Recursive sampling of random walks: self-avoiding walks in disordered media

1993 J. Phys. A: Math. Gen. 26 1023

(http://iopscience.iop.org/0305-4470/26/5/022)

J. Phys. A: Math. Gen. 26 (1993) 1023–1036. Printed in the UK

Recursive sampling of random walks: self-avoiding walks in disordered media

Peter Grassberger

Physics Department, University of Wuppertal D-5600 Wuppertal 1, Federal Republic of Germany

Received 13 July 1992, in final form 11 November 1992

Abstract. We present a new algorithm for simulating random walks which is simple, versatile and efficient. It uses recursive function calls and can be used to obtain unbiased samples with any given length distribution. This makes it particularly useful in disordered geometries where the effective connectivity constant is not known *a priori*. When applying it to self-avoiding random walks on two-dimensional media with quenched randomness, we find evidence for at least two new renormalization group fixed points.

1. Introduction

Self-avoiding random walks (SAWs) are one of the simplest critical phenomena. Being a model for randomly bent linear polymers, they have been studied in great detail with various methods, among them exact enumerations [1–3] and Monte Carlo simulations [4]. The most important parameters describing SAWs are the critical exponents ν and γ, and the connectivity constant μ. They are defined via the number Z_N of different SAWs of length N and their average end-to-end distance $R_N = [\langle R^2 \rangle]^{1/2}$ as

$$Z_N \sim \mu^N N^{\gamma-1} \qquad R_N \sim N^\nu. \tag{1}$$

In two dimensions, the critical exponents are known exactly to be $\nu = 3/4$ and $\gamma = 43/32$ [5]. This agrees with the values obtained from extrapolations of exact enumerations which also give the most precise value for μ.

Monte Carlo simulations, on the other hand, seem the most efficient method for specific geometries and in three dimensions. Among the numerous proposed methods, the pivot algorithm and its modifications [4] seem to be the fastest in the absence of walls or other obstacles, but they become very inefficient in complicated geometries. In such cases, the method proposed by Beretti and Sokal [6] seems to be the most efficient one known so far.

In the present paper we shall introduce a new algorithm which is similar to the incomplete enumeration method of [7, 8], which is itself similar to the Beretti–Sokal algorithm. However, using a recursive function seems more promising as it seems to be easier to apply, more flexible and more intuitive. Thus one of its main advantages will be that it can be easily applied in complex geometries.

To illustrate this, we shall apply it to the problem of SAWs in media with quenched randomness. This has attracted much attention in recent years [9–26]. On the one

0305-4470/93/051023+14$07.50

hand, it is of interest as a model for polymers in disordered media, on the other hand it serves as a playground for testing methods devised for dealing with random systems [10, 12]. The results of these studies so far have been frustratingly inconclusive, and they have not yet lead to any agreement between different researchers.

Following the above authors, we model the randomness by excluding randomly chosen sites on a regular lattice. The fraction of excluded sites is p. Let us denote by $Z_N(\mathcal{C})$ the number of walks starting from the origin in configuration $\mathcal{C}$ of the randomness. This 'partition sum' is a random variable whose distribution does not depend on the origin due to translational invariance. As noted in [10], its average value is given trivially by $\langle Z_N(\mathcal{C})\rangle = p^N Z_N^{(0)}$, where $Z_N^{(0)}$ is the 'ordered partition sum' used in equation (1). Also, the weighted average end-to-end distance ('annealed average')

$$\langle R^2\rangle = \frac{1}{\sum_{\mathcal{C}} Z_N(\mathcal{C})} \sum_{\mathcal{C}} \sum_{\text{SAWs}} R^2 \tag{2}$$

is trivially the same as that for the ordered case, for all p. In contrast to that, the 'quenched average'

$$\overline{R^2(p)} = \frac{1}{\sum_{\mathcal{C}}} \sum_{\mathcal{C}} \langle R^2\rangle_{\mathcal{C}} = \frac{1}{\sum_{\mathcal{C}}} \sum_{\mathcal{C}} \frac{\sum_{\text{SAWs}} R^2}{Z_N(\mathcal{C})} \tag{3}$$

is different, as is the geometric average of $Z_N(\mathcal{C})$. Here the sums are formed only over configurations which support at least one SAW†. The important question is whether the exponents describing the asymptotic behaviour of $\overline{R^2(p)}$ and $Z_N(p)$ are the same as for the ordered case, and whether they depend on p. *A priori* one might expect the value $p = p_c$ to be singled out, where p_c is the critical point for site percolation.

It is often said that the quenched average is the 'correct' average. As pointed out in [22], this has to be taken with a pinch of salt, since it depends on whether the location of the polymer (i.e. one of its ends) is fixed or not. In the latter case, one has to average over all locations and effectively the annealed average equation (2) applies. Thus, for example, the simulations performed in [14, 15] refer to the annealed average, and should not be compared with those presented below.

Besides choosing the proper weights, one also has to fix the set of configurations over which the averages are taken [25]. For the annealed averages, the set of *all* configurations seems most natural. For quenched averages, this does not make much sense. In that case, one can either take all configurations which support at least one SAW or one can take only those configurations which support arbitrarily long SAWs. The latter means that the starting point of the SAW has to be in an infinite cluster. While these two ensembles must give the same critical behaviour above p_c, it is not *a priori* clear whether this is also true for $p \leqslant p_c$.

The main reason why there is, as yet, no agreement about these scaling laws is that up to now there was no efficient Monte Carlo approach which would give the correct ensemble. The problem is that close to p_c the random variable $Z_N(\mathcal{C})$ has

† Instead of that, one could also take geometric averages of $Z_N(\mathcal{C}) + 1$ over *all* configurations, as done, for example, in [23]. Notice that equation (5) of that paper contains some misprints. Its correct version does not exclude that $\log Z_N(p)/N$ tends to 0 for $n \to \infty$.

an extremely wide variance. With any Monte Carlo sampling procedure with fixed acceptance probability, one either runs into the problem that most configurations do not give any long SAW or that in some configurations the number of accepted SAWs is so large that one runs out of CPU time. The same problem appears in exact enumerations as performed in [11, 20, 21, 23, 24]. These authors have enumerated all SAWs for small N (typically $N \approx 20$–30 in two dimensions) on randomly chosen configurations, and averaged these over typically some 10^3–10^4 samples. In [21] SAWs of length up to $N = 35$ were used. But as shown in [20], the author of [21] was not able to get an unbiased sample since he had to interrupt several runs for which $Z_N(C)$ was obviously too large had to be interrupted.

It is the purpose of the present paper to present a new Monte Carlo approach which works in any configuration, and which allows N-dependent acceptance probabilities to be chosen. For SAWs on percolation clusters, these acceptance probabilities are set such that the number of walks with length N is essentially independent of N (except for very small N since the number of accepted SAWs should not be larger than the number of different SAWs).

2. The algorithm

The basic ingredient in our algorithm is a recursively called subroutine taking as arguments a site `X` and the number `N` of steps already performed. Let us denote this subroutine '`WALK(X,N)`'. When called, it first checks whether the site `X` is accessible for a walker and not yet part of the present SAW. This is done using an array `S(X)`, each element of which can take two values: `S(X)=0` means that the site is blocked or was already visited by the present SAW. In that case, the subroutine is left immediately. If `S(X)=1`, indicating that the site is free, a random number `RND` is chosen between 0 and 1. If `RND` is larger than a pre-set value `P(N)`, the subroutine is also left. Only if `S(X)=1` and `RND<P(N)`, the site `X` is marked as being occupied by setting `S(X)=0`, and the statistics is updated to include the walk. After that, the subroutine `WALK` is called $2d$ times with `N` replaced by `N+1` and with `X` replaced by its $2d$ neighbours (we assume to work on a d-dimensional lattice). Finally, the site is again set free by putting `S(X)=1`, and the subroutine is left.

Calling this routine with arguments $(X, N) = (0, 0)$ for some given configuration $C = \{S\}$ gives a sample where each walk of length N is included with probability $\pi_N = \prod_{n=1}^{N} P(N)$. For the routine to stop in a finite time, one obviously needs $\lim_{N\to\infty} \pi_N \mu^N = 0$.

If $P(N)$ is chosen to be 1 for $N \leqslant N_0$ and 0 for $N > N_0$, this gives a very simple way of enumerating all SAWs of length up to N_0. After a couple of more or less obvious optimizations, it also seems to be the most efficient method for enumerating SAWs on the square lattice. On a DECstation 2100 (12 MIPS) it achieves approximately 3.98 SAWs μs^{-1} (independent of N), compared with, for example, 2.49 SAWs μs^{-1} obtained in [2] in a Masscomp 5700 computer, and 2347 SAWs μs^{-1} obtained in [3] on a Thinking Machines 64k CM-2 (where, however, R_N was not computed; according to the authors, computing R_N would have increased CPU time by a factor 3).

A very elegant and efficient hybrid method which enumerates all SAWs with $N \leqslant N_0$ and continues each randomly so as to get a grand canonical ensemble

for $N_0 < N \leqslant N_1$ is obtained by putting

$$P(N) = \begin{cases} 1 & N \leqslant N_0 \\ q & N_0 < N \leqslant N_1 \\ 0 & N > N_1. \end{cases} \tag{4}$$

We used the latter to test the method against the known values of μ, ν and γ on an ordered square lattice.

When applying the method to disordered systems, the main problem is that we cannot work with a fixed (i.e. configuration independent) $P(N)$. For most configurations this would give too few long SAWs, while it would need excessive CPU time for others. Thus we have to use some learning schedule which finds the optimal $P(N)$ for each configuration.

For walks with length $\leqslant N_1 = 100$ in two dimensions, we proceeded as follows. We first created disorder configurations by a Leath-type algorithm [27]. We then selected values for N_0 and q such that the algorithm with $P(N)$ set as in equation (4) terminated in some reasonable time for *all* configurations. This implied, in particular, that $q < (\mu p)^{-1}$. At the percolation threshold ($p_c = 0.592\,745$ [28], $\mu = 2.638\,158$ [1]) we used e.g. $N_0 = 11$, $q = 0.58$. With this choice, on most configurations we will not have any SAW with length 100 in our sample, but we will have enough SAWs of length $\gtrsim N_0$ to make a good estimate for the effective connectivity 'constant'

$$\mu_N(C) = Z_N(C)/Z_{N-1}(C) \tag{5}$$

for these intermediate values of N. We then replace $P(N)$ for these values of N by

$$P'(N) = 1/\mu_N(C) \tag{6}$$

increase N_0 by 1 unit if $Z_{N_0}(C)$ has not yet reached a preset value (approximately 4000), and call again `WALK(0,0)`. This is repeated until $Z_N(C)$ is larger than some other preset value (approximately 200) for all N or until $N_0 = N_1$. In cases where equation (6) would give $P'(N) > 1$, we replace it by $P'(N) = 1$ and increase N_0 by 1 unit. The latter guarantees that we do not get blocked by configurations where medium-length SAWs are abundant but long ones are very rare. For the above parameter settings, the average number of iterations for this procedure to yield an accepted sample was ≈ 5.

In this way we obtained an unbiased sample of SAWs for each configuration which did support at least one SAW. It took approximately 16 s on a DECstation 2100 to produce a sample which contained in the average approximately 5000 SAWs for every $N \leqslant 100$. This allowed to take averages over more than 10^4 configurations for each value of p. This seems to be by far the largest unbiased sample available so far, though it was obtained with rather modest computational effort (approximately 200 h on low-cost work stations). The only samples of comparable size were those used in [18, 19, 25], but it is far from obvious that they represent unbiased samples of the quenched ensemble†.

† It was pointed out to me by J Machta that my estimate of $\overline{R^2(p)}$ need not be completely unbiased even if the sample itself is without bias. This is due to the particular averaging in equation (3), which would suppress in one and the same configuration regions where the walks proliferate at late stages, compared with regions where the number of walks had already increased earlier. I am indebted to J Machta for this remark, but I do not think that it lead to any substantial bias in the present results.

The efficiency of our method increases with p, and it is highest for $p = 1$. When p is decreased below p_c, one might believe that the fluctuations should also decrease and the efficiency improve. But this is not true. We have not found any efficient way of dealing with configurations which allow many medium-long SAWs but none of length N_1. As we have pointed out, our algorithm is not completely blocked by such configurations, but it still has to try *all* medium-long SAWs in order to be sure that none of them gives a continuation of full length. This can be very time consuming. It virtually never happens for $p \geqslant p_c$, but it frequently happens for $p < p_c$, and slows down the overall performance there.

Before leaving this section I want to suggest that recursive algorithms like the present one should be used more frequently in statistical mechanics. Previously, recursive algorithms were found to be useful for sandpile models [29] and for percolation [30], but many more applications exist. First of all, the present algorithm can be trivially modified to apply to random media with imperfect exclusion of some sites as well. If we attribute a potential v_i to site i, then we have just to replace the space-independent array $P(N)$ by a space-dependent array $P(N, i) \propto e^{-\beta v_i}$. A similar modification could be used for polymers at the theta point or for stretched polymers. Another obvious extension (respectively simplification) of the present algorithm produces directed SAWs in random media [31] which can then be used to model growth phenomena in such media. Finally, a last obvious modification would lead to non-self avoiding walks (i.e. to ordinary diffusion) in random absorptive media [32].

The modifications needed to obtain algorithms for a variety of models are minimal. To show this, we show in the appendix a menu driven and fully commented BASIC program. Although it provides state-of-the-art algorithms for simulating models in at least five different universality classes and displays the clusters on the screen, it is less than one page long! Among others, it can simulate SAWs, percolation and Wolff's algorithm for the Ising model [33]. Model 5 is a stochastic variant of the Abelian sandpile model [34, 29] similar to the models studied in [35] and [36]. It seems to be the simplest such model and is critical only when mass is conserved in the average, i.e. for $p = 1/2$. This shows that criticality in sandpile models is a codimension 1 phenomenon as in other systems, even if the order parameter is hidden in the more common variants. Up to now, no detailed studies of models 5 and 6 have been made, and the criticality of the latter is not clear.

3. Numerical results

3.1. End-to-end distances

Our results for $\overline{R^2(p)}$ at $p = p_c = 0.592\,745$ are given in figure 1. Since the main question is whether the model falls into the same universality class as ordinary SAWs, and since the latter are known to involve substantial corrections to scaling, we chose to show the ratio $\overline{R^2(p)}/R_N^2$. Here R_N is the mean end-to-end distance on the ordered square lattice. Besides the data for the quenched average and for all (SAW-supporting) configurations we also show in figure 1 the results for the annealed average, and for 'infinite' clusters only. The latter are clusters with 'chemical' radius $l_c > 200$, which should be sufficiently large. Here the chemical radius l_c is defined as the number of steps during which a Leath-type algorithm does not die out. Formally,

it is defined as $l_c = \sup_i r_i$, where r_i is the 'chemical distance' of site i from the origin, i.e. r_i is the minimal number of steps *on the cluster* which are needed to reach site i from the origin.

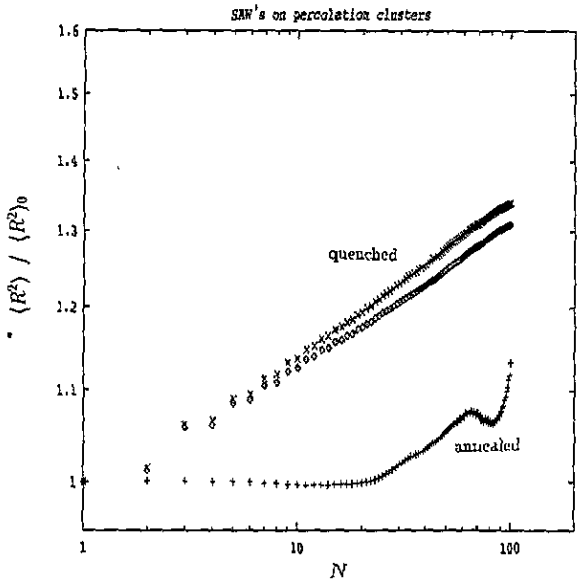

Figure 1. Log–log plot of mean square end-to-end distances at $p = p_c$ on randomly diluted square lattices, divided by the corresponding averages in the absence of randomness. Diamonds represent quenched averages, $\overline{R^2(p)}/R_N^2$, taken over all configurations which support at least one SAW of the required length. Crosses ($\times$) represent quenched averages over percolating clusters only, pluses ($+$) represent annealed averages. For the latter the statistical errors are very large for large N, and are responsible for the deviation from unity. For the quenched averages, the errors are roughly of the size of the symbols.

In figure 1 we see a linear increase of $\log[\overline{R^2(p)}/R_N^2]$ with $\log N$, indicating a power behaviour with exponent

$$\nu' = \nu + 0.035 \pm 0.003 = 0.783 \pm 0.003. \tag{7}$$

This shows unambiguously that the quenched average does indeed give a different universality class.

Our result agrees with theoretical predictions [22], but contradicts most recent numerical investigations. In particular, it disagrees with the conclusion drawn in [20] from very short SAWs ($N \leqslant 20$), though our results for such short SAWs essentially agree with those of [20] (for $N = 20$, e.g., we find $R^2 = 86.4 \pm 0.2$ instead of $86.95 \pm .4$). Our conclusion agrees qualitatively with that of [21], though it was shown in [20] that these simulations involved two serious mistakes: first, annealed averages were estimated; and second, configurations needing very much CPU time were dropped. What was not realized in [20] is that these two mistakes partly cancel, so the simulations of [21] were not too inaccurate after all. Finally, the disagreement of our results with those of [18, 19, 25] indicate that the samples used in these Monte Carlo simulations were probably not unbiased.

In contrast to the quenched average (where the statistical error bars are much smaller than the size of the symbols), the annealed average shows very large statistical fluctuations, but is compatible with $\langle R^2 \rangle / R_N^2$ being constant (see figure 1). The fluctuations are easily understood from the fact that only very few configurations give the dominant contributions to this average. We should also point out the strong even/odd oscillations which are typical for SAWs on square lattices.

Finally, the quenched averages over the 'infinite' clusters are consistently somewhat larger than the total averages, as expected [25]. But the exponent (best estimate $\nu'' - \nu = 0.036 \pm 0.003$) is compatible with being the same as at $p = p_c$.

Values of $\overline{R^2(p)}/R_N^2$ for different values of p are compared in figure 2. There we show only quenched averages, but we checked that the annealed averages gave $\langle R^2 \rangle = R_N^2$ independently of p. If we restrict the averaging over 'infinite' clusters (which in practice meant $l_c > 2N$), we see a clear increase in $\overline{R^2(p)}/R_N^2$ both for $p = 0.5$ and for $p = 0.75$, indicating that the problem is also in a different universality class from ordinary SAWs. While the exponent could be the same for $p > p_c$ as for $p = p_c$, we clearly see a larger exponent, $\nu''' - \nu = 0.065 \pm 0.005$, for $p = 0.5$. Indeed, both data sets (in particular that for $p = 0.75$) show upward curvatures, indicating that the correct asymptotic exponent could even be larger and the same for $p > p_c$ and $p < p_c$.

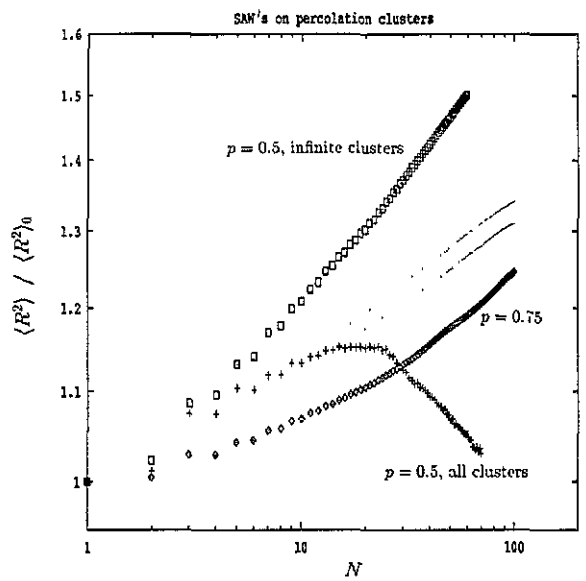

Figure 2. Scaled mean square end-to-end distances, $\overline{R^2(p)}/R_N^2$. The dots are the data from figure 1 for $p = p_c = 0.592\,745$, and the other data are: for all SAW-supporting clusters at $p = 0.75$ (diamonds), for 'infinite' clusters at $p = 0.5$ (rectangles), and for all SAW-supporting clusters at $p = 0.5$ (crosses). In all cases, quenched averages are shown.

When taking the average over all clusters which support at least one SAW, we get the same behaviour when $p > p_c$ (not shown in figure 2). But for $p < p_c$, we see in figure 2 a clear cross-over to a much smaller value for the exponent, ruling out an exponent larger than for SAWs on regular lattices. But since we were not able

to obtain very long SAWs for $p < p_c$ (see the discussion in section 2), we do not know whether the asymptotic behaviour is that of SAWs on regular lattices or whether $\overline{R^2(p)}$ increases even slower.

These surprising results do not seem to have been predicted by any of the existing theories.

3.2. Numbers of SAWs

We first checked that the (annealed) averages of $Z_N(\mathcal{C})$ were indeed simply given by $p^N Z_N^{(0)}$, where $Z_N^{(0)}$ had been obtained by our algorithm on regular lattices. Here the average is over *all* configurations, i.e. the normalization is done by dividing with the total number of configurations. But statistical fluctuations were again very large except for very small N.

These anomalous statistical fluctuations should be absent in the quenched entropy

$$S_N(p) = \langle \log Z_N(\mathcal{C}) \rangle \tag{8}$$

where the summation runs over all configurations with $Z_N(\mathcal{C}) > 0$, both in the denominator and in the numerator necessary for normalization.

To show this, we plotted $S_N(p) - \log(p^N Z_N^{(0)})$ and $\log[\langle Z_N(\mathcal{C}) \rangle / (p^N Z_N^{(0)})]$ against $\log N$ in figure 3, for $p = p_c$. According to the theoretical expectation the latter should be exactly equal to 1, which is indeed the case. In contrast, $S_N(p)$ is much smaller than $p^N Z_N^{(0)}$.

If we define $S_N(p)$ by restricting ourselves to SAW-supporting configurations (dots in figure 3), we find that it scales linearly with N. If, however, we include *all* configurations, then $S_N(p)$ has to be smaller by a factor $< N^{2-\tau}$ with $\tau = 2.055$ [37] due to the normalization. A sublinear increase $S_N(p) \sim N^{1-\alpha}$ when taken over *all* configurations (and with $\log 0$ replaced by 0) is indeed seen in figure 3 (diamonds), but with $\alpha = 0.075 \pm .003$. This is definitely larger than $\tau - 2$, and shows that clusters must have much more than N points in order to support an N-step SAW. We should mention that a stretched exponential increase of $\log Z_N(\mathcal{C})$ had been predicted in [17], but with a very different exponent.

This difference in the asymptotic behaviour of $S_N(p)$ and $\log[\langle Z_N(\mathcal{C}) \rangle]$ reflects the fact that the distribution of $Z_N(\mathcal{C})$ is extremely wide. In figure 4 we show the distribution of its logarithm for $N = 100$ and $p = p_c$. We see an approximately log-normal distribution. The distribution is narrower for $p \neq p_c$. For $p > p_c$, the lower cutoff of the distribution increases faster than the upper, while for $p < p_c$ the upper cutoff decreases to 0.

To show that the width of the entropy distribution also scales roughly linearly with N, we show in figure 5 the variance of the entropy per monomer at $p = p_c$,

$$[\delta S_N(p)/N]^2 = \frac{1}{N^2}[\langle \log Z_N(\mathcal{C})^2 \rangle - S_N(p)^2]. \tag{9}$$

We see very strong finite-length corrections, which give a decrease in the variance at small N, but asymptotically it seems to stay constant or even to increase.

The wide distribution of $Z_N(\mathcal{C})$ is suggestive of a multifractal distribution [38]. This is typically obtained in multiplicative random processes. In the present case, $Z_N(\mathcal{C})$ would be produced by a multiplicative random processes if the effective

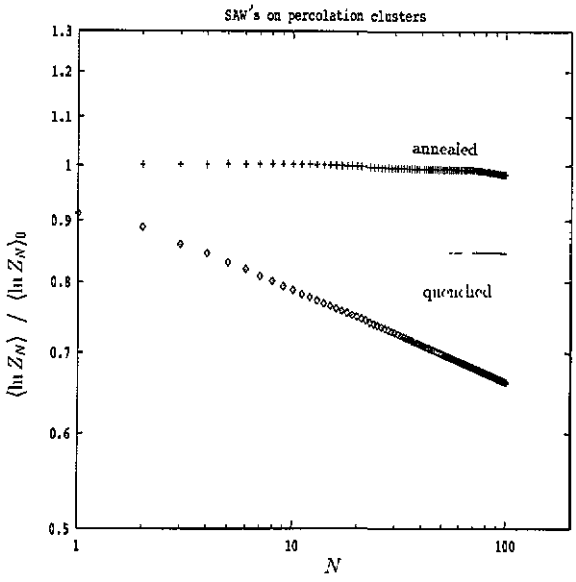

Figure 3. Quenched (dots) and annealed (crosses) entropy differences (with respect to the ordered square lattice) per monomer at $p = p_c$, plotted against N on a log–log plot. The errors of the quenched averages are of the size of the dots, while the errors of the annealed averages can be estimated from the deviations from a horizontal line. The diamonds represent quenched averages normalized by dividing over *all* configurations, including those which do not support any SAW.

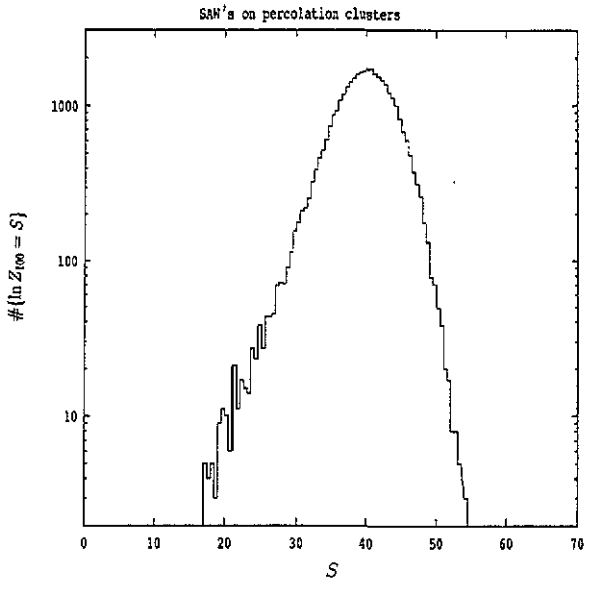

Figure 4. Distribution of the entropy S for $N = 100$ and $p = p_c$.

connectivity constants μ_N defined in equation (5) were to form a random sequence, as obtained typically in a mean-field treatment. In that case we would have

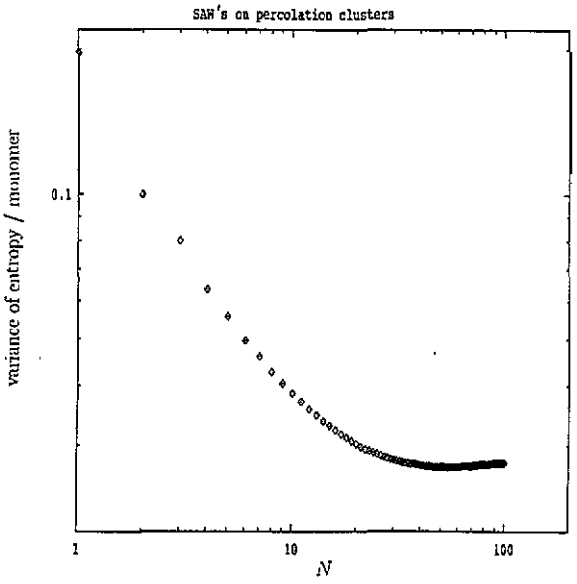

Figure 5. Variance of the entropy per monomer, equation (9), against N on doubly logarithmic plot, for $p = p_c$.

$$P(S) \equiv \text{prob}\{\log Z_N(\mathcal{C}) = S\} \propto N^{-1/2} e^{N f(S/N)} \tag{10}$$

where $f(x)$ is a cap-convex function with $\max_x f(x) = 0$. In order to test this, we plotted in figure 6 the quantity $N^{-1} \log P(S)$ against S/N, for three different values of N. The heights of the three curves are, in principle, fixed by normalization, and were chosen here so that the three curves have the same maxima. The agreement between them is far from perfect. It would be much better if we had used $S/N^{0.94}$ as the scaling variable instead of S/N, but we believe that this is a finite-length correction as also manifested in figure 5. Thus we interpret figure 6 as an indication that equation (10) is indeed correct.

4. Conclusions

Using a new Monte Carlo procedure, we have been able to resolve some of the problems concerning SAWs in two-dimensional quenched random media. In particular, we have shown that the end-to-end distance scales with a new exponent $\nu' > \nu$ at the percolation threshold $p = p_c$, but not only there. If we restrict ourselves to infinite clusters, then the end-to-end distance increases for $p < p_c$ even faster than at $p = p_c$, while it seems to increase for $p > p_c$ at least with the same exponent. Otherwise, if we average over all clusters which carry at least one SAW, we obtain a larger exponent than for ordered lattices only for $p \geq p_c$. All this refers to the quenched ensemble, and we checked that the annealed averages of the end-to-end distance are the same as on ordered lattices.

For the quenched average of the entropy (i.e. the logarithm of the number of SAWs), we found a stretched exponential as predicted in [22] only if we take averages

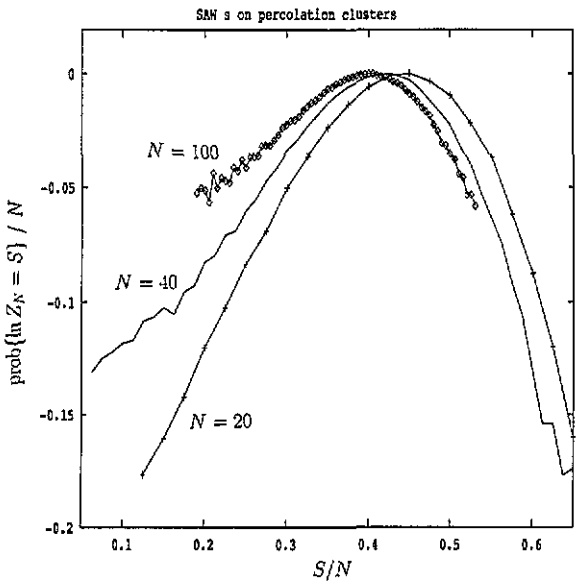

Figure 6. Plot of $N^{-1}\log P(S)$ against S/N, where S is the entropy and $P(S)$ its probability distribution, for $p = p_c$ and for three different values of N: $N = 20$ (crosses), $N = 40$ (no symbols), $N = 100$ (diamonds). The curves are arbitrarily shifted vertically so that they have the same maximal value.

over *all* clusters, including those which do not support any SAW. The distribution of the entropy seems to be multifractal.

Apart from these results on SAWs in random media, we believe that the main result of this paper is the Monte Carlo procedure itself. Its main aspect is an extremely simple recursively called subroutine. It is most simple in applications to non-self-avoiding walks. When called at point x and length N, it just calls itself at the neighbouring points of x and with length $N - 1$. When used for self-avoiding walks as in the present paper, it first sets a flag at x immediately after entering, checks that no flags have been set before going to neighbouring points, and clears the flag before returning to the calling routine.

This algorithm can be regarded as a recursive implementation of the incomplete enumeration of [7,8] or as a statistical variant of enrichment methods. It is also similar to the method in [6]. Its main advantage over these methods is its flexibility, simplicity and intuitiveness. It was the latter which made feasible the somewhat tricky schedule for finding optimal enhancement factors to counterbalance the attrition.

Variants of this recursive algorithm should be very usefull in all situations where one needs simulations of random walks with absorption or attrition, be they self-avoiding or not. More generally, we propose that recursive algorithms should find many more applications in statistical physics. Applications to percolation and to sandpile models, with routines surprisingly similar to that for SAWs, were given in [29,30]. The simplicity and similarity of the different algorithms is illustrated by the BASIC routine in the appendix.

```
CLEAR ,400000                          '   allocate 400 KB stack
DEFSNG "P"                             '   declare "P" as REAL

PRINT " Multi - purpose routine for Stat Mech on 100 x 100 lattice"
PRINT " Which model:"
PRINT "    self-avoiding random walks  (p_c = .37905) ..... 1"
PRINT "    SAW's on crit. percolation clusters (p_c = .6395) .. 2"
PRINT "    bond percolation  (p_c = .5) ..................... 3"
PRINT "    Ising - Wolff model  (p_c = .58579) ............... 4"
PRINT "    self-organised criticality  (p_c = .5) ............ 5"
PRINT "    self-organised Ising  (p_c = .5) .................. 6"
INPUT " ? ";Q                          '   ask for model selection
PRINT "": PRINT ""

INPUT " p = ";P                        '   ask for control parameter

BOX 259,159 TO 460,360                 '   draw frame around lattice
FILL COLOR =0                          '   "PBOX" will later fill box
                                       '            with white color
IF Q>3 OR Q=2 THEN Hot_Start           '   create random configuration

REPEAT
   I=260+ 2*RND(100)                   '   random x-coord. in [260,458]
   J=160+ 2*RND(100)                   '   random y-coord. in [160,358]
   IF Q<>5 THEN C=POINT(I,J) ELSE C=1  '   give color at point (I,J)
   IF Q=3 THEN MODE =1: PBOX 260,160 TO 459,359      '   reset lattice
   MODE =3                             '   set "XOR" mode for graphics
   Connect(I,J)                        '   start cluster
UNTIL INKEY$<>""                       '   stop when any key is pressed

END

DEF PROC Connect(I,J)
   IF POINT(I,J)=C THEN                '  color at (I,J) = start color ?
      BOX I,J,2,2                      '  change color black <--> white
      IF RND(1)<P THEN IF J<358 THEN Connect(I,J+2)
      IF RND(1)<P THEN IF J>160 THEN Connect(I,J-2)
      IF RND(1)<P THEN IF I<458 THEN Connect(I+2,J)
      IF RND(1)<P THEN IF I>260 THEN Connect(I-2,J)
      IF Q<3 THEN BOX I,J,2,2          '  change color back (for SAW's)
   ELSE
      IF Q>4 THEN BOX I,J,2,2          '  change color anyhow (for SOC)
   ENDIF
RETURN

DEF PROC Hot_Start
   FOR I=260 TO 458 STEP 2
      FOR J=160 TO 358 STEP 2
         IF RND(1)>.59724 THEN DRAW BOX I,J,2,2
      NEXT                             '  reverse spin at (i,j) with
   NEXT                                '  probability  1-p_crit for
RETURN                                 '  site percolation
```

Figure A1. A multi-purpose routine written in OMICRON BASIC for the Atari ST computer, for simulating six different models.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft, SFB 237. It grew out of a course on numerical physics on home computers for undergraduate and high school students, where, in particular, the routine shown in the appendix was worked out. I am indebted to Dieter Stauffer for convincing me that research-oriented numerical physics can be done by such students. I am also indebted for interesting correspondence with J Machta, H Nakanishi and C Vanderzande.

Appendix

In figure A1 we give a multi-purpose routine written in OMICRON BASIC for the Atari ST computer, for simulating six different models. At least the first five models are each in a different universality class. The clusters are displayed on the screen, using the video memory instead of an array to store the configuration, and using 2×2 pixels per site. The function `RND(x)` gives a pseudorandom real number in $[0, 1]$ if $x = 1$, while it gives an integer $\in [0, x)$ if x is integer and > 1. `PBOX` fills a rectangle with the color set by `FILL COLOR`, while `BOX` draws a 2×2 box. `MODE=1` sets the usual graphics mode while `MODE=3` sets the `XOR` graphics mode. Finally, the function `POINT(i,j)` gives back the colour at this point.

References

[1] Guttmann A J and Enting I G 1988 *J. Phys. A: Math. Gen.* **21** L165
[2] Guttmann A J and Wang J 1991 *J. Phys. A: Math. Gen.* **24** 3107
[3] Masand B, Wilensky U, Massar J P and Redner S 1991 *Preprint*
[4] Caracciolo S, Pelissetto A and Sokal A D 1992 *J. Stat. Phys.* **67** 65
[5] Nienhuis B 1982 *Phys. Rev. Lett.* **49** 1062
[6] Beretti A and Sokal A D 1985 *J. Stat. Phys.* **40** 483
[7] Redner S and Reynolds P J 1981 *J. Phys. A: Math. Gen.* **14** 2679
[8] Dhar D and Lam P M 1986 *J. Phys. A: Math. Gen.* **19** L1057
[9] Kremer K 1981 *Z. Phys.* B **45** 149
Lyklema J W and Kremer K 1984 *Z. Phys.* B **55** 41
[10] Harris A B 1983 *Z. Phys.* B **49** 347
[11] Rexakis J and Argyrakis P 1983 *Phys. Rev.* B **28** 5323
[12] Derrida B 1984 *Phys. Rep.* **103** 29
[13] Meir Y and Harris A B 1989 *Phys. Rev. Lett.* **63** 2819
[14] Honeycutt J D and Thirumulai D 1989 *J. Chem. Phys.* **90** 4542
[15] Honeycutt J D and Thirumulai D 1990 *J. Chem. Phys.* **93** 6851
[16] Machta J 1989 *Phys. Rev.* A **40** 1720
[17] Machta J and Kirkpatrick T 1990 *Phys. Rev.* A **41** 5345
[18] Lee S B and Nakanishi H 1988 *Phys. Rev. Lett.* **61** 2022
[19] Lee S B, Nakanishi H and Kim Y 1989 *Phys. Rev.* B **39** 9561
[20] Nakanishi H and Lee S B 1991 *J. Phys. A: Math. Gen.* **24** 1355
[21] Lam P M 1990 *J. Phys. A: Math. Gen.* **23** L831
[22] Doussal Le and Machta J 1991 *J. Stat. Phys.* **64** 541
[23] Vanderzande C and Komoda A 1991 *Europhys. Lett.* **14** 677
[24] Vanderzande C and Komoda A 1992 *Phys. Rev.* A **45** R5335
[25] Woo K Y and Lee S B 1991 *Phys. Rev.* A **44** 999
[26] Kim Y 1992 *Phys. Rev.* A **45** 6103
[27] Leath P L 1976 *Phys. Rev.* B **14** 5046
[28] Ziff R M and Sapoval B 1986 *J. Phys. A: Math. Gen.* **19** L1169

[29] Grassberger P and Manna S 1990 *J. Physique* **51** 1077
[30] Grassberger P 1992 *J. Phys. A: Math. Gen. Preprint* WU-B 92-05 to appear
[31] Derrida B 1990 *Physica* A **163** 71
[32] Grassberger P and Procaccia I 1982 *J. Chem. Phys.* **77** 6281; 1982 *Phys. Rev.* A **26** 3686
[33] Wolff U 1989 *Phys. Rev. Lett.* **62** 361
[34] Bak P, Tang C and Wiesenfeld K 1987 *Phys. Rev. Lett.* **59** 381
[35] Manna S S 1991 *J. Phys. A: Math. Gen.* **24** L363
[36] Manna S S, Kiss L B and Kertesz J 1990 *J. Stat. Phys.* **61** 923
[37] Stauffer D 1985 *Introduction to Percolation Theory* (London: Taylor and Francis)
[38] Paladin G and Vulpiani A 1987 *Phys. Rep.* **156** 147