Square lattice self-avoiding walks and polygons.

A J Guttmann,
Department of Mathematics and Statistics, The University of Melbourne
Parkville, Vic. 3010, Australia
and A R Conway,
Silicon Genetics,
2601 Spring St., Redwood City, CA 94063, U.S.A

September 17, 2001

Abstract

We give an algorithm for the enumeration of self-avoiding walks on the (anisotropic) square lattice. Application of the algorithm on a 1024 processor Intel Paragon supercomputer resulted in a 51 term series. For (isotropic) square lattice self-avoiding polygons, a related algorithm has produced a 90 term series. Careful analysis provides compelling evidence for simple rational values of the exponents in both the dominant and subdominant terms in the asymptotic form of the coefficients. We also advance compelling arguments—but not a proof—that the generating function for SAW is *not* differentiably finite. The corresponding result for SAP has recently been proved.

1 Introduction

The enumeration of self-avoiding walks (SAW) and polygons (SAP) has, for half a century, been among the most challenging problems in enumerative combinatorics. This is partly because they are such simple problems to define, yet have resisted all theoretical attacks designed to calculate their generating functions.

We first review briefly the basic facts and conjectures about SAW and SAP that will be used in the remainder of the paper. Let $\mathcal{L}$ be some regular d-dimensional lattice, though the detailed numerical results we will provide later in this paper apply particularly to the two-dimensional case. Then an *N-step self-avoiding walk* (SAW) ω on $\mathcal{L}$ is a sequence of *distinct* points $\omega_0, \omega_1, ..., \omega_N$ in $\mathcal{L}$ such that each point is a nearest neighbor of its predecessor. We assume all walks to begin at the origin ($\omega_0 = 0$) unless stated otherwise.

First we define the quantities relating to the *number* (or "entropy") of SAWs. Let c_N [resp. $c_N(\mathbf{x})$] be the number of N-step SAWs on $\mathcal{L}$ starting at the origin and ending anywhere [resp. ending at $\mathbf{x}$]. Then c_N and $c_N(\mathbf{x})$ are believed to have the asymptotic behavior

$$c_N \sim \mu^N N^{\gamma-1} \tag{1.1}$$

$$c_N(\mathbf{x}) \sim \mu^N N^{\alpha-2} \qquad (\mathbf{x} \text{ fixed} \neq 0) \tag{1.2}$$

as $N \to \infty$; here μ is called the connective constant of the lattice, and γ and α_{sing} are *critical exponents*. The critical exponents are believed to be universal among lattices of a given dimension d. Rigorous results concerning the asymptotic behavior of c_N and $c_N(\mathbf{x})$ are described in Sec. 2, and further in [38, 24, 25, 26].

If the end-point $\mathbf{x}$ of the walk is adjacent to the origin, then with one additional step the SAW can form a closed loop. Such unrooted loops, equivalent up to a translation, are known as *self-avoiding polygons* and the cardinality of the number of N-edge SAP is denoted p_N.[1] It is believed that p_N has the asymptotic behaviour

$$p_{2N} \sim \mu^{2N} N^{\alpha-3} \tag{1.3}$$

as $N \to \infty$. (The difference between (1.2) and (1.3) is just due to the fact that in the former case we are dealing with *rooted* loops, and in the latter case with *unrooted* loops.)

We will frequently refer to the generating functions for SAW and SAP, $C(x) = \sum c_n x^n$ and $P(x) = \sum p_n x^n$ respectively.

Next we define several measures of the *size* of an N-step SAW:

- The *squared end-to-end distance*
$$R_e^2 \;=\; \omega_N^2 \;. \tag{1.4}$$

- The *squared radius of gyration*
$$R_g^2 \;=\; \frac{1}{2(N+1)^2} \sum_{i,j=0}^{N} (\omega_i - \omega_j)^2 \;. \tag{1.5}$$

- The *mean-square distance of a monomer from the endpoints*
$$R_m^2 \;=\; \frac{1}{2(N+1)} \sum_{i=0}^{N} \left[\omega_i^2 \;+\; (\omega_i - \omega_N)^2\right] \;. \tag{1.6}$$

We then consider the mean values $\langle R_e^2 \rangle_N$, $\langle R_g^2 \rangle_N$ and $\langle R_m^2 \rangle_N$ in the probability distribution which gives equal weight to each N-step SAW. Very little has been proven rigorously about these mean values, but they are believed to have the leading asymptotic behavior

$$\langle R_e^2 \rangle_N,\, \langle R_g^2 \rangle_N,\, \langle R_m^2 \rangle_N \;\sim\; N^{2\nu} \tag{1.7}$$

as $N \to \infty$, where ν is another (universal) critical exponent, and is widely believed to be exactly $\nu = 3/4$ for two-dimensional SAW. Moreover, the amplitude ratios

$$A_N \;=\; \frac{\langle R_g^2 \rangle_N}{\langle R_e^2 \rangle_N} \tag{1.8}$$

$$B_N \;=\; \frac{\langle R_m^2 \rangle_N}{\langle R_e^2 \rangle_N} \tag{1.9}$$

$$C_N \;=\; \frac{\langle R_g^2 \rangle_N}{\langle R_m^2 \rangle_N} \tag{1.10}$$

[1] For the hypercubic lattice, among others, $p_{2N+1} = 0$.

are expected to approach universal values in the limit $N \to \infty$. Many other universal amplitude combinations (notably involving SAP) are discussed in [3]. We will have little more to say here about these measures of size, except to point out that they all have the same leading asymptotic behaviour. They are of interest for several reasons, perhaps most particularly because they are the quantities that can most readily be compared to measurements of polymer size. Apart from their intrinsic mathematical interest, the elucidation of the SAW generating function has, for a quarter of a century been arguably the most powerful method for studying the critical (long-chain) behaviour of polymer molecules in a good solvent [38, 30]. Indeed, it is this application that was responsible for perhaps the first paper on this problem, the posthumous work of Orr [42], describing work carried out in 1946.

It has also become a paradigm of algorithm design for other graph-enumeration problems arising in combinatorics and statistical mechanics. In this paper we give details of a radical extension of the square lattice SAW series, from 39 steps to 51 steps[2], and we use it to provide compelling evidence that the first non-analytic correction term to the asymptotic form of the coefficients is $O(\mu^n/n^{\frac{3}{2}})$ as predicted by Nienhuis [40]. While most of our activity is focussed on the singular behaviour of the generating function at $x = 1/\mu$, we also determine the nature of the singularity at another singular point, $x = -1/\mu$.

The first substantial enumeration of SAW was carried out in 1959 by Fisher and Sykes [11], entirely by hand, based on Sykes' Counting Theorem [46], which expresses SAW in terms of less numerous graphs. This resulted in the calculation of the first 16 terms in the square-lattice SAW generating function. The start of the modern era, using computers, appears to have been in 1961, when Hiley and Sykes [28] obtained 18 terms. In 1972 a 24 term series [47] was obtained, again using the chain counting theorem. In 1987 the series was extended [13] by three terms using a direct enumeration algorithm, essentially a back-tracking algorithm, while in 1991 a dimerisation algorithm [19] was used to extend the series by two further terms. In 1992 the series was extended [35] by an additional term, to 30 terms, using a similar algorithm, and later that year, using what was then one of the world's largest supercomputers, the Thinking Machines CM-2 with 65536 processors, the series was extended [39] to 34 terms, in a program that took about 100 hours to run.

All the above calculations were based on algorithms of exponential complexity, with the time taken growing asymptotically as λ^n, where $\lambda \approx 2.638$, the connective constant for SAW on the square lattice. That is to say, the time grows in direct proportion to the number of configurations.

The enumeration of SAP was primarily a by-product of SAW enumeration in earlier times, with identical computational complexity. In 1977 Enting [7] developed a powerful new technique, based on the idea of applying transfer matrix methods to finite lattices. In this way the complexity is reduced to $3^{n/4}$, at the cost of greatly increased program complexity and computer memory requirements. Nevertheless, Enting extended the series to SAP of perimeter 38. In subsequent years this method has been refined, and computer technology has improved so that 46 terms were obtained [8], then 56 terms [9], then 70 terms [10], and very recently with a refinement of the algorithm due to Jensen, 90 terms have been obtained [31], using an algorithm with complexity 1.2^n.

In 1993 we [5] introduced an analogous algebraic technique for SAW enumeration, utilising the finite-lattice method (FLM), coupled with transfer matrices. This is the same basic idea introduced by Enting [7] for SAP, but its application to SAW is much more difficult. The

[2] A preliminary account of this work appeared in [6]

algorithm is challenging to implement efficiently, and requires large amounts of physical memory, but affords the advantage of an exponential increase in efficiency over previous algorithms, being of complexity κ^n, where $\kappa = 3^{\frac{1}{4}} \approx 1.316$, as for the original implementation of the FLM for SAP.

The first implementation of this algorithm produced 35 step SAW on an IBM 6000/530 workstation with 256MB of memory, in approximately 50 hours. Later we obtained 39-step SAWs, using an IBM 3090 with more than 500MB of memory.

In the present study, we have parallelised the algorithm in order to run it on a 1024 processor Intel Paragon XP/S 150 supercomputer. While the parallelisation is not, in principle, too difficult, an efficient implementation that balances the load across all processors is more demanding. The underlying algorithm is as described in [5], and the details of the parallelisation scheme are described below.

The calculation took about 12 hours, required about 10GB of memory, and was performed modulo different primes (seven times), with the final results being reconstituted using the Chinese Remainder Theorem. The scale of the calculation can be gauged by the fact that something like 5TB of data is moved between processors on each run. Additional processing was required to construct the final series. The series coefficients of the SAW generating function, $c_0 \cdots c_{51}$ are given in Table 1.

In the next section we outline the rigorous results known for SAW and SAP, while in Section 3 we give details of the improved FLM for SAP. In Section 4 we outline the modifications to the FLM method for SAW which are necessary for efficient parallelization, and in Section 5 we analyse the SAP and SAW series, and provide compelling evidence in support of Nienhuis' prediction of the values of the critical exponents γ and α [40]. We also address the controversial question of the correction-to-scaling exponent Δ [16] [44] [2] [4], (defined in (5.3)) and investigate the possible analytic structure of the SAP and SAW generating functions [18].

2 Rigorous Results

There is surprisingly little known rigorously about the behaviour of the generating functions, and much of what is known was proved half a century ago. Indeed, it was not until 1990 that the seemingly obvious result $c_{n+1} > c_n$ was proved [41].

In 1954 Hammersley and Morton [22] proved that

$$\lim_{n\to\infty} \frac{1}{n} \log c_n = \log \mu \tag{2.1}$$

exists, with $1 \leq \mu < \infty$, and further, that the limit is approached from above—that is to say, for each n, we have the inequality $\frac{1}{n} \log c_n > \log \mu$. The proof relies on a lemma on sub-additive sequences due to Hille and Phillips [29]. It is widely used to establish the existence of such a limit for a variety of problems in enumerative combinatorics. Concatenating SAW leads immediately to the submultiplicative inequality $c_{n+m} \leq c_n c_m$. As $c_n \geq 1$ we may take logarithms, converting the inequality to a sub-additive inequality, and applying the lemma of Hille and Phillips we obtain the stated result.

This result implies that $c_n \sim \mu^n o(n)$. The precise form of the sub-dominant term has still not been proved. In 1962 Hammersley and Welsh [23] proved that

$$1 \leq c_n/\mu^n < d \exp(\gamma n^{\frac{1}{2}}). \tag{2.2}$$

n	c_n	n	c_n
0	1	1	4
2	12	3	36
4	100	5	284
6	780	7	2172
8	5916	9	16268
10	44100	11	120292
12	324932	13	881500
14	2374444	15	6416596
16	17245332	17	46466676
18	124658732	19	335116620
20	897697164	21	2408806028
22	6444560484	23	17266613812
24	46146397316	25	123481354908
26	329712786220	27	881317491628
28	2351378582244	29	6279396229332
30	16741957935348	31	44673816630956
32	119034997913020	33	317406598267076
34	845279074648708	35	2252534077759844
36	5995740499124412	37	15968852281708724
38	42486750758210044	39	113101676587853932
40	300798249248474268	41	800381032599158340
42	2127870238872271828	43	5659667057165209612
44	15041631638016155884	45	39992704986620915140
46	106255762193816523332	47	282417882500511560972
48	750139547395987948108	49	1993185460468062845836
50	5292794668724837206644	51	14059415980606050644844

Table 1: Coefficients $c_0 \cdots c_{51}$ of the walk generating function of square lattice SAWS.

Kesten [32, 33] subsequently proved that

$$|c_{n+2}/c_n - \mu^2| \leq b/n^{\frac{1}{3}}, \tag{2.3}$$

and sharpened the upper bound in (2.2) for higher dimensional SAW.

In fact, as remarked in the introduction, a much stronger result is universally believed (though not proved.) It is generally accepted that

$$c_n \sim \mu^n n^g, \tag{2.4}$$

where $g = \frac{11}{32}$ for all regular two-dimensional lattices. This result was first obtained by Neinhuis [40], and all subsequent calculations have lent weight to this belief. We emphasis however that even the *existence* of a sub-dominant term of the form n^g has yet to be rigorously shown. For 3-dimensional walks similar behaviour is believed to hold, with $g \approx 0.1585$ [36]. There is no reason to believe that the exact value of this exponent is rational. In dimension greater than or equal to 4 it is believed that $g = 0$, where in dimensionality $d = 4$ this corresponds to a logarithmic correction. The fact that $g = 0$ for sufficiently high dimension has been proved by Hara and Slade [25].

Analogous results for SAP are known. Hammersley's proved [21] that SAP have the same connective constant as SAW for d-dimensional hypercubic lattices. Further, if p_n denotes the number of n-step SAP equivalent up to a translation, then

$$\exp(-c\sqrt{n})/2n \leq p_n/\mu^n < n \tag{2.5}$$

The lower bound follows from Hammersley's theorem described above, whereas the upper bound was proved by Kesten [32]. This result has only recently been sharpened (for two-dimensional SAP) by Madras [37], who proved that

$$p_n/\mu^n \leq C/\sqrt{n}, \tag{2.6}$$

for all (even) $n \geq 1$. Kesten [32] also proved that

$$b_1/n^{\frac{1}{3}} \leq |p_{2n+2}/p_{2n} - \mu^2| \leq b_2/n^{\frac{1}{4}}, \tag{2.7}$$

and sharpened the lower bound in (2.5) for higher dimensional SAW.

As for SAW, a much stronger result is universally believed (though not proved.) It is generally accepted that

$$p_{2n} \sim \mu^{2n} n^{\alpha-3}, \tag{2.8}$$

where $\alpha = \frac{1}{2}$ for all regular two-dimensional lattices. This result was first obtained by Neinhuis [40], and all subsequent calculations have lent weight to this belief. We emphasis however that even the *existence* of a sub-dominant term of the form n^h has yet to be rigorously shown. For 3-dimensional SAP similar behaviour is believed to hold, with $\alpha \approx 0.2375$ [36]. There is no reason to believe that the exact value of this exponent is rational. In dimension greater than or equal to 4 it is believed that $\alpha = 0$, where in dimensionality d= 4 this corresponds to a logarithmic correction. The fact that $\alpha = 0$ for sufficiently high dimension has been proved by Hara and Slade [25].

The exact value of the connective constant μ for the square lattice is not known. However for the hexagonal lattice, Neinhuis has given powerful arguments that suggest that the connective

constant $\mu_H = \sqrt{(2+\sqrt{2})}$, though this has not been proved. In 1984 Guttmann [12] pointed out that the square lattice connective constant could not be numerically distinguished from the unique positive root of the polynomial $13x^4 - 7x^2 - 581$. Sixteen years later, with the best numerical estimate of μ having been improved by 4 or 5 significant digits, (as described in Section 5) the connective constant still cannot be distinguished from that root. No analogous results for the triangular lattice are known. It seems plausible that for both the square and triangular lattice that the value of μ should be an algebraic number, as is the case for the hexagonal lattice. There is however no reason to believe in the algebraicity of μ for higher dimensional SAW.

Unlike random walks, SAW in any dimension may trap themselves, and the mean number of steps that occur before they are trapped is finite. In 1958, Lehman and Weiss [34] proved that the probability that a SAW lasts for n steps, p_n satisfies $c_L \exp(-a_L n) \leq p_n \leq c_U \exp(-a_U n)$, where c_L, c_U, a_L, a_U are positive constants. In fact, numerical work suggests that the mean number of steps of a SAW on the square lattice is 71 [27].

3 Self-avoiding polygons

Recently Jensen [31] developed and implemented an improved version of the FLM for SAP. Counting SAP by the finite-lattice method is intrinsically easier than counting SAW, as a SAP of N-steps has a maximum extension in any direction of $N/2 - 1$, whereas a SAW has a maximum extension of N. Thus for the same size lattice one can, roughly speaking, count SAP of twice the length of SAW. The improved algorithm, which we now describe, has computational complexity $\approx 1.2^N$, in both time and space, for evaluating SAP of N-steps.

The method used to enumerate SAP on the square lattice is an enhancement of the method devised by Enting [7] in his pioneering work. The first terms in the series for the polygon generating function can be calculated using transfer matrix techniques to count the number of polygons in rectangles $W+1$ edges wide and $L+1$ edges long. The transfer matrix technique involves drawing a line through the rectangle intersecting a set of $W+2$ edges. For each configuration of occupied or empty edges along the intersection we maintain a (perimeter) generating function for loops to the left of the line cutting the intersection in that particular pattern. Polygons in a given rectangle are enumerated by moving the intersection so as to add one vertex at a time, as shown in Fig. 1. The allowed configurations along the intersection are described in [7]. Each configuration can be represented by an ordered set of edge states $\{n_i\}$, where

$$n_i = \begin{cases} 0 & \text{empty edge,} \\ 1 & \text{lower part of loop closed to the left,} \\ 2 & \text{upper part of loop closed to the left.} \end{cases} \tag{3.1}$$

Configurations are read from the bottom to the top. So the configuration along the intersection of the polygon in Fig. 1 is $\{0112122\}$.

The rules for updating the partial generating functions as the intersection is moved are identical to the original work, so we refer the interested reader to [7] for further details regarding this aspect of the transfer matrix calculation.

Due to the obvious symmetry of the lattice one need only consider rectangles with $L \geq W$. Valid polygons were required to span the enclosing rectangle in the lengthwise direction. So it is clear that polygons with projection on the y-axis $< W$, that is polygons which are narrower than

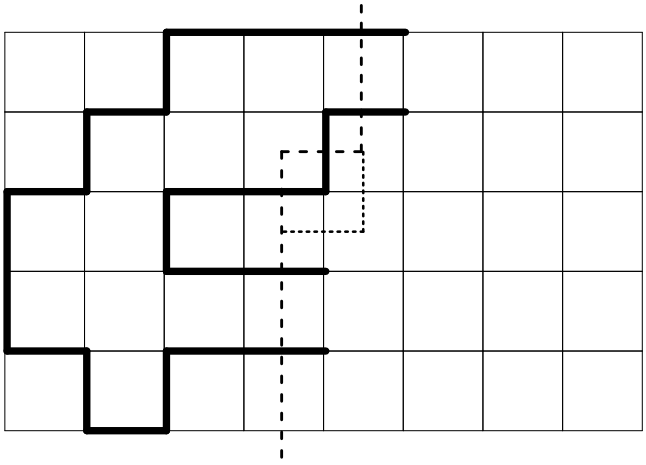

Figure 1: A snapshot of the intersection (dashed line) during the transfer matrix calculation on the square lattice. Polygons are enumerated by successive moves of the kink in the intersection, as exemplified by the position given by the dotted line, so that one vertex at a time is added to the rectangle. To the left of the intersection we have drawn an example of a partially completed polygon.

the width of the rectangle, are counted many times. It is however easy to obtain the polygons of width exactly W and length exactly L from this enumeration [7]. Any polygon spanning such a rectangle has a perimeter of length at least $2(W + L)$. By adding the contributions from all rectangles of width $W \leq W_{\max}$ (where the choice of $W_{\max}$ depends on available computational resources, as discussed below) and length $W \leq L \leq 2W_{\max} - W + 1$, with contributions from rectangles with $L > W$ counted twice, the number of polygons per vertex of an infinite lattice is obtained correctly up to perimeter $4W_{\max} + 2$.

The major improvement of the method used to enumerate polygons in this paper is that we require valid polygons to span the rectangle in *both* directions. In other words we directly enumerate polygons of width exactly W and length L rather than polygons of width $\leq W$ and length L as was done originally. The only drawback of this approach is that for most configurations we have to use four distinct generating functions since the partially completed polygon could have reached neither, both, the lower, or the upper boundaries of the rectangle. The major advantage is that the memory requirement of the algorithm is exponentially smaller.

Realizing the full savings in memory usage requires two enhancements to the original algorithm. Firstly, for each configuration we must keep track of the current minimum number of steps N_{cur} that have been inserted to the left of the intersection in order to build up that particular configuration. Secondly, we must calculate the minimum number of additional steps N_{add} required to produce a valid polygon. There are three contributions, namely the number of steps required to close the polygon, the number of steps needed (if any) to ensure that the polygon touches both the lower and upper boundary, and finally the number of steps needed (if any) to extend at least W edges in the length-wise direction. If the sum $N_{\text{cur}}+N_{\text{add}} > 4W_{\max}+2$ we can discard the partial generating function for that configuration because it won't make a contribution to the polygon count up to the perimeter lengths we are trying to obtain. For instance polygons spanning a rectangle with a width close to $W_{\max}$ have to be almost convex, so very convoluted polygons are not possible. Thus configurations with many loop ends (non-zero

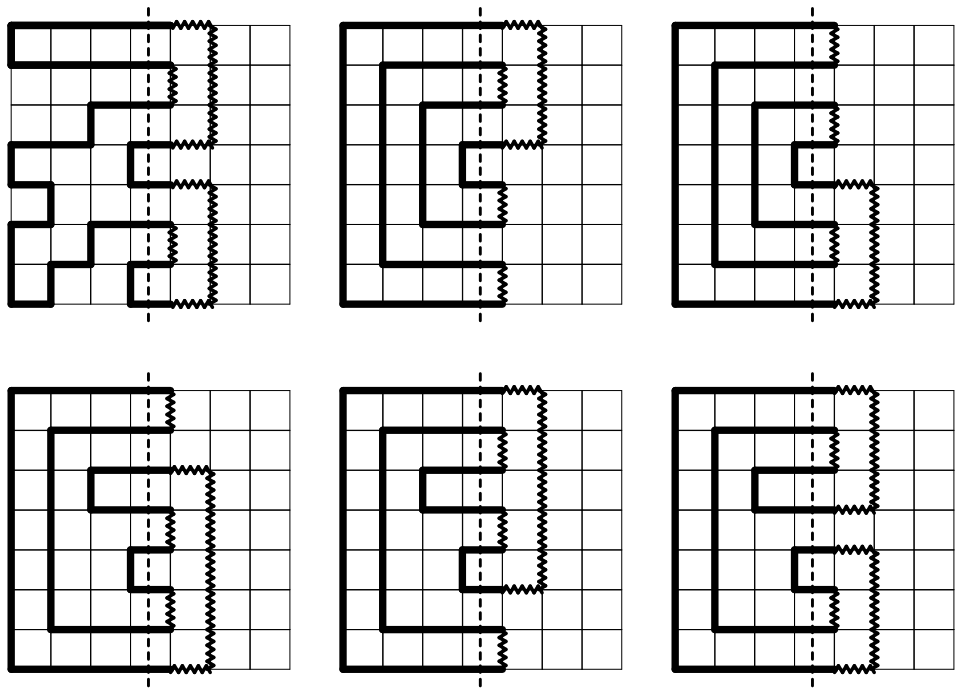

Figure 2: Examples of partially generated polygons (thick solid lines) to the left of the intersection (dashed line) and how to close them in a valid way (thick wavy line). Upper left panel shows how to close the configuration {12112212}. The upper middle and right panels show the two possible closures of the configuration {11112222}. The lower panels show the three possible closures of the configuration {11121222}.

entries) make no contribution at perimeter length $\leq 4W_{\max} + 2$.

The number of steps needed to ensure a spanning polygon is straightforward to calculate. The complicated part of the new approach is the algorithm to calculate the number of steps required to close the polygon. There are very many special cases depending on the position of the kink in the intersection and whether or not the partially completed polygon has reached the upper or lower boundary of the bounding rectangle. So in the following we will only briefly describe some of the simple contributions to the closing of a polygon.

Firstly, if the partial polygon contains separate pieces these have to be connected as illustrated in Fig. 2. Separate pieces are easy to locate since all we have to do is start at the bottom of the intersection and moving upwards we count the number of 1's and 2's in the configuration. Whenever these numbers are equal a separate piece has been found and (provided one is not at the last edge in the configuration) the currently encountered 2-edge can be connected to the next 1-edge above. N_{add} is incremented by the number of steps (the distance) between the edges and the two edge-states are removed from the configuration before further processing.

It is a little less obvious that if the configuration start (end) as $\{112\ldots2\}$ ($\{1\ldots122\}$) the two lower (upper) edges can safely be connected (note that there can be any number of 0's interspersed before the $\ldots$). Again N_{add} is incremented by the number of steps between the edges, and the two edge-states are removed from the configuration – leading to the new configuration $\{001\ldots2\}$ ($\{1\ldots200\}$) – before further processing. After these operations we may be left with a configuration which has just one 1- and one 2-edge, in which case we are

done since these two edges can be connected to form a valid polygon. This is illustrated in Fig. 2 where the upper left panel shows how to close the partial polygon with the intersection {12112212}, which contain three separate pieces. After connecting these pieces we are left with the configuration {10012002}. We now connect the two 1-edges and note that the first two-edge is relabeled to a 1-edge (it has become the new lower end of the loop). Thus we get the configuration {00001002} and we can now connect the remaining two edges and end up with a valid completed polygon.

Note that in the last two cases, in addition to the steps spanning the distance between the edges, an additional two horizontal steps had to be added in order to form a valid loop around the intervening edges. If the transformation above doesn't result in a closed polygon we must have a configuration of the form $\{111\ldots222\}$. The difficulty lies in finding the way to close such configurations with the smallest possible number of additional steps. Suffice to say that if the number of non-zero entries is small, one can easily devise an algorithm to try all possible valid ways of closing a polygon and thus find the minimum number of additional steps. In Fig. 2 we show all possible ways of closing polygons with 8 non-zero entries. Note that we have shown the generic cases here. In actual cases there could be any number of 0-edges interspersed in the configurations and this would determine which manner of closing would require the least number of additional steps.

With the original algorithm the number of configurations required as $W_{\max}$ increased grew asymptotically as $3^{W_{\max}}$ [16]. Our enumerations indicate that the computational complexity is reduced significantly. While the number of configurations still grows exponentially as $\lambda^{W_{\max}}$ the value of λ is reduced from $\lambda = 3$ to $\lambda \simeq 2$ with the improved algorithm. Furthermore, for any W we know that contributions will start at $4W$ since the smallest polygons have to span a $W \times W$ rectangle, so for each configuration we need only retain $4(W_{\max} - W) + 2$ terms of the generating functions while in the original algorithm contributions started at $2W$ because the polygons were required to span only in the length-wise direction. Combining these improvements allowed us to extend the series for the square lattice polygon generating function from 70 terms to 90 terms using the same computational resources. Obtaining a series this long with the original algorithm would have required at least 200 times as much memory, or close to a terabyte!

In Table 2 we have listed the new terms obtained in this work. The number of polygons of length ≤ 56 can be found in [16].

4 Parallelization of SAW counting algorithm

Details of the FLM algorithm used for SAW enumeration are given in [6]. Here we restrict ourselves to a description of the additional features necessary for an efficient parallelization of the algorithm. SAW are built up of partial SAW, with the connectivity of the partial SAW being represented by its *signature*. The connectivity represents the way in which a partial SAW may be joined to another.

The main time and memory consumptive process in counting SAWs consists of effectively running several matrix multiplication operations. Each matrix multiplication corresponds to adding one point to the finite array already processed (see the dotted line in Fig. 1 or Fig.3 of [5]). The vector being multiplied consists of a large number of polynomials with associated signatures indicating the connectivity of the partial SAW being represented by the signature. The polynomial represents the number of partial SAW of various various lengths, with the same

Table 2: The number, x_n, of embeddings of n-step polygons on the square lattice. Only non-zero terms are listed.

n	x_n	n	x_n
58	59270905595010696944	76	1158018792676190545425711414
60	379108737793289505364	78	7554259214694896127239818088
62	2431560774079622817356	80	49360379260931646965916677280
64	15636142410456687798584	82	323028185951187646733521902740
66	100792521026456246096640	84	2117118644744425875029583096670
68	651206027727607425003232	86	13895130612692826326409919713700
70	4216407618470423070733556	88	913197296505888161980048016984 00
72	27355731801639756123505014	90	6009314427575554688629703539417 00
74	177822806050324126648352460		

connectivity. The matrix is huge and sparse, and is represented implicitly.

The general idea of the parallelisation is to find an ordering of the elements in the vector which makes the matrix block diagonal, and then to compute each block on separate processors. The matrix multiplying the vector varies according to the point being added in the finite array. Different orders are needed for different matrices in order to get the required block diagonal form. We describe below an ordering which is very easy to re-order to make the subsequent ordering simple and efficient.

Each signature can be represented by a string, with one character corresponding to each point on the growing boundary of the finite lattice, as described for SAP in Sec. 3. Points can exist in one of four states, (for SAP only three states are needed) and are coded accordingly. Points without a bond are coded by "0". Unconnected chains are coded by a "3". Chains that are the leftmost (or rightmost) edge of a chain with both ends going across the border are represented with a "1" or a "2" respectively. Further details of this coding scheme are given in [5], Section 4, but it is a straightforward extension of the scheme described in Sec. 3 for SAP.

Adding a site to the processed lattice consists of taking each signature, adding the new point, and then seeing what resultant signatures can be produced. This is analogous to one row of the matrix multiplication. The signatures that can be produced from a given signature can have significantly changed the two bonds going into (in the old signature) and out of (in the new signatures) the point being added. Some other elements of the signature can also change. For instance, if a "2" bond coming in is not taken out of the point being added, then the corresponding "1" will change into a "3" as it is now disconnected with respect to the rest of the signature.

There is an invariant in the signatures in this operation—a non-zero can never become a zero or *vice versa* other than for the two points being changed. This allows a simple block diagonalization as follows: group together all the signatures into classes, where two signatures are in the same class if they have either both zeros or both non-zeros for each element in the signature, other than the two elements being changed directly. Call all the signatures in each class a *block*.

One can implicitly order the signatures such that this block diagonalization is simple to keep track of. Define the *id* for the block as a sub-signature, where each element in the sub-signature is 0 if the corresponding element in the signatures in each block is zero, and 1 otherwise. Order

these elements in the subsignature in the order that they will be processed next, so that the most significant element is the latest signature bit changed in the last matrix multiplication operation, and the least significant bit is the next to be altered. Order the blocks by this signature.

This ordering has the property that if a given block is processed, the resultant signatures will fall into one of two blocks for the next matrix multiplication operation depending upon the bit produced in this operation that will not be altered in the next operation. When doing the processing, we make two implicit lists and put the signatures into one of the two lists based upon this bit. When all the blocks have been processed, we concatenate these two lists, and the resulting list will then be in a perfect order for the next iteration of the matrix multiplication.

On a multi-processor system, each processor can be assigned some of the blocks, and the ordering can be made implicitly, partly as a result of the ordering kept on each processor, and partly due to an implicit arbitrary ordering of the processors. Imagine each processor doing the matrix multiplication on its block, and then sending the results off to a different processor, based upon the uppermost bit generated.

In practice, we typically have more blocks than processors typically, and so each processor will deal with several signatures. For efficiency of network traffic, we can arrange things such that, most of the time, the processor that will process the next block generated is usually the current processor.

Unfortunately, some blocks contain more signatures than other blocks. For instance, the block with *id* of all zeros can only contain the signature of all zeros, while the block with *id* of all ones can contain a mixture of 1s, 2s and 3s. This means that in order to have reasonable load balancing, it is necessary to take into account the size of each block when working out which block to put on which processor. An acceptable preselection can all be precalculated in a reasonable amount of time.

Depending upon the length of the signatures and the number of processors, it may still not be possible to effect a good load balance, particularly with a large number of processors and a small length of signatures. This can be worked around by dividing some of the blocks up into sub-blocks, based upon whether some small number of the signature bits are 1s, 2s, or 3s. This spoils the block diagonality, and causes a significant increase in inter-processor traffic in the middle of the matrix multiplication, but this traffic is not too bad, as the terms in the matrix that go between various sub-blocks are relatively rare.

In practice, an implementation of this algorithm on an Intel Paragon had roughly fifty percent processor utilization.

5 Analysis of the series

5.1 Method of differential approximants

We analyzed both the SAP and SAW series by the numerical method of differential approximants [14]. In this method, the available series coefficients are fitted to a linear ordinary differential equation, with polynomial coefficients. Denote $\mathcal{D} = z\frac{d}{dz}$. Then one fits the available series coefficients to a differential equation of the form

$$\sum_{i=0}^{K} Q_i(z)\mathcal{D}^i f(z) = P(z) \tag{5.1}$$

where Q_i and P_i are polynomials:

$$Q_i(z) = \sum_{j=0}^{N_i} Q_{i,j} z^j P(z) = \sum_{j=0}^{L} P_j z^j. \tag{5.2}$$

Such fitting is numerically fast and stable, involving only the solution of linear systems of equations. With appropriate normalization it is also unique.

Clearly the solution of such a differential equation will reproduce those coefficients used in the construction of the d.e., and, further, will have a singularity structure that will be characterised by singularities and associated exponents. These singularities are given by the zeros of $Q_K(z)$, and the exponents follow from the associated indicial equation. By varying the degree of the o.d.e. and the degrees of the polynomials, a variety of estimates of the singularity positions and exponents can be obtained. These are then appropriately averaged [14], yielding final estimates of the singularity position and exponent, which are often very precise. Indeed, in the case of an underlying D-finite generating function, the exact generating function can be discovered, and hence conjectured, in this way—see [17] for an example of this.

With the above notation, we denote the approximating o.d.e. (which are called *differential approximants*) by $[L/N_0; N_1]$ and $[L/N_0; N_1; N_2]$ for the (usual) cases of $K = 1$ and $K = 2$ respectively. A normalization condition, that forces the point at the origin to be a regular singular point is made. This is $Q_{K,0} = 1$ unless we are dealing with a homogeneous o.d.e. ($P(z) = 0$), in which case we impose the additional constraint $Q_{0,0} = 0$.

If we find singular points and critical exponents in this way, we call these *unbiased* estimates. If, as is sometimes the case, the critical point is exactly known, this can be imposed on the o.d.e. (simply by making $(z - z_c)$ a factor of $Q_K(z)$, where z_c is the known critical point singularity). The resulting critical exponents are then called *biased*. It is also possible to bias with both a critical point *and* the associated exponent. While it is not possible to directly bias with a critical exponent alone (that is, without specifying the critical point), this can be done by utilising the high degree of linearity observed between estimates of the critical exponent and critical point. Thus the "correct" critical point can be read off from the straight line of best fit corresponding to the known exponent. We will see an example of this later in this section.

5.2 Self-avoiding polygon analysis

For square-lattice polygons, the generating function $P(x) = \sum p_{2n} x^{2n} \sim A(x^2) + B(x^2)(1 - x^2/x_c^2)^{2-\alpha}$ is an even function. In Table 3 we have listed estimates of the critical point x_c^2 and exponent $2-\alpha$ of the series for the square lattice SAP generating function. The estimates were obtained by averaging values obtained from first order $[L/N; M]$ and second order $[L/N; M; K]$ inhomogeneous differential approximants. For each order L of the inhomogeneous polynomial we averaged over those approximants to the series which used at least the first 35 terms of the series (that is, polygons of perimeter at least 74), and used approximants such that the difference between N, M, and K didn't exceed 2. These are therefore "diagonal" approximants. Some approximants were excluded from the averages because the estimates were obviously spurious. The error quoted for these estimates reflects the spread (basically one standard deviation) among the approximants. Note that these error bounds should *not* be viewed as a measure of the true error as they cannot include possible systematic sources of error. We discuss further the systematic error when we consider biased approximants. Based on these estimates we conclude that $x_c^2 = 0.1436806289(5)$ and $\alpha = 0.5000005(10)$.

Table 3: Estimates for the critical point x_c^2 and exponent $2-\alpha$ obtained from first and second order inhomogeneous differential approximants to the series for square lattice polygon generating function. L is the order of the inhomogeneous polynomial.

L	First order DA		Second order DA	
	x_c^2	$2-\alpha$	x_c^2	$2-\alpha$
1	0.14368062897(17)	1.50000074(35)	0.14368062883(45)	1.50000092(92)
2	0.14368062902(14)	1.50000068(26)	0.14368062943(29)	1.49999957(80)
3	0.14368062878(35)	1.50000107(71)	0.14368062914(20)	1.50000034(51)
4	0.14368062910(29)	1.50000038(61)	0.14368062914(16)	1.50000038(44)
5	0.14368062890(43)	1.50000085(93)	0.14368062911(53)	1.5000002(12)
6	0.14368062863(49)	1.5000014(10)	0.14368062901(54)	1.5000005(12)
7	0.14368062886(39)	1.50000094(80)	0.14368062881(52)	1.5000009(10)
8	0.14368062885(64)	1.5000008(13)	0.143680629210(97)	1.50000021(24)

As stated earlier, there is very convincing evidence that the critical exponent $\alpha = 1/2$ exactly. If we assume this to be true we can obtain a refined estimate for the critical point x_c^2, by making use of the observed linearity in the plot of estimates of the critical point *vs.* the critical exponent. In Fig. 3 we have plotted estimates for the critical exponent $2-\alpha$ against estimates for the critical point x_c^2. Each point in this figure represents a pair of estimates obtained from a second order inhomogeneous differential approximant. The order of the inhomogeneous polynomial was varied from 0 to 10. Note that there is an almost linear relationship between the estimates for $2-\alpha$ and x_c^2 and that for $2-\alpha = 3/2$ we get $x_c^2 \simeq 0.14368062928\ldots$.

In order to get some idea as to the effect of systematic errors, we carried out this analysis using polygons of length up to 60 steps, then 70, then 80 and finally 90 steps. The results were $x_c^2 = 0.1436806308$ for $n = 60$, $x_c^2 = 0.14368062956$ for $n = 70$, $x_c^2 = 0.14368062930$ for $n = 80$, and $x_c^2 = 0.14368062928$ for $n = 90$. This is a rapidly converging sequence of estimates, though we have no theoretical basis that would enable us to assume any particular rate of convergence. However, observing that the differences between successive estimates are decreasing by a factor of at least 5, it is not unreasonably optimistic to estimate the limit at $x_c^2 = 0.14368062927(1)$.

This leads to our final estimate $x_c^2 = 0.14368062927(1)$ and thus we find the connective constant $\mu = 1/x_c = 2.63815853034(10)$. As mentioned above, some years ago we [5] pointed out that since the hexagonal lattice connective constant is given by the zero of a quadratic in x^2, it is plausible that this might be the case also for the square lattice connective constant. On the basis of an estimate of the connective constant that was 4 orders of magnitude less precise, we pointed out then that the polynomial

$$581x^4 + 7x^2 - 13 = 0$$

was the only polynomial we could find with "small" integer coefficients consistent with our estimate. The relevant zero of this polynomial is 0.1436806292698685.. in complete agreement with our new estimate—which, as noted above, contains four more significant digits! Unfortunately the other zero is at $x_c^2 = -0.1557288\ldots$, and we see no evidence of such a singularity. Nevertheless, the agreement is so astonishingly good that we are happy to take this as a good algebraic approximation to the connective constant. An argument as to why we might not expect to see the singularity on the negative real axis from our series analysis would make the

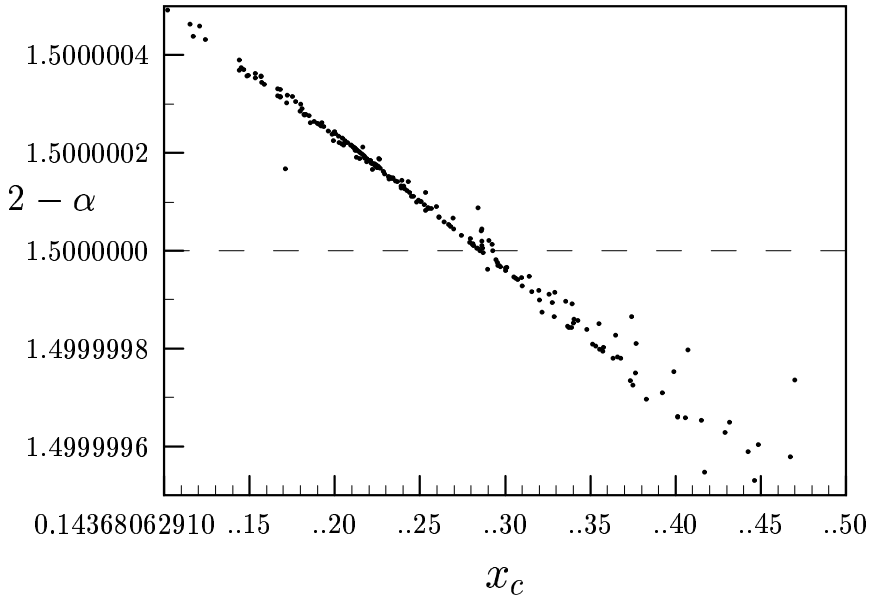

Figure 3: Estimates for the critical exponent $2-\alpha$ vs. estimates for the critical point x_c^2 of the square lattice polygon generating function.

root of the above polynomial a plausible conjecture for the exact value, but at present such an argument is missing.

Two further analyses were carried out on the data. Firstly, a study of the location of non-physical singularities, and secondly, a study of the asymptotic form of the coefficients—which is relevant to the identification of any correction-to-scaling exponent. Singularities outside the radius of convergence give exponentially small contributions to the asymptotic form of the coefficients, so are notoriously hard to analyse. Nevertheless, we see clear evidence of a singularity on the negative real axis at $x^2 \approx -0.40$ with an exponent that is extremely difficult to analyse but could be 1.5, in agreement with the physical exponent. There is weaker evidence of a further conjugate pair of singularities. First-order approximants locate these at $-0.015 \pm 0.36i$, while second order approximants locates them at $-0.035 \pm 0.31i$. There is also evidence of a further singularity on the negative real axis at $x_c^2 = -0.7$. We are unable to give a useful estimate of the exponents of these singularities.

We turn now to the asymptotic form of the coefficients. We have argued previously [6] that there is no non-analytic correction-to-scaling exponent for the polygon generating function. This is entirely consistent with Nienhuis's [40] observation that there is a correction-to-scaling exponent of $\Delta = \frac{3}{2}$. Since for the polygon generating function exponent $\alpha = \frac{1}{2}$, the correction term has an exponent equal to a positive integer, and therefore "folds into" the analytic background term, denoted $A(x)$ in Eqn. (5.3). That is to say, if one writes the polygon generating function as

$$P(x) = \sum_{n \geq 0} p_n x^n \sim A(x) + B(x)(1 - x/x_c)^{2-\alpha}[1 + (1 - x/x_c)^{\Delta} + \cdots] \quad (5.3)$$

$$= A(x) + B(x)(1 - x/x_c)^{\frac{3}{2}}[1 + (1 - x/x_c)^{\frac{3}{2}} + \cdots] = \tilde{A}(x) + B(x)(1 - x/x_c)^{\frac{3}{2}}[1 + \cdots] \quad (5.4)$$

one sees that because $2 - \alpha + \Delta$ is an integer, the non-analytic correction-to-scaling term "folds

into" the analytic background term $A(x)$.[3] On the other hand, since $-\gamma + \Delta$ is not an integer, a non-analytic correction-to-scaling term is predicted to occur in the *walk* generating function, as we show below.

We assert that the asymptotic form for the polygon generating function is as given by Eqn. (5.4) above. In evidence of this, we remark that from (5.4) follows the asymptotic form

$$p_{2n} x_c^{2n} \sim n^{-\frac{5}{2}} [a_1 + a_2/n + a_3/n^2 + a_4/n^3 + \cdots]. \tag{5.5}$$

Using the algebraic approximation to x_c quoted above, we show in Table 4 the estimates of the amplitudes $a_1, \cdots, a_4$. From the table we see that $a_1 \approx 0.0994018$, $a_2 \approx -0.02751$, $a_3 \approx 0.0255$ and $a_4 \approx 0.12$, where in all cases we expect the error to be confined to the last quoted digit. The excellent convergence of all columns is strong evidence that the assumed asymptotic form is correct. If we were missing a term corresponding to, say, a half-integer correction, the fit would be far worse. This is explained at greater length in [6]. So good is the fit to the data that if we take the last entry in the table, corresponding to $n = 45$, and use the entries as the amplitudes, then $p_4 \cdots p_{16}$ are given exactly by the above asymptotic form (provided we round to the nearest integer), and beyond perimeter 20 all coefficients are given to the same accuracy as the leading amplitude.

Finally, to complete our analysis, we estimate the critical amplitudes $A(x_c^2)$ and $B(x_c^2)$, defined in eqn.(5.3). $A(x_c^2)$ has been estimated by evaluating Padé approximants to the generating function, evaluated at x_c^2. In this way we estimate $A(x_c^2) \approx 0.036$, while $B(x_c^2)$ follows from the estimate of a_1 in eqn.(5.5), since $B(x_c^2) = \frac{4\sqrt{\pi} a_1}{3} \approx 0.234913$.

5.3 Self-avoiding walk analysis

We turn now to a corresponding analysis of the SAW generating function, based on differential approximants, as just described for the analysis of the SAP generating function. Averaging the large number of differential approximant estimates of the critical point and exponent, we find the following unbiased estimates of the critical point and critical exponent from both first- and second-order differential approximants:

$$x_c = 0.3790522(3),\ \gamma = 1.34367(10)\ 1^{st} \text{ order}; x_c = 0.3790523(3),\ \gamma = 1.34372(10)\ 2^{nd} \text{ order}.$$

These are in perfect agreement with, though more precise than, the estimates given previously [5], and based on a 39 step SAW series. They are also in very good agreement with the believed exact (but non-rigorous) value [40] $\gamma = \frac{43}{32} = 1.34375$. Assuming this value for γ then permits us to make biased estimates of the critical point, using the observed linearity between estimates of the critical point and critical exponent, as discussed above. We find:

$$x_c = 0.37905225(10),\ x_c = 0.37905229(10),$$

or equivalently,

$$\mu = 2.6381587(7),\ \mu = 2.6381584(7),$$

in complete agreement with, though less precise than, the estimates based on the (longer) SAP generating function.

[3] It does not appear to produce a confluent logarithm, as might be expected. Our numerical analysis gives no evidence of such a term.

Table 4: A fit to the asymptotic form $p_{2n}x_c^{2n} \sim n^{-\frac{5}{2}}[a_1 + a_2/n + a_3/n^2 + a_4/n^3 + \cdots]$ Estimates of the amplitudes a_1, a_2, a_3, a_4.

n	a_1	a_2	a_3	a_4
20	0.09940085	-0.02745705	0.02476376	0.11822181
21	0.09940118	-0.02747548	0.02511347	0.11601107
22	0.09940140	-0.02748880	0.02537979	0.11423855
23	0.09940154	-0.02749767	0.02556592	0.11293766
24	0.09940164	-0.02750397	0.02570426	0.11192457
25	0.09940170	-0.02750829	0.02580364	0.11116357
26	0.09940174	-0.02751137	0.02587757	0.11057283
27	0.09940177	-0.02751355	0.02593211	0.11011880
28	0.09940179	-0.02751510	0.02597236	0.10977030
29	0.09940180	-0.02751619	0.02600168	0.10950667
30	0.09940181	-0.02751694	0.02602273	0.10931043
31	0.09940182	-0.02751745	0.02603734	0.10916929
32	0.09940182	-0.02751777	0.02604692	0.10907354
33	0.09940182	-0.02751795	0.02605254	0.10901552
34	0.09940182	-0.02751802	0.02605500	0.10898929
35	0.09940182	-0.02751802	0.02605494	0.10898993
36	0.09940182	-0.02751796	0.02605285	0.10901358
37	0.09940182	-0.02751785	0.02604913	0.10905699
38	0.09940182	-0.02751771	0.02604408	0.10911757
39	0.09940182	-0.02751755	0.02603796	0.10919302
40	0.09940182	-0.02751736	0.02603097	0.10928158
41	0.09940182	-0.02751717	0.02602327	0.10938160
42	0.09940181	-0.02751696	0.02601500	0.10949174
43	0.09940181	-0.02751675	0.02600629	0.10961079
44	0.09940181	-0.02751653	0.02599720	0.10973796
45	0.09940181	-0.02751631	0.02598785	0.10987195

In the preceding analysis of the *polygon* generating function we pointed out that there was no numerical evidence for a non-analytic correction-to-scaling term. On the other hand, since $-\gamma + \Delta$ is not an integer, a non-analytic correction-to-scaling term is predicted to occur in the *walk* generating function.

We have analysed the series for the correction-to-scaling exponent by a wide variety of methods, all of which give consistent results, though inspiring different degrees of confidence. The most convincing, and arguably simplest method is a direct fit of the data to the assumed asymptotic form. To do this, we first need an accurate estimate of the connective constant, which, as noted above, we have.

As proved in [20], the SAW generating function $C(x)$ has, in addition to the "ferromagnetic" singularity at $x = x_c$, an "anti-ferromagnetic" singularity at $x = -x_c$. The exponent at the antiferromagnetic singularity is believed to be equal to the internal energy exponent $2 - \alpha = \frac{3}{2}$. The exponent at the ferromagnetic singularity [40] is known to be $\frac{43}{32}$.

We therefore expect the generating function for walks to behave like:

$$C(x) = \sum c_n x^n \sim A(x)(1 - x/x_c)^{-\frac{43}{32}}[1 + B(x)(1 - x/x_c)^{\Delta} + \cdots] \tag{5.6}$$

$$+ D(x)(1 + x/x_c)^{\frac{1}{2}}[1 + E(x)(1 + x/x_c)^{\Lambda} + \cdots] \tag{5.7}$$

where A, B, C, D, E are smooth functions.

Hence the asymptotic form of the coefficients is given by:

$$c_n x_c^n \sim n^{\frac{11}{32}}[a_1 + a_2 n^{-1} + a_3 n^{-\Delta} + a_4 n^{-2}] + (-1)^n n^{-\frac{3}{2}}[b_1 + b_2 n^{-\Lambda} + b_3 n^{-1} + b_4 n^{-2}]. \tag{5.8}$$

The correction-to-scaling exponent Λ, associated with the antiferromagnetic singularity, is one about which we have no *a priori* knowledge. Two obvious candidates are $\Lambda = 1$ and $\Lambda = 1.5$, corresponding respectively to an analytic correction and to the same correction as expected for the ferromagnetic singularity. A third possibility is $\Lambda = 0.5$, given that square roots are already present in the antiferromagnetic singularity. We have investigated all three possibilities (as well as others), and the evidence is strongly in favour of only an analytic correction, that is, $\Lambda = 1$.

A fit to the data of the above form with $\Delta = \frac{3}{2}$ and $\Lambda = 1$ (which we believe to be the correct values) is shown in Table 5. The coefficient b_3 is redundant since $\Lambda = 1$.

Convergence is seen to be excellent (and quite comparable to a corresponding analysis of the 54 term square lattice Ising susceptibility series [14], where both Δ and Λ are known to be 1). We estimate $a_1 \approx 1.177043, a_2 \approx 0.5500, a_3 \approx -0.140, a_4 \approx -0.12, b_1 \approx -0.1899, b_2 \approx 0.175$, and $b_3 \approx -1.51$. Without undertaking a detailed error analysis, we expect errors to be confined to the last quoted digit.

A corresponding fit with $\Delta = \frac{3}{2}$ and $\Lambda = 0.5$ was made (not shown) and the fit to the antiferromagnetic singularity amplitudes was significantly worse than with the exponent set used in Table 5. More significantly, the amplitude b_2 of the term associated with the assumed correction-to-scaling exponent was small in magnitude, and tending towards zero as n grows: this is a sign of its absence. A similar analysis (not shown) was made with $\Lambda = 1.5$, and again a poorer fit was observed, with the amplitude of the assumed correction term decreasing rapidly. Thus we conclude that there is no evidence for a non-analytic correction at the antiferromagnetic singularity.

Returning to the physical singularity, if we now set $\Delta = \frac{11}{16}$, for which there is some theoretical justification [44], and also allow for a second correction-to-scaling exponent $\Delta_1 = \frac{3}{2}$, the asymptotic form of the coefficients becomes:

n	a_1	a_2	a_3	a_4	b_1	b_2	b_4
39	1.1770421	.55020	-.14117	-.11819	-.18984	.17410	-1.49499
40	1.1770421	.55021	-.14118	-.11816	-.18984	.17411	-1.49511
41	1.1770422	.55018	-.14100	-.11857	-.18984	.17422	-1.49725
42	1.1770421	.55019	-.14104	-.11848	-.18984	.17424	-1.49770
43	1.1770422	.55016	-.14082	-.11899	-.18984	.17439	-1.50054
44	1.1770422	.55017	-.14089	-.11884	-.18984	.17443	-1.50142
45	1.1770423	.55015	-.14071	-.11927	-.18985	.17455	-1.50389
46	1.1770423	.55015	-.14073	-.11922	-.18985	.17456	-1.50414
47	1.1770423	.55014	-.14061	-.11951	-.18985	.17464	-1.50585
48	1.1770423	.55014	-.14060	-.11954	-.18985	.17463	-1.50566
49	1.1770424	.55013	-.14051	-.11976	-.18985	.17469	-1.50703
50	1.1770424	.55013	-.14049	-.11980	-.18985	.17468	-1.50674
51	1.1770424	.55012	-.14042	-.11998	-.18985	.17473	-1.50791

Table 5: A fit to the asymptotic form with $\Delta = 1.5$ and $\Lambda = 1$.

$$c_n x_c^n \sim n^{\frac{11}{32}}[a_1 + a_2 n^{-\Delta} + a_3 n^{-1} + a_4 n^{-2\Delta} + a_5 n^{-\Delta_1}] + (-1)^n n^{-\frac{3}{2}}[b_1 + b_2 n^{-1} + b_3 n^{-2}].$$

The results of this fit are shown in Table 6. The extremely small estimated values of the amplitude a_2 strongly suggest that this term is in fact absent. (Including even higher order terms in the above equation strengthens this conclusion). Furthermore, the values of the other amplitudes are consistent with the values of the corresponding terms in the previous table.

A variety of other combinations of values for Δ and Λ were tried, with equally persuasive numerical evidence in favour of just one correction-to-scaling exponent, given originally [40] as $\Delta = 1.5$, and only analytic corrections at the antiferromagnetic singularity, a result we believe to be new. The estimate of Δ is also in agreement with a finite-size scaling analysis [2].

6 Anisotropic Lattice and Exact Solutions

In a recent paper [15] a numerical procedure was given that clearly indicates whether or not a given statistical-mechanical system is solvable, in the sense that it is expressible in terms of D-finite functions. A D-finite function is one which may be expressed as the solution of a linear ordinary differential equation of finite order with polynomial coefficients (that is, precisely the family of differential equations used in the method of differential approximants described in the previous section. As discussed in [18], many of the solved combinatorial models in statistical mechanics fall into this class.

We can show, on the basis of the behaviour of the SAW and SAP generating functions on the *anisotropic* square lattice, that they are almost surely not D-finite.[4]

We can write the walk generating function as

$$C(x,y) = \sum_{m,n=0}^{\infty} c_{m,n} x^m y^n = \sum_{n=0}^{\infty} H_n(x) y^n.$$

[4] For SAP this has been recently proved [43]

n	a_1	a_2	a_3	a_4	a_5	b_1	b_2	b_3
39	1.1770289	.00375	.51572	.31466	-.51252	-.18984	.17400	-1.49325
40	1.1770159	.00555	.50439	.37207	-.56479	-.18984	.17410	-1.49495
41	1.1770253	.00423	.51280	.32902	-.52546	-.18984	.17417	-1.49624
42	1.1770174	.00536	.50556	.36644	-.55976	-.18984	.17423	-1.49738
43	1.1770293	.00362	.51676	.30799	-.50601	-.18984	.17432	-1.49918
44	1.1770185	.00523	.50633	.36297	-.55673	-.18984	.17440	-1.50089
45	1.1770295	.00356	.51724	.30492	-.50301	-.18985	.17449	-1.50271
46	1.1770225	.00464	.51010	.34325	-.53859	-.18985	.17455	-1.50393
47	1.1770286	.00368	.51651	.30850	-.50624	-.18985	.17460	-1.50505
48	1.1770253	.00420	.51301	.32764	-.52411	-.18985	.17463	-1.50568
49	1.1770290	.00361	.51701	.30558	-.50346	-.18985	.17466	-1.50640
50	1.1770270	.00393	.51479	.31796	-.51508	-.18985	.17468	-1.50681
51	1.1770297	.00349	.51783	.30093	-.49905	-.18985	.17471	-1.50739

Table 6: A fit to the asymptotic form with $\Delta = \frac{11}{16}$, $\Delta_1 = \frac{3}{2}$ and $\Lambda = 1$. Note that the amplitude a_2 is small, and consistent with a limit of zero.

Here $c_{m,n}$ is the number of SAW with m steps in the x direction and n steps in the y direction. From our enumerations we have calculated the first 12 values $H_n(x), n = 0, \cdots, 11$.

The first few values are:

$$H_0(x) = \frac{1+x}{1-x}$$

$$H_1(x) = \frac{2(1+x)^2}{(1-x)^2}$$

$$H_2(x) = \frac{2(1+7x+14x^2+16x^3+9x^4+3x^5)}{(1-x)^3(1+x)^2}$$

$$H_3(x) = \frac{2(1+10x+29x^2+44x^3+41x^4+22x^5+7x^6)}{(1-x)^4(1+x)^2}$$

$$H_4(x) = \frac{2(1+15x+74x^2+214x^3+366x^4+448x^5+370x^6+220x^7+81x^8+19x^9)}{(1-x)^5(1+x)^4},$$

the rest are given in a more compact but obvious notation below.

$H_5 =$ [2,38,252,978,2490,4572,6256,6572,5322,3258,1500,454,90]/
$(1-x)^6(1+x)^4(1+x+x^2)$

$H_6 =$[2,50,466,2662,10204,29184,64254,113238,161856,190836,185822,
150066,99436,53664,22850,7458,1678,230]$/(1-x)^7(1+x)^6(1+x+x^2)^2$

$H_7 =$[2,58,640,4330,20300,72154,202962,468428,905856,1493120,2120444,
2614760,2811800,2640700,2164180,1540708,946866,495778,218008,77950,
21960,4366,558]$/(1-x)^8(1+x)^6(1+x+x^2)^3(1+x^2)$

$H_8 =$[2,70,958,8190,48726,221426,800194,2384758,5986362,12893158,24113042,
39565126,57341056,73810864,84648452,86675952,79248094,64644810,46903890,

30145922,17040710,8397906,3557238,1272702,372890,85918,14114,1406]/
$(1-x)^9(1+x)^8(1+x+x^2)^4(1+x^2)$

H_9 =[2,78,1202,11572,78842,414326,1755590,6211684,18774920,49374676,114487320,
236608436,439425770,738259610,1127758362,1572720200,2008219064,2353023276,
2533306576,2507629760,2282049678,1907698086,1462698478,1026096828,656425728,
381180108,199768804,93728352,38986566,14167546,4419170,1146744,238382,35870, 3434]/
$(1-x)^{10}(1+x)^8(1+x+x^2)^5(1+x^2)^3$

H_{10} =[2,92,1716,20212,169342,1098858,5772030,25397848,95802292,315540718,
920175594,2402977678,5670463958,12182510444,23975573364,43445416402,
72794482536,113178105126,163751179506,220994268608,278708247274,
328926791288,363624287492,376761857554,365954755408,333164638274,
284128186626,226765011996,169135915738,117675890184,76188834320,
45767212250,25413673850,12985628802,6071485858,2579605224,987196680,
336508130,100644194,25883626,5551016,947932,117236,8610]/
$(1-x)^{11}(1+x)^{10}(1+x+x^2)^6(1+x^2)^3(1+x+x^2+x^3+x^4)$

H_{11} =[2,100,2036,26126,240232,1717830,9997194,48965896,206606824,764755582,
2518129404,7459383368,20060438350,49346520078,111727028868,234057469446,
455685997692,827584611676,1406494287002,2242889526364,3363600690048,
4752808975856,6337611832644,7985123901712,9515938057224,10733921580332,
11466227905864,11602604094800,11121966683392,10097592996872,8679254387460,
7058100526048,5425453935522,3937389995556,2693738938176,1734120542610,
1048131482672,593203450666,313368227330,153921122508,69972510256,
29274593670,11194294868,3878510196,1204305530,330217170,78404072,
15640502,2508152,294188,21090]/
$(1-x)^{12}(1+x)^{10}(1+x+x^2)^7(1+x^2)^5(1+x+x^2+x^3+x^4)^2$

The rational functions $H_n(x)$ are observed to have unimodal asymmetric (for $n>1$) numerators with positive coefficients, and denominators of the same degree as the numerator. The denominators appear to form a regular pattern, though we cannot be certain that this pattern persists. However, if we define $t_{n-1}=\frac{1-x^n}{1-x}$ and $s_n=(1+x^n)$ then the denominator of $H_n(x)$ appears to be given by:

$$\begin{aligned}(1-x)^{n+1}s_1^n t_2^{n-4}s_2^{n-7}t_4^{n-9}s_4^{n-10}t_6^{n-14}s_6^{n-17}t_8^{n-19}s_8^{n-20}t_{10}^{n-24}\cdots, &\quad \text{(n even)}\\ (1-x)^{n+1}s_1^{n-1} t_2^{n-4}s_2^{n-6}t_4^{n-9}s_4^{n-11}t_6^{n-14}s_6^{n-16}t_8^{n-19}s_8^{n-21}t_{10}^{n-24}\cdots, &\quad \text{(n odd).}\end{aligned}$$

In the above expressions, any negative exponents should be replaced by zero. If this is indeed the pattern of the denominators, this regularity must say something profound about the analytic structure of the generating function. It is beyond the scope of current algorithms and computers to calculate the next 20 or so denominators, which would be needed before we could be quite confident that the pattern persists. However what *is* clear is that, as n increases, the zeros of the denominator are becoming dense on the unit circle in the complex x-plane. As previously discussed [18], this then imposes severe restrictions on the $C(x,y)$. In particular, it excludes all algebraic functions, all D-finite functions, all constructible differentiably algebraic

functions, as well as a large class of other functions that are common in mathematical physics. (Just what it implies for $C(x,x)$ is less clear).[5] Regrettably, no obvious regularity has been observed in the numerators. Further, evaluating the numerator polynomials at $x=1$ does not lead to any recognizable sequence, which is sometimes the case with simpler problems, nor is there any feature that suggests the existence of an inversion relation for this problem, unlike the analogous case of the Ising model susceptibility [18], where the numerator polynomials are symmetric, as well as unimodal.

Similar behaviour is observed for SAP. The SAP generating function $P(x)$ may be written as

$$P(x,y) = \sum_{m,n\geq 1} p_{m,n} x^{2m} y^{2n} = \sum_{n\geq 1} R_n(x) y^{2n},$$

where $p_{m,n}$ is the number of square lattice polygons, equivalent up to a translation, with $2n$ horizontal steps and $2m$ vertical steps. We [10] found the first 9 R functions, $R_1(x), \cdots, R_9(x)$, and these were found to behave in a manner characteristic of D-unsolvable problems—that is, the zeros appear to build up on the unit circle. They first few are:

$$\begin{aligned}
R_1(x) &= x/(1-x), && (6.1)\\
R_2(x) &= x(1+x)^2/(1-x)^3, && (6.2)\\
R_3(x) &= x(1+8x+17x^2+12x^3+3x^4)/(1-x)^5, && (6.3)\\
R_4(x) &= x(1+18x+98x^2+204x^3+178x^4+70x^5+11x^6)/(1-x)^7, && (6.4)\\
R_5(x) &= xN_5/(1-x)^9(1+x)^2, && (6.5)\\
R_6(x) &= xN_6(x)/(1-x)^{11}(1+x)^4, && (6.6)\\
R_7(x) &= xN_7(x)/(1-x)^{13}(1+x)^6(1+x^2+x^4). && (6.7)\\
& && (6.8)
\end{aligned}$$

where, in an obvious notation,

$N_5 = [1,34,400,2154,6169,10302,10458,6492,2398,490,44]$
$N_6 = [1,54,1066,10200,55210,186288,416286,638978,686469,518356,272434,97126,22372,3034,188]$
and N_7 is also a unimodal polynomial, with positive coefficients, but of degree 22.

In this case, unlike the situation for SAW, the occurrence of new terms in the denominator, corresponding to higher roots of unity, can be identified with the first occurrence of specific graphs. In this way [43] the denominator pattern can be predicted, and the expectation, based on the build-up of zeros on the unit circle in the complex x-plane as the permitted number of vertical bonds increases, that the underlying generating function is D-unsolvable *can be proved* [43].

7 Further Applications

In this section we briefly review further applications of SAW and SAP. The model may be enhanced by taking into account the number of nearest neighbour contacts. This extension of the model has been used to explain a variety of complex and important phenomena, including the collapse transition of polymers, the folding of proteins, the "knottedness" of ring polymers (in dimension greater than 2), and the enzyme action on DNA.

[5] It also suggests, but does not prove, that the generating function $C(x,y)$ has a natural boundary at $|x|=1$.

By associating a weight with the number of such contacts, configurations with more contacts can be made energetically favourable or unfavourable. If such contacts are made energetically highly favourable, this will clearly cause the SAW to "collapse". For two-dimensional SAW it is observed in this case that $\langle R \rangle_n^2 \sim n$. In the energetically neutral or unfavourable case, one observes $\langle R \rangle_n^2 \sim n^{\frac{3}{4}}$ and at a particular value of this energy, at the so-called *theta temperature*, a third type of behaviour is observed in which $\langle R \rangle_n^2 \sim n^{\frac{4}{7}}$. None of the above has been proved, but is widely accepted on the basis of compelling numerical evidence and theoretical arguments.

For a SAW on the square lattice, label the vertices $i = 0, 1, 2, \ldots$, and denote the coordinates of the i^{th} vertex $\mathbf{r}_i$. Two vertices are said to form a *contact* if $|\mathbf{r}_i - \mathbf{r}_j| = 1$ and $i \neq j$. We denote the number of SAW and SAP with n-steps and k contacts as $c_n(k)$ and $p_n(k)$ respectively. The corresponding generating functions, usually called partition functions are

$$Z_n(\beta) = \sum_k c_n(k) \exp(\beta k) \tag{7.1}$$

$$Z_n^0(\beta) = \sum_k c_n(k) \exp(\beta k). \tag{7.2}$$

It has been proved [48, 49] that the corresponding limiting free-energies

$$\kappa(\beta) = \lim_{n \to \infty} \frac{1}{n} \ln Z_n(\beta) \tag{7.3}$$

$$\kappa_0(\beta) = \lim_{n \to \infty} \frac{1}{n} \ln Z_n^0(\beta) \tag{7.4}$$

exist, and are equal to each other for $\beta \leq 0$. (In fact this has been proved for three-dimensional SAW and SAP, but the proof holds in any dimension.) For $\beta > 0$ the existence of $\kappa_0(\beta)$ has been proved, as has the fact [1] that it is monotonic and convex. Very recently [45], it has been proved that it is independent of k. Further, there exist functions $B_1(k)$ and $B_2(k)$, independent of n, such that

$$B_1(k) n^k p_n(0) \leq p_n(k) \leq B_2(k) n^k p_n(0) \tag{7.5}$$

for k fixed and n large. If the limit exists [45], then $0 \leq \lim_{n \to \infty} \frac{<k>_n}{n} < 1$. If $k = \alpha n$, the connective constant $\kappa(\alpha)$ exists, and is a concave function of α.

8 Conclusion

We have described improved algorithms for the enumeration of self-avoiding polygons and self-avoiding walks on the square lattice. The computational complexity of the SAP algorithm is estimated to be 1.2^n, and the complexity of the SAW algorithm is 1.316^n. Implementing these algorithms has enabled us to obtain polygons up to perimeter length 90, and SAW up to length 51. Decomposing the coefficients into prime factors reveals frequent occurrence of very large prime factors, supporting the widely held view that there is no "simple" formula for the coefficients. For example, p_{78} contains the prime factor 7789597345683901619. Our extended series enables us to give an extremely precise estimate of the connective constant, and we give a simple algebraic approximation that agrees precisely with our numerical estimate. An alternative analysis provides very strong evidence for the absence of any non-analytic correction terms to the proposed asymptotic form for the generating function for SAP, and provides compelling evidence for a correction-to-scaling exponent $\Delta = 1.5$ for SAW. We give asymptotic representations for the coefficients which we believe to be very precise.

We have shown how further numerical work can provide compelling support for the belief that the generating functions are not D-finite, which has subsequently been proved for SAP [43]. We have also outlined some further extensions of this rich model.

9 Acknowledgements

It is a pleasure to acknowledge Ian Enting's concept of the original algorithm that preceded this work, and Iwan Jensen's permission to describe his SAP algorithm in detail. This work was supported by the Australian Research Council. The authors acknowledge the use of the Intel Paragon XP/S 150 computer, located in the Oak Ridge National Laboratory CCS, funded by the DoE's MICS Division of the Office of Computational and Technology Research.

References

[1] R. Brak, A. J. Guttmann and S. G. Whittington, *On the behaviour of collapsing linear and branched polymers,* J. Math. Chem. **8**, (1991) 255-68.

[2] I Guim, H W J Blöte and T W Burkhardt, *Universality class of trails in two dimensions,* J. Phys A: Math. Gen. **30**, (1997) 413-21.

[3] J.L. Cardy and A.J. Guttmann, *Universal amplitude combinations for self-avoiding walks, polygons and trails,* J. Phys. **A26**, (1993) 2485-94.

[4] S R Shannon, T C Choy and R J Fleming, *Corrections to scaling in two-dimensional polymer statistics* Phys. Rev Rev. B **53** (1996) 2175-8.

[5] A. R. Conway and A. J. Guttmann, *Algebraic techniques for enumerating self-avoiding walks on the square lattice,* J. Phys. A:Math. Gen., **26**, (1993) 1519-34.

[6] A. R. Conway and A. J. Guttmann, *Square lattice self-avoiding walks and corrections to scaling,* Phys. Rev. Letts., **77**, (1996) 5284-7.

[7] I. G. Enting, *Series expansions from the finite lattice method,* J. Phys. A. **10**, (1977) 801-5.

[8] I. G. Enting and A. J. Guttmann, *Polygons on the square, L and Manhattan lattices,* J. Phys. A:Math. Gen., **18**, (1985) 1007-1017.

[9] I. G. Enting and A. J. Guttmann, *Polygons on the honeycomb lattice,* J. Phys. A:Math. Gen., **22**, (1989) 1371-84.

[10] I. G. Enting and A. J. Guttmann, (unpublished).

[11] M. E. Fisher and M. F. Sykes, *Excluded-volume problem and the Ising model of ferromagnetism*, Phys. Rev., **114** (1959) 45-58.

[12] A. J. Guttmann, *On two dimensional self-avoiding random walks,* J. Phys. A:Math. Gen., **17**, (1984) 455-68.

[13] A J Guttmann, *On the critical behaviour of self-avoiding walks,* J. Phys. A:Math. Gen., **20**, (1987) 1839-1854.

[14] A J Guttmann, *"Asymptotic analysis of coefficients in phase transitions and critical phenomena"* Vol 13. Eds. C. Domb and J. Lebowitz (Academic Press, London) (1989) 1-234.

[15] A. J. Guttmann, *Indicators of solvability for lattice models,* Disc. Maths., **217**, (2000) 167-189.

[16] A J Guttmann and I G Enting, *The size and number of rings on the square lattice,* J. Phys. A:Math. Gen., **21**, (1988) L165-172.

[17] A J Guttmann and I G Enting, *The number of convex polygons on the square and honeycomb lattices,* J. Phys. A:Math. Gen., **21**, (1988) L467-74.

[18] A J Guttmann and I G Enting, *On the solvability of some statistical mechanical systems,* Phys. Rev. Lett., **76**, (1996) 344-7.

[19] A J Guttmann and J S Wang, *The extension of self-avoiding random walk series in 2 dimensions,* J. Phys. A:Math. Gen., **24**, (1991) 3107-9.

[20] A J Guttmann and S G Whittington, *Two-dimensional lattice embeddings of connected graphs of cyclomatic index two,* J. Phys. A, **11**, (1978) 721-29.

[21] J. M. Hammersley, *The number of polygons on a lattice*, Proc. Camb. Phil. Soc., **57** (1961) 516-23.

[22] J. M. Hammersley and J. W. Morton, *Poor Man's Monte Carlo*, J. Royal. Stat. Soc. Ser. B, **16**, (1954) 23-38.

[23] J. M. Hammersley and D. J. A. Welsh, *Further results on the rate of convergence to the connective constant of the hypercubical lattice,* Q. J. Math. (Oxford, 2nd series), **13**, (1962) 108-10.

[24] T. Hara and G. Slade, *Self-avoiding walk in five or more dimensions. I. The critical behaviour*, Commun. Math. Phys. **147**, (1992) 101-36.

[25] T. Hara and G. Slade, *The lace expansion for self-avoiding walks in five or more dimensions*, Reviews in Math. Phys. **4**, (1992) 235-327.

[26] T. Hara, G. Slade and A.D. Sokal, *New lower bounds on the self-avoiding walk connective constant*, J. Stat. Phys. **72**, (1993) 479-517.

[27] S. Hemmer and P.C. Hemmer, *An average self-avoiding random walk on the square lattice lasts 71 steps*, J. Chem. Phys. **81**, (1984) 584-5.

[28] B.J. Hiley and M. F. Sykes, *Probability of initial ring-closure in the restricted random walk of a macromolecule,* J. Chem. Phys **34**, (1961) 1531-37.

[29] E. Hille and R. S. Phillips, *Functional Analysis and Semigroups*, A.M.S. Colloq. Publ. 31, (Providence, R.I.)

[30] B D Hughes, *Random walks and random environments, vol. 1* (1995) (Clarendon Press: Oxford).

[31] I. Jensen and A. J. Guttmann, *Self-avoiding polygons on the square lattice,* J. Phys. A:Math. Gen., f 32, (1999) 4867-76.

[32] H. Kesten, *On the number of self-avoiding walks,* J. Math. Phys. **4**, (1963) 960-9.

[33] H. Kesten, *On the number of self-avoiding walks II*, J. Math. Phys. **5** (1964) 1128-37.

[34] R. S. Lehman and G. H Weiss, *A study of the restricted random walk,* J. SIAM **6**, (1958) 257-78.

[35] D MacDonald, D L Hunter, K Kelly and N Jan, *Self-avoiding walks in two to five dimensions. Exact enumerations and series study,* J. Phys A: Math. Gen. **25** (1992) 1429-40.

[36] D. MacDonald, S. Joseph, D. L. Hunter, L. L. Mosely, N. Jan and A. J. Guttmann, *Self-avoiding walks on the simple-cubic lattice,* J. Phys. A, **33**, (2000) 5973-83.

[37] N Madras, *A rigorous bound on the critical exponent for the number of lattice trees, animals and polygons*, J. Stat. Phys., **78**, (1995) 681-99.

[38] N. Madras and G. Slade, *The Self-Avoiding Walk* (Birkhäuser, Boston–Basel–Berlin, 1993).

[39] B Masand, U Wilensky, J P Maffar and S Redner, *An extension of the two-dimensional self-avoiding walk series on the square lattice*, J. Phys A: Math. Gen. **25** (1992) L365-9.

[40] B Nienhuis, *Exact critical points and critical exponents of $O(n)$ models in two dimensions,* Phys. Rev. Lett. **49** (1982) 1062-5.

[41] G. L. O'Brien, *Monotonicity of the number of self-avoiding walks,* J. Stat. Phys., **59**, (1990) 969-79.

[42] W. J. C. Orr, *Statistical treatment of polymer solutions at infinite dilution,* Trans. Faraday Soc., **43**, (1947) 12-27.

[43] A. Rechnitzer, Ph. D. thesis, *Some problems in the counting of lattice animals, polyominoes, polygons and walks,* The University of Melbourne, (2001).

[44] H Saleur, *Conformal invariance for polymers and percolation,* J. Phys A: Math. Gen. **20** (1987) 455-70.

[45] C. E. Soteros and S. G. Whittington, *Contacts in self-avoiding walks and polygons*, J. Phys. A: Math. Gen. **34** (2001) 4009-40.

[46] M. F Sykes, *Some counting theorems in the theory of the Ising model and the excluded volume problem,* J. Math. Phys, **2** (1961) 52-62.

[47] M F Sykes, A J Guttmann, P D Roberts and M G Watts, *The asymptotic behaviour of self-avoiding walks and returns on a lattice,* J. Phys. A:Math. Gen., **5**, (1972) 653-60.

[48] M. C. Tesi, E. J. J. van Rensburg, E. Orlandini and S. G. Whittington, *Interacting self-avoiding walks and polygons in three dimensions*, J. Phys. A:Math. Gen. **29**, (1996) 2451-64.

[49] T. C. Yu, D. S. Gaunt and S. G. Whittington, *Free-energy of self-interacting uniform stars*, J. Phys. A:Math. Gen., **30**, (1997) 4607-14.