Degenerate Rosenbluth Monte Carlo scheme for cluster counting and lattice animal enumeration

C. M. Care
Materials Research Institute, Sheffield Hallam University, Pond Street, Sheffield S1 1WB, England
(Received 2 January 1997)

We describe an algorithm for the Monte Carlo enumeration of clusters and lattice animals. The method may also be used to calculate associated properties such as moments or perimeter multiplicities of the clusters. The scheme is an adaptation of the Rosenbluth method for growing polymer chains and may be used to estimate the number of distinct lattice animals on any lattice topology. The method is validated against exact and Monte Carlo enumerations for clusters of size 30, on two- and three-dimensional simple-cubic lattices. The method may be readily adapted to yield Boltzmann weighted averages over clusters. [S1063-651X(97)04807-1]

PACS number(s): 02.70.Lq, 05.50.+q

I. INTRODUCTION

The enumeration of lattice animals has attracted considerable attention over the last 20 years. They are important in a variety of physical problems including nucleation [1], percolation [2], and branched polymers [3]. In the latter case it has been established that dilute branched polymers and lattice animals belong to the same universality class, having an upper critical dimension of 8 [4].

A site (bond) lattice animal is a cluster of N connected sites (bonds) on a lattice with given symmetry and dimensionality and we seek to enumerate all distinct animals with a given number of sites (bonds). Exact enumeration has been carried out for small lattice animals using a variety of methods [2,5,6] but the methods become computationally prohibitive for large animals since the number of animals a_N behaves asymptotically as [6]

$$a_N = A N^{\theta} \lambda^N, \tag{1}$$

where A and λ depend on the type of lattice and θ is a universal constant dependent on the dimensionality. Many techniques have been used to enumerate larger lattice animals including various Monte Carlo growth schemes [2,7–9], a constant fugacity Monte Carlo method [10], an incomplete enumeration method [4], and reaction limited cluster-cluster aggregation [3]. Many of these schemes generate estimates which include a small, but unquantifiable, bias.

In the following paper we describe a method based on an extension of the scheme proposed by Rosenbluth and Rosenbluth [11] for enumerating self-avoiding polymer chains. The scheme provides an estimate of the number of lattice animals and can also yield estimates of any other desired properties of the animals such as their radius of gyration or perimeter multiplicities [2]; the estimates are unbiased in the limit of large samples. A merit of the scheme is that for thermal systems it may be easily adapted to include Boltzmann weightings following, for example, the arguments used by Siepmann and Frenkel [12] in the development of the configurational bias technique. A possible numerical limitation of the method arises from the highly skewed probability distribution of Rosenbluth weights which occurs for large cluster sizes. However, if only compact clusters are sampled in a Boltzmann weighted scheme, it is likely that the skewness will present less of a problem.

II. ROSENBLUTH SCHEME

In this section we summarize the original Rosenbluth method in order to clarify how it may be extended. The method is a Monte Carlo technique which generates a sample ensemble of N_E self-avoiding chains of length N on a lattice which may be of arbitrary dimensionality and coordination number. As each member of the ensemble is grown, a weight is constructed which subsequently allows the calculation of ensemble averages of quantities associated with the chains such as their average radii of gyration $\overline{R_N^2}$ or the total number c_N of self-avoiding chains of length N.

Each chain in the ensemble is grown by placing an initial segment on the lattice and then adding successive segments to the end of the chain. If we consider a chain which already has i segments, the $(i+1)$th segment is added to the end of the chain by (a) assigning a normalized probability p_i^{ω} for the addition of the $(i+1)$th segment to one of the ω sites adjacent to the end segment of the chain; (b) selecting one of these sites, ω_i, by simple Monte Carlo sampling, with probability $p_i^{\omega_i}$; (c) repeating steps (a) and (b) until the chain has grown to the required length N; and (d) a weight W_{α}, defined below, is associated with the construction of each member, α, of the ensemble. If it is impossible to complete the growth of the chain, because all directions are blocked, the weight for the chain is set to zero. This chain must be included in the counting associated with the weighted average defined below.

Provided that $\Sigma_{\omega} p_i^{\omega} = 1$, there is no necessity for all of the p_i^{ω} to be equal, but this is usually adopted for the construction of athermal chains. It is important that the choice of the p_i^{ω} allows the growth of all possible chains (i.e., is ergodic) and it should be noted that the choice affects the way in which the method converges to the required averages. A weighted average over the ensemble may be defined for any property O_{α} of the chains as follows:

$$\langle O \rangle_W = \frac{1}{N_E} \sum_{\alpha=1}^{N_E} W_{\alpha} O_{\alpha}. \tag{2}$$

The expectation value of $\langle O\rangle_W$ is given by

$$E[\langle O\rangle_W]=\frac{1}{N_E}\sum_{\alpha=1}^{N_E}E[W_\alpha O_\alpha] \tag{3}$$

$$=\sum_{\nu=1}^{c_N}(P_\nu W_\nu O_\nu) \tag{4}$$

where we have used α to index the members of the ensemble and ν to index each of the c_N possible chains of length N. P_ν is the normalized probability of growing the chain ν given by

$$P_\nu=\prod_{i=2}^{N}p_i^{\omega_i}, \tag{5}$$

where $p_i^{\omega_i}$ is the probability of selecting adjacent site ω_i during the growth of the chain. Rosenbluth and Rosenbluth essentially chose

$$W_\nu=\frac{1}{P_\nu} \tag{6}$$

and with this choice it follows that

$$E[\langle O\rangle_W]=\sum_{\nu=1}^{c_N}O_\nu. \tag{7}$$

Hence

$$E[\langle 1\rangle_W]=c_N, \tag{8}$$

$$E[\langle R_\nu^2\rangle_W]=\sum_{\nu=1}^{c_N}R_{N\nu}^2=c_N\overline{R_N^2}. \tag{9}$$

If the $p_i^{\omega_i}$ and W_ν are chosen appropriately, it is possible to grow the chains with Boltzmann weightings and this is the technique used in algorithms such as the configurational bias scheme [12].

III. CLUSTER GROWTH ALGORITHM

A. Simple scheme

The result (8) raises the possibility that a Rosenbluth scheme could be adapted to enumerate lattice animals and properties associated with lattice animals. We thus consider using the Rosenbluth scheme described above to generate an ensemble of lattice animals by successive addition of ''bricks'' to form a connected cluster of N occupied sites. The only difference from the previous scheme is that during the growth process, the $(i+1)$th brick may be added to *any* of the sites adjacent (i.e., vacant and connected) to the i cluster. This process will sample all possible lattice animals but unfortunately introduces a degeneracy which must be evaluated if the exact number of lattice animals is to be enumerated. Thus at each point in the cluster growth a choice is made of (a) the brick in the i cluster to which the $(i+1)$th brick is to be attached and (b) the site adjacent to that brick which is to be occupied. It is step (a) which introduces degeneracy because of the number of different ways in which the growth can proceed. In the process of *chain* growth in the original Rosenbluth scheme, the degeneracy is 1 since the next segment (i.e., brick) can only be added at the end of the chain. The calculation of the degeneracy associated with the process of selecting the site for attachment in this simple scheme is not trivial since it depends on the connectivity of the final cluster. It is possible to evaluate the degeneracy by using the Rosenbluth scheme a second time, once the connectivity of the final cluster is known. However, although tractable, the time required to obtain ensemble averages to a given accuracy using this ''double'' Rosenbluth scheme makes the method unusable for any but the smallest clusters.

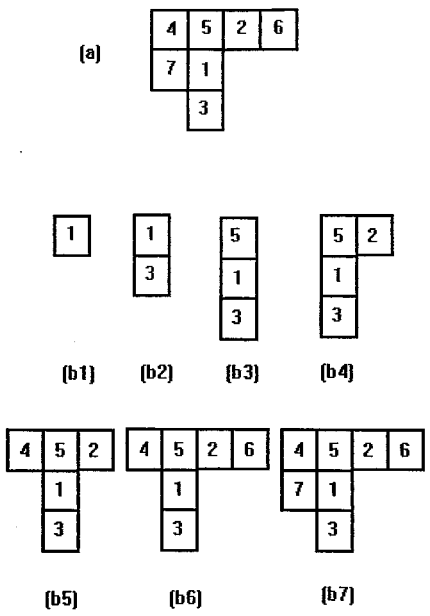

FIG. 1. Example of growth sequence for a cluster of seven labeled bricks. (a) Final cluster with arbitrary labeling. (b1)–(b7) Unique growth sequence to achieve given labeled cluster using the algorithm described in Sec. III B 1.

B. Degenerate Rosenbluth scheme

1. A unique cluster growth sequence

The problem of the degeneracy may be resolved if we consider each of the N bricks used to build the cluster to have a label κ which is an integer in the range $1,\ldots,N$ inclusive. We then construct a scheme which allows us to generate only once each of the $N!$ possible arrangements of the labeled bricks into a *given* cluster shape; i.e., we force the degeneracy to be $N!$. The algorithm described in the next section is based on the observation that for any given labeled N cluster it is possible to generate a unique growth sequence by the following scheme which is illustrated in Fig. 1.

(i) Select brick $\kappa=1$ as the first brick.

(ii) Identify all the bricks connected to $\kappa=1$ and select the brick with the lowest value of κ as the next brick.

(iii) Repeat (ii) at each stage of the cluster growth, i.e., identify all the bricks connected to the cluster of size i and select the one with the lowest κ as the next brick in the growth sequence.

Figure 1(a) shows an arbitrary cluster of seven labeled bricks and Figs. 1(b1)–1(b7) illustrate the unique growth se-

quence by which the above scheme would generate this cluster. In the following section we describe an algorithm which enforces this growth sequence.

2. The algorithm

We construct an ensemble of clusters and for each cluster we calculate a weight which we can subsequently use to calculate weighted averages of cluster properties. In particular, it can be seen from Eq. (8) that the average of the weights is an estimate of the number a_N of lattice animals of size N.

We construct each cluster from a set of labeled ''bricks''; the label on each brick is denoted by κ, an integer in the range $1,\ldots,N$ inclusive. As bricks are added to the cluster we maintain a record of the set of sites (ω in number) which are adjacent to the cluster, i.e., vacant and connected to the cluster. For each such adjacent site we record a quantity κ_m which is the minimum value of κ for a brick which may be placed in that adjacent site. The value of κ_m for any given adjacent site changes, in a manner described in detail below, as the cluster is grown.

We begin each cluster growth by placing the brick with $\kappa=1$ on the lattice and repeat the following steps until the cluster is fully grown.

(i) Select one of the adjacent sites as the site which is next to be occupied and delete the site from the list of available adjacent sites.

(ii) Select one of the remaining bricks with a κ value greater than the κ_m for that adjacent site.

(iii) Add the brick to the cluster and remove it from the set of available bricks.

(iv) adjust the record of adjacent sites and their associated κ_m values.

(v) Accumulate the data necessary to calculate the weight to be associated with the cluster.

We now comment on each of these steps in more detail.

(i) The adjacent site to be used for the attachment of a brick is chosen from the set of ω available adjacent sites by simple Monte Carlo sampling with a probability p_i^{ω}. This probability may simply be chosen to be $p_i^{\omega}=1/\omega$, but more powerful methods of constructing the p_i^{ω} are described in the next section. The value of p_i^{ω} associated with the selected adjacent site is recorded for the subsequent evaluation of the weight W_{α} to be associated with cluster α. The selected adjacent site is removed from the table of adjacent sites.

(ii) A brick is selected from the subset of remaining bricks which have $\kappa\geqslant\kappa_m$ where κ_m is the minimum allowed κ value for that adjacent site. The brick is chosen with a probability p_i^{κ}. Once again the brick may be selected with equal probability from the subset of allowed bricks; however, a more careful choice of the probabilities p_i^{κ} is preferable, as explained below. The value of p_i^{κ} for the chosen brick is recorded to calculate W_{α}. In the following we assume that the selected brick has $\kappa=\kappa_s$. In practice it is possible to combine the choice of adjacent site and brick [steps (i) and (ii)] into a single Monte Carlo decision. This is the method used to obtain the results reported below.

(iii) The brick is added to the record of the current cluster and removed from the list of available bricks.

(iv) For each *old* adjacent site in the *current* table of adjacent sites, set

$$\kappa_m|_{\text{new}}=\max(\kappa_m|_{\text{old}},\kappa_s+1). \tag{10}$$

If $\kappa_m|_{\text{new}}$ is greater than the κ value of any of the bricks remaining to be placed, the site is removed from the list of adjacent sites since no bricks could subsequently be placed at that site. Identify *new* adjacent sites associated with the brick placed at step (ii). In order to qualify as a new adjacent site, the site must not appear in the current table of adjacent sites. Add the new adjacent sites to the list of adjacent sites and for each new adjacent site, set κ_m to the lowest κ value of the remaining bricks.

The adjustment of the adjacent sites and their associated κ_m is the key process in enforcing the unique cluster growth sequence described in Sec. III B 1. Thus after a site and brick have been selected at each stage of the cluster growth, the cluster is surrounded by a set of old adjacent sites which have *not* been selected. It is essential in the future growth of the cluster that none of these sites be filled with a brick with a lower κ value than the brick which has just been placed; otherwise this brick would be the one selected under the growth sequence described in Sec. III B 1 and not the one that had just been placed. The adjustment in Eq. (10) prevents such a choice. There is no restriction on the bricks which may be placed on *new* adjacent sites and hence the associated κ_m is set equal to the lowest κ value of the remaining bricks.

(v) Associate a weight $W_{\alpha}=1/(d_N\Pi_{i=2}^{N}p_i^{\omega_i}p_i^{\kappa_i})$ with the cluster, where the degeneracy $d_N=N!$. If it is impossible to complete the growth of the cluster, because all directions are blocked, the weight for the cluster is set to zero. The degeneracy factor d_N for the N cluster arises from (a) a degeneracy factor of N associated with the placing of the first brick (which is the number of different ways of placing the first brick within the cluster) and (b) a factor of $(N-1)!$ associated with the number of different ways of placing the remaining $(N-1)$ bricks which have $\kappa=\{2,\ldots,N\}$. The method may be refined by noting that ensemble averages can be constructed for *all* clusters up to and including size N by setting the associated degeneracy d_M for a cluster of size M to $d_M=M(N-1)!/(N-M)!$. The degeneracy arises from a factor M associated with placing the first brick and a factor $(N-1)!/(N-M)!$ associated with the number of different ways of placing the remaining $(M-1)$ bricks. Thus data can be collected for all cluster sizes up to size N simultaneously, associating the weight $W_{\alpha}=1/(d_M\Pi_{i=2}^{M}p_i^{\omega_i}p_i^{\kappa_i})$ with each cluster up to $M=N$.

Once an ensemble of clusters has been generated, weighted averages can be calculated.

3. An example cluster growth

The way in which the algorithm achieves the growth sequence described in Sec. III B 1 is perhaps best seen by following through a simple example as shown in Fig. 2 for a cluster of size 7 on a two-dimensional square lattice. Cluster sites are shown as large shaded squares and adjacent sites are shown as small unshaded squares; *new* adjacent sites are identified by a shaded triangle in the upper right hand corner of their square. The adjacent site chosen for the placement of the next brick is identified by the black circle.

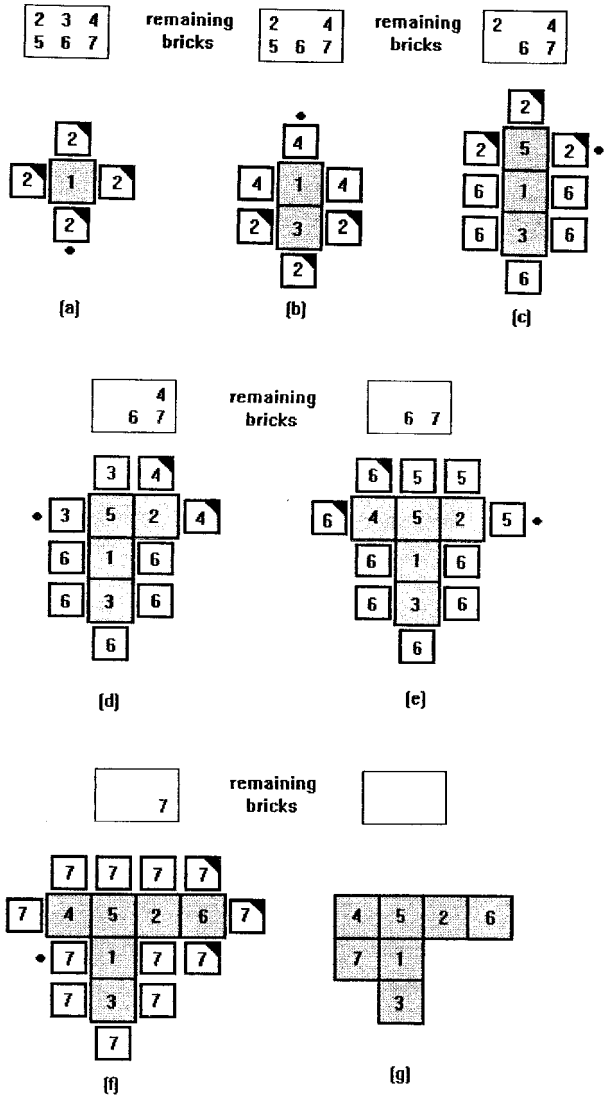

FIG. 2. Example of the growth of a cluster described in detail in Sec. III B 3. Each diagram shows the assigned bricks as large shaded squares, adjacent sites as small unshaded squares, and the *new* adjacent sites are identified by a shaded triangle in the upper right hand corner. The bricks are labeled with their κ values and the adjacent sites are labeled with their κ_m. The selected adjacent site at each step is identified by the black dot.

(i) Figure 2(a): The brick with $\kappa=1$ is placed on the lattice and the four *new* adjacent sites have their κ_m values set to 2, which is the κ value of the lowest available brick. One of the four adjacent sites is chosen by Monte Carlo sampling from the available sites for the placement of the next brick.

(ii) Figure 2(b): Brick number 3 is chosen by Monte Carlo sampling from the available bricks and placed in the selected adjacent site. The old adjacent sites have their κ_m values adjusted to be $3+1$. The new adjacent sites have their κ_m value set to 2, the κ value of the lowest available brick. An adjacent site is chosen by Monte Carlo sampling.

(iii) Figure 2(c): The chosen adjacent site has $\kappa_m=4$ and in this case only the bricks with $\kappa=4,5,6,7$ are available for selection; brick 5 is selected by Monte Carlo sampling. The old adjacent sites have their κ_m values set to $5+1$ and the new adjacent sites have their κ_m set to 2, the κ value of the smallest available brick. An adjacent site with $\kappa_m=2$ is selected by Monte Carlo sampling.

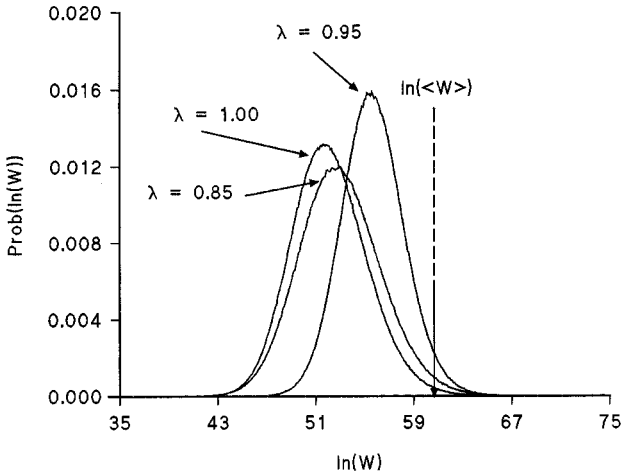

FIG. 3. Probability distribution $P(\ln(W))$ of the weights W determined from the growth of 1.6×10^6 clusters of size $N=32$ for different values of the parameter λ. The natural logarithm of the expectation value of the weights is also shown.

(iv) Figure 2(d): A brick may be selected from any of the remaining bricks; the brick with $\kappa=2$ is selected. If the old adjacent sites have κ_m values less than $2+1$, their κ_m values are adjusted to be $2+1$, otherwise their κ_m values are left unchanged. New adjacent sites have their κ_m set to 4, the lowest κ value of the remaining bricks. Note that κ_m values of 3 and 4 are in fact equivalent since the lowest available κ value is 4. An adjacent site with $\kappa_m=3$ is selected.

(v) Figures 2(e)–2(g): The algorithm is repeated until the cluster is complete.

The algorithm can only construct the given labeling of the cluster in the sequence shown. Similarly, any other labeling of this particular cluster shape can only be constructed by the algorithm in one way. Hence, since there are $N!$ ways of labeling the cluster, we have ensured that an overall degeneracy of $N!$ is associated with every unique cluster shape.

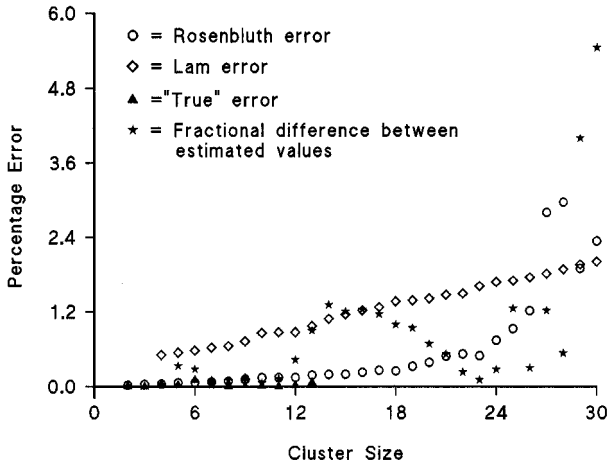

FIG. 4. Comparison of errors for the degenerate Rosenbluth method and the incomplete enumeration scheme of Lam [4]. The ⋆ symbol shows the absolute value of the fractional difference between the estimates generated by the two methods.

TABLE I. Degenerate Rosenbluth estimate of the number of lattice animals of size N on a three-dimensional square lattice using 1.8×10^7 sample clusters, each grown to $N=32$ with $\lambda=0.90$; exact values from [4]; estimated values and associated errors from the incomplete enumeration method of Lam [4]; calculation of error estimate described in text; ''true'' error is the fractional difference between Rosenbluth estimate and exact value; χ and ξ are defined in the text.

N	Rosenbluth estimate	Exact value	Lam [4] estimate	e^{est} (% error)	True (% error)	Lam [4] (% error)	χ	ξ
2	3.000×10^{0}	3		0.02	0.01		0.42	−0.02
3	1.500×10^{1}	15		0.04	0.00		0.00	0.06
4	8.597×10^{1}	86	8.594×10^{1}	0.04	0.04	0.51	0.86	−0.14
5	5.339×10^{2}	534	5.321×10^{2}	0.06	0.02	0.54	0.39	0.55
6	3.485×10^{3}	3 481	3.475×10^{3}	0.06	0.11	0.58	1.64	0.40
7	2.352×10^{4}	23 502	2.353×10^{4}	0.08	0.09	0.63	1.16	−0.46
8	1.629×10^{5}	162 913	1.631×10^{5}	0.09	0.01	0.65	0.12	−0.11
9	1.154×10^{6}	1 152 870	1.155×10^{6}	0.12	0.13	0.73	1.11	0.44
10	8.296×10^{6}	8 294 738	8.291×10^{6}	0.14	0.01	0.86	0.10	0.24
11	6.050×10^{7}	60 494 540	6.042×10^{7}	0.15	0.01	0.87	0.04	0.35
12	4.461×10^{8}	446 205 905	4.442×10^{8}	0.15	0.02	0.87	0.15	0.17
13	3.321×10^{9}	3 322 769 129	3.291×10^{9}	0.18	0.07	0.97	0.35	0.32
14	2.493×10^{10}		2.461×10^{10}	0.20		1.09		0.60
15	1.884×10^{11}		1.862×10^{11}	0.20		1.16		0.13
16	1.434×10^{12}		1.416×10^{12}	0.23		1.22		0.22
17	1.095×10^{13}		1.082×10^{13}	0.26		1.27		0.82
18	8.412×10^{13}		8.329×10^{13}	0.25		1.37		0.02
19	6.507×10^{14}		6.446×10^{14}	0.33		1.38		0.72
20	5.036×10^{15}		5.002×10^{15}	0.39		1.41		0.16
21	3.917×10^{16}		3.897×10^{16}	0.49		1.47		1.12
22	3.059×10^{17}		3.052×10^{17}	0.52		1.49		0.90
23	2.388×10^{18}		2.391×10^{18}	0.49		1.61		0.10
24	1.872×10^{19}		1.877×10^{19}	0.74		1.68		0.89
25	1.461×10^{20}		1.480×10^{20}	0.93		1.70		0.92
26	1.165×10^{21}		1.168×10^{21}	1.22		1.75		0.93
27	9.321×10^{21}		9.209×10^{21}	2.80		1.81		4.33
28	7.251×10^{22}		7.290×10^{22}	2.97		1.88		4.72
29	5.555×10^{23}		5.786×10^{23}	1.90		1.96		0.66
30	4.359×10^{24}		4.610×10^{24}	2.34		2.01		1.17

4. Choice of sampling probabilities

In principle, the choice of the *normalized* probabilities p_i^{ω} and p_i^{κ} can be of any form consistent with ergodicity. However, it is found that the probability distribution $P(W)$ of weights associated with clusters of a given size is approximately log normal as might be expected since it arises from the multiplication of a large number of random variables. The log-normal distribution is highly skewed and the skewness becomes increasingly marked as the cluster size increases. Unless a careful choice is made of p_i^{ω} and p_i^{κ}, large errors arise in the sampling of the mean and this makes the calculation of results for large clusters computationally prohibitive. The aim of any scheme must therefore be to minimize the variance of the probabilities used in constructing the weights.

If every brick is given equal weight there is a reasonable chance that early bricks attached to the cluster will have high κ values and this leads to a significant reduction in the number of available sites for growth and consequently $P(W)$ becomes dominated by low weights. A number of schemes were explored for attaching higher probabilities to the bricks with lower κ values. It was found empirically that the skewness of the distribution could be significantly reduced if the p_i^{κ} was chosen such that $p_i^{\kappa}\propto\lambda^{\kappa}$ where $1\geqslant\lambda\geqslant0$. It is important to note that λ should not be made too small otherwise the distribution once again becomes highly skew because the bricks with high κ occur with only very small probabilities but correspondingly very large weights. The probability p_i^{ω} of an adjacent site being selected was made proportional to the sum of the weights, λ^{κ}, for the bricks available at that site. In the collection of the data presented below, the parameter λ was chosen empirically, for given N, to minimize the skewness of $P(W)$ and the values of λ associated with each brick were not changed as the cluster grew. The effect of changing λ is illustrated in Fig. 3.

5. Evaluation of the algorithm

Table I shows the results of using the method to enumerate lattice animals up to size 32 on a simple cubic three-dimensional lattice by sampling a total of 1.8×10^7 clusters

TABLE II. Degenerate Rosenbluth estimate of the number of lattice animals of size N on a two-dimensional square lattice using 3.5×10^7 sample clusters, each grown to $N=32$ with $\lambda=0.92$; exact results from [6]; calculation of error estimate described in the text; ''true'' error is the fractional difference between Rosenbluth estimate and the true value; χ and ξ are defined in the text.

N	Rosenbluth estimate	Exact value	e^{est} (% error)	True (% error)	χ	ξ
2	2.000×10^0	2	0.01	0.01	0.83	0.01
3	6.000×10^0	6	0.02	0.01	0.29	−0.51
4	1.900×10^1	19	0.02	0.01	0.51	−0.14
5	6.301×10^1	63	0.02	0.02	0.75	0.09
6	2.160×10^2	216	0.02	0.02	0.75	−0.04
7	7.602×10^2	760	0.03	0.03	1.06	−0.60
8	2.725×10^3	2 725	0.03	0.01	0.44	0.19
9	9.915×10^3	9 910	0.04	0.05	1.19	−0.30
10	3.645×10^4	36 446	0.04	0.01	0.32	−0.33
11	1.353×10^5	135 268	0.05	0.05	0.90	−0.30
12	5.055×10^5	505 861	0.05	0.07	1.40	−0.37
13	1.903×10^6	1 903 890	0.06	0.05	0.94	0.12
14	7.204×10^6	7 204 874	0.06	0.01	0.23	0.39
15	2.740×10^7	27 394 666	0.08	0.02	0.30	0.91
16	1.046×10^8	104 592 937	0.07	0.01	0.09	0.09
17	4.004×10^8	400 795 844	0.10	0.10	0.98	0.30
18	1.536×10^9	1 540 820 542	0.11	0.31	2.94	0.18
19	5.923×10^9	5 940 738 676	0.13	0.30	2.24	−0.42
20	2.293×10^{10}		0.20			0.25
21	8.899×10^{10}		0.29			0.36
22	3.461×10^{11}		0.38			0.94
23	1.343×10^{12}		0.40			0.99
24	5.208×10^{12}		0.44			0.59
25	2.048×10^{13}		0.75			1.25
26	7.990×10^{13}		0.85			1.94
27	3.131×10^{14}		0.98			1.31
28	1.229×10^{15}		2.02			4.36
29	4.767×10^{15}		1.80			1.73
30	1.809×10^{16}		3.27			5.06

with $\lambda=0.90$. Results are not quoted for $N=31$ and $N=32$ since the errors were unacceptably large. It took approximately two hours to collect the data on an R5000 Silicon Graphics workstation using code written in the language C. The results are quoted together with a standard error e^{est} calculated by breaking the data into 50 blocks and determining the variance of the block means for each cluster size. If the number of samples in each block is sufficient, it follows from the central limit theorem that the sampling distribution of the means should become reasonably symmetrical. We therefore also quote a *skewness* ξ defined by [13]

$$\xi=m_3/m_2^{3/2}, \tag{11}$$

where m_i is the ith moment about the mean of the sampling distribution. It is expected that $\xi\lesssim0.5$ for a symmetrical distribution and $\xi>1$ for a highly skew distribution. The statistic ξ should be treated with some caution since it is likely to be subject to considerable error because it involves the calculation of a third moment from a limited number of data points.

Exact results are known for clusters up to size 13 [4] and in the table we quote the values for the quantity χ defined by

$$\chi_M=\left|\frac{a_M^{\text{exact}}-a_M^{\text{est}}}{a_M^{\text{exact}}e_M^{\text{est}}}\right| \tag{12}$$

and it can be seen that all the values of χ are $O(1)$. Thus e^{est} represents an acceptable method of estimating the error in the simulation for the smaller clusters. However, it is likely that the e^{est} will underestimate the true error if the distribution becomes more skewed. We also quote in Table I the values of a_N calculated by Lam [4] using a Monte Carlo incomplete enumeration method together with his error estimates for this method. It can be seen from Fig. 4 that the degenerate Rosenbluth method described in this paper gives significantly smaller errors, except for the very largest clusters. It should be noted that the Lam data were calculated with 2 h of CPU time on a Cyber 76 computer.

In Table II we quote data collected from a square two-dimensional lattice by collecting data from 3.5×10^7 clusters up to size 32 with $\lambda=0.92$. These data took the same time to

collect as the data for the three-dimensional lattice and comparison is given with exact results [6] up to clusters of size 19. It is of interest to note that the errors associated with the degenerate Rosenbluth method appear to grow with cluster size at essentially the same rate for both the two- and three-dimensional lattices. This suggests that the technique may work well in higher dimensions.

IV. CONCLUSIONS

We have described a degenerate Rosenbluth scheme for the enumeration of lattice animals and demonstrated its viability for clusters up to size 30 on two- and three-dimensional simple-cubic lattices. The results suggest that the method is computationally more effective than the incomplete enumeration method of Lam except for the largest clusters where the former method may be superior because the errors appear more controlled. Thus it appears that the errors associated with the incomplete enumeration method rise linearly whereas those associated with the Rosenbluth method increase in a nonlinear fashion. It is possible that an alternative scheme for the choice of p_i^{ω} and p_i^{ω} can be found which will reduce the skewness of the probability distribution $P(W)$ of the Rosenbluth weights and hence increases the size of N for which the method can be employed. A possible method may include multilink additions (e.g., [14]) in which several bricks are added simultaneously.

However, the degenerate Rosenbluth method can be readily extended to include Boltzmann weights following, for example, the techniques used in configurational bias [12]. It is likely that the errors associated with compact clusters in the presence of Boltzmann weight may extend the range of usable N because of the more limited range of clusters which is being sampled. The method may also be adapted to count clusters of chains and such an algorithm is currently being developed by the author.

[1] G. Jacucci, A. Perini, and G. Martin, J. Phys. A **16**, 369 (1983).
[2] B. F. Edwards, M. F. Gyure, and M. Ferer, Phys. Rev. A **46**, 6252 (1992).
[3] R. C. Ball and J. R. Lee, J. Phys. (France) I **6**, 357 (1996).
[4] P. M. Lam, Phys. Rev. A **34**, 2339 (1986).
[5] H. P. Peters, D. Stauffer, H. P. Hölters, and K. Loewenich, Z. Phys. B **34**, 399 (1979).
[6] M. F. Sykes and M. Glen, J. Phys. A **9**, 87 (1976).
[7] P. M. Lam and F. Family, Physica A **231**, 369 (1996).
[8] D. Stauffer, Phys. Rev. Lett. **41**, 1333 (1978).
[9] P. L. Leath, Phys. Rev. Lett. **36**, 921 (1976).
[10] S. Redner and P. J. Reynolds, J. Phys. A **14**, 2679 (1981).
[11] M. N. Rosenbluth and A. W. Rosenbluth, J. Chem. Phys. **23**, 356 (1955)
[12] J. I. Siepmann and D. Frenkel, Mol. Phys. **75**, 59 (1992).
[13] M. G. Bulmer, *Principles of Statistics* (Oliver and Boyd, London, 1965).
[14] A. D. Mackie, E. M. O'Toole, and D. A. Hammer, Fluid Phase Equilibria **82**, 251 (1993).