figure 1) need not be straight. We do not have to build up the rectangle one column at a time—we need only add one site at a time. The form of the section line is as shown in figure 7, where adding the circled site in each case takes us on to the next case. The circled sites are regarded as having two edges 'in' (those which cut

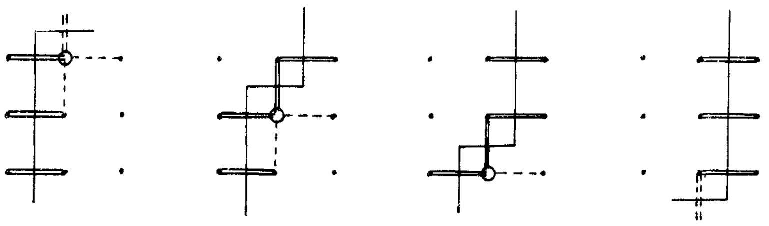

Figure 7. Successive section lines corresponding to adding one site at a time to a $3\times n$ rectangle. Adding the circled site in each diagram corresponds to moving the section line (the single solid line) into the position shown in the succeeding diagram. The double lines indicate the edges cut by the section line (the double broken edges are dummy edges introduced for uniformity). New steps added as the circled site is added lie along the single broken lines.

the old section line but not the new section line) and two edges 'out' (those which cut the new section line but not the old). Polygon steps on the 'in' edges are not regarded as being connected until the circled site is added. Adding a site can thus connect only two loop segments, and so the cases shown in figures 5 and 6 are the only ones that need be considered.

In figure 7 the single solid line is the section line, and the double lines are the edges in the rectangle that are cut by the section line (the double broken edges are 'dummy' edges that never include steps of the polygon, but are considered so that all vertices may be treated equivalently). The output edges (on which steps can be added in the process of going to the next position of the section line) are shown as single broken lines.

Using '0' (no step), '1' (lowermost end of a loop) and '2' (uppermost end of a loop) notation, figure 8 shows all the possible states of the 'in' edges and the corresponding states of the 'out' edges and the basis of the constraints picture in figure 6. The transformation back to the beginning of figure 8 is simply a renumbering of the edges cut by the section line so that the dummy edge is at the top.

Once the construction is reduced to the cases considered in figure 8, the procedure is easy to implement on a digital computer. For large counts it was necessary to use the arithmetic of residues modulo various primes ($2^{15}-19$, $2^{15}-49$, $2^{15}-51$ and $2^{15}-55$), the integer counts being reconstructed from the residues at the completion of the calculations.

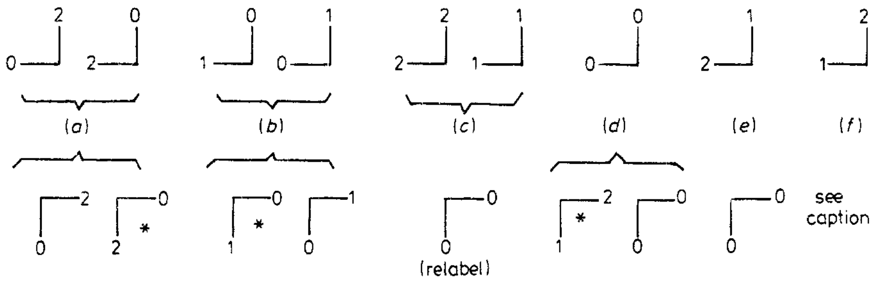

Figure 8. The various states of the 'in' edges for each new site are shown in the upper row. The corresponding 'out' states are shown in the lower row. Configurations marked with a * are forbidden if the site is at the bottom of a column. Configuration (*f*) cannot contribute to any larger rectangles, but does contribute to the current $G_{m,n}$ if no other steps cut the section line.

Table 2 shows the numbers of n-step polygons for $n \leq 38$. This is a significant improvement on the results for $n \leq 26$ given by Sykes *et al* (1972). For each n, table 2 also shows the maximum width (the number of sites) of rectangle that need be considered to obtain polygon counts for that number of steps. (For example at ten steps we need consider only polygons of width three as seen in § 2.) The bent section lines shown in figure 7 cut $m+1$ edges in a rectangle m sites wide. The final column in table 2 shows the number of ways in which the $m+1$ edges can form the ends of self-avoiding loops. This is the number of distinct partial generating functions that need be considered. It is the growth in this number that limits the size of polygons that can be enumerated using these techniques. Rectangles 11, 12 and 13 sites wide would need 15 511, 41 835 and 113 634 partial generating functions respectively.

Table 2. The numbers of embeddings of n-step polygons on the square lattice. The maximum width (specified by number of sites) is for the widest rectangle needed for the particular number of steps. The number of partial generating functions (PGFs) is the number used in constructing rectangles of that width.

Number of steps	Number of polygons	Maximum width	Number of PGFs
4	1	2	4
6	2	2	
8	7	3	9
10	28	3	
12	124	4	21
14	588	4	
16	2 938	5	51
18	15 268	5	
20	81 826	6	127
22	449 572	6	
24	2 521 270	7	323
26	14 385 376	7	
28	83 290 424	8	835
30	488 384 528	8	
32	2 895 432 660	9	2 188
34	17 332 874 364	9	
36	104 653 427 012	10	5 798
38	636 737 003 384	10	

6. Conclusions

The main point of this paper is to illustrate the way in which generating function techniques can be applied to enumerations in lattice statistics even when strong non-local connectivity constraints apply. The self-avoidance constraint allows a conditional independence. It is not as strong as the purely local conditional independence occurring in Ising-like models (models which are described in statistical terms as Markov Random fields, see Dobrushin (1968) and references cited by Enting and Welberry (1978)) and so the polygon enumeration is more complicated than deriving Ising model series.

The enumeration techniques are relevant for a number of other interesting problems in statistical physics.

(i) Site percolation: each cluster is bounded externally by a polygon on the dual lattice and so the connection is quite direct.

(ii) Colouring polynomials: the connectivity constraint requires a more general labelling but the principles are similar.

(iii) Quantum XY model: the time ordering operator introduces connectivity constraints.

(iv) Certain subsets of the partial generating functions sum to give generating functions for self-avoiding walks whose ends are on the edge of a bounded square lattice, but the use of these generating functions seems to give at most a slight improvement over direct enumeration techniques.

(v) In principle the polygon enumeration and the various generalisations could all be applied to triangular lattice systems.

The generating function techniques have given a greatly improved means of enumerating square lattice polygons. It is to be expected that similar improvements will be achieved when some of the other problems mentioned above are tackled using these techniques.

References

Dobrushin R L 1968 *Theory Prob. Appl.* **13** 197–224
Domb C 1969 in *Stochastic Processes in Chemical Physics* ed Schuler (Wiley) pp 229–59
—— 1972 *J. Phys. C: Solid St. Phys.* **5** 1417–28
—— 1974 *J. Phys. A: Math., Nucl. Gen.* **7** 1335–48
Enting I G 1978 *J. Phys. A: Math. Gen.* **11** 563–8
Enting I G and Guttmann A J 1980 *J. Phys. A: Math. Gen.* **13** 1043–50
Enting I G and Welberry T R 1978 *Suppl. J. Appl. Prob.* **10** 65–72
de Gennes P G 1972 *Phys. Lett.* **38A** 339
Hammersley J M 1957 *Proc. Camb. Phil. Soc.* **53** 642–5
Hiley R J and Sykes M F 1961 *J. Chem. Phys.* **34** 1533–7
Kim D and Enting I G 1979 *J. Comb. Thry.* A **26** 327–36
de Neef T and Enting I G 1977 *J. Phys. A: Math. Gen.* **10** 801–5
Sykes M F 1961 *J. Math. Phys.* **2** 52–62
Sykes M F, McKenzie D S, Watts M G and Martin J L 1972 *J. Phys. A: Gen. Phys.* **5** 661–6