A LETTERS JOURNAL EXPLORING THE FRONTIERS OF PHYSICS

OFFPRINT

Scale-free enumeration of self-avoiding walks on critical percolation clusters

NIKLAS FRICKE and WOLFHARD JANKE

EPL, **99** (2012) 56005

epl

A Letters Journal Exploring the Frontiers of Physics

AN INVITATION TO SUBMIT YOUR WORK

www.epljournal.org

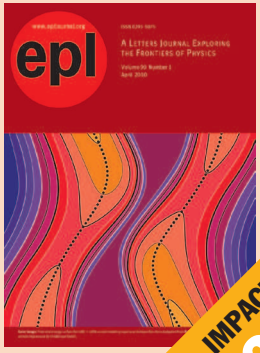
The Editorial Board invites you to submit your letters to EPL

EPL is a leading international journal publishing original, high-quality Letters in all areas of physics, ranging from condensed matter topics and interdisciplinary research to astrophysics, geophysics, plasma and fusion sciences, including those with application potential.

The high profile of the journal combined with the excellent scientific quality of the articles continue to ensure EPL is an essential resource for its worldwide audience. EPL offers authors global visibility and a great opportunity to share their work with others across the whole of the physics community.

Run by active scientists, for scientists

EPL is reviewed by scientists for scientists, to serve and support the international scientific community. The Editorial Board is a team of active research scientists with an expert understanding of the needs of both authors and researchers.



IMPACT FACTOR 2.753*

*As ranked by ISI 2010

IMPACT FACTOR
2.753*
* As listed in the ISI® 2010 Science Citation Index Journal Citation Reports

OVER
500 000
full text downloads in 2010

30 DAYS
average receipt to online publication in 2010

16 961
citations in 2010
37% increase from 2007

"We've had a very positive experience with EPL, and not only on this occasion. The fact that one can identify an appropriate editor, and the editor is an active scientist in the field, makes a huge difference."

Dr. Ivar Martinv
Los Alamos National Laboratory, USA

Six good reasons to publish with EPL

We want to work with you to help gain recognition for your high-quality work through worldwide visibility and high citations.

1. **Quality** – The 40+ Co-Editors, who are experts in their fields, oversee the entire peer-review process, from selection of the referees to making all final acceptance decisions
2. **Impact Factor** – The 2010 Impact Factor is 2.753; your work will be in the right place to be cited by your peers
3. **Speed of processing** – We aim to provide you with a quick and efficient service; the median time from acceptance to online publication is 30 days
4. **High visibility** – All articles are free to read for 30 days from online publication date
5. **International reach** – Over 2,000 institutions have access to EPL, enabling your work to be read by your peers in 100 countries
6. **Open Access** – Articles are offered open access for a one-off author payment

Details on preparing, submitting and tracking the progress of your manuscript from submission to acceptance are available on the EPL submission website **www.epletters.net**.

If you would like further information about our author service or EPL in general, please visit **www.epljournal.org** or e-mail us at **info@epljournal.org**.

EPL is published in partnership with:

European Physical Society | Società Italiana di Fisica | EDP Sciences | IOP Publishing

A Letters Journal Exploring the Frontiers of Physics

September 2012

EPL, **99** (2012) 56005
doi: 10.1209/0295-5075/99/56005

www.epljournal.org

Scale-free enumeration of self-avoiding walks on critical percolation clusters

Niklas Fricke[a] and Wolfhard Janke[b]

Institut für Theoretische Physik and Centre for Theoretical Sciences (NTZ), Universität Leipzig Postfach 100920, D-04009 Leipzig, Germany, EU

received 6 May 2012; accepted in final form 17 July 2012
published online 12 September 2012

PACS 64.60.De – Statistical mechanics of model systems (Ising model, Potts model, field-theory models, Monte Carlo techniques, etc.)
PACS 36.20.-r – Macromolecules and polymer molecules
PACS 64.60.ah – Percolation

Abstract – We present a new method for exact enumeration of self-avoiding walks on critical percolation clusters. It can handle very long walks by exploiting the clusters' low connectivity and self-similarity. We have implemented the method in 2D and used it to enumerate walks of more than 1000 steps with over 10^{170} conformations. The exponents ν and γ, governing the scaling behavior of the end-to-end distance and the number of configurations, as well as the connectivity constant μ could thus be determined with unprecedented accuracy. The method will help answering long-standing questions regarding this particular problem and can be used to check and gauge other methods, analytical and numerical. It can be adapted to higher dimensions and might also be extended to similar systems.

Self-avoiding walks (SAWs) on critical percolation clusters (CPCs) have been the subject of numerous studies since the 1980s (see [1,2]). They reflect the qualitative behavior of linear polymers in highly packed disordered environments such as porous rocks or biological cells. The model is also of relevance from a theoretical point of view, not least because of the fundamental importance of the two constituents (see [3,4]).

Despite its simplicity, the lattice SAW features interesting scaling behavior with universal exponents that are representative for a range of more complicated systems, including long polymers in good solvents [5,6]. The main focus is usually on the scaling of the end-to-end distance R and the number of chain conformations Z with the number of steps N, described by the exponents ν and γ:

$$\sqrt{\langle R^2 \rangle} \sim N^{\nu}, \qquad Z \sim \mu^N N^{\gamma-1}. \tag{1}$$

Only the so-called connectivity constant μ depends on the details of the lattice. In the presence of disorder, one usually considers quenched averages of $\langle R^2 \rangle$ and Z over all possible realizations of the background. Whether and how the scaling behavior changes in that case has been a widely discussed issue [7–9].

A convenient implementation of disorder is random site dilution. Most intriguing is the case at the critical occupation probability p_c where the (fractal) dimension of the substrate shifts to a slightly smaller, non-integer value. How the fractal structure of the substrate influences the SAWs' scaling behavior is still only partly understood.

Different analytical and numerical techniques have been used to investigate the problem. They include mean-field (Flory) approximations [10,11], renormalization in real space [8,12–14], field theory [15–18], various chain-growth Monte Carlo (MC) methods [9,19–22], and exact enumeration (EE) [7,23–28]. While much insight has been gained, questions and controversies have remained. There is general agreement that the scaling exponents at the percolation threshold differ from those on the full lattice, but the accuracy with which they could be determined so far is quite limited. Also, the results from different studies deviate significantly, and some authors [18] even doubt the existence of an asymptotic scaling limit. The main problem of the numerical methods is that they can only handle small systems, leading to finite-size errors which are hard to control. EE studies are most affected: Due to the exponential increase in computation time, only very few steps (up to 45 in 2D [28]) could be afforded so far. With MC longer walks (up to 200 steps [22]) could be

[a] E-mail: niklas.fricke@itp.uni-leipzig.de
[b] E-mail: wolfhard.janke@itp.uni-leipzig.de

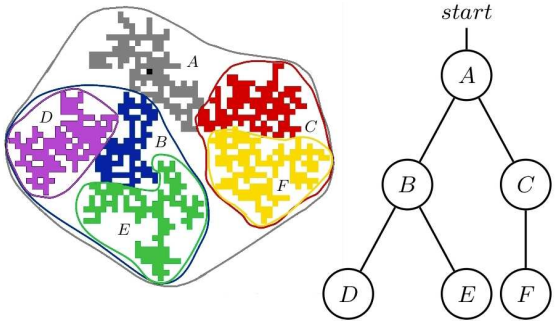


Fig. 1: (Color online) Decomposition of a critical percolation cluster into nested blobs and corresponding tree hierarchy. The SAW's starting position is marked in black (in blob A).

simulated, but the results are not as reliable. Indeed, some methods are known to yield biased results if sample sizes are too small (see [9]).

We devised a method for exact enumeration of SAWs on CPCs that appears to have overcome the seemingly inherent problem of exponential complexity. This is possible by exploiting the fragility and self-similarity of the CPCs. These two properties allow factorization of the problem on all length scales and the use of a recursive, "renormalizing" treatment. Since our method relies explicitly on the structural properties of CPCs, it is only applicable at or close to the percolation threshold. There, however, it outperforms even the most efficient MC methods.

The aim of this letter is to convey the key ideas of this new method and to demonstrate that it allows for a much more profound treatment of the problem at hand and related ones. Estimates for the values of ν, γ, and μ in two dimensions are also presented and discussed. We omit some of the technical details. In particular, we shall only explain how to obtain the number of chains Z, which is the central part of the method. The additional measurement of $\langle R^2 \rangle$ is straightforward.

Normally, the computational effort for EE scales exponentially with the system size, *e.g.*, the number of steps N. That problem may be avoided by dividing the system into parts that can be treated separately. The self-similar, weakly connected structure of a critical percolation cluster allows for just that.

At the percolation threshold, the average number of bonds between sites that one would need to cut in order to divide the cluster into separate pieces is on the order of one, independent of the system size. This scale-free property makes it possible to decompose the cluster into nested pieces (*blobs*) of all sizes in such a way that each one has only a small number of connections (*links*) to the larger blob containing it (*parent*) and a small number of *children* contained by it (see fig. 1).

To see how the low connectivity can be used, consider a blob A from which a child B of about half its size can be dissevered by cutting a single bond l. Imagine we want to determine the number $Z_{a\to B}[n]$ of SAWs of n steps starting from some point a outside of the child B and

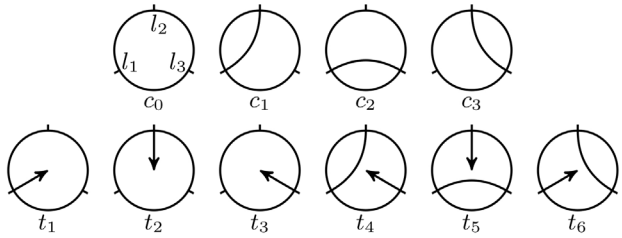


Fig. 2: Schematic depiction of all states for a blob with three links. A c-state represents paths that traverse a blob, for a t-state they terminate within.

terminating within. Instead of counting directly, we can calculate that number from the number of segments from a to l and from l to anywhere in B:

$$\begin{aligned} Z_{a\to B}[n] &= \sum_{i=0}^{n} Z_{a\to l}[i] \cdot Z_{l\to B}[n-i] \\ &:= \left(Z_{a\to l} \circledast Z_{l\to B}\right)[n]. \end{aligned} \tag{2}$$

The number of counted paths is thus only the sum (and not the product) of the numbers of segments. Such a factorization is more difficult when there are more links and children, as paths may then leave and re-enter the pieces. To accommodate for this we need to distinguish "topologically different" ways to traverse a blob (meaning that different links are connected). An ensemble of topologically equivalent ways through a blob we call a *state*. We furthermore distinguish *terminal* (t-) states, where the chains end within the blob, from *conductive* (c-) states and represent them graphically as shown in fig. 2. Knowing the maximal number of states S as a function of the number of links L is important for understanding the limitations of the method and helps to decide how to decompose a cluster. To derive $S(L)$, we start by considering the c-states with $2i$ connected links. Their number is equal to the number of c-states when the unconnected links are disregarded (as if $L = 2i$), $C[2i]$, times the number of ways to intersperse the $L - 2i$ unconnected links: $C[2i]\binom{L}{2i}$. In 2D, where the lines in the diagrams may not cross, the c-states without unconnected links can be mapped bijectively to the set of binary trees with i nodes, for which the number of configurations is given by the Catalan numbers (see [29]): $C[2i]_{D=2} = \binom{2i}{i}/(i+1)$. When states with crossed lines are included, the situation is easier, and $C[2i]_{D>2} = \frac{(2i)!}{i!2^i}$ can be obtained using elementary combinatorics. For each c-state, there are $L - 2i$ t-states, one for each unconnected link. Summing over all i for which $2i \leqslant L$ leads to the total number of states:

$$S(L) = S_c(L) + S_t(L) = \sum_{i=0}^{\lfloor L/2 \rfloor} C\,[2i] \binom{L}{2i} (1 + L - 2i). \tag{3}$$

The more states a blob can assume, the more memory and time will be needed for its treatment. While

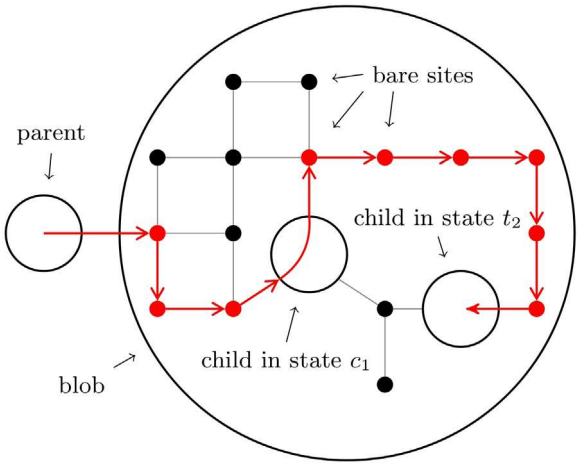


Fig. 3: (Color online) Snapshot of the enumeration procedure: a path of length 11 (red arrows) belonging to the blob's t_1-state is found; the corresponding entry $Z^{comb}_{t_1;c_1,t_2}[11]$ is incremented.

ten links are still manageable ($S_{D=2}(10) = 10538$ and $S_{D>2}(10) = 35696$), twenty are clearly too many (approximately 4×10^8 and 1×10^{11} states, respectively). This has to be kept in mind when we partition the percolation cluster. We shall discuss the partitioning procedure later on; for now let us assume that we have a suitable decomposition of a cluster.

We begin the enumeration by counting the number of paths for each state of the blobs at the bottom of the hierarchy. Once this is done, they can be "renormalized" to single points when we count the paths through the next larger blobs. This goes on until we arrive at the *root*, whose only link is at the starting location.

The central challenge is to *efficiently determine the paths for each state of a blob containing children whose paths (for each state) are already known.* This is done using a depth-first recursion similar to the standard backtracking method (see [7]). The procedure takes place on a graph whose nodes are the blob's parent, children, and "*bare sites*", *i.e.*, sites that are not contained within the children (see fig. 3). The paths always start at the parent node. Regardless of their true size, children and parent behave as pointlike sites. However, they differ from normal sites in that they change their state when visited. If, for example, a child with three links in state c_0 (see fig. 2) is accessed via link l_1, it will transit to t_1. When the child is then left via link l_2, it changes to c_1. Likewise, each path belongs to a state of the blob itself, which changes whenever the parent is visited. Note that accessing or leaving a child or the parent is only one move (we use this term here instead of "step" to avoid confusion), as for any normal site. For each move, we increment a counter $Z^{\mathrm{comb}}[i]$ determined by the "length" i of the path (in moves) and the combination of states (of blob and children). Once the routine has terminated, the paths for each state combination Z^{comb} are connected (as in eq. (2)) with the paths for the involved states of the children. The result is added to the overall number of paths for the respective state of the blob, *e.g.*:

$$Z^{\mathrm{blob}}_{t_1} = \ldots + Z^{\mathrm{comb}}_{t_1;c_1,t_2} \circledast Z^{\mathrm{child}_1}_{c_1} \circledast Z^{\mathrm{child}_2}_{t_2} + \ldots .$$

Here $Z^{\mathrm{comb}}_{t_1;c_1,t_2}$ stands for the number of t_1-paths through the blob for each number of moves i with the children in states c_1 and t_2. After this has been done for all combinations of states that have occurred, we can discard all information concerning the children, and the blob will effectively shrink to a point when we treat its parent.

The computation time needed for counting the paths through a blob depends on the number of its nodes and how densely they are connected. The effort needed to connect the paths and the required memory, on the other hand, will depend on the number of different state combinations. Finding a decomposition where each blob is satisfactory in both regards is not trivial. We pursued a bottom-up *amalgamation* strategy.

First, individual sites are defined as the smallest blobs. Then all possibilities to merge any blob with a fraction of its neighbors are considered, and the one deemed best is performed. The choice is primarily based on the number of links of the resultant blob minus the maximum number of links of any of its components. A secondary criterion is the *height* of the new blob, recursively defined as the height of its highest component plus one (initial blobs have zero height). Smaller values are preferred, and the height only decides in case of a tie for the first criterion. If two options are judged as equally good, one is chosen at random. The newly formed blob is linked to its constituents in the emerging tree. This procedure is repeated until only one blob remains. The root of the tree is then defined as the blob which is the SAW's starting position. Finally we dissolve some of the blobs and give their bare sites and children to the parent. This is done in order to get rid of blobs with many links and children, which would lead to many different state combinations. However, dissolving a blob will increase the number of nodes of its parent, and it is crucial to find the right balance. To this end, we make use of the following empirical observations: 1) The peak memory on average depends on the maximal product of the number of states of a blob and its children: $M_{\mathrm{peak}} \approx f\left(\max\left[\mathrm{S}_{\mathrm{blob}} \cdot \prod \mathrm{S}_{\mathrm{child_i}}\right]\right)$, where f increases monotonously. 2) The typical computation time per blob increases exponentially with the number of *faces*, F_{blob}, of the corresponding graph and also depends on the number of states: $T_{\mathrm{blob}} \approx a \cdot b^{F_{\mathrm{blob}}} + g\left(S_{\mathrm{blob}} \cdot \prod S_{\mathrm{child_i}}\right)$. a, b are positive constants and g is again monotonously increasing. The first term estimates the time for counting the paths and the second for connecting them (as in eq. (2)). f, g, a, and b have to be determined empirically. One can then decide which blobs need to be dissolved to minimize the estimated computation time for the whole hierarchy, while a fixed limit for the predicted peak memory is kept. Further justifications for these heuristics, along with our

empirical estimates, shall be supplied in a forthcoming publication. Note that the decomposition will only affect the efficiency (or feasibility) of the enumeration and not the results.

Apart from eq. (3), the method does not explicitly depend on the dimension of the system, but our current implementation is only for the 2D case. Decomposition of a 2D CPC and enumeration of $N = 10^3$ steps took about ten minutes on average (on a 2.6 GHz PC). Theoretically, there are cases where a satisfactory decomposition cannot be achieved, and the enumeration will take extremely long. One example is the full lattice, which is a valid —if unlikely— disorder configuration. In practice, however, these cases are extremely rare, and none occurred during our simulations. We cannot yet specify the theoretical and practical limitations of the method. At present it appears that we might go up to much longer chains, although the computation will eventually become quite time consuming. The method also works slightly above p_c (≈ 0.592746 [30]), but only up to some finite length. For example, we found that for $p = 0.62$ one can still go to about $N = 300$. This becomes relevant if we want to study the crossover to the non-critical regime. For $p < p_c$ the connectivity is even lower than at p_c, so the method will work fine (but finding large enough clusters can be difficult).

To generate the percolation clusters, we used the Leath method [31]. The lattice size was chosen large enough to exclude the possibility of a walker reaching the edges. A cluster was defined as *percolating* when it wrapped around the lattice in at least one dimension (see [30]).

We enumerated SAWs of up to 10^3 steps on 2×10^5 disorder configurations, more than in any other work. On each we measured the number of SAWs $Z(N)$ and their average squared end-to-end distance $\langle R^2(N)\rangle$. Finally, we took the quenched averages where each disorder configuration contributes equally.

We determined ν via a least-square fit to the scaling law (1). The first 500 steps were excluded to reduce finite-size effects. The error was estimated using *bootstrap resampling* [32]. We thus found $\nu = 0.7754(15)$, which is slightly smaller than the values obtained in the latest studies: 0.778(15) (ref. [26]) and 0.782(3) (ref. [22]). This may be due to the fact that in these studies the backbone was used instead of the full cluster, relying on the argument that the "dangling ends" should not affect the asymptotic behavior [33]. However, the correctness of this assumption is questionable, see [20]. As our method works equally well for both cases, we hope that we can clarify this issue soon.

Estimating γ and μ is difficult due to large deviations of Z. In fact, the distribution of Z has log-normal character since Z is effectively a product of random variables. Most measured values are therefore much smaller than the average $[Z]$, which, as a result, is likely underestimated: For $N = 10^3$, we obtained $\tilde{Z} \approx 10^{164}$ as median and $\bar{Z} \approx 10^{180}$ as mean value while a lognormal fit suggests $[Z] \approx 10^{193}$.

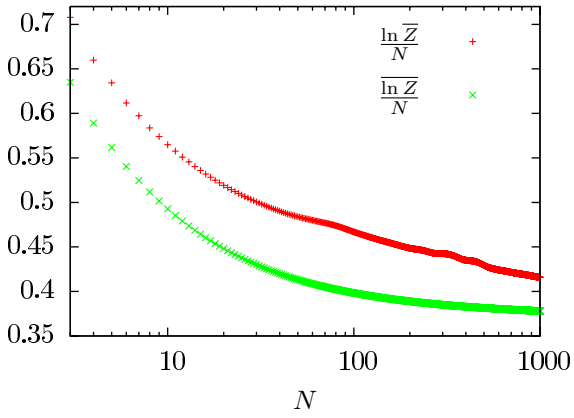


Fig. 4: (Color online) $\frac{\ln \overline{Z}}{N}$ (upper curve) and $\overline{\frac{\ln Z}{N}}$ (lower curve) *vs.* N.

Table 1: Results for γ_0, μ_0, γ, and μ for different fit ranges. The first line shows the results for the backbone from ref. [26].

Range	γ_0	μ_0	γ	μ
4–30	1.26(5)	1.456(5)	1.34(5)	1.565(5)
4–30	1.307(4)	1.4652(5)	1.34(1)	1.568(2)
10–50	1.352(7)	1.4614(5)	1.35(3)	1.567(3)
50–100	1.50(3)	1.457(7)		
100–500	1.79(2)	1.454(2)		
200–700	2.01(4)	1.453(2)		
500–1000	2.9(1)	1.451(5)		

This underestimation is a problem if we want to determine γ and μ via a least-square fit of

$$\frac{\ln \overline{Z}}{N} = \frac{\ln A}{N} + \ln \mu + (\gamma - 1)\frac{\ln N}{N}, \tag{4}$$

where A is a constant amplitude. As can be seen in fig. 4, the estimator $\frac{\ln \overline{Z}}{N}$ starts to decline much faster around $N = 70$. It also becomes noisy for large N because the value is dominated by a small number of configurations. By analyzing the dependence of $\overline{Z}$ on the number of clusters, we concluded that only the results for $N \leqslant 50$ can be trusted. For a fit range of $10 \leqslant N \leqslant 50$ we obtained $\gamma = 1.35(3)$ and $\mu = 1.567(3)$. We also determined the so-called zeroth moments γ_0 and μ_0. These are defined via $\exp([\ln Z]) \sim \mu_0^N N^{\gamma_0 - 1}$ and can be obtained by exchanging $\ln \overline{Z}$ with the mean entropy, $\overline{\ln Z}$, in eq. (4), see [26]. Here the whole measured range could be used since there are no large deviations in the distribution of $\ln Z$. For the same interval as before we got $\gamma_0 = 1.352(7)$ and $\mu_0 = 1.4614(5)$. However, when we shifted the range to larger N, the value for μ_0 slowly declined while γ_0 increased drastically; see table 1. It seems plausible that the normal γ would behave similarly. Hence we conclude that the scaling behavior of Z is much less understood than previously thought and that further research is needed here.

Nonetheless, we think that the usefulness of our method has been demonstrated. The advantage over previous methods is drastic: Using the standard enumeration method, counting of 10^{170} chains would take more than

10^{150} years, and even the most capable MC method would need weeks to produce reliable results.

Certainly, the accuracy attained in older studies would be higher were they performed today. However, the gain would probably not be very significant, especially when using exact enumeration where the computation time increases exponentially with N. In contrast, preliminary investigations suggest that for our method it increases only polynomially with an exponent smaller than three. This implies that our accuracy can still be significantly improved in the future.

So far, our method has only been implemented for two-dimensional lattices but the adaptation to higher dimensions is in progress. It could furthermore be extended to more evolved polymer models based on the SAW. Easy examples are SAWs with persistence [34] or under stretching force [28,35], kinetic SAWs [18], edge-to-edge SAWs [14], directed SAWs [12], SAWs that interact with lattice sites [12,24], or self-avoiding rings [36]. More difficult are star polymers [37] or SAWs with nearest-neighbor interactions such as θ-polymers [27]. These cases would require some modifications, but the main ideas should still apply. Other substrates such as invasion percolation clusters or geometrical fractals [38] could also be (re)investigated. Besides ν, γ, and μ, many other properties can now be determined more accurately. Examples are (universal) shape parameters [39], exponents for higher moments ("multifractal exponents") [26,40], density or free-energy fluctuations [12,24], and the scaling of probability distributions [41]. As a final remark, we would like to point out that the idea of exploiting the fractal structure through such a renormalizing treatment is probably not restricted to the study of SAWs but might, for example, be used to investigate interacting spins or transport phenomena on CPCs.

* * *

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) via FOR 877 under Grant No. JA 483/29-1, the Graduate School GSC 185 "BuildMoNa", and by the Deutsch-Französische Hochschule (DFH) under Grant No. CDFA-02-07.

REFERENCES

[1] Chakrabarti B. K., *Statistics of Linear Polymers in Disordered Media* (Elsevier, Amsterdam) 2005.
[2] Barat B. and Chakrabarti B. K., *Phys. Rep.*, **258** (1995) 377.
[3] Madras N. and Slade G., *The Self-Avoiding Walk* (Birkhäuser, Boston) 1991.
[4] Stauffer D. and Aharony A., *Introduction to Percolation Theory* (Taylor and Francis, London) 1992.
[5] de Gennes P.-G., *Scaling Concepts in Polymer Physics* (Cornell University Press, Ithaca, NY) 1976.
[6] des Cloizeaux J. and Jannink G., *Polymers in Solution* (Clarendon, Oxford) 1990.
[7] Lam P. M., *J. Phys. A*, **23** (1990) L831.
[8] Meir Y. and Harris A. B., *Phys. Rev. Lett.*, **63** (1989) 2819.
[9] Lee S. B., Nakanishi H. and Kim Y., *Phys. Rev. B*, **39** (1989) 9561.
[10] Sahimi M., *J. Phys. A*, **17** (1984) L379.
[11] Roy A. K. and Blumen A., *J. Stat. Phys.*, **59** (1990) 1581.
[12] Le Doussal P. and Machta J., *J. Stat. Phys.*, **64** (1991) 541.
[13] Lam P. M. and Zhang Z. Q., *Z. Phys. B*, **56** (1984) 155.
[14] Hovi J. P. and Aharony A., *J. Stat. Phys.*, **86** (1997) 1163.
[15] Kim Y., *J. Phys. C*, **16** (1983) 1345.
[16] Blavatska V., von Ferber C. and Holovatch Y., *J. Phys.: Condens. Matter*, **14** (2002) 9465.
[17] von Ferber C., Blavatska V., Folk R. and Holovatch Y., *Phys. Rev. E*, **70** (2004) 035104(R).
[18] Janssen H. K. and Stenull O., *Phys. Rev. E*, **75** (2007) 020801(R); **85** (2012) 011123.
[19] Kremer K., *Z. Phys. B*, **45** (1981) 149.
[20] Woo K. J. and Lee S. B., *Phys. Rev. A*, **44** (1991) 999.
[21] Grassberger P., *J. Phys. A*, **26** (1993) 1023.
[22] Blavatska V. and Janke W., *EPL*, **82** (2008) 66006; *J. Phys. A*, **42** (2009) 015001.
[23] Nakanishi H. and Lee S. B., *J. Phys. A*, **24** (1991) 1355.
[24] Vanderzande C. and Komoda A., *Phys. Rev. A*, **45** (1992) R5335.
[25] Rintoul M. D., Moon J. and Nakanishi H., *Phys. Rev. E*, **49** (1994) 2790.
[26] Ordemann A., Porto M., Roman H. E., Havlin S. and Bunde A., *Phys. Rev. E*, **61** (2000) 6858.
[27] Barat K., Karmakar S. N. and Chakrabarti B. K., *J. Phys. A*, **25** (1992) 2745.
[28] Singh A. R., Giri D. and Kumar S., *J. Chem. Phys.*, **131** (2009) 065103.
[29] Knuth D. E., *The Art of Computer Programming*, Vol. **4A** (Addison-Wesley, Upper Saddle River, NJ) 2011.
[30] Newman M. E. J. and Ziff R. M., *Phys. Rev. E*, **64** (2001) 016706.
[31] Leath P. L., *J. Phys. B*, **14** (1976) 5046.
[32] Efron B., *Ann. Stat.*, **7** (1979) 1.
[33] Kim Y., *Phys. Rev. A*, **39** (1989) 4923.
[34] Giacometti A. and Maritan A., *J. Phys. A*, **25** (1992) 2753.
[35] Blavatska V. and Janke W., *Phys. Rev. E*, **80** (2009) 051805.
[36] Gersappe D. and Olvera de la Cruz M., *Phys. Rev. Lett.*, **70** (1993) 461.
[37] Blavatska V., von Ferber C. and Holovatch Y., *Phys. Rev. E*, **83** (2011) 011803.
[38] Ordemann A., Porto M and Roman H. E., *J. Phys. A*, **35** (2002) 8029.
[39] Blavatska V. and Janke W., *J. Chem. Phys.*, **133** (2010) 184903.
[40] Blavatska V. and Janke W., *Phys. Rev. Lett.*, **101** (2008) 125701.
[41] Ordemann A., Porto M., Roman H. E. and Havlin S., *Phys. Rev. E*, **63** (2001) 020104(R).