J. Phys. A: Math. Gen. 25 (1992) L365–L369. Printed in the UK

LETTER TO THE EDITOR

An extension of the two-dimensional self-avoiding walk series on the square lattice

Brij Masand†, Uri Wilensky†, J P Massar† and S Redner‡

† Thinking Machines Corporation, Cambridge, MA 02142, USA
‡ Center for Polymer Studies and Department of Physics, Boston University, Boston, MA 02215, USA

Received 20 September 1991

Abstract. We have adapted the dimerization algorithm for enumerating self-avoiding walks to exploit the parallelism of the Connection Machine System®. By this approach we have extended the series for the number of self-avoiding walks on the 2D square lattice by five terms. These data may permit more accurate estimates of critical properties.

The enumeration of self-avoiding walks in an efficient manner is a problem of continuing interest. A standard approach for enumeration is based on a 'backtracking' algorithm [1] in which all n-step SAWs are used as a base upon which all $(n+1)$-step SAWs are built (an example of Fortran code for enumeration can be found in [2]). A useful alternative approach is the 'dimerization' algorithm, first developed by Alexandrowicz [3] in the context of Monte Carlo simulations, and introduced by Torrie and Whittington for enumeration [4] and then recently rediscovered by Wang [5]. In dimerization, one constructs all SAWs of length n by the backtracking method, for example, and then constructs all SAWs of length $2n$ by joining together all n-step pairs of SAWs which do not lead to additional self-intersections. This method was used very recently by Guttman and Wang [6] to extend the enumeration of SAWs on the square lattice by two terms to $n=29$. In this letter, we report the further extension of the square lattice series for the number of SAWs by five terms up to $n=34$. Results for the mean-square end-to-end distance, and also series for other lattices will be discussed in a future publication.

Our calculations were performed on a Thinking Machines CM-2 massively parallel supercomputer§. Guttman and Wang report that their calculations (performed on a Masscomp 5700) took somewhat less than 700 hours for the square lattice. Taking this time to refer to the calculation of the data for $n=28$ and $n=29$ only, our calculation for the number of SAWs took approximately 2 hours. We estimate that the inclusion of the endpoint distances (performed by Guttman and Wang) would increase the computation time to approximately 6 hours for the equivalent calculation. To extend the series for the number of SAWs by 5 more terms, up to $n=34$, required another $\sim$100 hours of calculation.

Undoubtedly, the most significant factor contributing to the relative speed of our enumeration program is the exploitation of the massively parallel architecture of the CM-2. To this end, we parallelized the dimerization algorithm.

§ Actually, several CMs of various sizes (e.g. 4k, 8k, 16k processors) were used. All time calculations in this paper are normalized to a 64k CM.

0305-4470/92/070365+05$04.50

The CM-2 is a massively parallel supercomputer with a maximum of $64k$ physical processors. Communication to the CM is performed through a front-end computer (in this case a SUN workstation) which broadcasts instructions to all $64k$ processors simultaneously. Computations are done in a data-parallel manner by loading each processor with a different element of data and then having all the processors execute the same† instructions on their own data. The number of processors can be made to appear larger than $64k$ by dividing up each physical processor into v virtual processors‡ where v is called the *vp-ratio*. The CM supports CM-Fortran, C*, and *Lisp, parallel extensions of the corresponding scalar languages. Our program for enumerating SAWs is written in *Lisp. A further substantial increase in the speed of our program can likely be achieved if the program is rewritten in a lower-level language.

The algorithm consists of the following steps. To enumerate SAWs of length n, first a 'split' point, $m < n$, is selected. Then all SAWs of length m, generated by a parallelized version of the backtracking method, are loaded into the CM, one SAW per processor§. In this parallel version of backtracking, each processor independently constructs the next step and then branches to three (or fewer) new processors, corresponding to the allowable directions for the next step, until the CM is filled with complete walks. This version of backtracking takes very little time to fill the CM. After the processors are filled with SAWs of length m, the front end generates a single SAW of length $n-m$. This SAW is then broadcast simultaneously to each processor. Each processor, in parallel, performs an intersection check between its two segments of the n step walk to determine if indeed a valid SAW has been formed. The front end steps serially through all SAWs of length $n-m$. The simultaneous intersection check with the m-step SAWs already stored in the CM memory permits a substantial speedup over a scalar computation.

In the serial version of the dimerization process, m is in principle arbitrary, but we expect the greatest efficiency when m is close to $n/2$. However, in the parallel version of dimerization, we expect greater efficiency when m is large. This is due to two factors: (1) the parallel backtracking method fills the CM memory quickly, and (2) having more walks in the CM memory increases the speed of the computation since the check for intersection of one walk on the front end with all walks in memory requires a fixed amount of time. Essentially, the execution time of the algorithm is proportional to c_{n-m}. We therefore choose m to be as large as will fit in the CM memory. The number of SAWs that fit in memory is determined by the number of physical processors (in our case $64k$), the amount of memory on each processor (in our case $256k$), and the *vp*-ratio. For example the calculation for c_{29} was done with $m = 17$. A more detailed discussion of the algorithmic trade-offs is beyond the scope of this letter and will be discussed elsewhere. To manage the computation over time, we have also divided the computation into sub-tasks, each sub-task‖ consisting of the computation of the continutations of stored SAWs of length p, for some small p.

† The latest version of the Connection Machine, the CM5, allows different processors to execute different instructions.

‡ Virtual processors are an abstraction in a data-parallel language, that is transparent to the user and provides a view of physical processors in software that enables them to be seen as *vp*-ratio times the number of physical processors. They can be implemented by dividing up the memory of each processor by the *vp*-ratio and then looping over the different data elements in each sub-division.

§ Actually, symmetry considerations allow one to load only a quarter of the SAWs, but that will be ignored in the following.

‖ The actual *vp*-ratio was chosen with respect to the number of SAWs that were needed to be stored by each sub-task. For example for an $8k$ machine we used a *vp*-ratio of 16, with 71 sub-tasks.

The new terms for the number of SAWs are

$$c_{30} = 16\,741\,957\,935\,348$$

$$c_{31} = 44\,673\,816\,630\,956$$

$$c_{32} = 119\,034\,997\,913\,020$$

$$c_{33} = 317\,406\,598\,267\,076$$

$$c_{34} = 845\,279\,074\,648\,708.$$

As a check of our results, we verified the previously published data of Guttmann and Wang, as well as reproducing our results by using two different split points for dimerization.

Owing to the length of the series for the number of SAWs, a rudimentary, although biased, analysis method [7] yields an estimate for the connective constant which is comparable to that obtained by the more sophisticated analysis methods of Guttmann [8] based on the series with five fewer terms. We assume that

$$c_n \sim A n^{\gamma-1} \mu^n (1 + B_1/n^{\phi_1} + B_2/n^{\phi_2} + \ldots) \tag{1}$$

where $\phi_i > 0$, and where the correction terms are assumed to arise from lower order confluent singularities. Now using the conjectured value [9] of $\gamma = \frac{43}{32}$, we then analyse the reduced series $d_n = c_n/n^{\gamma-1}$, for which we focus on estimating the value of the connective constant μ.

For the analysis of the reduced series, we form the ratios of alternating successive terms $r_n = (d_{n+2}/d_n)^{1/2}$. We use the alternating terms in forming the ratio to reduce the magnitude of the even–odd oscillations, characteristic of a loose-packed lattice, in subsequent extrapolations. According to the hypothesized asymptotic form of the series d_n, the ratios r_n should vary as

$$r_n \sim \mu(1 + C_1/n^{\alpha_1} + \ldots) \tag{2}$$

as $n \to \infty$, with $\alpha_i = 1 + \phi_i$ strictly greater than unity. To estimate the value of μ, we now form the extrapolants

$$s_n = [n^\alpha r_n - (n-2)^\alpha r_{n-2}]/[n^\alpha - (n-2)^\alpha] \tag{3}$$

that is, the intercept at $1/n^\alpha = 0$ of the straight line that passes through r_n and r_{n-2} when the data is plotted against $1/n^\alpha$ (figure 1(*a*)). These points should converge to the value of μ as $n \to \infty$. In this plot, the averages of two successive terms, $s^{(1)}n = 1/2(s_n + s_{n-1})$ are also shown. In our extrapolation, we have found that the empirical choice $\alpha = 2$ gives the most rapid convergence of the extrapolants to the asymptotic limit. Already at this initial analysis stage, we estimate μ to be close to 2.638 15 with a subjective uncertainty, based on visually extrapolating the envelope of the oscillations in s_n, of 0.000 01 or less. Owing to the even–odd oscillations which still persist in $s^{(1)}n$, we perform two additional successive averages of consecutive terms in $s^{(1)}n$, thereby forming $s^{(2)}n$ and $s^{(3)}n$. The latter sequence has virtually no even–odd oscillations and this aspect facilitates additional extrapolations (figure 1(*b*)).

If r_n actually has the form $r_n \sim \mu(1 + C_1/n^2 + C_2/n^3)$, as suggested by the analysis of $s^{(1)}n$, then the sequence $s^{(3)}n$ will approach its limiting value as $1/n^3$. Extrapolating $s^{(3)}n$ in this way, we find that the estimates for the intercept at $1/n^3 = 0$ lie on a relatively straight line (figure 2). The linear extrapolation of the last three pairs of successive data points are all within 0.000 0002 of our estimate $\mu = 2.638\,1589$. This

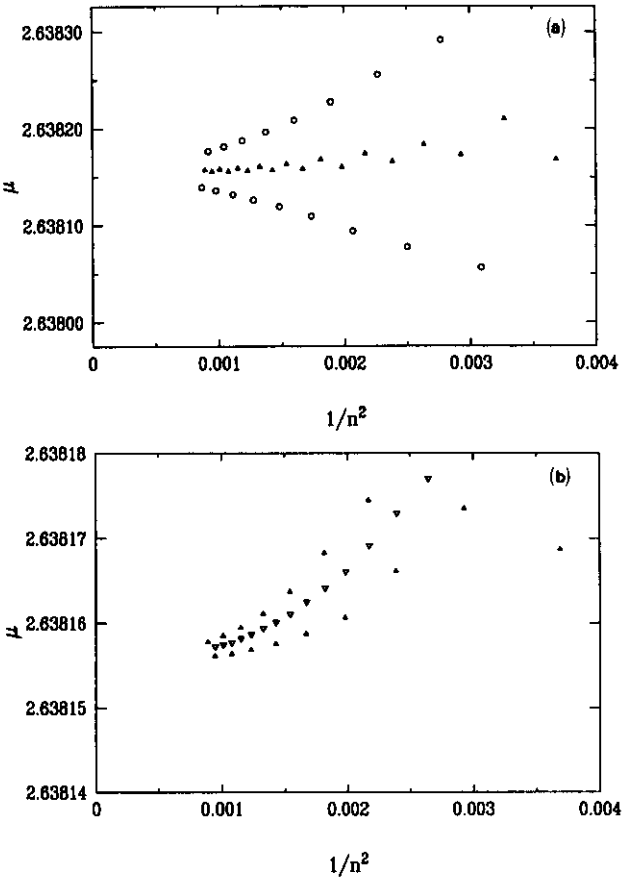

Figure 1. (*a*) Plot of s_n (○) and $s^{(1)}n$ (▲) against $1/n^2$, and (*b*) on a finer vertical scale a plot of $s^{(1)}n$ and $s^{(3)}n$ (▽).

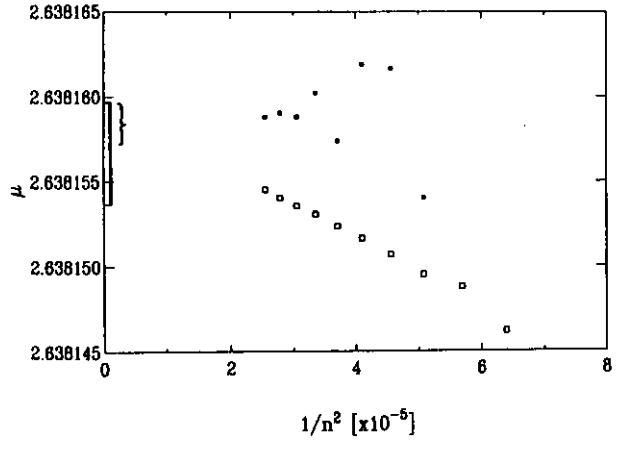

Figure 2. Extrapolation of $s^{(3)}n$ against $1/n^3$ (□), and a second extrapolation versus $1/n^3$ (■). The square bracket and the brace indicate, respectively, the estimated range of value for μ given by Guttmann and Wang and by Guttmann and Enting.

would be a reasonable estimate of the uncertainty in the value of μ, if our analysis is based on correct assumptions for the form of the various corrections to the asymptotic behaviour of the series. Our estimate for μ is consistent with the value $\mu = 2.638\,1563 \pm 0.000\,0034$, quoted by Guttmann and Wang [6], and also with the value $\mu = 2.638\,1584 \pm 0.000\,0014$ quoted by Guttmann and Enting [10] based on analysis of a 56-term series for the number of self-avoiding polygons, determined by an algorithm specialized specific to the polygon problem.

In conclusion, we have developed a new algorithm based on parallelizing the dimerization method to extend the series for the number of self-avoiding walks on the square lattice to 34 terms. These series appear to be long enough to give very accurate estimates for critical parameters with only rudimentary analysis methods. In future work, we will report on extended series for other lattices, as well as the results of a more thorough analysis.

The authors wish to acknowledge Alex Kushkuley, Roger Frye, and Mario Bourgoin for helpful discussions regarding the enumeration algorithm on the CM. SR thanks the ARO and NSF for partial support of this work.

Note added. After this work was completed we received a preprint by Mertens and Lautenbacher (to appear in *J. Stat. Phys.*) in which they describe a method for enumerating lattice animals by distributing the enumeration process over several workstations.

References

[1] Martin J L 1974 *Phase Transitions and Critical Phenomena* vol 3, ed C Domb and M S Green (New York: Academic)
[2] Redner S 1982 *J. Stat. Phys.* **29** 309
[3] Alexandrowicz Z 1969 *J. Chem. Phys.* **51** 561
[4] Torrie G and Whittington S G 1975 *J. Phys. A: Math. Gen.* **8** 1178
[5] Wang J 1989 *J. Phys. A: Math. Gen.* **22** L969
[6] Guttmann A and Wang J 1991 *J. Phys. A: Math. Gen.* **24** 3107
[7] Gaunt D S and Guttmann A J 1974 *Phase Transitions and Critical Phenomena* vol 3, ed C Domb and M S Green (New York: Academic)
[8] Guttmann A J 1989 *Phase Transitions and Critical Phenomena* vol 13, ed C Domb and J L Lebowitz (New York: Academic)
[9] Nienhuis B 1982 *Phys. Rev. Lett.* **49** 1062
Nienhuis B 1984 *J. Stat. Phys.* **34** 731
[10] Guttmann A J and Enting I G 1988 *J. Phys. A: Math. Gen.* **21** 1165