A direct electrifying algorithm for backbone identification

2007 J. Phys. A: Math. Theor. 40 14679

(http://iopscience.iop.org/1751-8121/40/49/004)

IOP Publishing Journal of Physics A: Mathematical and Theoretical

J. Phys. A: Math. Theor. **40** (2007) 14679–14686 doi:10.1088/1751-8113/40/49/004

A direct electrifying algorithm for backbone identification

Chunyu Li and Tsu-Wei Chou

Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA

E-mail: chou@udel.edu

Received 10 July 2007, in final form 10 October 2007
Published 21 November 2007
Online at stacks.iop.org/JPhysA/40/14679

Abstract

This paper proposes a new algorithm for identifying backbones in the application of percolation theory. This algorithm is based on the current-carrying definition of backbone and is carried out on the predetermined spanning cluster. It is fairly easy to implement and further parallelize. The efficiency is enhanced by the fact that the conductivity of a percolating system can be obtained in the same processing of backbone identification. The critical exponents of backbone mass, red bonds (sites) and conductivity obtained by this algorithm are in very good agreement with the existing results.

PACS numbers: 64.60.Ak, 05.10.−a, 05.70.Jk

(Some figures in this article are in colour only in the electronic version)

1. Introduction

The percolation theory has found applications in a wide variety of fields, such as materials science (polymers, concrete, composites and porous media) [1–4], geophysics (oil exploitation, geothermal power, groundwater pollution and earthquake prediction) [5], information technology (Internet [6], wireless communications [7]), sociophysics (social hierarchies, political persuasion and marketing) [8] and medical or biological studies (epidemics [9], species evolution [7]). Except for some very simple regular bond or site percolation problems which can be examined by analytical methods, the most complex percolating systems have to be simulated by the Monte Carlo (MC) method.

There are two main tasks in the MC simulations, which are of fundamental importance. One is to find the spanning cluster and the other is to identify the backbone. The Hoshen–Kopelman cluster labeling algorithm [11] along with its modifications [12] is most popular for finding the spanning cluster, though the Newman–Ziff algorithm [13] seems to be more efficient. The parallel implementations of the Hoshen–Kopelman algorithm have also been designed for large-scale MC simulations [14, 15]. In contrast, backbone identification is still a bottleneck in the simulation of large-scale percolation systems.

The backbone is a subset of the spanning cluster and essentially the biconnected nodes in computer science. It plays a fundamental role in any transport process in percolating systems. Tarjan's recursive depth-first-search (DFS) algorithm [16] is well known and most used for backbone identification. The burning algorithm [17] and the dual lattice algorithm [18] are much slower than Tarjan's and not easy to implement. The matching algorithm [19] is faster than the burning algorithm, but slightly slower than Tarjan's and also not easy to implement. The hull-generating algorithm [20, 21], which is twice as fast as Tarjan's, is also a depth-first-search algorithm but only suitable for strictly planar graphs. The fastest algorithm (four times as fast as Tarjan's) so far was proposed by Yin and Tao [22] based on the modified Hoshen–Kopelman algorithm. But its application is limited to two-dimensional graphs with open lateral boundary conditions, while the burning, matching and Tarjan's algorithms can be used for arbitrary graphs. Up to date, Tarjan's recursive DFS algorithm [16] seems to be the fastest algorithm for the general use of backbone identification. However, the intensive use of the stack is the major drawback of this algorithm. When dealing with a large system, the high number of consecutive recursive calls issued by this algorithm often causes a stack overflow [23]. Thus, the limitation of the recursive algorithm is obvious and its parallel implementation seems to be rather difficult [24].

In this paper, we propose an effective method, termed as a *direct electrifying algorithm*, for backbone identification by directly employing the definition of backbone which Kirkpatrick [25] described as the current-carrying part of a resistor network. There has been no report on identifying the backbone directly from its current-carrying definition. The effectiveness of this algorithm is demonstrated by a two-dimensional square lattice site and bond percolation problems. Its main advantage is in overcoming the drawback of recursive algorithms.

2. Direct electrifying method

Let us first consider the site percolation on a square lattice of size $L \times L$. Our algorithm for backbone identification requires three steps. The first step consists of randomly generating occupied sites based on a given probability p and determining if a spanning cluster has formed. The Hoshen–Kopelman algorithm is used for the identification of spanning clusters. Different from the traditional assumption of 'bus bar' geometry in which the two opposite sides of a lattice are entirely connected to superconducting electrodes, we only assume that the sites on two opposite sides are individually connected to superconductors. With this modification, the number of dangling arcs connected with the spanning cluster would be significantly reduced. The other two sides are assumed to be open as usual. The output of the first step includes the information of sites on the spanning cluster and the connectivity between these sites.

The second step is to calculate the electric current flowing through the spanning cluster. We assume that the connection between any pair of two neighboring sites on the spanning cluster represents a resistor with unit resistance (figure 1(*a*)). A voltage is applied to the superconducting electrodes by assuming the electric potential to be 0 at the bottom and 1 at the top. The voltage distribution at each site of the spanning cluster can be solved by establishing a system of algebraic equations based on the finite element method [26].

For a typical resistor element *i*–*j* (figure 1(*b*)), the elemental matrix representing the relation between the current (I) entering the element at the ends and the end potential (V) is

$$\begin{Bmatrix} I_i^e \\ I_j^e \end{Bmatrix} = \left[K_{ij}^e\right] \begin{Bmatrix} V_i \\ V_j \end{Bmatrix} = \frac{1}{R^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} V_i \\ V_j \end{Bmatrix}. \tag{1}$$

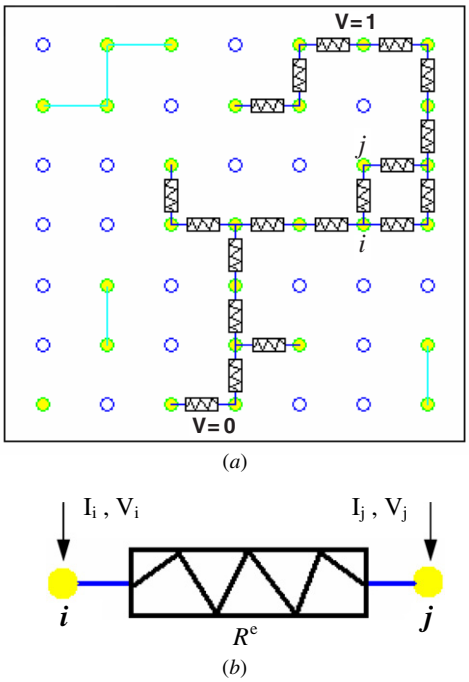

Figure 1. (*a*) Resistor network, (*b*) an isolated resistor element.

According to Kirchhoff's current law, a system of algebraic equations can be assembled for the entire spanning cluster:

$$\mathbf{I} = \mathbf{KV}, \tag{2}$$

where $\mathbf{V} = \{V_1, V_2, \ldots, V_n\}^T$ stands for the nodal potentials at n sites belonging to the spanning cluster, $\mathbf{I} = \{I_1, I_2, \ldots, I_n\}^T$ is the vector of external input current at the n sites (here $\mathbf{I} = 0$, because we are not inputting any current at any site into the spanning cluster) and the global coefficient matrix

$$\mathbf{K} = \sum_{e=1}^{m} \left[K_{ij}^e\right], \tag{3}$$

where m is the number of resistor elements.

After applying the voltage boundary conditions to equation (2), the electric potentials at each site of the spanning cluster can be obtained. The current flowing through each resistor element can then be determined by

$$I^e = (V_i - V_j)/R^e. \tag{4}$$

The final step is to extract the backbone from the spanning cluster based on the current-carrying definition. If the current in a resistor element is nonzero, it means that this resistor is carrying current and its two ends must belong to the backbone. The backbone can be identified after all the resistors in the spanning cluster are scanned. As shown in figure 2, all of the dangling ends, loops and arcs carry no current. For bond percolation problems, the procedure for backbone identification is roughly the same as described above for site percolation.

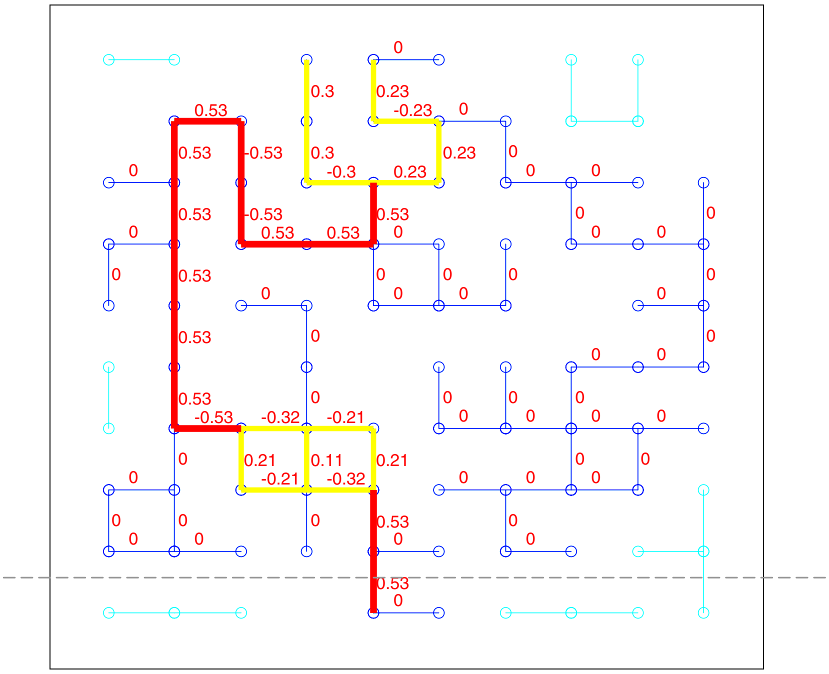

Figure 2. An example of identified backbone (text: electric current value, positive directions are ↓ and ←; Color: red—red bonds, yellow—blobs, blue—dangling bonds; dash line: section for determining the total current).

It should be pointed out that the backbone extracted based on the above algorithm would actually be the effective backbone [26]. Some so-called perfectly balanced bonds (PBBs), which carry no current as a result of the voltages on their ends being equal, belong to the geometrical backbone. Although the PBBs are often single bonds, sometimes they can be multiple bonds. If the finding of the geometrical backbone is the objective, then a small change should be made to the algorithm described above. Batrouni *et al* [27] once made a systematic study on the density of perfectly balanced bonds in the geometrical backbone by using a method of noisy resistors. We adopt the same tactic in our algorithm.

For finding out the geometrical backbone, we only need to add small random noisy resistances (uniformly distributed between -0.001 and 0.001) to the resistors R^e (unit resistance) in the elemental matrix $\left[K_{ij}^e\right]$ shown in equation (1). Then the PBBs will also have a current in addition to the effective backbone. Thus, based on the 'nonzero-current' criterion, the entire geometrical backbone can be accurately identified. The choice to use uniform unity resistances or noisy resistances depends on the objective of simulations. Usually, the backbone identification is for calculating transport properties. The PBBs are not really important under such circumstances. If so, then the algorithm without noisy resistances is accurate enough. If the geometrical backbone is the real objective, then the algorithm with noisy resistances should be used.

3. Conductivity calculation

The backbone is relevant to transport properties. Its structure consists of red bonds and blobs. The critical exponent of the backbone represents the scaling relationship of backbone mass

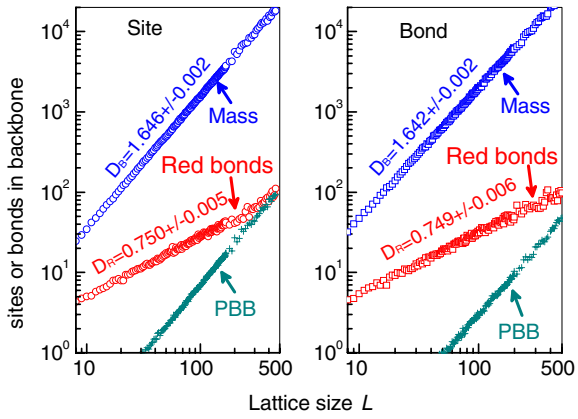

Figure 3. Log–log plot of sites or bonds in a backbone versus lattice size (PBB: perfectly balanced bonds).

with the systems size. The exact value of the critical exponent is not known, but a current numerical estimate is 1.6432 for 2D percolations [29]. The identification of the backbone enables us to calculate the scaling exponents of backbone mass and red sites or red bonds. The backbone mass represents the total number of sites or bonds in a geometrical backbone, which is extracted by using the noisy resistance method (actually, the uniform unity resistance method gives the same results because the PBBs have almost no effect on the average mass due to their very small percentage). The red bonds or red sites are figured out by using the uniform unity resistance method. The exponents D_B and D_R are defined by the relationship between the mass M_B, the number of red sites (bonds) N_R and the lattice size L as

$$M_B \sim L^{D_B}, \qquad N_R \sim L^{D_R}. \tag{5}$$

The direct method to determine the scaling exponents involves plotting the mass versus the lattice size in log–log scale and then determining the slope of the linear fitting line. We use lattice sizes ranging from 8×8 to 500×500. MC simulations are carried out 1000 times for each size. The percolation thresholds are taken as $p_c = 0.592\,746$ for site percolation and $p_c = 0.5$ for bond percolation. Figure 3 shows the average backbone mass variation with the lattice size. The slopes of the log–log lines give $D_B = 1.646 \pm 0.002$ for site percolation and $D_B = 1.642 \pm 0.002$ for bond percolation. This result is in very good agreement with the most extensive simulation performed by Grassberger [30], who gave $D_B = 1.6432 \pm 0.0008$ for both site and bond percolations. The exponent $D_R = 0.750 \pm 0.005$ for site percolation and $D_R = 0.749 \pm 0.006$ for bond percolation are also in excellent agreement with the exact value $1/\nu = 3/4$ given by Conniglio [31]. The number of PBBs in the backbone is proportional to the backbone mass at a percentage of $(0.156 \pm 0.002)\%$ for bond percolation, which is in very good agreement with $(0.159 \pm 0.004)\%$ given by Batrouni *et al* [26]. We also find that the number of PBBs is $(0.342 \pm 0.002)\%$ of the backbone mass for site percolation, which has not been reported before, if we assume that there is a bond between a pair of neighboring sites.

We also obtain the results of conductivity for site and bond percolations in the same procedure of backbone identification. The total electric current is first calculated by summing up the currents of backbone bonds that cross a selectively prepositioned line (shown in figure 2)

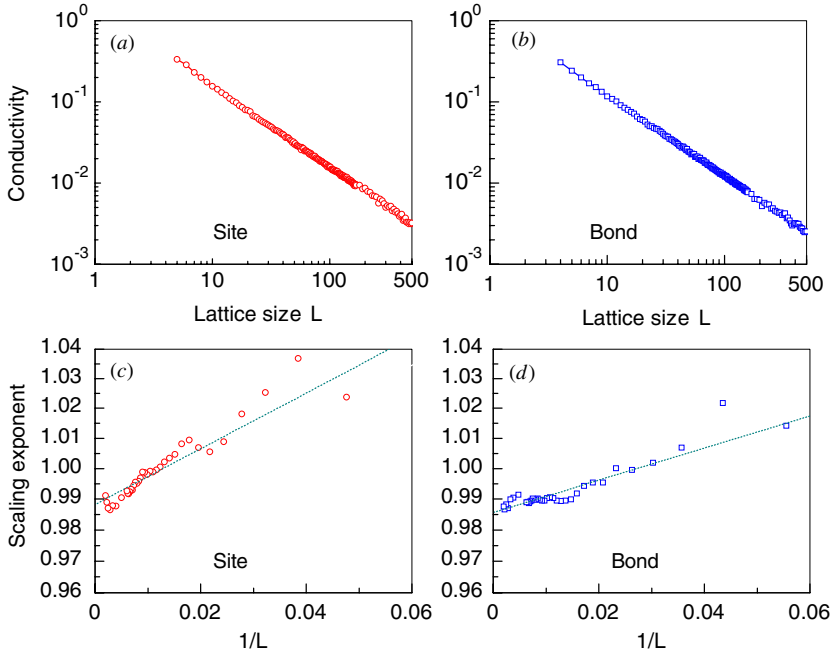

Figure 4. The conductivity of percolating systems and its critical exponent.

paralleling to the electrode border. The effective resistance of the percolating system is then given by

$$R_{\text{eff}} = (V_{\text{top}} - V_{\text{bottom}})/I_{\text{total}}, \tag{6}$$

and the conductance G_{eff} is just the reciprocal of the effective resistance

$$G_{\text{eff}} = 1/R_{\text{eff}}. \tag{7}$$

Because of the fact that the lattice sizes in the two directions are the same and the thickness of the lattice can be assumed to be unity, the conductivity σ is thus equal to the effective conductance. This is a very efficient and universal method to compute the conductivity, compared with the algorithm of Lobb and Frank [32]. Figures 4(*a*) and (*b*) show the dependence of conductivity on the lattice size. The almost straight log–log lines prove the scaling relationship between the conductivity and the lattice size, i.e. $\sigma \sim L^{-D_\sigma}$. In fact, the exponent gradually decreases with an increasing lattice size (figures 4(*c*) and (*d*)). The scaling exponent in figure 4 is calculated by using different numbers of data points. With an increasing size, the available data of conductivity increase. For a given size system, the scaling exponent is calculated using all available data produced before. It is obvious that a larger size gives a more accurate exponent, though with some fluctuations. The exponent D_σ can be linearly extrapolated to 0.988 for site percolation and 0.986 for bond percolation, which is in very good agreement with the 0.9826 of Grassberger [30] and 0.9745 of Normand *et al* [33]. The conductivity exponent for 2D percolation is thus $\nu D_\sigma = \frac{4}{3} D_\sigma$ =1.315–1.317, which is in excellent agreement with theoretical and experimental results [30, 34].

4. Discussions

The present algorithm has been tested in our MC simulations for both site and bond percolation problems and has demonstrated superb accuracy. The advantage is in the simplicity of the present algorithm, which is essentially a one-dimensional FEM. The implementation is fairly straightforward and the know-how accumulated in the studies of the FEM is readily transferable. For example, the parallel implementation of the FEM has been explored for decades and many effective algorithms have been successfully developed [28]. Thus, researchers interested in the algorithm proposed in this paper can readily take the advantage of parallel computing for FEM.

For general graphs, the popular algorithm for identifying backbone seems to be Tarjan's depth-first-search algorithm, which has a time complexity $O(N)$ for a graph of N nodes. But its main drawback is the recursion. Dealing with a large system may cause a computer stack overflow. The present algorithm for backbone identification has the advantage in overcoming the drawback of recursive algorithms. Its time complexity depends on the method used for solving the system of linear equations. For small or medium systems, the direct methods are usually used and have time complexity around $O(N^3)$. Thus, for small or medium systems, the present algorithm seems to be not competitive. But for large systems, iterative methods are more suitable and the time complexity of some iterative methods is believed to in be in the order of $O(\log N)$–O(N) [35]. Thus the present algorithm is attractive for large systems, especially its implementation in parallel computers because of its non-recursive feature. For demonstrating the feasibility, we only simulated 500×500 site and bond systems. But theoretically, there is no size limitation in the present algorithm. Further studies, such as the calculation time compared with other backbone identification methods, the implementation on parallel computers, the optimal solver for the system of linear equations, etc., are necessary for fully taking advantage of the power of the present algorithm.

5. Conclusion

In summary, a straightforward algorithm for backbone identification is proposed in this paper. The algorithm is based on the current-carrying definition of backbone and is carried out on the predetermined spanning cluster. The conductivity of the percolating system can also be obtained in the same processing of backbone identification. The Monte Carlo simulations on site and bond percolation problems based upon this algorithm indicate that it is very effective. The parallel implementation of the present algorithm can be easily realized if necessary. It is expected that the present algorithm will overcome the drawback of recursive algorithms and greatly expand the limit on the simulation size of percolating systems.

Acknowledgments

This work has been supported by the US National Science Foundation NIRT Program, grant no. 0304506 (Dr Ken P Chong, Program Director), the US Air Force Office of Scientific Research, grant no FA9550-06-1-0489 (Dr Byung-Lip Lee, Program Director) and the US Office of Naval Research, grant no N00014-07-1-0345 (Dr Yapa Rajapakse, Program Director).

References

[1] Clarke D R 1992 Interpenetrating phase composites *J. Am. Ceramic. Soc.* **75** 739–59
[2] Strumpler R and Glatz-Reichenbach J 1999 Conducting polymer composites *J. Elelctroceram.* **3** 329–46

[3] Thostenson E T and Chou T W 2006 Carbon nanotube networks: sensing of distributed strain and damage for life prediction and self healing *Adv. Mater.* **18** 2837–41
[4] Li C Y and Chou T W 2007 Continuum percolation of nanocomposites with fillers of arbitrary shapes *Appl. Phys. Lett.* **90** 174108
[5] Sahimi M 1994 *Applications of Percolation Theory* (Bristol, PA: Taylor and Francis)
[6] Newman M E J 2003 The structure and function of complex networks *SIAM Rev.* **45** 167–256
[7] Glauche I, Krause W, Sollacher R and Greiner M 2003 Continuum percolation of wireless ad hoc communication networks *Physica* A **325** 577–600
[8] Stauffer D 2003 *Sociophysics—a review of recent Monte Carlo simulations Fractals* **11** 313–8
[9] Moore C and Newman M E J 2000 Epidemics and percolation in small-world networks *Phys Rev.* E **61** 5678–82
[10] Lipowski A and Lopata M 1999 Model of biological evolution with threshold dynamics and infinitely many absorbing states *Phys. Rev.* E **60** 1516–9
[11] Hoshen J and Kopelman R 1976 Percolation and cluster distribution: I. Cluster multiple labeling technique and critical concentration algorithm *Phys. Rev.* B **14** 3438–45
[12] Babalievski F 1998 Cluster counting: the Hoshen–Kopelman algorithm versus spanning tree approaches *Int. J. Modern Phys.* C **9** 43–60
[13] Newman M E J and Ziff R M 2001 Fast Monte Carlo algorithm for site and bond percolation *Phys. Rev.* E **64** 016706
[14] Constantin J M, Berry M W and VanderZanden B T 1997 Parallelization of the Hoshen–Kopelman algorithm using a finite state machine *Int. J. Supercomp. Appl. High Perform Comp.* **11** 34–48
[15] Teuler J M and Gimel J C 2000 A direct parallel implementation of the Koshen-Kopelman algorithm for distributed memory architectures *Comp. Phys. Commun.* **130** 118–29
[16] Tarjan R E 1972 Depth-first search and linear graph algorithms *SIAM J. Comput.* **1** 146–60
[17] Roux S and Hansen A 1987 A new algorithm to extract the backbone in a random resistor network *J. Phys. A: Math. Gen.* **20** L1281–5
[18] Herrmann H J, Hong D C and Stanley H E 1984 Backbone and elastic backbone of percolation clusters obtained by the new method of burning *J. Phys. A: Math. Gen.* **17** L261–6
[19] Moukarzel C 1998 A fast algorithm for backbones *Int. J. Mod. Phys.* C **9** 887–90
[20] Ziff R M, Cummings P T and Stell G 1984 Generation of percolation cluster perimeters by a random-walk *J. Phys. A: Math. Gen.* **17** 3009–17
[21] Grassberger P 1992 Spreading and backbone dimensions of 2D percolation *J. Phys. A: Math. Gen.* **25** 5475–84
[22] Yin W G and Tao R B 2003 Rapid algorithm for identifying backbones in the two-dimensional percolation model *Int. J. Mod. Phys.* C **14** 1427–37
[23] Martin-Herrero J 2004 Hybrid cluster identification *J. Phys. A: Math. Gen.* **37** 9377–86
[24] Fleischer L K, Hendrickson B and Pinar A 2000 On identifying strongly connected components in parallel *Lect. Notes Comput. Sci* **1800** 505–11
[25] Kirkpatrick S 1978 *Electrical Transport and Optical Properties of Inhomogeneous Media* ed Garland J C and Tanner D B (New York: AIP) p 99
[26] Zienkiewicz O C 1974 *The Finite Element Method* 3rd edn (London: McGraw-Hill)
[27] Batrouni G G, Hansen A and Roux S Negative moments of the current spectrum in the random-resistor network *Phys. Rev.* A **38** 3820–3
[28] Mackerle J 2003 FEM and BEM parallel processing: theory and applications—a bibliography (1996–2002) *Eng. Comput.* **20** 436–84
[29] Barthélémy M, Buldyrev S V, Havlin S and Stanley H E 1999 Scaling for the critical percolation backbone *Phys. Rev.* E **60** R1123–5
[30] Grassberger P 1999 Conductivity exponent and backbone dimension in 2-d percolation *Physica* A **262** 251–63
[31] Coniglio A 1989 Fractal structure of Ising and Potts clusters—exact results *Phys. Rev. Lett.* **62** 3054–7
[32] Lobb C J and Frank D J 1984 Percolative conduction and the Alexander–Orbach conjecture in 2 dimensions *Phys. Rev.* B **30** 4090–2
[33] Normand J M, Herrmann H J and Hajjar M 1988 Precise calculation of the dynamical exponent of two-dimensional percolation *J. Stat. Phys.* **52** 441–6
[34] Yoon S and Lee S I 1990 Possible breakdown of the universality of the conductivity critical exponent in an anisotropic percolation system *Physica* B **167** 133–7
[35] Huang G, Tsai W K and Lu W 1987 Fast parallel equation solvers based on textured decomposition *Proc. 26th Conf. on Decision and Control (Los Angeles, CA)* pp 1450–4