Highly efficient algorithm for percolative transport studies in two dimensions

D. J. Frank
IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598

C. J. Lobb
Division of Applied Sciences and Department of Physics, Harvard University, Cambridge, Massachusetts 02138

(Received 2 March 1987)

We present a fast algorithm using the Y-∇ transformation for calculating the conductance of large two-dimensional random resistor lattices. The procedure can also be used to calculate other quantities for which Y-∇ transformations can be defined. As an example of the method's usefulness, we present extensive data on square random resistor lattices as large as 1000 bonds on a side. Using this data, we estimate that the conductivity exponent $t = 1.303^{+0.004}_{-0.014}$ in two dimensions.

I. INTRODUCTION

Since the problem was first discussed extensively in the early 1970s, there has been considerable interest in the conductivity of randomly diluted resistor lattices. Even an incomplete sampling of the literature contains numerous papers using Monte Carlo approaches,[1–7] analog experiments,[8–14] and renormalization-group calculations[15–21] and their extensions to larger cells in the form of finite-size scaling arguments.[21–27]

In this paper, we will mostly consider a square lattice, occupied by bonds with conductance g with probability p, and empty with probability $1-p$. In Sec. II, an *exact* algorithm for solving any given realization of a large lattice is described. In Sec. III, the algorithm is applied to square lattices at the percolation threshold $p=p_c$; the data obtained are used, via finite-size scaling, to obtain an accurate estimate of the critical exponent t. Finally, we conclude the paper in Sec. IV, and outline how other problems and other lattices may be solved using this technique.

II. THE ALGORITHM

In the absence of an exact solution, the bulk conductance of large random resistor lattices has been widely studied using numerical methods. To find the bulk conductance of such lattices, we have developed a method which consists of applying a sequence of transformations to the bonds of the lattice.[28] The final result of this sequence of transformations is to reduce the lattice to a single bond that has the same conductance as the entire lattice. First we discuss the transformations, and then the algorithm.

The simplest two transformations of conductances G are the well-known series and parallel reductions, which are shown in Fig. 1. For the series case,

$$G_s = G_1 G_2 / (G_1 + G_2) , \tag{1}$$

and for the parallel case,

$$G_p = G_1 + G_2 . \tag{2}$$

In both cases, the resulting single conductor is completely equivalent to the two original conductors as far as any external circuit is concerned.

Also well known is the Y-∇ transformation,[29] shown in Fig. 1(c). This transformation is defined in both directions, the conductances given as follows:

$$Y \rightarrow \nabla: \begin{cases} G_A = \dfrac{G_2 G_3}{G_1+G_2+G_3} \\ G_B = \dfrac{G_3 G_1}{G_1+G_2+G_3} \\ G_C = \dfrac{G_1 G_2}{G_1+G_2+G_3} , \end{cases} \tag{3}$$

and

$$\nabla \rightarrow Y: \begin{cases} G_1 = G_B G_C \left[\dfrac{1}{G_A} + \dfrac{1}{G_B} + \dfrac{1}{G_C} \right] \\ G_2 = G_C G_A \left[\dfrac{1}{G_A} + \dfrac{1}{G_B} + \dfrac{1}{G_C} \right] \\ G_3 = G_A G_B \left[\dfrac{1}{G_A} + \dfrac{1}{G_B} + \dfrac{1}{G_C} \right] . \end{cases} \tag{4}$$

Note that, just as in the series and parallel cases, the conductor configurations are completely equivalent from the point of view of the external circuit.

We next introduce the primary transformation used in this paper. This transformation, which we call the propagator transformation, is illustrated in Figs. 2(a)–2(d) and can be decomposed into three separate parts: first a $\nabla \rightarrow Y$ transformation, then a redefinition of the lattice point, and finally a $Y \rightarrow \nabla$ transformation. (The middle step is included primarily for clarity.) In transforming from Fig. 2(a) to Fig. 2(b), conductances 6, 7, and 8 are determined from 1, 2, and 3 by Eq. (4). Then, in the

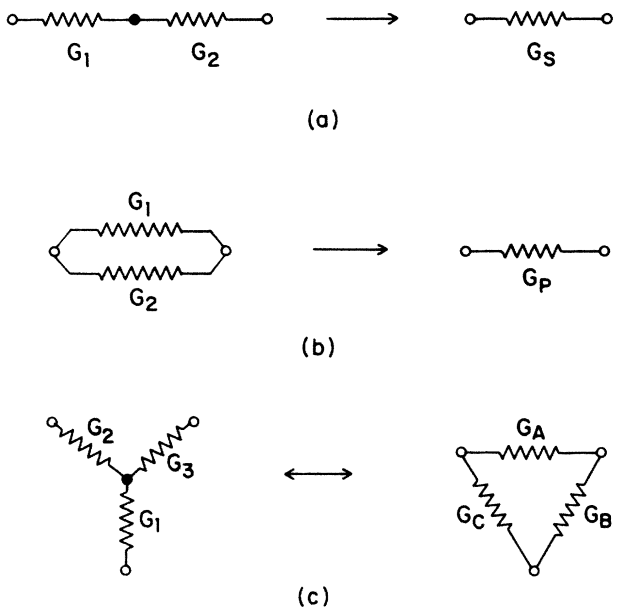

FIG. 1. Transformations for conductances in (a) series, (b) parallel, and (c) the Y-∇ transformation. All three transformations have no effect on the overall conductance of a circuit connected to the open circles.

transformation from Fig. 2(c) to Fig. 2(d), conductances 9, 10, and 11 are determined by Eq. (3). When the conductances in Fig. 2(d) are calculated in this way from those in Fig. 2(a), the two conductor configurations are completely equivalent from the point of view of the external circuit, and net topological effect is to propagate the diagonal bond past the lattice point in the middle.

When some of the conductors are missing, as is often the case in percolation problems, the propagator usually becomes simpler, and often does not result in a new diagonal conductor being generated. An example of this is shown in Figs. 2(e)–2(h), where one bond is missing. (This case also corresponds to what occurs at the edges of a lattice.) In this case, the first two parts of the propagator are the same as before, but the final $Y \rightarrow \nabla$ is reduced to a simple combination of two conductors in series. There is no longer any diagonal bond.

The algorithm for reducing a two-dimensional (2D) square lattice using this transformation is illustrated in Fig. 3 for a $b=3$ cell at $p=1$, where p is the probability that a bond is present and b is the length of a cell in the x direction. Starting with a full lattice in Fig. 3(a), the two conductors attached to the top node of the left-hand column are combined in series to form a diagonal bond. This bond is then propagated through the lattice, as indicated in Fig. 3(b), until there is no longer a diagonal bond. In this case, the diagonal bond terminated at an edge, following the example given in Figs. 2(e)–2(h). (In a larger cell with $p \neq 1$, it would be more likely that the diagonal bond would terminate somewhere inside the lattice due to missing bonds.) The same procedure is again followed in Figs. 3(c)–3(g); the uppermost node of the leftmost column is removed by a series reduction and the resulting diagonal conductor is propagated until it terminates. After the last node is eliminated, the remaining series string of conductors in Fig. 3(h) is easily reduced to a single conductor, as shown in Fig. 3(i). The conductance of this single conductor is equal to the conductance of the original lattice.

The procedure outlined above was implemented in FORTRAN for general b and was run first on a DEC LSI 11/2, and then, later, for the larger lattices, on a VAX11-780. More recently, it has been adapted to run on an IBM 3081 mainframe computer. Proper operation of the program and its algorithm were verified by having it solve lattices with known conductances. We were able to evaluate accurately the conductance of any size lattice that would fit in the memory of the machine being used. The program is a straightforward reduction of the algorithm to program statements. [The only subtlety not mentioned above is that after all nodes are removed from a column, the program checks to see if the single

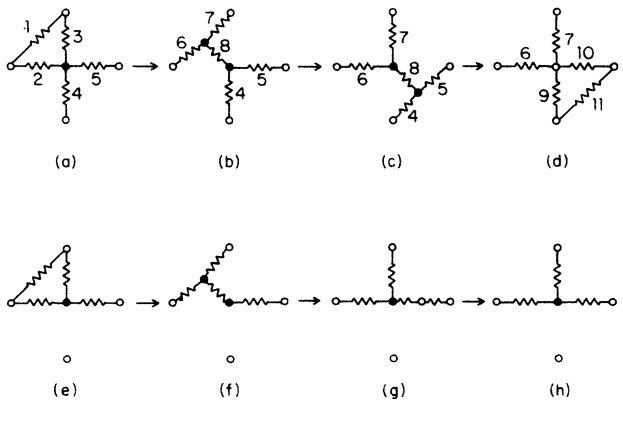

FIG. 2. The propagator transformation, which moves a diagonal bond through the lattice. The general case is illustrated in (a)–(d). When some bonds are missing, as they are at the boundary of a lattice, the propagator becomes simpler, as in (e)–(h).

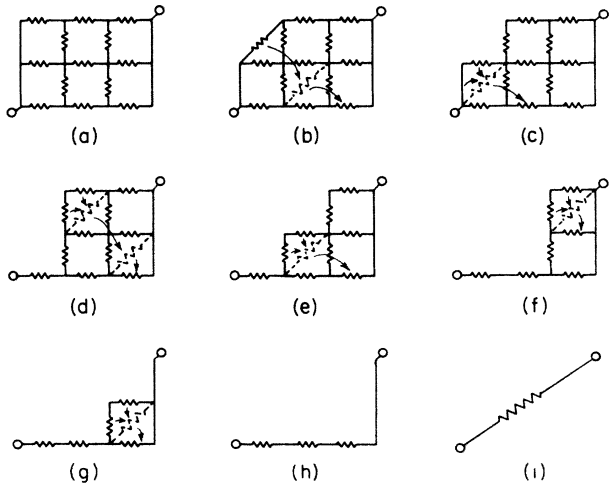

FIG. 3. Reduction of a small cell to a single conductance, using the bond propagator and series transformations. Arrows indicate direction in which bonds are moved.

remaining conductance for that column is zero. If it is, then the conductance of the entire lattice must be zero (i.e., the lattice is not connected from left to right), and the reduction of the rest of the lattice is not necessary.]

The efficiency of the algorithm has been analyzed in two ways. The first is related to the total number of operations that must be performed to reduce a lattice, which is proportional to the amount of computer time used. It is easiest to do this analysis in terms of the number of propagations required, since we do not have any direct way of counting the operations. The algorithm is fastest for p near p_c, and for smaller lattices, as expected. Empirically, the average number of propagations per realization is approximately given by $0.42b^2\ln(b/1.6)$ for $p=p_c=\frac{1}{2}$, and is exactly $b(b+1)(b-1)/3$ for $p=1$. The amount of computer time was found to be nearly linearly dependent on the number of propagations required, the constant of proportionality being about 4 sec/1000 propagations for the LSI-11, about 0.14 sec/1000 propagations for the VAX, and about 0.013 sec/1000 propagations on the IBM 3081. Thus, for example, a $b=1000$ lattice at $p=p_c$, which contains, on the average, 10^6 conductors, typically takes 31 sec per realization on the IBM 3081.

The $b^2\ln(b)$ dependence near p_c can be semiquantitatively understood from the following considerations. There are three distinct types of conductances for the bonds in the lattice: normal conductors, opens, and shorts. The conductors and the opens are the usual percolation elements, and the shorts are used to define equipotentials at the edges, and are also generated in the interior of the lattice by the propagator transformations [e.g., when conductor 2 or 3 in Fig. 2(a) is open]. Thus, there are $2\times3^4=162$ possible combinations of elements in Fig. 2(a). Of these 162, only 32 actually propagate the diagonal; the rest all terminate it. When p is near unity, the combinations that result in a propagated diagonal predominate, while when p is near p_c, the nonpropagating combinations are much more common. Thus, when p is near p_c, the diagonal will not be propagated very many times before terminating. Since the data in this regime indicates that the total number of propagations is of order $b^2\ln(b)$, it must be that on average the diagonal is propagated $\ln(b)$ times before it terminates.

The $b^2\ln(b)$ dependence also has implications for the amount of memory needed for the algorithm, which is the second measure of efficiency we have considered. Since for $p\neq1$ the propagations do not occur to the full depth of the lattice, it is not necessary to store the entire lattice in the computer's memory. If the maximum depth to which a propagation can be expected to go is n, then only the leftmost n columns need be available in memory. When the leftmost column is transformed away, a new column on the right can be added, thus keeping the number of columns in memory constant until the right-hand side is reached. (This makes the algorithm useful for the long-strip geometry used in transfer-matrix calculations.[26])

This approach was actually used to solve our largest lattices. The peak memory required (floating point numbers) never exceeds $2b^2-1$, the memory required to store a $p=1$ cell. The average memory needed at $p=p_c$ is given approximately by $5.2b\ln(b/1.8)$; the worst-case cells that we encountered required that an additional $16b$ of memory be added.

The literature contains several other methods that have been used to solve for the conductance of square 2D resistor lattices, including the relaxation approach,[1-3,5-7] Fogelholm's generalized Y-∇ approach,[4] a Gaussian elimination method[5,25] which involves the simultaneous solutions of the node voltages as given by Kirchhoff's laws, and the related transfer-matrix approach.[26] The method used in our earliest work[20,21] is essentially a very restricted form of the approach later developed by Fogelholm; our later work[22] used the method outlined in this paper.

In comparing with these other methods, it appears that the present approach is the most well-behaved for problems where the conductivity must be calculated for the entire range of p values, from 1 to p_c. The relaxation method is better near $p=1$, provided one does not need exact answers, since it only requires of order b^2 operations. Near $p=1$, the Fogelholm, Gaussian, and transfer-matrix methods are much more costly than our method, since they require at least of order b^4 operations. (Lb^3 operations are required for the transfer-matrix approach for a strip of length L.) Near p_c, we would expect that the Fogelholm method might have a b dependence similar to ours, but it may be more computationally costly because of the linked data structure that is necessary. The minimum number of operations required for the Gaussian approach near p_c is unknown to us, but seems likely to be more than for our algorithm.

In addition to the above specific considerations, there are three general comparisons among the algorithms which can be made. First, the algorithm we have developed is restricted to certain well-behaved 2D lattices, while the other methods can be applied to any network in any number of dimensions. Second, our algorithm, the Gaussian, and the Fogelholm methods are exact, while the relaxation approach is approximate. Third, the algorithm we have used is not restricted to resistor lattices, but can be used on any 2D system in which there are well-defined Y-∇ and ∇-Y transformations,[21,30] as will be discussed in Sec. V. The other algorithms are not, in general, extendable in this way.

III. THE CRITICAL EXPONENT t

The exponent t can be determined most accurately from our data by using a large-cell renormalization-group approach or, in what amounts to the same thing for this problem, finite-size scaling.[31,32,20-27] For an infinite sample, it is expected that the bulk conductance G varies as

$$G\sim g(p-p_c)^t\ , \tag{5}$$

where g is the conductance of a bond which is present with probability p, and where Eq. (5) is true for a small range of $p\geq p_c$. It can then be argued that the average conductance $\langle G\rangle$ of large finite samples at $p=p_c$ is given by

$$\langle G \rangle = b^{-t/\nu}[c_1 + c_2 f_2(b) + \cdots] , \quad (6)$$

where $f_2(b) \to 0$ as b, the sample size measured in units of the lattice constant, approaches infinity, and where c_1 and c_2 are constants.[20–22] The exponent ν is the percolation correlation-length exponent.[31,32]

A considerable saving of computer time can be realized by carefully choosing the shape and boundary conditions used on the lattice. Figure 3(a) shows the type of cell we used for $b=3$. As has been shown,[33,18] cells of this type correctly predict that $p_c = \frac{1}{2}$ for all b. In addition, rapid convergence to the infinite-sample limit occurs for the "thermal" percolation exponents[34,35] and for transport properties.[20–22,36]

We used Eq. (6) to determine t numerically. $\langle G \rangle$ was calculated for $b=2$ to $b=1000$ by generating different realizations at the various sizes and keeping track of arithmetic, geometric, and harmonic means. To give some examples of the statistics of our data, we solved every distinct realization for $b=2$ and $b=3$, 2×10^6 randomly generated realizations for $b=4$, 10^6 for $b=9$, 10^4 for $b=30$, 14400 for $b=100$, and 5000, 1000, and 400 for $b=200$, 500, and 1000. The data for $b \geq 4$ are plotted in Fig. 4.

After the data were generated, they were fit to Eq. (6) with a variety of choices for $f_2(b)$. Our first choice was $f_2(b)=b^{-\Delta/\nu}$; that is, the correction is a power law, by analogy to the standard percolation case.[37] A least-squares fitting was done, varying t/ν, Δ/ν, and the number of points included in the fit to minimize χ^2. [The number of points was varied by omitting data for small b because it is not known how large b must be for Eq. (6) to be valid.] We obtained the best fits when the $b=2$ and $b=3$ data were not included; excluding points beyond $b=6$ did not improve the fit. This suggests that the scaling assumption is reasonable for $b \geq 4$ or, perhaps, $b \geq 5$ for this system and, indeed, the shape of the distribution of conductances is qualitatively scale invariant for larger sample sizes. Our fits to arithmetic, geometric, and harmonic means gave values which are included in the estimates $t/\nu = 0.977 \pm 0.002$ and $\Delta/\nu = 1.8 \pm 0.5$.

Another form for the correction which we considered explicitly was $f_2(b) = 1/\ln b$, which corresponds to the limit $\Delta \to 0$. Contrary to our earlier findings,[22] which were based on smaller cells with fewer realizations, this procedure does not give as satisfactory a fit as the power-law corrections. Although the best log-correction fits work reasonably well, they do not fit as well overall,

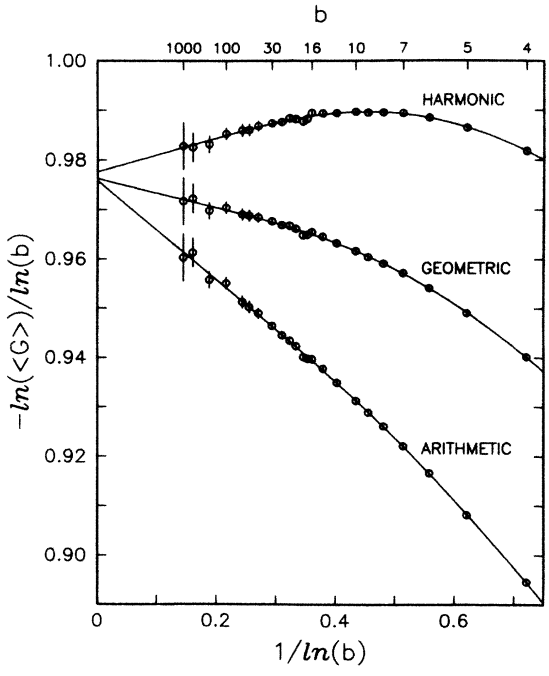

FIG. 4. Plot of $\ln(\langle G \rangle^{-1})/\ln b$ against $1/\ln b$. According to (6), the y intercept of these curves should be t/ν. The curves are three independent fits to (6) for $b \geq 4$ using the power-law correction described in the text. Error bars correspond to one standard deviation from the mean; points without error bars have uncertainties which are less than the size of their circles.

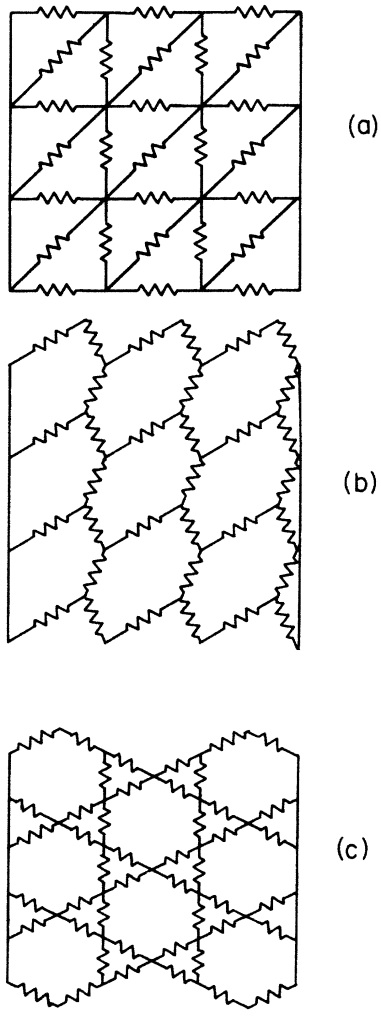

FIG. 5. Other common two-dimensional lattices which can be reduced to a single conductance using algorithms described in the text. Part (a) is a hexagonal lattice, drawn to clarify the transformations described in the text, (b) is a honeycomb lattice, and (c) is a Kagome lattice.

and they deviate systematically from the larger cell data. The log-correction fits give values that fall in the range $t/\nu = 0.970 \pm 0.014$.

Based on our preference for the power-law correction, our preferred value and subjective estimate of uncertainty is

$$t/\nu = 0.977^{+0.003}_{-0.010} . \quad (7)$$

The uncertainty is asymmetric in order to enclose the log-correction value. Using the value $\nu = \frac{4}{3}$, (Refs. 31, 32, 34, 35, 38, and 39) we obtain

$$t = 1.303^{+0.004}_{-0.014} . \quad (8)$$

IV. DISCUSSION

We have shown that an algorithm using $Y \rightarrow \nabla$, $\nabla \rightarrow Y$, series, and parallel transformations, makes it possible to reduce a square lattice of conductors to a single equivalent conductor. Using similar algorithms, the other common 2D lattices can also be reduced. Assuming propagation from the upper left to the lower right, as above, the hexagonal lattice, Fig. 5(a), can be reduced by first propagating out the diagonals, starting at the lower right and working to the upper left. For the honeycomb lattice, Fig. 5(b), there are two approaches: create the dual lattice, which is hexagonal, and solve it as above, or perform Y-∇ transformations on alternate rows of nodes, and then solve the resulting hexagonal lattice as above. As another example, the Kagome lattice, Fig. 5(c), can be solved by performing ∇-Y transformations on all of the small triangles, and then solving the resulting honeycomb lattice by either of the two methods outlined above.

A number of other physical quantities have $Y \rightarrow \nabla$, $\nabla \rightarrow Y$, series, and parallel transformations, and can thus be studied using our method. In an earlier publication,[21] we developed the necessary transformations for a model superconductor to find the critical current of percolative superconducting systems; the dual of this problem is a model for dielectric breakdown in a metal-dielectric mixture. This approach is currently being used to study breakdown in systems where a broad distribution of bond strengths are present.[40]

A novel application of this technique is to two-dimensional Ising models. Ising spins have a "star-triangle" transformation[30] which is topologically equivalent to the Y-∇ transformation used here. This algorithm should allow the study of randomly diluted Ising lattices or Ising spin glasses. (There are some difficulties in the latter case because combinations of positive and negative coupling energies can lead to complex-valued coupling energies after a transformation. Whether this leads to a breakdown of the method, or whether fully solved lattices will always have real couplings, is currently being studied.)

ACKNOWLEDGMENTS

We are grateful to Michael Dubson and M. G. Forrester for useful conversations regarding this work. This research was supported in part by the National Science Foundation through Grant No. NSF-DMR86-14003.

[1]Scott Kirkpatrick, Phys. Rev. Lett. **27**, 1722 (1971).
[2]Scott Kirkpatrick, Rev. Mod. Phys. **45**, 574 (1973), and references cited therein.
[3]A. K. Sarychev and A. P. Vinogradoff, J. Phys. C **12**, L681 (1979).
[4]Rabbe Fogelholm, J. Phys. C **13**, L571 (1980).
[5]Scott Kirkpatrick, in *Proceedings of the Summer School on Ill-Condensed Matter*, edited by Roger Balian, Roger Maynard, and Gerard Toulouse (North-Holland, Amsterdam, 1979), p. 340, and references cited therein.
[6]Itzhak Webman, Joshua Jortner, and Morrel H. Cohen, Phys. Rev. B **16**, 2593 (1977).
[7]Joseph P. Strately, Phys. Rev. B **15**, 5733 (1977).
[8]B. P. Watson and P. L. Leath, Phys. Rev. B **9**, 4893 (1974).
[9]N. T. Liang, Yueh Shan, and Shou-yih Wang, Phys. Rev. Lett. **37**, 526 (1976).
[10]Michael E. Levinshtein, J. Phys. C **10**, 1895 (1977).
[11]G. Deutscher and M. L. Rappaport, J. Phys. (Paris) Lett. **40**, L219 (1979).
[12]L. N. Smith and C. J. Lobb, Phys. Rev. B **20**, 3653 (1979).
[13]J. P. Clerc, G. Giraud, S. Alexander, and E. Guyon, Phys. Rev. B **22**, 2489 (1980).
[14]Michael A. Dubson and James C. Garland, Phys. Rev. B **32**, 7621 (1985).
[15]R. B. Stinchcombe and B. P. Watson, J. Phys. C **9**, 3221 (1976).
[16]R. Rosman and B. Shapiro, Phys. Rev. B **16**, 5117 (1977).
[17]Joseph P. Straley, J. Phys. C **10**, 1903 (1977).
[18]J. Bernasconi, Phys. Rev. B **18**, 2185 (1978).
[19]J. M. Yeomans and R. B. Stinchombe, J. Phys. C **11**, 4095 (1978).
[20]C. J. Lobb and D. J. Frank, J. Phys. C **12**, L827 (1979).
[21]C. J. Lobb and D. J. Frank, in *Inhomogeneous Superconductors–1979 (Berkeley Springs, WV), Proceedings of the Conference on Inhomogeneous Superconductors*, AIP Conf. Proc. **58**, edited by D. U. Gubser, T. L. Francavilla, J. R. Leibowitz, and S. A. Wolf (AIP, New York, 1980), p. 308.
[22]C. J. Lobb and D. J. Frank, Phys. Rev. B **30**, 4090 (1984).
[23]A. K. Sarychev and A. P. Vinegradoff, J. Phys. C **14**, L487 (1981).
[24]C. D. Mitescu, M. Allain, E. Guyon, and J. P. Clerc, J. Phys. A **15**, 2523 (1982).
[25]M. Sahimi, B. D. Hughes, Le. E. Scriven, and H. T. Davis, J. Phys. C **16**, L521 (1983).
[26]B. Derrida and J. Vannimenus, J. Phys. A **15**, L557 (1982); B. Derrida, D. Stauffer, H. J. Hermann, and J. Vannimenus, J. Phys. (Paris) Lett. **44**, L701 (1983); J. Zabolitzky, Phys. Rev. B **30**, 4077 (1984); H. J. Hermann, B. Derrida, and J. Vannimenus, *ibid.* **30**, 4080 (1984).
[27]A. K. Sarychev and A. P. Vinogradoff, J. Phys. C **16**, L1073 (1983).
[28]D. J. Frank and C. J. Lobb, Bull. Am. Phys. Soc. **27**, 388 (1982); also see AIP document No. PAPS PRBMDO-37-302-6 for six pages of a program listing of our version of the

algorithm. This program runs on an IBM 3081 using IBM VS FORTRAN Release 3.0. Order by PAPS number and journal reference from American Institute of Physics, Physics Auxiliary Publication Service, 335 East 45th Street, New York, NY 10017. The price is $1.50 for each microfiche (98 pages) or $5.00 for photocopies of up to 30 pages, and $0.15 for each additional page over 30 pages. Airmail additional. Make checks payable to the American Institute of Physics.

[29]Stephen B. Senturia and Bruce D. Wedlock, *Electronic Circuits and Applications* (John Wiley and Sons, New York, 1975), p. 75.

[30]Itiro Syozi, in *Phase Transitions and Critical Phenomena,* edited by C. Domb and M. S. Green (Academic, London, 1972), Vol. 1, p. 269.

[31]Peter J. Reynolds, H. Eugene Stanley, and W. Klein, J. Phys. A **11**, 1199 (1978).

[32]Peter J. Reynolds, H. Eugene Stanley, and W. Klein, Phys. Rev. B **21**, 1223 (1980).

[33]Peter J. Reynolds, W. Klein, and H. Eugene Stanley, J. Phys. C **10**, L167 (1977).

[34]C. J. Lobb and Keith R. Karasek, J. Phys. C **13**, L245 (1980).

[35]C. J. Lobb and Keith R. Karasek, Phys. Rev. B **25**, 492 (1982).

[36]C. J. Lobb, D. J. Frank, and M. Tinkham, Phys. Rev. B **23**, 2262 (1981).

[37]D. Stauffer, Phys. Rep. **54**, 1 (1979).

[38]M. P. M. den Nijs, J. Phys. A **12**, 1857 (1979).

[39]B. Derrida and L. de Seze, J. Phys. (Paris) **43**, 475 (1982).

[40]M. Octavio, A. Octavio, J. Aponte, R. Medina, and C. J. Lobb (unpublished).