

## Nonuniform star polymers in two dimensions

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

1994 J. Phys. A: Math. Gen. 27 L721

(<http://iopscience.iop.org/0305-4470/27/19/001>)

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

Download details:

IP Address: 171.66.16.68

The article was downloaded on 01/06/2010 at 22:35

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

LETTER TO THE EDITOR

Non-uniform star polymers in two dimensions

Peter Grassberger

Physics Department, University of Wuppertal D-42097 Wuppertal, Germany

Received 17 August 1994

**Abstract.** We present simulation results for non-uniform 2D star polymers, using a recursive implementation of the enrichment method. We show that the choice of parameters in this method is not trivial, and that a good choice can lead to extremely efficient code. We find that the exponents  $\gamma_f$  governing the increase of the  $f$ -star partition function with  $N$  are compatible with the prediction  $\gamma_f^{\text{uniform}} + f - 1$  by Duplantier.

During recent years, enormous progress has been made in understanding 2D critical phenomena, mainly due to arguments based on conformal invariance. This also applies to polymeric systems in two dimensions.

In particular, using such arguments Duplantier [1] has conjectured gamma exponents for uniform networks made up of flexible chains with arbitrary topology. By ‘uniform’ we mean that all chains connecting any two vertices have identical length. In particular, for chain polymers with  $f$  branches the conjecture of [1] (see also [2]) is

$$\gamma_f^{\text{uniform}} = \frac{86 - 9(f - 2)(f - 1)}{64} \tag{1}$$

where  $\gamma_f^{\text{uniform}}$  is defined via the number of distinct configurations with fixed central vertex,  $Z_N^{(f),\text{uniform}} \sim \mu^N N^{\gamma_f^{\text{uniform}} - 1}$ . The attrition constant  $\mu$  is the same as for ordinary (unbranched) SAWs, and  $N$  is the total number of monomers.

Equation (1) gives, in particular, the well known prediction  $\gamma = \frac{43}{32}$  for unbranched polymers ( $f = 1$  or  $f = 2$ ) [3], and it gives

$$\gamma_3^{\text{uniform}} = \frac{17}{16} \quad \gamma_4^{\text{uniform}} = \frac{1}{2} \tag{2}$$

for 3- and 4-stars. All these predictions are nicely confirmed by exact calculations [4–6].

The situation is more confused for non-uniform (or ‘polydisperse’) star polymers. These are stars where the arms are allowed to have different lengths, the sum over all lengths being  $N$ . There, enumerations [7] have led to the conjecture

$$\gamma_f = \gamma + f - 1. \tag{3}$$

However, theoretically it is predicted [8] that

$$\gamma_f = \gamma_f^{\text{uniform}} + f - 1. \tag{4}$$

For  $f = 3$  and 4 this would give  $\gamma_3 = 3 + \frac{1}{16} = 3.0625$ ,  $\gamma_4 = 3.5$ .

Finally, recent simulations by Zhao and Lookman [9] have given values in between,

$$\gamma_3 = 3.15 \pm 0.05 \quad \gamma_4 = 3.65 \pm 0.05. \tag{5}$$

Both values agree within three standard deviations with the predictions of [8], such that there is no strict disagreement given the usual uncertainties in estimating critical exponents. But Zhao and Lookman say that they believe that such agreement 'is not true for general non-uniform networks. For 3-stars, exact enumerations by Camacho *et al* [10] gave a similar result,  $\gamma_3 = \gamma + 1.79 \pm 0.05$ .

In this letter we want to present much larger simulations than those of [9] which suggest strongly that (4) is indeed correct for  $f = 3$  and 4. We, of course, cannot say anything about more general networks.

In [9], inversely restricted sampling (Rosenbluth and Rosenbluth method [11]) was used to generate stars with  $N$  up to 180. For this largest  $N$ , 50 000 configurations were generated. In the present study we generated stars for *all* values of  $N$  up to 1000, and for each  $N$  the number of chains was  $> 10^8$ . This large number should, of course, be understood with some caution. Due to the method used in our simulations (enrichment [12]; see below) most of our configurations are statistically dependent. But also the numbers of statistically strictly independent chains are rather large. For  $N = 180$  they are  $> 10^6$  (all numbers quoted hold for  $f = 3$  and for  $f = 4$ ), and for  $N = 1000$  we still have  $> 10^5$  independent chains.

But even these numbers are not really significant, just as the numbers quoted in [9] have to be properly interpreted in order to judge their significance. In the Rosenbluth method the main problem is that chains generated by the algorithm must be weighted when forming averages, with weights which become increasingly inhomogeneous with increasing  $N$ . Thus, for large  $N$  all statistical averages are dominated by only a very few configurations even if the sample is huge, and thus statistical errors are much larger than one could expect naively. In our implementation of enrichment the problem is slightly different, but—unless special care is taken—with the same net effect. As we have already mentioned, in this method configurations are not generated independently. If the amount of correlation between successive configurations is (roughly) constant, this is not a big problem: we just have to count the number of independent configurations as we did above. But it might happen (and it does, unless the implementation is done carefully) that the correlation time varies strongly. In that case the total sample might be dominated by only a few clusters (or 'tours' in the notation of [13]) of strongly correlated configurations, and the statistical errors are again large.

In our implementation [13] of the enrichment method for linear chains [12] we use a recursively called subroutine  $\text{STEP}(x, N)$ , the arguments of which are the actual end-point of the chain and its actual length  $N$ . When called, it first marks the site  $x$  as occupied and selects a random† neighbouring site  $x'$  of  $x$ . If this site is not yet marked as occupied and if  $N$  is smaller than the maximal chain length  $N_{\text{max}}$ , STEP is called recursively with new arguments  $x'$  and  $N + 1$ . After returning from this call, a uniformly distributed random number  $r \in [0, 1]$  is drawn, and  $\text{STEP}(x', N + 1)$  is called again if  $r + 1$  is less than some prescribed number  $P_N > 1$ . The last is the enrichment step proper, and it implies that we try, on average,  $P_N$  extensions of each chain in the sample in order to overcome attrition.

In the generalization to non-uniform star polymers, we first chose an arbitrary but fixed numbering of the branches, labelling them by numbers  $k = 1, \dots, f$ . The central vertex is located at  $x_0$ , and the branches start at neighbouring sites  $x_k$ . We add  $k$  as an additional argument in the subroutine STEP, and modify  $\text{STEP}(x, N, k)$  by including the following steps: after having marked  $x$  as occupied and before calling itself again with the same argument  $k$ , we choose a random number  $q$  and call  $\text{STEP}(x_{k+1}, N + 1, k + 1)$  if

† Except that we forbid immediate reversals of the chain.

$q < Q_k P_N$ , where the constants  $Q_k$  are chosen such that  $0 < Q_k P_N \leq 1$  for all  $N$  and  $f$ . This we do, of course, only if  $k < f$  and  $N < N_{\max}$ , and it corresponds to starting the next branch. The first call from the main program is with  $k = 1$ , and configurations are counted only if all  $f$  branches have really been generated, i.e. if the subroutine is called with  $k = f$ . Indeed, by counting how often the subroutine was called with each value of  $k$ , we can estimate the numbers of  $k$ -star stars for all values of  $k$  (and for all  $N \leq N_{\max}$ ) in a single run†.

In this algorithm we have two sources of freedom: the probabilities  $P_N - 1$  for making a second attempt to prolongate a chain, and the probabilities  $Q_k$  to start a new chain. One can show that  $P$  is not allowed to depend on  $k$ , and  $Q$  has to be independent of  $N$ , if one wants to avoid any bias. This bias could be compensated only if one keeps detailed statistics for the numbers of stars with  $N_k$  branches, which would have not been easy on the sort of workstation where the present work was done (the 'bias' introduced by having  $P_N$  depend on  $N$  and  $Q_k$  on  $k$  is removed trivially, in contrast). A possible criterion in choosing these constants could be the same as with unbranched SAWs [14, 15]: we want to compensate attrition as neatly as possible, i.e. we want to generate roughly the same number of configurations for each  $N$ . Otherwise we would not generate enough long chains (if  $P_N$  is too small), or we would generate too many strongly correlated stars (if  $P_N$  is too large). With  $Q_k \approx 1$  this would imply  $P_N \approx (3/\mu)(1 + 1/N)^{1-\nu}$  [15].

But it is not clear *a priori* that  $Q_k \approx 1$  is a good choice. We had tried it first, with disastrous results: although the numbers of independent configurations were large when  $P_N$  was chosen according to the above criterion, we ran into the problem that only very few large clusters of correlated configurations dominated even huge samples. It is indeed not too difficult to understand that the best choice is  $Q_k \gg 1$ : with  $P_N Q_k = 1$  we start a new branch at every step, which means that a very large number of high- $k$  branches are generated in the same 'background' of low- $k$  branches. In an optimal implementation, the number of branches generated should be roughly the same for each  $k$ . This is achieved with  $Q_k \approx k/N_{\max}$ . Actually we got best results with slightly larger values,  $Q_k = k/150$ , and with  $P_N = 3(1 + 1/(N + 2))^{-1.6}/\mu$ . With this choice, we were able to reduce CPU times by factors  $\approx 10^3$ , compared to simulations with  $Q_k \approx 1$ . This is in spite of the fact that the number of configurations increases quite strongly with  $N$ . It seems to be more important to have a sample well balanced over  $k$  than over  $N$ . We should, however, stress that this choice need not be optimal. We did not attempt any systematic optimization, thus further improvements might have been possible.

Our final results, based altogether on  $\approx 150$  h CPU time on DEC ALPHA workstations, are shown in figure 1. There we plotted effective exponents

$$\gamma_f^N = \frac{1}{\ln 2} \ln \frac{Z_N^{(f)}}{\mu^{N/2} Z_{N/2}^{(f)}} \quad (6)$$

against  $1/N$ . For  $\mu$  we used the value of [16]. The curves connect points with successive values of  $N$ , and the thickness of the curves correspond to  $\pm 1$  standard deviation.

If there were only analytic corrections to scaling, these lines would be straight, and the extrapolations to  $1/N \rightarrow 0$  would give  $\gamma_f$ . We see that there are obviously non-analytic

† We should mention that this is not the only way to implement enrichment recursively. A different structure is, for example, better suited for stars with the end-point fixed. But all these implementations are equivalent as far as efficiency is concerned. We should also point out that uniform star polymers can be generated in a very similar way: we just call STEP( $x_{k+1}, N + 1, k + 1$ ) only when the  $k$ th branch is finished. We finally point out that our algorithm generates a sample where configurations are considered different if they involve different *bonds*, even if *sites* are the same ('weak embedding').

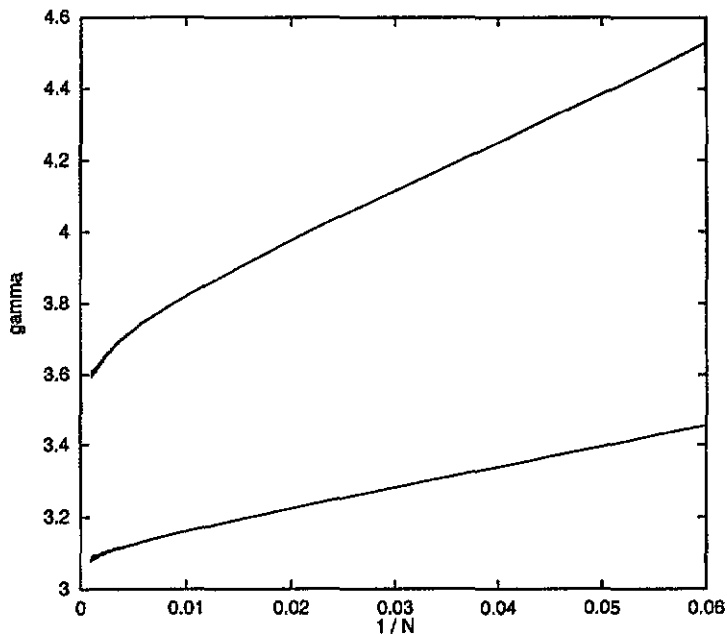


Figure 1. Effective exponents (see equation (6)) for non-uniform 3-stars (lower curve) and 4-stars (upper curve), plotted against the inverse total number of monomers. For large  $N$ , the thickness of the curves indicates  $\pm 1$  standard deviation, while the statistical errors are much smaller than the thickness for small  $N$ .

corrections to scaling. They are particularly strong for  $f = 4$ , but they are also significant for  $f = 3$ . Both curves cannot be fitted by simple power laws, whence a single correction term would not be enough. For that reason also plotting the data against any other negative power of  $N$  would not give straight lines, and extracting the leading correction to scaling becomes very difficult. As a consequence, the extrapolation is not trivial. Our final error bars are essentially educated guesses and are much larger than the statistical errors of our finite- $N$  data. We obtain

$$\gamma_3 = 3.065 \pm 0.010 \quad \gamma_4 = 3.53 \pm 0.02. \quad (7)$$

This suggests that (4) is indeed satisfied.

I am indebted to C v Ferber and L Schäfer for pointing out [8] to me. This work was supported by DFG, SFB 237.

## References

- [1] Duplantier B 1986 *Phys. Rev. Lett.* **57** 941
- [2] Ohno K and Binder K 1988 *J. Physique* **49** 1329
- [3] Nienhuis B 1982 *Phys. Rev. Lett.* **49** 1062
- [4] Wilkinson M K, Gaunt D S, Lipson J E G and Whittington S G 1986 *J. Phys. A: Math. Gen.* **19** 789
- [5] Gaunt D S, Lipson J E G, Whittington S G and Wilkinson M K 1986 *J. Phys. A: Math. Gen.* **19** L811
- [6] Lipson J E G, Gaunt D S, Wilkinson M K and Whittington S G 1986 *Macromolecules* **20** 186
- [7] Gaunt D S, Lipson J E G, Torrie G M, Whittington S G and Wilkinson M K 1984 *J. Phys. A: Math. Gen.* **17** 211
- [8] Duplantier B 1989 *J. Stat. Phys.* **54** 581
- [9] Zhao D and Lookman T 1994 *Europhys. Lett.* **26** 339

- [10] Camacho C J, Fisher M E and Straley J P 1992 *Phys. Rev. A* **46** 6300
- [11] Rosenbluth M N and Rosenbluth A W 1955 *J. Chem. Phys.* **23** 356
- [12] Wall F T and Erpenbeck J J 1959 *J. Chem. Phys.* **30** 634, 637
- [13] Grassberger P and Hegger R 1994 *Preprint* WU-B 94-01
- [14] Beretti A and Sokal A D 1985 *J. Stat. Phys.* **40** 483
- [15] Grassberger P 1993 *J. Phys. A: Math. Gen.* **26** 2769
- [16] Conway A R, Enting J G and Guttmann A J 1993 *J. Phys. A: Math. Gen.* **26** 1519