PHYSICAL REVIEW B VOLUME 27, NUMBER 7 1 APRIL 1983

Monte Carlo study of weighted percolation clusters relevant to the Potts models*

Mark Sweeny
The Enrico Fermi Institute and the Department of Physics, The University of Chicago, Chicago, Illinois 60637
(Received 30 August 1982)

The two-dimensional Potts models were simulated with the use of a Monte Carlo method. This study is unique in that we simulated the weighted percolation clusters first described by Kasteleyn and Fortuin in 1969. We describe an auxiliary data structure which enables us to determine the connectedness of large clusters very efficiently. Our simulation of the percolation problem was found not to be affected by a critical slowing down. Critical exponents were found by studying the size and shape of large clusters. Our results agree with previous studies done on integer values of q, and with the conjecture of den Nijs.

I. INTRODUCTION

The Potts model[1] is defined as a lattice of spins that can take on q different values $\sigma_i = 1,2,\ldots,q$ and interacting by the Hamiltonian

$$H = -K \sum_{\langle i,j \rangle} \delta_{\sigma_i \sigma_j} . \tag{1.1}$$

δ is the Kronecker delta and $\langle i,j \rangle$ denotes summation over nearest neighbors. The constant K includes a factor of $1/k_B T$. The two-dimensional Potts model undergoes a phase transition which is still not completely understood.

More than a decade ago, Fortuin and Kasteleyn[2] showed that the Potts model is equivalent to a weighted percolation problem. Their construction allows the Potts model to be generalized to nonintegral values of q. Temperley and Lieb[3] used the result of Fortuin and Kasteleyn to prove the equivalence of the Potts model to the six-vertex or square-ice model, with staggered polarizations. Baxter, Kelland, and Wu[4] (BKW) have since found a very elegant derivation of the result of Temperley and Lieb. They use a construction hereafter known as the BKW construction which makes many exact results obvious, including the critical temperature, self-duality, and energy at criticality of the Potts model.

Baxter used the equivalence to the six-vertex model to show that the Potts model has a first-order phase transition for $q > 4$, and he also computed the latent heat.[5]

The weighted percolation problem is a two-dimensional square-lattice-bond percolation problem with weight functions

$$W = q^r \lambda^b , \tag{1.2}$$

where r is the number of connected regions, λ is e^{K-1}, and b is the number of bonds connecting nearest-neighbor pairs.

This is a difficult problem to simulate, because, with each proposed change, connectedness must be determined. If large regions must be traced out, then huge amounts of central processing unit (CPU) time will be consumed. We have implemented an algorithm which determines connectedness quickly using an auxiliary data structure. This data structure turns out to be based upon the BKW construction. Although the data structure appears at first to be relatively complicated, the Amdahl 470 V/7 at the University of Chicago can effect a change in less than 50 μs when simulating a heat bath. This algorithm would be useful in the study of "lattice animals."[6,7]

We measured the exponents Y_H and Y_T, where Y_H is the magnetic exponent and Y_T is the temperature exponent.

II. THE METROPOLIS METHOD

The METROPOLIS method[8] is a simulation of the manner in which a solid-state experimentalist would study a real system. If he wanted to study a ferromagnetic at temperature T, then he would put a sample in an oven held at temperature T. While in the oven, the ferromagnet would exchange energy with the surroundings. As the surroundings are presumed to follow a Boltzmann distribution, the relative probability of our system gaining a quantity E of energy in a single interaction is proportional to $\exp(-E/k_B T)$.

In the case of graph configurations, the single interactions are taken to be the changing of a single link. Our simulation proceeds as follows.

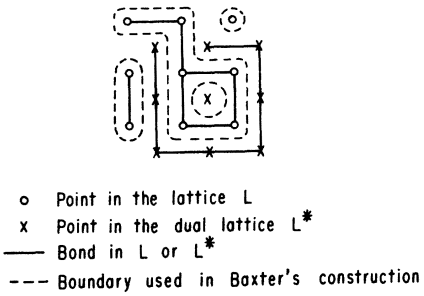

FIG. 1. Baxter's construction and the Euler equation. In this instance of a 3×3 lattice: $l=4$, $b=7$, $r=3$, $N=9$. The Euler equation, $b+2r=N+l$, holds.

Step 1: We pick a link and propose changing it.

Step 2: Find out by what factor δW, the weight function, changes if we change the link variable. Looking at Eq. (1.2) we see that $\delta W=q^{\delta r}\lambda^{\delta b}$ with δb the change in the total number of bonds. If the link originally has a bond, then $b=-1$. Otherwise $b=+1$. Similarly, δr is the change in the number of connected regions.

Step 3: Generate a pseudorandom number X uniformly distributed from 0 to 1. Change the link if $X<(\delta W)/(\delta W+1)$.

Step 4: Return to step 1 unless the experimenter has decided that the system is sufficiently equilibriated.

Step 2 is nontrivial, as connectedness, a nonlocal property, must be determined with every iteration. The straightforward way to determine the connectedness of two points is to start at the first point and enumerate all points connected to it until either the second point is reached, or an entire connected region has been enumerated. This is adequate if the clusters are small (tens of points), but if large clusters are involved then the CPU time requirements can grow unreasonable. A method to quickly determine the connectedness of large two-dimensional clusters is described next.

III. THE FAST ALGORITHM

The fast algorithm is based upon the BKW construction. They consider a lattice L, with N points, together with its dual lattice L^*. If a bond is present on L, then its dual bond L^* is absent, and vice versa. The boundaries between graphs on L and their dual graphs on L^* will form a collection of closed loops, as shown in Fig. 1. One then has the Euler relation,

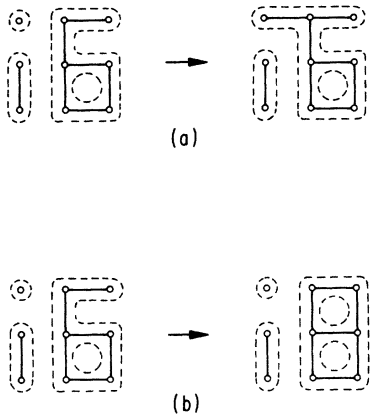

FIG. 2. (a) Addition of a bond to a 3×3 lattice with the destruction of a connected region and a loop is demonstrated. (b) Addition of a bond with no change in the number of connected regions, and the creation of a loop.

$$b+2r=N+l\,, \tag{3.1}$$

which relates the number of regions and bonds to the number of loops l. Baxter, Kelland, and Wu use the Euler relation as part of their demonstration of the equivalence of the Potts model and the six-vertex model. By representing the loops in a computer, one transforms the problem of determing connectedness to the problem of determining if two loop segments are part of the same loop. Figures 2(a) and 2(b) illustrate the addition of a bond with and without the destruction of a region.

The boundaries are represented in the computer as a chain of pointers. A pointer is a memory location containing the address of an object, in this case the address of the next pointer in the chain. The concept of "express pointers" is used to traverse out loops more quickly than one step at a time.

Figure 3 shows a graph on a lattice with pointers, represented by arrows. Pointer A points to pointer B, B points to C, etc. In this manner the loops are represented by chains of pointers. Because of the Euler relation, [Eq. (3.1)], we can determine δr if we can determine δl, the change in the number of loops. Link L_1 illustrates the case in which removing the bond will divide a loop into two loops, while line L_2 illustrates the case in which two loops will be joined if the bond is removed. In every case, two cuts must be made in our chains and the four ends rejoined if we are to change a link. In the case of L_1, the $A\rightarrow B$ and the $E\rightarrow F$ connections must be severed and we need to set $A\rightarrow F$ and $E\rightarrow B$. Finally, we note that in order to determine the case of L_1, it suf-

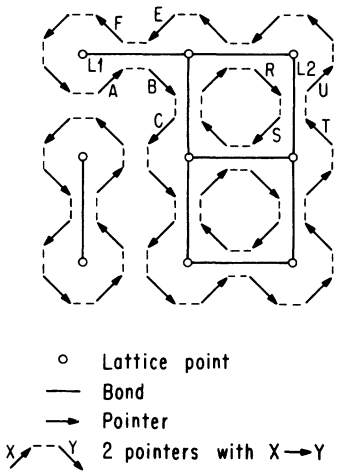

FIG. 3. 3×3 lattice with chains of pointers, as discussed in the text. $A\rightarrow B$, $E\rightarrow F$, but when bond L_1 is removed then $A\rightarrow F$, and $E\rightarrow B$ and one loop is split into two. Removal of L_2 changes $R\rightarrow S$ and $T\rightarrow U$ to $R\rightarrow U$ and $T\rightarrow S$. In this case two loops are joined into one.

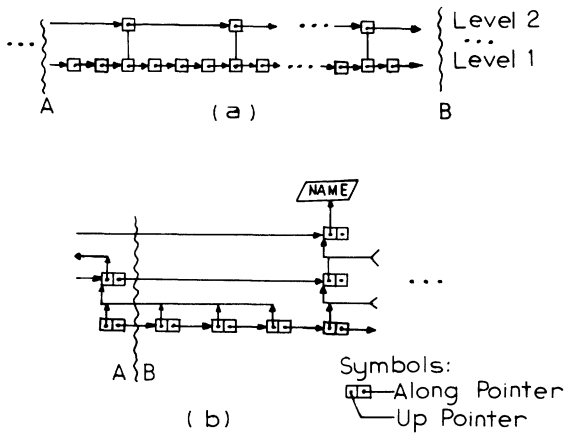

FIG. 4. (a) Illustration of an express pointer. (b) Data structure used by our program. The up pointers allow one to trace from level 1 to Name. The along pointers allow one to follow the loop and are used in updating the up pointers.

fices to have determined that A and E are in the same chain. In summary, pointers in chains can be used to represent the boundaries used by the BKW construction, and such data structure can be used to find δW needed for our Monte Carlo simulation provided we can do two things: (1) determine if two given pointers belong to the same chain, and (2) be able to cut and then rejoin our chains of pointers. The concept of express pointers enables us to traverse our pointer chains very quickly. After describing the general nature of express pointers, we will present the actual data structure used to simulate the weighted percolation problem.

Look at Fig. 4(a) and consider the problem of traversing the chain of pointers from A to B. If the chain has N links, then the time required to traverse the chain at level 1 grows linearly with N. If every fourth pointer is associated with an "express" pointer, analogous to the express stations in a transit system, then the time to traverse at level 2 grows only as $N/4$. We can traverse level 2 four times as fast as level 1. Of course we must first traverse along at level 1 until we reach an express station, but this is usually only a few stops. Similarly, we can add a third level and traverse long chains 16 times as fast. Ultimately, if enough levels are added, then the time to traverse a chain becomes dominated by time spent at lower levels trying to reach the next express stop. The time to traverse a chain then grows only as $\log(N)$.

Figure 4(b) illustrates the actual data structure used in our implementation. Pointers are assembled into pairs, one part of each pair is called the along pointer and is analogous to the pointers in Fig. 2(a) discussed previously. The other part of each pair is called the up pointer. This pointer points to the previous express station at the next highest level. The top-level up pointers contain a unique name which identifies each different loop.

The data structure of Fig. 4(b) allows us to easily find out if any two given level-1 pointers belong to the same loop. One simply traces the up pointers until the name is reached, then compares names.

When loops are cut and rejoined as described earlier, then the names are updated differently depending on whether two loops are being joined, or whether a loop is being split, but otherwise the problem is simply that we must join cut-end A, whose along pointers contain wrong values, to cut-end B, whose up pointers contain the wrong values. Figure 5(a) illustrates this. There is an $\times$ in each pointer which has the wrong value. In order to update the data structure in this case, one must do the following. First, load the along pointers on end A with the proper values. Figure 5(b) shows our data structure after this has been done. Next, the up pointers on end B. This starts at B and continues until an express station is encountered on the B end. Usually only a few up pointers need to be changed. Lastly, if two loops are being joined, then the names on end B must be made to be the same as the names on end A. If a loop is being split, then a new name must be made up and this name applied to one of the two new loops. When two loops are to be joined and each has a different number of levels, then the situation is slightly more complex but can still be han-

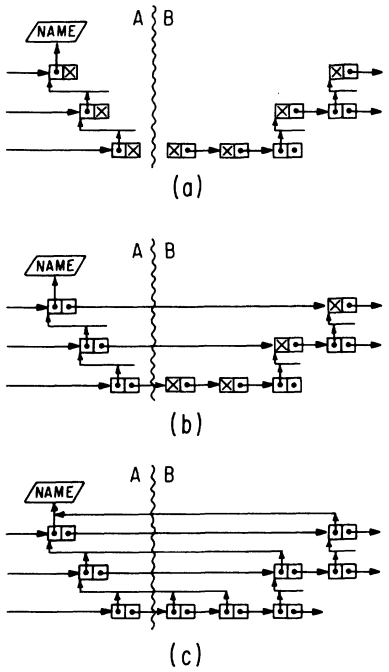

FIG. 5. Illustration of the joining of two chains. (a) End A is to be joined to end B, nothing has yet been done, so the along pointers of end A and the up pointers of end B contain improper values. (b) The along pointers of end A have been filled. (c) The fully updated chain.

dled in a straightforward manner.

Figure 6 shows the relation of the stations to the lattice L. There are four level-1 pointers for each lattice point. Each level-1 pointers for each lattice point. Each level-1 pointer lies between a lattice point and a dual-lattice point. In this way, every boundary between a lattice cluster, and the dual-lattice points, such as the loops of Fig. 1, is naturally associated with a closed chain of pointers at level 1. One in four level-1 pointers is associated with a level-2 pointer. In this case, only the highest level pointer is explicitly shown in Fig. 6. Similarly, every fourth level-2 pointer is associated with a level-3 pointer, etc. In our simulations we used a 256×256 lattice with 9 levels of pointers. In order to save memory, however, the level-1 pointers were not stored and a subroutine traced along the graph until a level-2 pointer was encountered.

This entire procedure was programmed in assembly language and run on the Amdahl 470 V/7 at the University of Chicago. When the weighted percolation problem is simulated, it takes less than 50 μs to compare the names of the two loops and, if needed, make two cuts and rejoin the four ends. All of our work, other than debugging, was done using a 256×256 lattice.

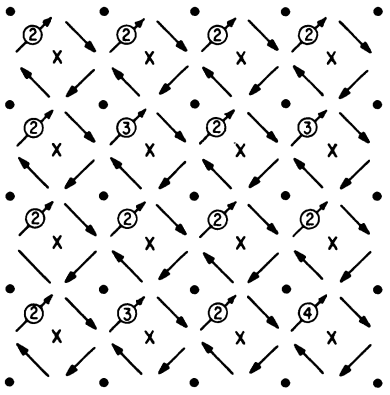

FIG. 6. Location of pointers on the lattice L.

IV. A COMPARISON OF METHODOLOGIES

The Potts model has been studied extensively by numerical simulation. Monte Carlo renormalization-group (MCRG) calculations have yielded accurate values for critical exponents. These simulations can only work when q is an integer. They are also plagued by a critical slowing down when close to criticality.[9] Although updating a conventional Monte Carlo simulation or a MCRG simulation is simple and fast, the critical slowing down makes the process more CPU intensive than the percolation method whenever the correlation length exceeds ~ 10 lattice lengths. At the time this work was being performed, Nightingale and Blöte[10] studied the Potts model with noninteger q by calculating the transfer matrix for lattices of various widths. They obtained results that are essentially exact for the $n\times\infty$ lattice, but the work involved grows in a nonpolynomial fashion with increasing n, so that calculations with $n > 10$ become prohibitively expensive. Nonetheless, extrapolation in n (by Pade approximates and other methods) has yielded results for Y_H and Y_T which appear to be accurate to 0.1% and agree with the conjectures of den Nijs, of Pearson, and of Nienhuis *et al.*[11,12] This study used the loop data structures based upon the BKW construction. An alternative would have been to use a straightforward tracing procedure. Tracing out even a small cluster is generally no faster than following its boundary, so the loop data structure im-

poses no overhead in CPU time. In practice, however, the overhead in memory usage is large, and the added complication of the data structure makes programming and debugging more difficult. The CPU savings are greatest for configurations close to criticality.

The mean cluster size (weighted by size) was found to be roughly 5000 spins. Thus to check connectedness by tracing requires thousands of iterations. The loop method requires roughly $(4+1)\times\log(5000)=30$ iterations to perform an update. Therefore, it is up to 2 orders of magnitude faster. In practice, a tracing alogrithm which is optimized is faster than this would indicate but the loop data structure still gains an order of magnitude in speed. As with the conventional simulations, there is no longer an advantage of the loop data structure once the correlation length is <10 lattice lengths.

V. ADVANTAGES OF METHOD

A. No critical slowing down

When $q<4$ the graph configuration equilibrates very quickly. A 256 lattice has no observables with a correlation time greater than one pass through the lattice. Thus, after 10 to 20 passes, the system has lost all memory of the starting configuration and can be said to be completely equilibrated.

In contrast, when the Potts model is simulated conventionally changing the state of one spin at a time, then the slowest moving observables have a correlation time which increases as a power of the lattice size. For $q=2$, this power has been determined to be approximately 2,[9] Swendson *et al.*[13] have to equilibrate their system with tens of thousands of passes through the lattice before they start to collect data. The block spins used in his MCRG studies are relatively slow moving, so that he must make 16 passes through the lattice between measurements. Sixteen steps is still not enough to remove a relatively strong time correlation between samples for the largest block spins. They have typically used a 96 by 128 lattice. Thus, the simulation of weighted clusters could even be useful for generating configurations for a Monte Carlo study of the Potts model, if one wants to perform many decimations.

B. Clusters are interesting

The weighted percolation clusters are interesting in themselves. They are not accessible by other methods.

C. q need not be an integer

From the point of view of simulating the graph configurations, q is simply a real number. Obviously, the Potts model does not even make sense, unless q is an integer. At this time there are two reasons to be especially interested in noninteger values of q.

At this time, there are conjectures for both the heat exponent Y_T,[11] and the magnetic exponent Y_H.[12] The first is due to den Nijs; the second has been formulated independently by both Nienhuis *et al.* and by Pearson. These conjectures are in agreement with what is known about the Potts model with integer q. The work of Nightingale, Blöte, and Derrida[10] is the only other work to study noninteger q. Their work agrees with the conjectures.

The four-state Potts model is of great interest to theorists largely due to the logarithmic corrections to scaling which occur largely due to the logarithmic corrections to scaling which occur there.[14] The four-state Potts model is not well understood, and the study of models with q near to four could give insight into the behavior of these systems.

D. Clusters are more useful

A single conventional Potts-model configuration can be obtained from a graph configuration by choosing a random number from 1 through q for each cluster and assigning each spin in the cluster the random number as its value. Thus, for one graph configuration, one can obtain q^r different Potts-model configurations. For a 256×256 lattice r, the number of clusters is generally several thousand. Thus one graph corresponds to a very large number of Potts configurations. Anything which is measured directly from the graph represents an average over all of the spin configurations which could be obtained from the graph.

Consider the case of measuring the two-point correlation function. If one looks at spin configurations, then one counts how often two spins a given distance apart take on the same value. In the case of random spin configurations, two spins will take on the same value one time in q. Thus counts obtained once for every q measurement do not represent an effect of the physics, but are just a background which is theoretically uninteresting. From the point of view of measuring the two-point correlation function, these "accidental" counts are a background of random noise which must be subtracted out. In the case in which the real correlations are smaller than $\frac{1}{2}$, then the measurement will be very difficult. In contrast, if one looks at graph configurations, then one simply counts how often two spins a given dis-

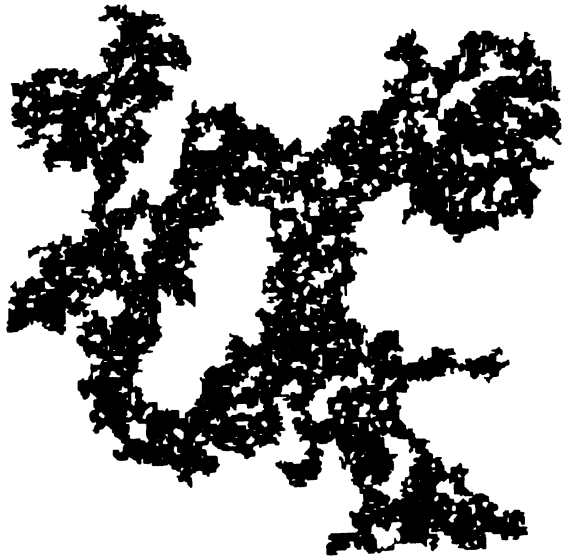

FIG. 7. This is a large cluster corresponding to $q=3$ and $K=K_c$.

tance apart are in the same cluster. The case of a random spin configuration corresponds to the graph with each point in its own cluster; there is no background to be subtracted out. Furthermore, by studying one cluster at a time, one can spend computer time counting spin pairs which are correlated, rather than grinding through the many spin pairs which are in different clusters.

VI. RESULTS OF THE SIMULATION

When the simulator was programmed in assembly language and run on the Amdahl 470 V/7 at the University of Chicago, the time needed to examine and change, if needed, one link was found to be less than 50 μs. This is far longer than an Ising simulator, but it is fast enough to make the simulation of large lattices practical. When the lack of critical slowing down is taken into account, then it compares well with the more conventional simulation of spin systems.

For $q<4$, we obtained large clusters after only 10–20 Monte Carlo steps per link even when the starting configuration contained no links. Figure 7 is a cluster obtained in the case of $q=3$. Thus long-range correlations build up very quickly, in contrast to the dynamics of a conventional simulation. Also, the relaxation time for the total number of bonds was found to be only about one Monte Carlo step per link.

This simulation produces percolation configurations which are expected to exhibit scaling, as does simple percolation.[7] At criticality, $n(s)$, the number of clusters of size s is expected to be a power of s. Figure 8 is a log-log plot of $n(s)$ vs s for the Ising model ($q=2$). The very linear behavior over 4 orders of magnitude indicates that we have fully equilibrated our configuration and that scaling is occurring.

With $q=4$, the relaxation time for the total number of links was found to be very long, at least hundreds of steps per link. For $q>4$, the transition is first order, so one would expect to find that a graph that began with no bonds would evolve into one with a number of bonds corresponding to the upper energy state of the system. When we tried $q=5$, however, we found that the total number of bonds changed very slowly and in 100 steps per link, never reached the neighborhood of the high-energy state.

From the study of our clusters, we were able to measure the exponents Y_H and Y_T. The details of the methods will be discussed later. The accuracy was just over 0.3% in the case of Y_H, and just over 3% in the case of Y_T. In comparison, the MCRG study[13] of Swendson *et al.* obtained 1% accuracy for Y_H and 2% for Y_T. Our measurements of Y_H were based on only 30 configurations for each q value and those of Y_T on only a total of roughly 70 configurations for each q value. We can generate and study roughly five configurations per minute using the Amdahl 470 V/7. To determine a set of exponents, Swendson *et al.*, with the use of MCRG, typically used 50 000 passes through a lattice that was about $\frac{1}{5}$ the size of ours and an algorithm that is unlikely to be more than ten times faster.

VII. FRACTAL DIMENSION— THE MEASUREMENT OF Y_H

Fractal is a word coined by Mandelbrot[15] to denote objects of fractional dimension. He uses the Hausdorff-Besicovitch definition of dimension. This definition tells us to cover the object with spheres of radius r. If at least V_r spheres are needed

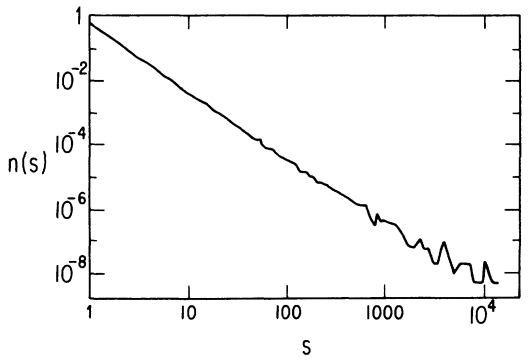

FIG. 8. Graph of $n(s)$ vs s for the Ising model.

TABLE I. Measured values of Y_H.

q	Method 1	Method 2	Method 3	Theory	Other	Other method
0.5	1.918	1.868	1.894	1.924		
1.0	1.900	1.827	1.878	1.898	1.895	$Y_H=2/(1+1/\delta)$, $\delta=18$ from Table II in Ref. 7
1.5	1.880	1.820	1.869	1.892		
2.0	1.882	1.800	1.857	1.875	1.875	Exact (Onsager solution)
2.5	1.875	1.782	1.844	1.877		
3.0	1.877	1.808	1.853	1.870	1.8666	Exact?[a] Hard hexagon model (Ref. 18)
					1.870±0.1%	MCRG (Swendson, Ref. 13)
3.5	1.870	1.750	1.840	1.868	1	
4.0	1.90				1.83	MCGR (Ref. 13)
					1.875	exact[a] (Ref. 19)

[a]These models are believed to be in the same universality class. Theory is conjecture of Nienhuis *et al.* and Pearson: $\beta=(2-Y_H)/(Y_T)$, $\beta=\frac{1}{12}(1+2u/\pi)$, $\cos(u)=\frac{1}{2}\sqrt{q}$. I used my measured value of Y_T from Table II in computing Y_H from these formulas. Methods 2 and 3 were not used for $q=4$ as they were used with open boundary conditions and there were no large 2-4 clusters with open boundary conditions.

and if $V_r \propto r^{-d}$, then d is said to be the dimension of the object. We expect that $d=Y_H$. To implement this definition exactly is difficult, so an effective fractal dimension was measured. Three different definitions of effective fractal dimension were tried. The first is very close to the Hausdorff-Besicovitch definition, the other two are an adaptation of the Kadanoff-Migdal recursion relations used in the study of scaling of spin systems.[16] Only the first method gave values of Y_H in agreement with other work. In his review article Stauffer[7] gives nine different ways to measure fractal dimensions of percolation clusters, stating that different methods generally give different results. The seemingly simple concept of fractal dimension involves subtleties which are not yet fully understood. I will return to this after describing the methods and results.

A. Method 1

Method 1 is based on the assumption that if the number of spheres of radius r needed to cover a cluster behaves as r^{-d}, then the number of points within a sphere of radius r behaves as r^d. The detailed shape of the "sphere" should not matter, so I used squares of side r. If one were trying to cover a cluster with a minimum covering, then one would not place spheres so that only a few spins near the circumference were included in the sphere. There is reason to suspect that the failure of methods 2 and 3 is due to a boundary effect. With this in mind, I formulated method 1 to use only squares whose center is also an element of the cluster being studied (to do this, r, the side of my squares, was always odd).

Method 1 is very simple. One picks at random a single point of the large cluster under study and draws a square of side r about the point. Next, one counts the number of points belonging to the cluster that are also within the square. Lastly, one picks more points and does more counting until an accurate average, denoted D_r, is obtained for the number of points belonging to the cluster which fall within a square of side r and centered about an element of the cluster. If $D_r \sim r^{+d}$, then we say that the cluster has dimension d according to method 1. A quick look at Table I, where the results of this method are listed together with the values of Y_H known through other methods, will show that this method works well.

Figure 9 is a log-log plot of D_r/r^2 vs r for $q=2$. Twelve different r values ranging from 3 through 155 and forming an approximately geometric pro-

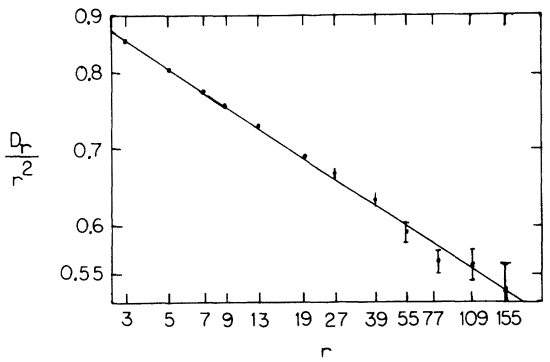

FIG. 9. log-log plot of D_r/r^2 (relative "area") vs r from method 1 of measuring fractal dimension.

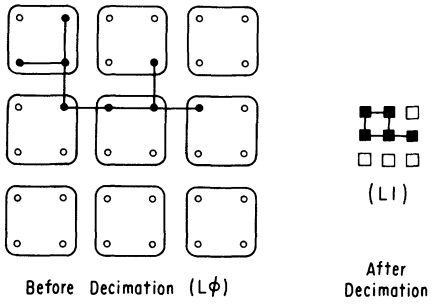

FIG. 10. An example of method 2 for the measure of fractal dimension—a block-spin decimation.

gression were used. Only clusters larger than 10^4 points were used. The relatively large fluctuations for large values of r are statistical. One can fit this data to straight lines with roughly 5% accuracy. Because the slope is Y_H-2 and Y_H is close to 2, we find Y_H to within 0.3%. The results of these measurements are included in Table I. Although the agreement is never far from 0.3% there may be some systematic error, as the measured values of Y_H are slightly higher than expected for the *integer* q values. This is believed to be caused by the finite size of the clusters studied. Our method should only work when the size of the squares is small compared to the size of the cluster being studied. In the future we hope to be able to correct for this effect. The agreement with previous determinations of Y_H for the integer q values gives us confidence that this method is reliable at the 0.5% level. The data available at this time is consistent with the conjecture of Nienhius *et al.* and of Pearson.

B. Methods 2 and 3

Methods 2 and 3 are very similar and will be described together. Both are based on a decimation scheme similar to the real-space renormalization-group methods used to study spin systems. We assume that we have a cluster C_0 on an $N\times N$ lattice L_0 and that there are V_0 points in the cluster. Form the $N/2\times N/2$ lattice L_1 by assigning one block spin in L_1 to every four spins of L_0 as illustrated in Fig. 10. In the case of method 2, form the cluster C_1 on the lattice L_1 by assigning a point of L_1 to C_1 if at least one point of L_0 is in C_0. Method 3 differs from method 2 only in that we require that at least two of the points in L_0 corresponding to the point of L_1 be in C_0 before assigning the point in L_1 to C_1. C_2 is to be constructed out of C_1 and put on the $N/4\times N/4$ lattice L_2 in the same manner in which C_1 was constructed from C_0. When we have constructed $C_1,C_2,C_3,\ldots,$ and measured $V_1,V_2,V_3,\ldots,$ then if $V_k=2^{-dk}$, one says d is the effective fractal dimension of the cluster. Figures 11(a) and 11(b) are log-log plots of data obtained using these methods, again, as d is known to be close to 2, we show not V_k directly but $V_k/2^k$. The results of these two methods are included in Table I. As one can see, the values of Y_H are much too low, and method 2 is worse than method 3. Y_H too low means that the volume V_k is not decreasing fast enough with successive decimations. This is evidently caused by the boundary. After k decimations, a $2^k\times 2^k$ square, which barely touches the cluster, will end up being included, so that our decimated cluster will not be a coarse-grained version of the original, but rather a coarse-grained version of the original plus all points within 2^k lattice spacings of the original. This would not be a problem if, as is the case for normal sets such as a large disk, the number of boundary points were small compared to the number of interior points. Percolation clusters have just as many points on the boundary as on the interior.[17] Method 3 is better than method 2, because the stricter criteria used keeps the decimated clusters from spreading as fast.

In conclusion, we have a method, method 1, of measuring the fractal dimension of clusters, which is

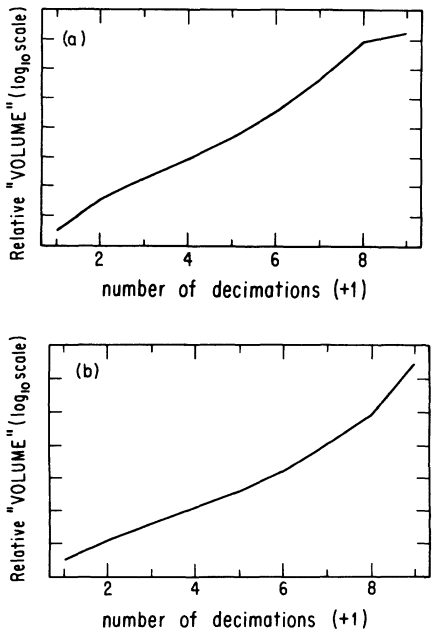

FIG. 11. (a) and (b) Representative data for methods 2 and 3 for the measure of fractal dimension, respectively. In each case $q=2$ and the relative "volume" has been multiplied by r^2 $(=4^k)$ with k as the number of decimations. Thus the slope of the line measures the difference of Y_H from 2.

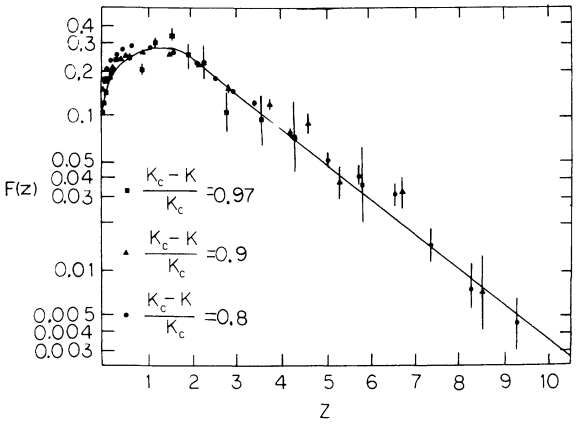

FIG. 12. Measured values of $F(x)$ ($\log_{10}$ scale) vs the dimensionless variable Z with $q=2$ and $K/K_c=0.8,0.9,0.97$.

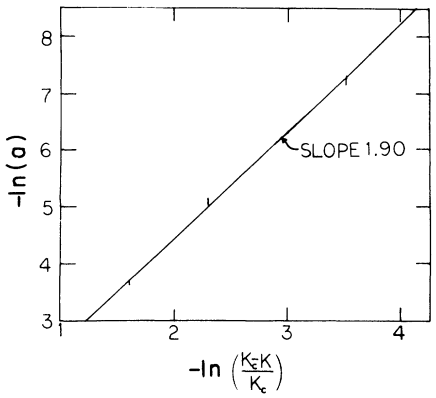

FIG. 13. Illustration of the slope of the exponential tail of $G(z)$ vs $\ln[(K_c-K)/K_c]$ for $K/K_c=0.8$, 0.9, and 0.97.

reliable. On the other hand, the examples of two unreliable methods, which intuitively appear to be correct, demonstrate that fractal objects can behave counter to our intuition, so that one must be very careful. At this time the failure of methods 2 and 3 is believed to be due to a boundary effect, an effect which in large is due to the demonstrated fact that percolation clusters have as many boundary points as interior points.

VIII. MEASURE OF Y_T

Y_T is the temperature exponent which describes the scaling behavior, or dimensionality of energy. Thus $z=S(K-K_c)^{(Y_H/Y_T)}$ is dimensionless.

If scaling works,[7] then

$$N(S,K) \underset{\substack{K\to K_c \\ S\to\infty}}{\to} S^{-\tau}F(z)\ , \tag{8.1}$$

where $\tau=1+2/Y_H$, S is the cluster size, K is the interaction strength, and $z=S(K-K_c)^{(Y_H/Y_T)}$. $N(S,K)$ is the mean number of clusters containing S spins (per unit area).

Figure 12 compares $F(z)$ as measured with $Q=2$ and $K/K_c=0.8$, 0.9, and 0.97. The error bars are statistical. The large errors for large Z and K close to K_c result because one must count large ($\sim 10^4$ spin) clusters to measure these points and such clusters occur infrequently.

TABLE II. Measured values of Y_T.

q	Values of β/β_c compared	This work	Theory	Other	Other method
0.5	0.8, 0.7	0.52	0.561		
1.0	0.93, 0.87, 0.8	0.73	0.75	0.733±0.008	MCRG (Ref. 20)
				0.75	MCRG (Ref. 23) site percolation on a triangle lattice
1.5	0.97, 0.9, 0.8	0.86	0.8867		
2.0	0.97, 0.9, 0.8	0.99	1	1	exact (Onsager)
2.5	0.97, 0.9, 0.8	1.04	1.102		
3.0	0.97, 0.9, 0.8	1.17	1.20	1.21−1.22	MCRG (Ref. 13)
				1.20	exact?[a] Hard hexagon model
3.5	0.99, 0.97, 0.9, 0.8	1.29	1.305		
4.0	0.99, 0.97, 0.9, 0.8	1.41	1.5	1.34	MCRG (Refs. 13 and 21)
				1.50	exact?[a] Triplet Ising model (Ref. 22)

[a]These models are believed to be in the same universality class. Theory—the conjecture of den Nijs: $Y_T=\frac{3}{2}[2+\pi/(u-\pi)]$; $\cos(u)=\frac{1}{2}\sqrt{q}$.

Y_T can be determined by comparing the functions $G(S)=S^{\tau}N(S,K)$ for different values of K. $\log[G(S)]$ was found to behave as $\log[G(S)]=-as+b$ for large S. Figure 13 shows the values of $\log(a)$ vs $\log(K-K_c)$ obtained by a least-squares fit to the data of Fig. 12. The line through the points corresponds to $Y_T=0.987$. Y_T was measured by matching the exponential tails of the functions $G(S)$. Although logarithmic corrections are expected to be present when $Q=4$, no unusual behavior was observed. The value of Y_T is much lower than the predicted $\frac{3}{2}$ when $q=4$ and this may be the result of logarithmic corrections. The $q=\frac{1}{2}$ case is special because K had to be far from K_c in order to make the clusters smaller than the lattice. The measured value of Y_T may be distorted by the need to work near $K/K_c=0.7$.

Table II lists the measured values of Y_T together with the values known from other measurements and the values predicted by the conjecture of den Nijs. Other than $q=4$, which is not a good test case, there is good agreement both with other methods and with the conjecture of den Nijs. As with the measurement of Y_H, there appears to be a systematic error, as the measured values are slightly low. This may be caused by the use of a finite lattice. Correction for these effects will allow for an order of magnitude improvement in accuracy of the measurement of the critical exponents. The prospects for being able to make such corrections are good.

IX. CONCLUSION AND FUTURE PROSPECTS

Simulation of the weighted percolation problem is a practical way of equilibrating large Potts-model configurations near criticality if our auxiliary data structure is used to quickly determine connectedness of percolation clusters. We demonstrated the utility of this simulation by measuring the critical exponents of the Potts model and found the measurements to be clean at the 1% level. The prospects for improving the accuracy and order of magnitude are good. However, in addition to simply using more computer time to obtain better statistics, we need to compensate for the effects of finite-lattice size. It is hoped that our auxiliary data structure, based upon closed chains of pointers, will find other applications in addition to the study of the Potts model.

ACKNOWLEDGMENTS

I would like to express my appreciation to M. P. M. den Nijs, T. Eguchi, and L. P. Kadanoff for helpful discussions and encouragement. This work was supported in part by an NSF predoctoral fellowship, NSF Grant No. PHY 79-23669, and the Louis B. Block Foundation.

---

*Submitted to the Department of Physics, The University of Chicago, in partial fulfillment of the requirements for the Ph.D. degree.

[1]R. B. Potts, Proc. Cambridge Philos. Soc. 48, 106 (1952).

[2]C. M. Fortuin and P. W. Kasteleyn, J. Phys. Soc. Jpn. Suppl. 26, 11 (1969); C. M. Fortuin and P. W. Kasteleyn, Physica 57, 536 (1972).

[3]J. N. V. Temperley and E. H. Lieb, Proc. R. Soc. London Ser. A 322, 251 (1971).

[4]R. J. Baxter, S. B. Kelland, and F. Y. Wu, J. Phys. A 9, 397 (1976).

[5]R. J. Baxter, J. Phys. C 6, 445 (1973).

[6]D. Stauffer, Phys. Rev. Lett. 41, 1333 (1978); H. J. Herrmann, Z. Phys. B 32, 335 (1979).

[7]D. Stauffer, Phys. Rep. 54, 1 (1978); J. W. Essam, Rep. Prog. Phys. 43, 830 (1980).

[8]*Monte Carlo Methods in Statistical Physics,* Vol. 7 of Topics in Current Physics, edited by K. Binder (Springer, Heidelberg, 1979).

[9]H. E. Stanley, *Introduction to Phase Transitions and Critical Phenomena* (Oxford University Press, New York, 1971).

[10]M. P. Nightingale and J. W. J. Blöte, Physica A 104, 352 (1980); M. P. Nightingale, J. W. J. Blöte, and B. Derrida, J. Phys. A 14, L45 (1981).

[11]M. P. M. den Nijs, Ph.D. Thesis, Katholieke Universiteit te Nijmegen, 1979 (unpublished).

[12]B. Nienhuis, E. K. Riedel, and M. Schick, J. Phys. A 13, 631 (1980); R. B. Pearson, Phys. Rev. B 22, 2579 (1980).

[13]C. Rebbi and R. H. Swendsen, Phys. Rev. B 21, 4094 (1980).

[14]L. P. Kadanoff, Phys. Rev. B 22, 2560 (1980); M. Nauenberg and D. J. Scalapino, Phys. Rev. Lett. 44, 837 (1980).

[15]B. B. Mandelbrot, *Fractals: Form, Chance and Dimension* (Freeman, San Francisco, 1977).

[16]*Phase Transitions and Critical Phenomena,* edited by C. Domb and M. S. Green (Academic, London, 1972), Vols. 1 and 6.

[17]H. Kunz and B. Souillard, Phys. Rev. Lett. 40, 133 (1978).

[18]R. J. Baxter, J. Phys. A 13, L61 (1980).

[19]M. N. Barber and R. J. Baxter, J. Phys. C 6, 2913

(1973).

[20]S. Kirkpatrick, in *Ill Condensed Matter,* edited by R. Balian, R. Maynard, and G. Toulous (North-Holland, Amsterdam, 1979), p. 340.

[21]P. D. Eschbach, D. Stauffer, and H. J. Herrmann, Phys. Rev. B 23, 422 (1981).

[22]R. J. Baxter and F. Y. Wu, Phys. Rev. Lett. 31, 1294 (1973).