A Monte Carlo method for series expansions

J. Phys. A: Math. Gen. 19 (1986) L1057-L1061. Printed in Great Britain

LETTER TO THE EDITOR

A Monte Carlo method for series expansions

Deepak Dhar† and P M Lam‡

Institüt fur Theoretische Physik, Universitat zu Köln, D-5000 Koln 41, West Germany

Received 7 July 1986, in final form 20 August 1986

Abstract. We describe a simple and general algorithm to calculate series expansions in enumeration problems to large orders approximately by a Monte Carlo method. It can be used to generate unbiased samples in cluster studies, e.g. linear or branched polymers, random surfaces, etc, in any dimension. We calculate the number of site animals of size n on the square lattice for $n \leqslant 50$ and their average size to better than 1% accuracy.

The analysis of series expansions has been a very powerful technique in statistical mechanics. These calculations have been pushed to quite high orders for many problems, and in many cases yield the most accurate known estimates of physical quantities of interest, e.g. the critical exponents [1]. However, the computational effort needed to calculate the nth-order term increases exponentially with n, and one is usually restricted to studying $n \leqslant 30$ in two dimensions [2]. In higher dimensions the situation is even worse, and for three-dimensional systems, series with more than 20 known terms are quite rare. Even if the asymptotic behaviour of the series is known theoretically in parametric form, precise estimates of the parameters such as critical exponents are hard to obtain from a short series because of significant corrections to scaling [3].

Often the calculation of series coefficients can be posed as an enumeration problem of some kinds of objects such as graphs or configurations (e.g. the linear or branched polymer problems). The exponential increase of the number of such configurations with their size n makes a brute-force enumeration infeasible for large n.

In this letter we describe a simple algorithm, to be called the incomplete enumeration method, that allows an approximate determination of the number of configurations in enumeration problems by a Monte Carlo technique. The results obtained are only of finite accuracy, and are complementary to the exact series analysis because of the much higher orders reachable. While approximate values of high-order terms can be obtained by existing extrapolation methods such as exact series analysis or phenomenological renormalisation, it should be stressed that the error bars in these estimates are purely subjective. The magnitude of correction to the assumed extrapolation form is completely unknown. By contrast, the errors in the high-order coefficients calculated by the present method are purely statistical, objectively determined and can be made arbitrary small. Of course, extrapolated quantities such as critical parameters still have unknown, though smaller, systematic errors. The computation time increases only as a

† Permanent address: Tata Institute of Fundamental Research, Homi Bhabha Road, Bombay-400005, India.

‡ Permanent address: Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing, China.

0305-4470/86/171057+05$02.50

power of n, and an accuracy of better than 1% is easily obtained for n as large as 50. The algorithm generates an unbiased sample [4], and other configuration averages can also be computed. It gives a direct estimate of the number of configurations, and hence of entropy.

As a test case, we studied the number of site animals A_n on a square lattice, and their average moment of inertia I_n as a function of the number of sites n for $1 \leqslant n \leqslant 50$. An accuracy of about 0.5% for A_{50} and about 0.13% for I_{50} required approximately 20 h of CPU time on a CDC Cyber 175 machine. In contrast, the exact enumeration of A_{24} already needs 10 months of CPU time on a PDP-11/70 computer [5]†, and each successive order requires a four-fold increase in CPU time. On our machine, an exact calculation of A_{50} would require approximately 4×10^{20} h.

For concreteness, we shall describe the algorithm for this special case. Other enumeration problems can be handled similarly. For the enumeration of lattice animals, there is a well known deterministic algorithm using backtracking [5, 1], and an elegant and short FORTRAN program using this algorithm has been published [6]. We describe it very briefly here.

One starts by choosing a rule for designating for each animal configuration one of its sites as the last added site. Deleting the last added site from an n-site animal A, we obtain an $(n-1)$-site animal A', called the parent of A. The animal configurations are thus classified into a tree structure according to their lineage. The unique 1-site animal forms the root of the tree. The n-site configurations are at a height $(n-1)$ connected to their parents at height $(n-2)$ by single bonds. The exhaustive enumeration of all animals having n sites proceeds by a systematic exploration of the genealogical tree to height $(n-1)$ [6].

In the incomplete enumeration method, we arbitrarily choose a set of $(n-1)$ real numbers p_i with $0 < p_i \leqslant 1$ and $i = 2$ to n. Any configuration with r sites $(2 \leqslant r \leqslant n)$ is defined to be 'stillborn' with probability $(1-p_r)$ and normal with probability p_r, independent of the state of all other configurations. We delete the stillborn configurations from the genealogical tree. All configurations which are descendants of stillborn configurations are also deleted. We then systematically enumerate the remaining n-site configurations in the genealogical tree. This can be done very efficiently using backtracking. When a particular r-site configuration is first generated, one chooses to ignore it and all its descendants in the enumeration with probability $(1-p_r)$.

The probability that a particular r-site animal will be enumerated in a given trial is $p_2 p_3 \ldots p_r \overset{\text{def}}{=} \Pi_r$ (say), and is the same for all configurations with the same r. The algorithm thus generates an *unbiased* sample of configurations. The number of n-site animals generated in one trial is a random variable X_n. Averaging over several trials, we can estimate $\langle X_n \rangle$. Since this equals $\pi_n A_n$, we can estimate A_n using Monte Carlo methods.

By judiciously choosing p_i, we can get $\langle X_n \rangle$ to be close to 1. Then the percolation process on the genealogical tree is close to threshold, and the variance of X_n increases slowly with n, and may be approximated as increasing like n^a, where a is some exponent. For a genealogical tree with constant branching number, a can be calculated analytically, and one finds $a = 1$. For trees with variable branchings, a is harder to calculate exactly. To obtain the fractional error in A_n less than ε, the number of trials has to increase as $\varepsilon^{-2} n^a$ for large n. Since the time required to generate an animal

† The computer time could be decreased considerably by using a suitable generating function technique (see Sykes [5]).

increases linearly with n, the computer time to obtain a fractional error in A_n increases as $\varepsilon^{-2}n^{a+1}$.

The method may be viewed as constructing a *site percolation process on the genealogical tree*, with the stillborn configurations as the blocked nodes on the tree. The enumerated configurations form the connected part of the tree.

It is useful to contrast the Monte Carlo algorithm outlined here with those commonly used. The conventional method, as developed by Metropolis *et al* [7] relies on the construction of a Markov process whose time averages replace the ensemble averages sought. Averaging over the initial state may also be done [8]. This procedure gives rise to strong time correlations in samples and, in many problems of physical interest, the approach to equilibrium distribution can become very slow due to hydrodynamic slowing, or critical slowing or metastability [8]. Also it is not possible to determine thermodynamic quantities such as entropy or free energy directly from the simulations and these must be calculated by integration over temperatures.

The present algorithm does not construct a Markov process, and the problem of correlations is much more manageable. Different configurations produced in a single trial are, of course, correlated with each other being more likely to share a common lineage. There is also a small anticorrelation since a configuration can occur at most once in a single trial (sampling without replacement).

The variability of X_n in different trials is crucial in keeping the sample unbiased, as is the case in the simple algorithm for generating self-avoiding walks by randomly generating an ideal random walk of n steps, and rejecting it if it is found to be self-intersecting. This algorithm is inefficient for large n, as the fraction of accepted configurations is very small. In the incomplete enumeration approach discussed here, a step that would lead to rejection makes the algorithm backtrack to attempt another allowed completion, and does not lead to a rejection of the full configuration.

The basic idea of combining an enumeration algorithm with indeterminism has been discussed earlier by Redner and Reynolds [9] in the case of self-avoiding walks and by Yang *et al* [10]. For the animals problem a similar algorithm was proposed by Lam [11]. However, the detailed algorithm used by Lam is not completely free of bias. The present algorithm differs from his by having a different rule for backtracking. His algorithm may try to generate some configurations more than once during the execution of the program (see figure 1). This makes the sample biased in favour of such configurations. Another source of bias in it is introduced by terminating a trial before completion as soon as some preset number of configurations has been generated.

The approach is easily generalised to other enumeration problems such as linear or branched polymers in restricted geometries, random surfaces, etc. In each case the exponential growth of the number of configurations is curtailed by randomly pruning off branches in the genealogical tree of configurations. The technique can also be

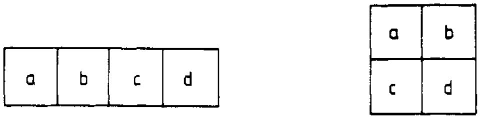

Figure 1. An example to illustrate the difference between the present algorithm and that given in [11]. We show two animals of size 4 on a square lattice. By Lam's algorithm [11], the second animal may be generated by sequential occupation of sites a b c d or a c d b or a b d c, and thus is three times as likely to be generated in a single trial as the first animal. This bias is avoided in the present algorithm.

generalised to generate configurations with prescribed (unequal) weights (needed, for example, in study systems like the Ising model in the canonical ensemble). It is not suitable for determining the behaviour of non-leading singularities. A detailed analysis of the algorithm and additional numerical results will be published elsewhere.

For the numerical study of site animals on the square lattice, we chose $p_r = (r+1)/4r$ for $r \geqslant 2$, and averaged over 4×10^7 independent trials. The mean number $\langle X_n \rangle$ obtained for $n = 3$-50 is shown in table 1. The standard error of estimate shown was obtained by grouping the data into 150 equal parts and calculating fluctuations about the mean value. The average moment of inertia I_n was calculated in a separate run of 4×10^6 trials.

Table 1. The mean number of animals generated per trial $\langle X_n \rangle$ as a function of n. The estimated total number of n-site animals is $A_n = \langle X_n \rangle 4^{n-1} 2/(n+1)$.

n	$\langle X_n \rangle$	n	$\langle X_n \rangle$	n	$\langle X_n \rangle$
3	0.7497 ± 0.0004	19	0.8641 ± 0.0019	35	1.0943 ± 0.0040
4	0.7423 ± 0.0005	20	0.8765 ± 0.0020	36	1.1120 ± 0.0043
5	0.7383 ± 0.0006	21	0.8888 ± 0.0022	37	1.1287 ± 0.0044
6	0.7382 ± 0.0007	22	0.9019 ± 0.0023	38	1.1461 ± 0.0046
7	0.7423 ± 0.0008	23	0.9158 ± 0.0023	39	1.1632 ± 0.0048
8	0.7479 ± 0.0008	24	0.9292 ± 0.0024	40	1.1812 ± 0.0050
9	0.7559 ± 0.0009	25	0.9427 ± 0.0026	41	1.1997 ± 0.0052
10	0.7645 ± 0.0010	26	0.9568 ± 0.0027	42	1.2185 ± 0.0054
11	0.7741 ± 0.0011	27	0.9704 ± 0.0029	43	1.2368 ± 0.0056
12	0.7844 ± 0.0012	28	0.9853 ± 0.0030	44	1.2561 ± 0.0058
13	0.7944 ± 0.0013	29	0.9997 ± 0.0031	45	1.2753 ± 0.0060
14	0.8053 ± 0.0014	30	1.0149 ± 0.0032	46	1.2950 ± 0.0062
15	0.8160 ± 0.0015	31	1.0295 ± 0.0034	47	1.3146 ± 0.0065
16	0.8275 ± 0.0016	32	1.0454 ± 0.0035	48	1.3344 ± 0.0068
17	0.8393 ± 0.0018	33	1.0611 ± 0.0037	49	1.3549 ± 0.0071
18	0.8515 ± 0.0019	34	1.0775 ± 0.0038	50	1.3759 ± 0.0074

The values $\langle X_n \rangle$ vary approximately exponentially with n for $n \geqslant 15$, and a linear plot of $\log \langle X_n \rangle$ against n gives

$$\ln\langle X_n \rangle \simeq \alpha n + \beta$$

with $\alpha = 0.0153 \pm 0.0004$ and $\beta = -0.455 \pm 0.020$. No term proportional to $\log n$ is expected, as the exactly known exponent $\theta = 1$ has been incorporated in our choice of p_n. These values agree with the result $\alpha = 0.015\,52 \pm 0.000\,04$ obtained by phenomenological renormalisation [12], and $\alpha = 0.015\,53 \pm 0.000\,05$, $\beta = -0.456 \pm 0.010$ obtained by extrapolation of the exact series coefficients [13]†. A plot of $n \log(I_{n+1}/I_n)$ against $1/n$ is an approximate straight line which extrapolated to $n \to \infty$ gives $I_n \sim n^{2\nu+1}$ with $\nu = 0.646 \pm 0.004$. This value of ν is also in agreement with earlier, more precise estimates. The advantage of the present technique is that it can be used with equal ease in higher dimensions, where the existing estimates are much less precise [14].

We would like to thank Dietrich Stauffer for useful advice, critical comments on the manuscript and references to earlier literature. DD thanks Sonder Forschung Bereich 125 and PML the Max Planck Gesellschaft for financial support.

† Our parameter $\alpha = -\log(4x_c)$ is in the notation of the authors in [12].

References

[1] Domb C and Green M S (ed) 1974 *Phase Transitions and Critical Phenomena* vol 3 (New York: Academic)

[2] Baxter R J, Enting I G and Tsang S K 1980 *J. Stat. Phys.* **22** 465

[3] Adler J, Moshe M and Privman V 1983 *Ann. Israel Phys. Soc.* **5** 397

[4] Glaus U 1985 *J. Phys. A: Math. Gen.* **18** L609
Beretti A and Sokal A D 1985 *J. Stat. Phys.* **40** 483
Aragao de Carvalho C and Carcibolo S 1983 *J. Physique* **44** 323

[5] Redelmeier D H 1981 *Discrete Math.* **36** 191
Sykes M F 1986 *J. Phys. A: Math. Gen.* **19** 1007

[6] Redner S 1982 *J. Stat. Phys.* **29** 309
Demme E S and Diemer K 1984 *J. Undergrad. Res. in Phys.* **3** 25

[7] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 *J. Chem. Phys.* **21** 1087

[8] Binder K (ed) 1979 *Monte Carlo Methods* (Berlin: Springer)

[9] Redner S and Reynolds P J 1981 *J. Phys. A: Math. Gen.* **14** 2679
Fisher M E, Privman V and Redner S 1984 *J. Phys. A: Math. Gen.* **17** L569
Privman V and Redner S 1985 *J. Phys. A: Math. Gen.* **18** L781

[10] Yang Y S, Liu Y and Lam P M 1985 *Z. Phys.* B **59** 445

[11] Lam P M 1986 *J. Phys. A: Math. Gen.* **19** L155

[12] Derrida B and Stauffer D 1985 *J. Physique* **46** 1623
Kertész J 1986 *J. Phys. A: Math. Gen.* **19** 599

[13] Guttman A J 1982 *J. Phys. A: Math. Gen.* **15** 1987

[14] Stanley H E, Redner S and Yang Z R 1982 *J. Phys. A: Math. Gen.* **15** L569