Continuum percolation of discs by a cluster growth method on random lattices

1987 J. Phys. A: Math. Gen. 20 6053

(http://iopscience.iop.org/0305-4470/20/17/038)

J. Phys. A: Math. Gen. **20** (1987) 6053-6057. Printed in the UK

Continuum percolation of discs by a cluster growth method on random lattices

J F McCarthy

Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, UK

Received 5 May 1987

Abstract. We simulate the continuum percolation of discs in two dimensions and calculate estimates for the threshold parameter and some critical exponents. We use a cluster growth method which relies on our recent discovery that the continuum percolation of discs can be formulated as a bond percolation problem on a planar triangulated random lattice. The algorithm is very efficient.

1. Introduction

In a recent letter [1] we showed that the continuum percolation of discs in two dimensions is equivalent to a bond percolation problem on a random lattice. This raised the possibility of doing computer simulations of continuum percolation using techniques which have been developed for lattice structures. In particular, since nearest neighbours are well defined on the random lattice, we could use cluster growth methods. In this paper we develop an algorithm for growing continuum percolation clusters at threshold using the technique of invasion percolation on random lattices [2]. We report the results of our computations of the threshold parameter and critical exponents obtained using this new algorithm.

2. The continuum percolation of discs and the random lattice

The random lattice which we will be discussing in this paper is of the type described by us in [3]. The sites of the lattice are points distributed randomly in the plane and correspond to the centres of the discs in the continuum percolation problem. The sites are linked up to form triangles according to the criterion that the circumcircle of any three sites forming a triangle in the lattice has no other site inside it. It is known that this can be done in such a way that the triangles fill the plane with no overlap. The dual of the random lattice is a lattice consisting of Voronoi polygons. The geometrical properties of these random lattices have been thoroughly researched.

The problem of the continuum percolation of discs involves placing discs of equal radius at random in the plane. If two discs overlap they are said to be in the same cluster and the problem is to study the distribution of these clusters and to find the critical density at which an infinite cluster is formed through the system. We showed in [1] that this is equivalent to the bond problem on a random lattice in which all bonds with length less than or equal to the diameter of the discs are coloured red and two sites are defined to be in the same cluster if they are linked either by a red bond

0305-4470/87/176053+05$02.50

or by a connected path of red bonds. Hence it is possible to study the continuum percolation of discs by doing simulations on an ensemble on random lattices.

3. Invasion percolation on random lattices

In our recent paper [2] we discussed the simulation of percolation of random lattices. We decided that cluster growth methods (see, for example, [4, 5]) were probably the best to use since they are not very dependent on lattice structure. Actually, we used the technique of invasion percolation to compute the critical points and critical exponents of uncorrelated site and bond percolation on random lattices. We refer you to that paper [2] for details of the method.

Briefly, invasion percolation clusters for uncorrelated bond percolation can be grown by the following algorithm.

(1) Choose a bond to be the seed of the cluster.

(2) Assign random numbers between 0 and 1 to the nearest-neighbour bonds of the seed. These bonds form the boundary of the cluster to start with.

(3) Find the bond on the boundary which has the smallest random number, r, assigned to it.

(4) Accept that bond into the cluster and assign random numbers to any of its nearest neighbours which are empty, thereby adding them onto the boundary of the cluster.

(5) Go to step (3).

It is clear that the invasion percolation clusters are never constrained to terminate and can be grown to any size. It is not so clear, but nevertheless seems to be the case, that these clusters bear an interesting relationship to the clusters of ordinary percolation at threshold which allows invasion percolation to be used to study the critical behaviour of the ordinary percolation model. Although most of the rigorous evidence for this relationship is based on analyses of uncorrelated site percolation on a two-dimensional square lattice [6] we assume in this paper that it holds equally well for a random lattice and for the bond problem discussed earlier which is different from an uncorrelated bond problem. Certainly the results which we obtain seem to validate the use of the invasion percolation technique. Invasion percolation is used to compute an estimate for the critical density, χ_c, the fractal dimension, D, of critical clusters and the gap exponent, $\Delta \equiv \beta + \gamma$, of ordinary percolation at threshold.

4. Growing continuum percolation clusters at threshold

When we performed our simulation of uncorrelated bond percolation we used a periodic two-dimensional random lattice of 10 000 sites and, starting at a bond near the centre, grew invasion percolation clusters out into repetitions of the basic lattice. Different clusters were the result of different assignments of random numbers to the lattice.

The bond problem which we wish to discuss in this paper is slightly different from the uncorrelated bond problem. If you refer back to the algorithm which we gave for growing invasion percolation clusters in the uncorrelated bond problem, then the algorithm for the bond problem of interest would be obtained by replacing steps (2)-(5) by the steps (2′)-(5′) given below.

(2′) Calculate the lengths of the nearest-neighbour bonds of the seed. These bonds form the boundary of the cluster to start with.

(3′) Find the bond on the boundary which has the smallest length.

(4′) Accept that bond into the cluster and calculate the lengths of any of its neighbours which are not already on the boundary, thereby adding them onto the boundary of the cluster.

(5′) Go to step (3′).

The clusters grown using this algorithm depend only on the geometry of the random lattice—there is no independent probability distribution imposed on the bonds. Hence the results are obtained by averaging over an ensemble of random lattices. To grow, say, 1000 clusters by the methods used in [2] it would be necessary to make 1000 random lattices with different realisations of the random sites. This would be quite time consuming and the size of the clusters grown would be limited by the size of the lattices.

Fortunately we have been able to get around this problem by finding an algorithm in which the bonds of the random lattice and the invasion percolation cluster are constructed in a very similar manner, starting at a central point and spiralling outwards, so that the two processes can be superimposed. To understand this it is necessary to consider how to construct a random lattice in two dimensions.

The problem is, given a set of N points distributed at random in a square Ω, to link them up into a lattice of triangles according to the prescription that the circumcircle of any three points forming a triangle in the lattice has no other point inside it. A simple sequential algorithm which we devised for doing this is given below.

Choose a central point and join it to its nearest neighbour. These points can be labelled 1 and 2 respectively and form the first link of the lattice. Find the closest point to link 12 by finding the point to which the largest angle is subtended. Label this point 3. Form links 13 and 23. Now consider the link 13. It divides the plane into two parts. On one side we have already formed a triangle, i.e. $\triangle 123$. We must now form the triangle on the other side by finding the closest point to link 13 on that side. This will be point 4. Now consider link 14 and find its triangle point on the side which does not contain point 3. Circle around point 1 in this way until the new triangle point for link $1n$ takes us back to point 2. Then we have completely enclosed point 1 with triangles.

Now we will complete the circle of triangles around point 2. To do this we first take link $2n$ which was the last link formed. We know that it is contained in $\triangle 12n$. Find the triangle point on the other side and label it $n+1$. Consider link $2(n+1)$ and keep circling around point 2 until we get back to point 3. Then circle around point 3, returning to point 4, etc. The process ends when the Nth point has been circled around.

We have defined a process in which one triangle of the lattice is formed near the centre and is built upon in a spiralling manner to encompass all the points of the lattice. This algorithm can be combined with the algorithm for growing invasion percolation clusters. A very useful extra ingredient is the method of overlaying an 'imaginary covering mesh' onto the area Ω, each mesh cell being square and of unit area. We quote the method from [7].

'For each cell select a random number ξ between 0 and 1. Determine an integer D by requiring

$$\sum_{k=0}^{D+1} P(\chi, k) > \xi > \sum_{k=0}^{D} P(\chi, k)$$

where $P(\chi, k) = e^{-\chi}\chi^k/k!$ in the Poisson distribution. Then place D points in the

current cell, with their positions within the cell chosen randomly. Note that the statistics of the spatial distribution of discs thus obtained are identical to the statistics of a distribution of discs whose positions are chosen to be anywhere in the area Ω, and hence the imaginary covering mesh does not distort the continuum nature of the simulation.'

With this extra ingredient it is possible to grow continuum percolation clusters at threshold from 'next to nothing'.

Start with the central nine cells of the imaginary covering mesh and fill them with random points according to the method outlined above (the average number of points in any cell is an adjustable parameter and we found that a sufficient number was 15). Choose a point in the central cell and circle around it to find its nearest-neighbour bonds in the random lattice. These bonds form the boundary of the invasion percolation cluster to start with. Find the bond with the smallest length and accept its free endpoint into the cluster (it may happen at some stage that both endpoints of the bond have already been accepted into the cluster in which case the next step can be skipped). Determine which cell the new cluster site is in. Call it the current cell. If necessary, make some more cells of the imaginary covering mesh so that the current cell is completely surrounded. Circle around the new cluster site in the current cell finding any of its nearest neighbours which are not already on the boundary, thereby adding them onto the boundary of the cluster. Continue the process.

Using this algorithm it is possible to grow critical continuum percolation clusters of any size. The algorithm is efficient both in speed and in storage because the random lattice and the cluster are constructed at the same time and because only as much of the random lattice is constructed as is required.

5. Discussion of results

We grew 1200 clusters of size 10 000 sites. All computations were done on the ICL DAP at Queen Mary College, London. The time required to grow one cluster was 525 DAP seconds. (No doubt the DAP was a poor choice of machine to use for this problem since the algorithm is essentially sequential and the DAP is quite slow in computing sequential steps, gaining efficiency only if the algorithm is arranged so that many elements are processed in parallel. However, it was a convenient machine to use for other reasons.)

Our results are presented in table 1. The normalised threshold parameter, χ_c, which we compute, is the number of points per unit area (as in [7]). The lower end of the range of n which we used in calculating critical exponents was given by $n_{\min} = 3000$ (see [5]). The errors quoted are one standard deviation statistical errors estimated by dividing the data into ten groups and observing the standard deviation between the ten sets of results.

Table 1. Results of computer simulations of the continuum percolation of discs.

	χ_c	$1/\Delta$	$1/D$
Present work	0.720 ± 0.004	0.378 ± 0.028	0.529 ± 0.014
Gawlinski and Stanley [7]	0.718 ± 0.003	0.389 ± 0.009	0.523 ± 0.020
Lattice values		0.3956	0.5275

In table 1 we compare our results with those obtained by Gawlinski and Stanley [7] and with the generally accepted values for the critical exponents on regular lattices. The agreement is excellent.

In summary, we have devised an algorithm for growing continuum percolation clusters at threshold in two dimensions. We have used our new algorithm to compute estimates for the critical density and critical exponents of the continuum percolation of discs. The results obtained are competitive with those obtained using other methods and provide more evidence for the conjecture that the problems of continuum percolation and percolation on lattices have the same critical exponents.

Note added. I would like to draw the reader's attention to the work of Kerstein [8, 9] on the dual problem of 'void' percolation in terms of Voronoi networks.

References

[1] McCarthy J F 1987 *Phys. Rev. Lett.* **58** 2242
[2] McCarthy J F 1987 *J. Phys. A: Math. Gen.* **20** 3465-9
[3] McCarthy J F 1986 *Nucl. Phys.* B **275** [FS 17] 421
[4] Leath P L 1976 *Phys. Rev.* B **14** 5046-55
[5] Wilkinson D and Barsony M 1984 *J. Phys. A: Math. Gen.* **17** L129-35
[6] Chayes J T, Chayes L and Newman C M 1985 *Commun. Math. Phys.* **101** 383-407
[7] Gawlinski E T and Stanley H E 1981 *J. Phys. A: Math. Gen.* **14** L291-9
[8] Kerstein A R 1983 *J. Phys. A: Math. Gen.* **16** 3071-5
[9] Elam W T, Kerstein A R and Rehr J J 1984 *Phys. Rev. Lett.* **52** 1516-9