

The size and number of rings on the square lattice

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

1988 J. Phys. A: Math. Gen. 21 L165

(<http://iopscience.iop.org/0305-4470/21/3/009>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 129.252.86.83

The article was downloaded on 31/05/2010 at 15:35

Please note that [terms and conditions apply](#).

LETTER TO THE EDITOR

The size and number of rings on the square lattice

A J Guttmann[†] and I G Enting[‡]

[†] Department of Mathematics, University of Melbourne, Parkville, Victoria 3052, Australia

[‡] CSIRO Division of Atmospheric Research, Private Bag 1, Mordialloc, Victoria 3195, Australia

Received 10 November 1987

Abstract. We have calculated the number of self-avoiding polygons on the square lattice to 56 steps, and the caliper size to 54 steps. Analysis of the generating function permits estimates of the connective constant $\mu = 2.638\,1585 \pm 10^{-6}$ and the critical exponents $\alpha = 0.500\,06 \pm 0.000\,06$ and $\nu = 0.753 \pm 0.007$. The singularity structure of the polygon generating function is found to be consistent with a correction to scaling exponent $\Delta = 1.5$, as predicted by Nienhuis. The confluent part, however, maps into the additive analytic term due to the value of the exponent α .

This paper builds on earlier work of Enting (1980, hereafter referred to as I), Enting and Guttmann (1985, hereafter referred to as II) and Privman and Rudnick (1985). We previously obtained 46 terms of the polygon generating function (II), while Privman and Rudnick (1985) obtained 28 terms in the radius of gyration and caliper-size distribution series.

By rewriting our earlier program to most effectively utilise the virtual memory aspects of the operating systems on the various computers we used, and by judicious data structuring to reduce page faulting as far as possible, we have been able to obtain the number of square lattice polygons to 54 steps. Further, the number of 56-step polygons was obtained by adding the number of *convex* 56-step polygons, as discussed in Guttmann and Enting (1988, hereafter referred to as III). The sum of the spans of the polygons in a given lattice direction d_n and the sum of the square of the spans d_n^2 were obtained to 54 steps. The first and second moments of the caliper size are defined by

$$\langle D_n \rangle = d_n / p_n \quad \text{and} \quad \langle D_n^2 \rangle = d_n^2 / p_n \quad (1)$$

respectively, and one expects $\langle D_n^m \rangle \approx n^{m\nu}$.

The number of self-avoiding rings of n steps, u_n , per site of the square lattice includes only rings that cannot be mapped into one another by translation. Rings that are equivalent under rotations or reflections but not translations are regarded as distinct. The techniques of enumeration used in I and II classify rings according to the smallest rectangle that can be drawn around them. The notation $u_{n:i,j}$ specifies the number of rings for which the smallest circumscribing rectangle is $i \times j$ steps (in I we denoted the rectangles by the number of *sites* along each edge). The generating function g_{ij} is defined by

$$g_{ij} = \sum_n u_{n:i,j} x^n. \quad (2)$$

In I it was shown how transfer-matrix techniques could be used to evaluate G_{ij} , which is the generating function of rings that span the full length j of an $i \times j$ rectangle and fit within the width i . Thus

$$G_{ij} = \sum_{m \leq i} (i - m + 1) g_{mj} \quad (3a)$$

and

$$g_{ij} = G_{ij} - 2G_{i-1,j} + G_{i-2,j}. \quad (3b)$$

Because the symmetry $g_{mn} = g_{nm}$, determining G_{ij} (and hence g_{ij}) for $i \leq k$ will enumerate all rings of up to $4k + 2$ steps.

In order to determine the caliper or spanning moments, Privman and Rudnick (1985) list $s_{n:i}$ for $n \leq 28$ where

$$s_{n:i} = \sum_j u_{n:i,j}. \quad (4)$$

The corresponding generating function is

$$S_i(x) = \sum_n s_{n:i} x^n = \sum_j g_{ij}. \quad (5)$$

In order to determine $S_i(x)$ correctly to order x^{4k+2} the summation in (5) can be written as

$$S_i(x) = \sum_{j=1}^{2k+1-i} g_{ij} = \sum_{j=1}^{2k+1-i} (G_{i,j} - 2G_{i-1,j} + G_{i-2,j}) \quad i \leq k$$

and

$$S_i(x) = \sum_{j=1}^{2k+1-i} g_{ji} = G_{2k+1-i,j} - G_{2k-i,j} \quad i > k. \quad (6)$$

Hence the enumeration techniques described in I can be used to determine the $S_i(x)$ and consequently the caliper-moment series $\langle D \rangle$ and $\langle D^2 \rangle$ defined (Privman *et al* 1984) as

$$\langle D \rangle = \sum_i i S_i(x) \quad \text{and} \quad \langle D^2 \rangle = \sum_i i^2 S_i(x). \quad (7)$$

The determination of $S_i(x)$ does not require any more information than was required to determine the G_{ij} and in particular the sizes of the vectors involved in the transfer-matrix calculations are unchanged. We have thus been able to repeat and extend the calculations reported in II for rectangles up to width 13 steps, and so determine the caliper moments, in principle, to order x^{54} . The algorithm used is fully described in I and II. Because of the huge integers involved, the calculations have previously been done in modular arithmetic, modulo a large prime. Originally, this was repeated for distinct primes until sufficient information was obtained to reconstitute the coefficients. This meant that the program had to be run four or five times for each prime. As each run for 54-step polygons takes 30 d on a VAX 8650 (of this only 36 h CPU time is used, but with 6×10^7 page faults, the program is disc i/o bound) it is highly desirable to reduce the number of runs. This was done by using the method of differential approximants (originally called the recurrence relation method by its originators Guttman and Joyce (1972)) to predict subsequent coefficients. Given the series to 50 terms, say, the 52nd term is predicted by about ten different approximants. These

predictions are found to agree among themselves to 10 or 11 decimal digits, and this information is utilised to reduce the number of primes needed to two. Once the number of 52-step polygons was found in this way the process was iterated to find the number of 54-step polygons. Our procedure can be regarded as using differential approximants to give the most significant digits of p_n , while performing the transfer-matrix calculation modulo q gives the least significant digit in a base q representation of p_n . The validity of this approach is demonstrated by comparison with II. In that paper we predicted that $p_{48} \approx 5814\ 401\ 613\ 000\ 000$. From table 1 we now see that this is correct to all quoted non-zero digits.

However, the transfer-matrix calculations enumerate most of the polygons of 56 steps. The only exceptions are those that fit into rectangles whose minimal bounding rectangle has perimeter 56. We call polygons whose perimeter is equal to the perimeter of their minimal bounding rectangle *convex polygons* and they can be enumerated to any order as shown in III. We also accumulated the caliper spans d_n and d_n^2 . We give in table 1 a listing of all these quantities. Subsequently, a confirmatory run with a third prime verified all our predicted coefficients.

The resource demands of the enumeration program are quite considerable. Our program development (enumerating polygons to 50 steps) was carried out on a DEC Micro Vax II with 5 Mb of main memory and 160 Mb of disc capacity, running in single-user mode. The program was very slow to run, being limited by disc transfers.

Table 1. Coefficients of the polygon generating function p_{2n} , the sum of the linear spans of all n -step polygons d_{2n} and the sum of the squared linear spans d_{2n}^2 .

n	p_{2n}	d_{2n}	d_{2n}^2
2	1	1	1
3	2	3	5
4	7	14	30
5	28	70	186
6	124	370	1 164
7	588	2 028	7 344
8	2 938	11 452	46 732
9	15 268	66 172	299 604
10	81 826	389 416	1 932 900
11	449 572	2 326 202	12 537 542
12	2 521 270	14 070 268	81 705 782
13	14 385 376	86 010 680	534 663 812
14	83 290 424	530 576 780	3 511 466 838
15	488 384 528	3 298 906 810	23 136 724 382
16	2 895 432 660	20 653 559 846	152 888 000 934
17	17 332 874 364	130 099 026 600	1 012 925 595 468
18	104 653 427 012	823 979 294 284	6 726 766 841 438
19	636 737 003 384	5 244 162 058 026	44 767 880 394 634
20	3 900 770 002 646	33 523 117 491 920	298 522 284 224 824
21	24 045 500 114 388	215 150 177 410 088	1 994 187 071 224 056
22	149 059 814 328 236	1 385 839 069 134 800	13 343 458 013 982 572
23	928 782 423 033 008	8 956 173 544 332 434	89 420 609 914 983 270
24	5 814 401 613 289 290	58 056 703 069 399 056	600 088 861 692 380 354
25	36 556 766 640 745 936	377 396 656 568 011 618	4 032 377 457 462 571 694
26	230 757 492 737 449 636	2 459 614 847 765 495 754	27 129 080 022 839 863 318
27	1 461 972 662 850 874 916	16 068 572 108 927 106 202	182 726 605 700 552 001 998
28	9 293 993 428 791 901 042		

By changing to a Webster disc controller with 2 Mb of cache memory, execution speed was improved by a factor of 2.2. This still implied a prohibitive run time at the next level of 54-step polygons, which requires a single array of over 100 Mb.

The program was then transferred to a DEC VAX 8650, a general purpose time-sharing machine. The program ran in 30 d, though requiring only 36 h of CPU time, and amassed 60 million page faults. The program was run again, with another prime, on a newly installed Cyber 990, with an early, and slowish, version of the Fortran compiler. To keep storage within bounds, i.e. arrays no bigger than 100 Mb, requires 2-byte integers. These were created on the Cyber by equivalencing the array to an array of (2-byte) characters. Using regular integers would have required 0.5 Gb of virtual memory. The computer was only moderately loaded, and the program ran in 5 d, taking about 7.7 h of CPU time, and 13.5 million page faults were recorded.

To extend the series further is feasible in principle but, as the foregoing shows, massive computing resources are needed. In fact, the speed of a supercomputer is not the principal requirement; what is needed is rather a computer with large amounts of physical rather than virtual memory. With such an architecture, a microcomputer CPU chip such as a Motorola 68020 or Intel 80286/80386 could probably match or even outperform a supercomputer in terms of turnaround time, at least for the calculations to date. In the absence of such a machine it is not considered realistic to extend the series further unless some remarkable benefit were to be attached to two further terms. Additionally our existing program would require a number of minor modifications to enumerate longer series because control vectors and other parameters of the program would exceed the range of the usual 32-bit integer.

The remarkable length of these series follows from the computational complexity of the algorithm used. The complexity is dominated by the calculation of a vector of partial generating functions. Each component of the vector corresponds to a distinct allowable way in which a polygon can cut a line of cross section in the bounding rectangle. The components of the vector hold polynomials, or truncated power series. In topological terms, the definition of 'allowable' is given in I.

Algebraically, each allowable intersection corresponds to an ordered m -tuple $\langle n_1, \dots, n_m \rangle$ subject to $n_i = 0, 1, 2$ where $m = \text{width (in bonds)} + 2$. The number of 1 must equal the number of 2, and reading from left to right the number of 2 can never exceed the number of 1. The algorithm evolves by moving the cross-sectional line so as to include one additional site, and generating a new vector from the old. These algebraic constraints can be shown to give rise to a sequence which grows exponentially at a rate that is asymptotically 3^n , where n is the width of the containing rectangle. The precise size of the vectors can be determined by considering m -tuples of 0, 1, 2 such that, reading from left to right, the number of 2 never exceeds the number of 1. If we let r_{mn} be the number of such m -tuples with n more 1 than 2 then

$$r_{10} = 1 \quad (\langle 0 \rangle) \quad (8a)$$

$$r_{11} = 1 \quad (\langle 1 \rangle) \quad (8b)$$

$$r_{1n} = 0 \quad n > 1 \quad (8c)$$

$$r_{mn} = r_{m-1,n-1} + r_{m-1,n} + r_{m-1,n+1} \quad n \geq 1 \quad (9a)$$

$$r_{m0} = r_{m-1,0} + r_{m-1,1} \quad (9b)$$

The size of the vector required for the enumeration of polygons of width W is $r_{w+2,0}$. In practice we use vectors that are slightly larger because we are hash addressing, with

the m -tuples as a key, to access the vector elements. We can show that $r_{w+2,0}$ is the coefficient of x^{w+1} in the expansion of $y(x)$, defined by

$$y(x) = \frac{1}{2x^3} [1 - x - 2x^2 - (1 - 2x - 3x^2)^{1/2}]. \tag{10}$$

Increasing the width by 1 allows four additional steps in the polygon, and so the growth rate is $3^{1/4}$ or 1.316... This can be compared to the connective constant of 2.638..., which would be the growth rate of a conventional counting algorithm. Thus at 56 steps the algorithm we have used is faster by a factor of about 10^{12} . Thus a conventional algorithm would have to run for a time comparable to the lifetime of the universe on the world's fastest computer to achieve the same results as those presented here.

We have analysed the three series by the method of differential approximants, utilising the scheme developed in Guttman (1987). In table 2 we give the (unbiased) estimates of the exponent $2 - \alpha$, and critical point x_c of the polygon generating function, as given by inhomogeneous first- and second-order differential approximants, denoted $K = 1$ and 2 respectively. We combine these entries using the method described in Guttman (1987), to give the estimates

$$\begin{aligned} x_c^2 &= 0.143\ 680\ 56\ (6) & 2 - \alpha &= 1.500\ 06\ (5) & K &= 1 \\ x_c^2 &= 0.143\ 680\ 52\ (5) & 2 - \alpha &= 1.500\ 10\ (5) & K &= 2. \end{aligned} \tag{11}$$

These estimates include 'errors' of two standard deviations, shown as the parenthesised uncertainty in the last digit. They do not, however, take account of the trend evident

Table 2. Summary of exponent and critical point estimates using first- and second-order unbiased approximants. Biased critical point estimates were obtained by linear regression assuming the exponent is $\frac{3}{2}$ exactly. The right-hand column shows the number of distinct approximants used in the estimates given in the corresponding row.

n	Critical point x_c^2	Exponent $2 - \alpha$	Biased x_c^2	Number of approximants
First-order approximants				
18	0.143 680 26 (114)	1.500 30 (70)	0.143 680 74	12
19	0.143 680 07 (130)	1.500 41 (83)	0.143 680 70	12
20	0.143 680 09 (27)	1.500 39 (18)	0.143 680 65	11
21	0.143 680 48 (38)	1.500 13 (34)	0.143 680 66	7
22	0.143 680 60 (23)	1.500 05 (19)	0.143 680 66	10
23	0.143 680 09 (27)	1.500 39 (187)	0.143 680 64	11
24	0.143 680 397 (238)	1.500 181 (173)	0.143 680 646	6
25	0.143 680 610 (207)	1.500 034 (170)	0.143 680 651	7
26	0.143 680 509 (237)	1.500 107 (214)	0.143 680 627	8
27	0.143 680 563 (96)	1.500 062 (84)	0.143 680 637	6
28	0.143 680 624 (71)	1.500 007 (68)	0.143 680 631	10
Second-order approximants				
22	0.143 680 039 (900)	1.500 414 (550)	0.143 680 677	8
23	0.143 680 108 (384)	1.500 376 (250)	0.143 680 681	8
24	0.143 680 381 (86)	1.500 197 (65)	0.143 680 630	6
25	0.143 680 395 (212)	1.500 192 (161)	0.143 680 647	7
26	0.143 680 544 (224)	1.500 072 (194)	0.143 680 628	5
27	0.143 680 603 (96)	1.500 023 (83)	0.143 680 630	7
28	0.143 680 635 (92)	1.499 997 (95)	0.143 680 632	8

in table 2 in which the estimates of x_c increase and those of $2 - \alpha$ decrease with increasing order. These trends are slight, but clearly sufficient to move $2 - \alpha$ to precisely $\frac{3}{2}$. If we calculate x_c assuming this value of the exponent, we obtain, by linear regression as discussed in Guttman (1987) the results shown in the third column. These are combined to give

$$\begin{aligned} x_c^2 &= 0.143\,680\,64\,(3) & K &= 1 \\ x_c^2 &= 0.143\,680\,63\,(1) & K &= 1. \end{aligned} \quad (12)$$

This estimate gives for the connective constant $\mu = 2.638\,1585\,(1)$. This may be compared with our biased estimate based on the SAW series of $\mu = 2.638\,1559\,(15)$, and our earlier estimate, based on a 46-term series and an alternative method of analysis, of $\mu = 2.638\,155\,(25)$. These estimates do not quite include our latest estimate, which has an incredibly small uncertainty associated with it. This slight discrepancy is due to the aforementioned trends of increasing estimates with increasing order. However, we wish to use a reproducible algorithmic analysis method, and so we have not included the effect of this trend, which would involve some *ad hoc* correction to the above estimate. Rather, we propose to widen our error bars tenfold, which still gives the very precise estimate

$$\mu = 2.638\,1585 \pm 0.000\,0010. \quad (13)$$

In an earlier paper (Guttman 1984), one of us suggested various surds that could correspond to the exact connective constant. Our result (13) effectively rules out all of these.

We turn now to the analysis of the series for the caliper span. Privman and Rudnick obtained 28 terms in these series, and pointed out that non-analytic correction terms to the expected asymptotic forms can be expected. They therefore focused on the radius of gyration series for their estimate of ν . We have analysed the series for d_n and d_n^2 by using our estimate of μ above, and hence have obtained biased estimates of ν . The alternative, of dividing term-by-term by p_n , has the effect of increasing the influence of confluent corrections. Since we have a highly accurate value of μ , we believe that the method we used is to be preferred. The results of this biased analysis applied to the series for d_n are shown in table 3 for the first- and second-order approximants. From the series for d_n we obtain the exponent estimates 0.744 ± 0.002 ($K = 1$) and 0.748 ± 0.005 ($K = 2$). This exponent is $2 - \alpha - \nu$. Hence assuming $2 - \alpha = \frac{3}{2}$ exactly, as discussed above, we obtain $\nu = 0.752 \pm 0.006$ encompassing both sets of estimates. The same analysis applied to the series for d_n^2 yields for the exponent $2 - \alpha - 2\nu$ the estimate -0.0010 ± 0.0010 ($K = 1$) and -0.005 ± 0.005 ($K = 2$), so that $\nu = 0.755 \pm 0.005$. The results from the two series may be combined to give $\nu = 0.753 \pm 0.007$. This is in excellent agreement with the exact value of $\frac{3}{4}$ (Nienhuis 1982, 1984), and the result of Privman and Rudnick (1985) from the radius of gyration of 0.750 ± 0.0015 , and our earlier estimate of 0.750 from the mean-square end-to-end distance series. Clearly, however, these are not the best series to estimate ν .

In our earlier paper II, we devoted considerable effort to the analysis of the series in order to find a confluent exponent. Each method we tried seemed to give a different value for the correction to scaling exponent, though the value $\Delta = 0.84$ was marginally favoured. With our longer series, we no longer find any convincing evidence of a particular non-analytic correction to scaling term. Indeed, the differential approximants prefer a single root at x_c to a double root (which latter would correspond to a confluent exponent). This suggests that the only correction term is analytic, which is in precise

Table 3. Square lattice polygon linear-span generating function analysis. Biased differential approximants $[L/N + \Lambda; N]$, and $[L/N + \Lambda; N + \Lambda, N]$, $\Lambda = -1, 0, 1$, with critical point fixed as described in the text. Defective approximants are marked with an asterisk.

		<i>N</i>									
Λ		4	5	6	7	8	9	10	11	12	
First-order approximants ($K = 1$)											
$L = 1$	-1	0.7039*	0.7226	0.7273	0.7369	0.7607*	0.7422	0.6958*	0.7441	0.7349*	
	0	0.7159	0.7164*	0.7359	0.7376	0.7394	0.7233*	0.7389*	0.7443	0.7450	
	1	0.7231	0.7231	0.7375	0.7373	0.7386	0.7152*	0.7450	0.7450		
$L = 2$	-1	0.7564	0.7385	0.7364	0.7375	0.7368	0.7304	0.7405	0.7437	0.7326*	
	0	0.7387	0.7832	0.7367	0.7382	0.7398	0.7031*	0.7415	0.7430		
	1	0.7385	0.7366	0.7375	0.7379	0.7991*	0.7383	0.7560	0.7450		
$L = 3$	-1	0.7137	0.7287	0.7373	0.7402	0.7405	0.7457	0.7439	0.7431		
	0	0.7385	0.7337	0.7410	0.7408	0.7414	0.7444	0.7440	0.7434		
	1	0.7215	0.7368	0.7401	0.7405	0.7363	0.7441	0.7429			
$L = 4$	-1	0.7386	0.7361	0.7426	0.7403	0.7425	0.7257*	0.7439	0.7449		
	0	0.7383	0.7378	0.7401	0.7407	0.7434	0.7440	0.7404			
	1	0.7365	0.7417	0.7402	0.7422	0.7425	0.7448	0.7462			
$L = 5$	-1	0.7326	0.7373	0.7410	0.7415	0.7434	0.7435	0.7431			
	0	0.7364	0.7432	0.7408	0.7462	0.7434	0.7441	0.7461			
	1	0.7314	0.7404	0.7412	0.7434	0.7435	0.7795*				
$L = 6$	-1	0.7361	0.7399	0.7407	0.7412	0.7442	0.7444	0.7448			
	0	0.7360	0.7402	0.7415	0.7432	0.7435	0.7493				
	1	0.7508	0.7397	0.7443	0.7438	0.7454	0.7400				
$L = 7$	-1	0.7364	0.7403	0.7395	0.7434	0.7434	0.7489				
	0	0.7390	0.7404	0.7431	0.7440	0.7439	0.7479				
	1	0.7402	0.7407	0.7433	0.7435	0.7492					
$L = 8$	-1	0.7373	0.7398	0.7429	0.7468	0.7501	0.7446				
	0	0.7407	0.7415	0.7432	0.7432	0.7473					
	1	0.7271	0.7428	0.7419	0.7441	0.7454					
Second-order approximants ($K = 2$)											
$L = 1$	-1	0.7353	0.7381*	0.7431	0.7472	0.7460					
	0	0.7370	0.7415	0.7488	0.7436						
	1	0.7423	0.7487	0.7460	0.7457						
$L = 2$	-1	0.7376	0.7349*	0.7487	0.7565	0.7460					
	0	0.7362	0.7421	0.7466	0.7458						
	1	0.7389	0.7489	0.7441							
$L = 3$	-1	0.7359	0.7404	0.7429	0.8349*						
	0	0.7338	0.7408	0.7445	0.7459						
	1	0.7414	0.7439	0.7459							
$L = 4$	-1	0.7334	0.7434	0.7433	0.7449						
	0	0.7378	0.7428	0.7420							
	1	0.7455	0.7447	0.7456							
$L = 5$	-1	0.7330	0.7439	0.7395*	0.7422						
	0	0.7473	0.7436	0.7430							
	1	0.7432	0.7395*								

Table 3. (continued)

	Λ	N									
		4	5	6	7	8	9	10	11	12	
$L = 6$	-1	0.7356	0.7435	0.7424							
	0	0.7381	0.7449	0.7361*							
	1	0.7440	0.7451								
$L = 7$	-1	0.7424	0.7445	0.7455							
	0	0.7439	0.7474								
	1	0.7451	0.7456								
$L = 5$	-1	0.7452	0.7454	0.7469							
	0	0.7446	0.7394*								
	1	0.7636*									

agreement with Nienhuis's prediction of a correction-to-scaling exponent of $\Delta = 1.5$. This apparently contradictory statement is in fact entirely consistent, as the leading exponent is precisely $\frac{3}{2}$, so the confluent contribution simply adds in to the additive analytic background term.

Such a confluent term does, however, give rise to a non-analytic correction to scaling term in the SAW generating function. Fitting the SAW series to the implied asymptotic form gives entirely consistent and satisfactory results, and it is proposed to report on this analysis subsequently. For the moment, we believe that there is little more numerical work that is worthwhile doing on this problem. The critical exponents given by Nienhuis are undoubtedly correct, and the connective constant has been attained to a degree of accuracy rivalling that of the fine-structure constant.

We would like to thank Webster Computer Corp Pty Ltd for the loan of their excellent disc controller, and to record our gratitude for the considerable assistance rendered by Darryl Chivers, David Pennington and John Wastell of the University Computing Services in helping us run our resource-heavy program on a variety of computers. This work was supported by the Australian Research Grants Scheme through a grant to one of us (AJG).

References

- Enting I G 1980 *J. Phys. A: Math. Gen.* **13** 3713
 Enting I G and Guttman A J 1985 *J. Phys. A: Math. Gen.* **18** 1007
 Guttman A J 1984 *J. Phys. A: Math. Gen.* **17** 455
 — 1987 *J. Phys. A: Math. Gen.* **20** 1839
 Guttman A J and Enting I G 1988 *J. Phys. A: Math. Gen.* submitted
 Guttman A J and Joyce G S 1972 *J. Phys. A: Gen. Phys.* **5** L81
 Nienhuis B 1982 *Phys. Rev. Lett.* **49** 1062
 — 1984 *J. Stat. Phys.* **34** 731
 Privman V, Family F and Margolina A 1984 *J. Phys. A: Math. Gen.* **17** 2837
 Privman V and Rudnick J 1985 *J. Phys. A: Math. Gen.* **18** L789