Application of 'logical transport' to determine the directed and isotropic percolation thresholds



1987 J. Phys. A: Math. Gen. 20 L873

(http://iopscience.iop.org/0305-4470/20/13/011)

J. Phys. A: Math. Gen. 20 (1987) L873-L878. Printed in the UK

LETTER TO THE EDITOR

# Application of 'logical transport' to determine the directed and isotropic percolation thresholds

Alex Hansen† and Stéphane Roux‡

† Groupe de Physique des Solides de l'Ecole Normale Supérieure, 24 rue Lhomond, F-75231 Paris Cedex 05, France

‡ Laboratoire d'Hydrodynamique et Mécanique Physique de l'Ecole Supérieure de Physique et Chimie Industrielles, 10 rue Vauquelin, F-75231 Paris Cedex 05, France

Received 10 April 1987, in final form 19 June 1987

**Abstract.** We present a Monte Carlo algorithm based on 'logical transport' which determines the effective isotropic percolation and directed percolation thresholds of finite-size lattices. We apply the algorithm to bond and site directed percolation in two dimensions and determine the asymptotic percolation thresholds by finite-size scaling.

There already exist several numerical and analytical methods for determining the percolation threshold, $p_c$, on various lattices [1]. In this letter we present yet another Monte Carlo method, which is rather different in spirit from previous methods, even though related to the invasion percolation method of Wilkinson and Barsony [2] and which is numerically very efficient for direct percolation [3]. The method consists of an algorithm that determines the effective percolation threshold for each lattice in a given ensemble of lattices of a given size. By finite-size scaling [1] $p_c$ is then found. We use this method to estimate $p_c$ in directed site percolation on a square lattice to be

$$p_c = 0.7056 \pm 0.0002 \tag{1}$$

which is consistent with the previous series expansion estimate of De'Bell and Essam [4], $p_c = 0.7055(1)$. For directed bond percolation we obtain

$$p_c = 0.6448 \pm 0.0002 \tag{2}$$

which is consistent with the series expansion estimate of Essam *et al* [5], $p_c = 0.644\,701(2)$.

The idea underlying the Monte Carlo method we present here is that of 'Boolean transport'. All the static properties of percolation clusters in the thermodynamic limit are properties of the 'connectedness' of the clusters. This connectedness is best described by logical variables, i.e. variables that are either 'true' or 'false' as will be described in the following paragraphs where we adopt a 'logical Green function' approach. In this context, the static properties may be viewed as transport properties of logical variables, just as the dynamical properties may be viewed as transport of scalar (e.g. conductivity) or vector (e.g. elasticity) variables, and the 'logical Green functions' exhibit all the properties of 'ordinary' Green functions. The application of logical variables to study the geometrical properties of ordinary connectivity percolation is very natural and efficient. It may turn out that this formalism also is able to provide an algorithm for determining the geometric properties of such complicated systems as

  

the elastic central-force network near the rigidity threshold [6]. Not all readers will wish to go through the rather formal development of the method that follows. These readers may skip directly to equation (8).

Suppose we have a lattice where the sites are occupied with probability $p$ and empty with probability $1-p$. We define a path of length 1 in this lattice to be a connection between two occupied nearest-neighbour sites. This is now the standard site percolation problem. We introduce two *Boolean* matrices $A_{ij}$ and $D_{ij}$ where $i$ and $j$ refer to sites $i$ and $j$ on the given lattice. The diagonal element $D_{ii}$ is 1 (i.e. 'true') if site $i$ is occupied, and 0 ('false') if it is empty, and all off-diagonal elements $D_{ij}$ are zero. The off-diagonal element $A_{ij}$ is 1 if $i$ and $j$ both are occupied and if they are nearest neighbours, otherwise $A_{ij}=0$. All diagonal elements $A_{ii}$ are zero. A non-zero ('true') element of $A_{ij}$ corresponds to a connected path of length 1 on the lattice. The two matrices $A_{ij}$ and $D_{ij}$ completely specify the cluster structure of the original lattice without containing any redundant information apart from the symmetry of the matrix that results from the symmetry of the connections. In the 'diode' network of Redner [7], if an occupied site $i$ is connected to another occupied site $j$ by a path of length 1, $j$ will *not* necessarily be connected to $i$. In this case, $A_{ij}$ will not be symmetric.

A connected path of length 2 between sites $i$ and $j$ through their common nearest-neighbour site $k$ exists if and only if ($A_{ik}$ and $A_{kj}$) = 1, where 'and' is the logical 'and' operation. Now, suppose $i$ and $j$ have two common nearest-neighbour sites $k$ and $l$. Then there is a connected path between $i$ and $j$ of length 2 if ($A_{ik}$ and $A_{kj}$) or ($A_{il}$ and $A_{lj}$) = 1, where 'or' is the logical 'or' operation. We explain this expression in words. There is a path of length 2 between $i$ and $j$ if there are paths of length 1 between $i$ and $k$ and $k$ and $j$, or there are paths of length 1 between $i$ and $l$ and $l$ and $j$. With this in mind, we define the multiplication and addition rules for the Boolean matrices as follows: suppose we have two Boolean matrices $B$ and $C$. Then define the matrix product as $(B\otimes C)_{ij}=(B_{i1}$ and $C_{1j})$ or $(B_{i2}$ and $C_{2j})$ or ... or $(B_{iN}$ and $C_{Nj})$, where $N$ is the size of the matrices. Furthermore, in light of this definition, it is natural to define the matrix addition as $(B\oplus C)_{ij}=B_{ij}$ or $C_{ij}$ (where there is no summation over repeated indices). Now, with these definitions, all off-diagonal and non-zero elements of the matrix $A^2=A\otimes A$ represent paths of length 2 on the original lattice, and the matrix $D\oplus A\oplus A^2$ represents all connected paths of length 0, 1 or 2. In general $D\oplus A\oplus A^2\oplus A^3\oplus\ldots\oplus A^n$ contains all connected paths of lengths from 0 to $n$. Thus, a Boolean connectedness (or 'logical') Green function can now be defined as

$$G=D\oplus\sum_{k=1}^{\infty}A^k \tag{3}$$

where $G_{ij}$ is 1 if sites $i$ and $j$ belong to the same cluster, and zero otherwise. It may be amusing to note that the Green function in (3) obeys a Dyson equation of the form $G=D\oplus G\otimes A$. This follows trivially from (3). The symbol $\Sigma$ means as usual the repeated use of the $\oplus$ operation defined above. Suppose our finite lattice is part of an infinite lattice. Let Greek indices indicate the sites on the edge of the finite lattice, and let Latin indices indicate, as before, any site on the finite lattice. Suppose furthermore that we know which sites on the edges of the finite lattice belong to the infinite cluster. This information is contained in the Boolean function $P_\alpha$ which is 1 if $\alpha$, which is a border site, belongs to the infinite cluster, and 0 otherwise. Then

$$P_i=\mathrm{OR}_\alpha\, G_{i\alpha}P_\alpha \tag{4}$$

will be either 1 if $i$ belongs to the infinite cluster, or 0 otherwise. The symbol $\mathrm{OR}_\alpha$ means that the logical 'or' operation is done between all sites $\alpha$. We now give a different, and for our purpose more convenient, representation of the Green function, which is analogous to the path integral representation of 'standard' Green functions. Suppose $\pi(i, j)$ is a given path between sites $i$ and $j$. Then, $G$ may be written

$$G_{ij} = \mathrm{OR}_{\{\pi(i,j)\}}(\mathrm{AND}_{k\in\pi(i,j)} D_{kk}) \tag{5}$$

(the information contained in matrix $A$ is now hidden in the definition of paths $\pi(i, j)$) where the logical 'or' operation is taken between all possible paths between $i$ and $j$, and the logical 'and' operation is taken between all sites along each given path $\pi$.

To generate the finite lattice and the matrix $A$, we associate in the usual way a random number $r_i$ to each site $i$ [1]. Thus, if $r_i < p$, site $i$ is occupied, otherwise it is empty. The matrix $D$ is then given by

$$D_{ii} = (r_i < p) \tag{6}$$

and

$$A_{ij} = (r_i < p) \text{ and } (r_j < p) \tag{6'}$$

if $i$ and $j$ are nearest neighbours, zero otherwise. Combining (5) and (6), we get

$$G_{ij} = \mathrm{OR}_{\{\pi(i,j)\}}(\mathrm{AND}_{k\in\pi(i,j)}(r_k < p)). \tag{7}$$

We notice that it is the *largest* $r_k$ along a given path $\pi(i, j)$ that determines whether this path is connected or not; if this $r_k$ is less than $p$, all the others will also be less than $p$, and the path is connected. Furthermore, it is the *smallest* of the largest $r_k$ from each path that determines whether $i$ and $j$ are connected or not; if the smallest of the largest $r_k$ is less than $p$, then there is a connected path between $i$ and $j$. This leads to the following equation for determining the effective percolation threshold between sites $i$ and $j$. In (7) let us turn 'and' and 'or' operations into 'max' and 'min', to get

$$p_{\mathrm{eff}}(i, j) = \mathrm{MIN}_{\{\pi(i,j)\}}(\mathrm{MAX}_{k\in\pi(i,j)} r_k). \tag{8}$$

This equation forms the basis for our algorithm for calculating the percolation threshold $p_c$. The meaning of $p_{\mathrm{eff}}(i, j)$ in terms of the connectedness Green function defined in (3) is that if $p < p_{\mathrm{eff}}(i, j)$, $G_{ij} = 0$, and if $p > p_{\mathrm{eff}}(i, j)$, $G_{ij} = 1$, i.e. there is a connection between $i$ and $j$ along occupied bonds only if $p > p_c$. A more physical way of visualising the contents of this equation is to think of each assigned random number as the height of an obstacle at each node. Then the barrier of a given path connecting $i$ and $j$ is defined as the height of the highest obstacle along this path, and thus the effective threshold $p_{\mathrm{eff}}(i, j)$ may be interpreted as the height of the smallest barrier along any path between nodes $i$ and $j$. Now, if the distance between $i$ and $j$ goes to infinity, $p_{\mathrm{eff}}(i, j)$ will approach $p_c$.

In order to compute the effective percolation threshld for a given finite-size lattice, one can implement (8) through a transfer-matrix algorithm [8] to be explained below for the directed percolation case. In the present isotropic case, this method will require $N^2$ operations (and in the directed case, $N$ operations) where $N$ is the number of nodes in the lattice. This is as fast as one can do in order to exactly determine the effective threshold for a given lattice. However, to average the effective threshold over several lattices and then extrapolate to infinite-size lattices in order to determine $p_c$ is probably too expensive in terns of computer time when compared with other numerical methods [1]; there is more information in the effective threshold of a given lattice than

we need for determining $p_c$. In directed percolation, however, the determination of $p_{\text{eff}}(L)$ for a given lattice is very fast, and $p_c$ may be determined from an average over effective thresholds.

In a square lattice let us introduce a preferred direction (e.g. a 'time' axis) along one of the diagonals. Now, the directed percolation problem differs from the usual, isotropic one we have discussed so far by defining occupied nearest-neighbour sites to be connected by a path of length 1 only in the preferred direction of increasing 'time'. In terms of the matrix $A$, this can be expressed in the following way. If $j$ has a smaller 'time' coordinate than $i$, then $A_{ji}=0$ no matter what $A_{ij}$ is. This leads to (4) being no longer a boundary-value problem as in the isotropic case, but an *initial value* problem. Let us at this point change the notation for the lattice sites: a site is defined by a 'time' and a coordinate $i$ along one of the axes. Thus, (4) now becomes

$$P_i(T)=\text{OR}_j G_{ij}(T,t)P_j(t) \tag{9}$$

where $T>t$ are the 'time' coordinates. Using (7) and (9), we may write

$$\begin{aligned} P_i(t+1) &= \{P_i(t) \text{ and } (r_i(t+1)<p)\} \text{ or } \{P_{i+1}(t) \text{ and } (r_i(t+1)<p)\} \\ &= \{P_i(t) \text{ or } P_{i+1}(t)\} \text{ and } (r_i(t+1)<p). \end{aligned} \tag{10}$$

If we in this equation substitute 'and' by 'max' and 'or' by 'min' as done in (8), we get

$$\begin{aligned} p_{\text{eff}}(i,t+1) &= \min(\max(p_{\text{eff}}(i,t), r_i(t+1)), \max(p_{\text{eff}}(i+1,t), r_i(t+1))) \\ &= \max(\min(p_{\text{eff}}(i,t), p_{\text{eff}}(i+1,t)), r_i(t+1)). \end{aligned} \tag{11}$$

We interpret the contents of this equation in terms of the notion of barriers introduced after (8). $p_{\text{eff}}(i,t)$ may be interpreted as the minimum barrier along any path connecting the lower boundary to the node $(i,t)$ forwards in time. Then (11) states that the barrier to overcome to reach node $(i,t+1)$ is equal to the maximum height of the obstacle at node $(i,t+1)$ and the smallest of the barriers to overcome to get to either node $(i,t)$ or node $(i+1,t)$ from the lower boundary. By the same arguments as those given after (8), we find that $p_{\text{eff}}(i,t)$ approaches $p_c$ as $t\to\infty$ (if the lattice is wide enough). This equation is easily and efficiently implemented on a computer.

In our computer runs in order to determine the percolation threshold for directed site percolation on square lattices we used a (square) lattice of size $L\times L$ oriented so that the preferred direction was along the diagonals. The boundary conditions were periodic in the direction orthogonal to the 'time' axis and non-periodic in the direction parallel to it. Equation (11) was integrated from the lower ($t=1$) to the upper ($t=L$) edge of the lattice. As initial conditions we set $p_{\text{eff}}(i,t=1)=0$. The effective directed percolation threshold for the given lattice is then given by $\min(p_{\text{eff}}(i,t=L))$.

Next, this effective threshold is averaged over many lattices, and we obtain a $p_{\text{eff}}(L)$. In the corresponding directed *bond* problem, (11) is changed into

$$p_{\text{eff}}(i,t+1)=\min(\max(p_{\text{eff}}(i,t), u_i(t+1)), \max(p_{\text{eff}}(i+1,t), v_i(t+1))) \tag{12}$$

where $u_i(t+1)$ is a random number between 0 and 1 associated with the link between sites $(i,t)$ and $(i,t+1)$ and $v_i(t+1)$ is another random number associated with the link between sites $(i+1,t)$ and $(i,t+1)$. The determination of $p_{\text{eff}}(L)$ from this is identical to that of the site problem. However, in terms of computer time, the site problem is almost twice as fast as the bond problem since most of the CPU time is spent generating random numbers (i.e. 70% of the CPU time when using the built-in random number generator of an FPS-164 for the site problem), and the bond problem needs twice as many random numbers per lattice as the site problem.

The lattices we generated and included in our analysis ranged in size from $L = 10$ to 1000 on three different computers: an FPS-164 (with a speed of 12 $\mu$s per site in the site problem), a Cyber 76 (with a speed of 18 $\mu$s per site) and an IBM 3090. The averaged $p_{\text{eff}}(L)$ and the corresponding standard deviations for the site problem are listed in table 1. To analyse the data, we made the finite-size scaling assumption

$$p_{\text{eff}}(L) = p_c - aL^{-1/\nu_\parallel} + bL^{-1/\nu_\perp} \tag{13}$$

where $\nu_\parallel$ is the correlation length exponent in the direction parallel to the 'time' axis, and $\nu_\perp$ is the orthogonal correlation length exponent. Essam *et al* [5] have calculated these two exponents by series expansion methods to be $\nu_\parallel = 1.7334(5)$ and $\nu_\perp = 1.0972(4)$. (We have assumed that these exponents are equal for the site and bond problem.) Given these exponents we fitted (13) to the data of table 1 by a least-squares method where we choose a prefactor $b$ and then determine which prefactor $a$ and constant term $p_c$ minimise $\chi^2$. Then we choose $b$ which gives the smallest minimum $\chi^2$. Exponents other than $1/\nu_\perp$ such as $2/\nu_\parallel$, and 1 were tried for the second correction-to-scaling term, but $1/\nu_\perp$ gave clearly the best result. For the site problem we found the value $p_c = 0.7056(2)$, quoted in (1); for the bond problem we found $p_c = 0.6448(2)$ as quoted in (2). These best-fit curves are shown in figure 1.

**Table 1.** $p_{\text{eff}}(L)$ and its standard deviation for site directed percolation in two dimensions for some of the sizes generated. The number of lattices generated for each $L$ is indicated.

| $L$ | Realisations | Effective thresholds | Standard deviation |
|---|---|---|---|
| 10 | 1000 | 0.5041 | 0.0973 |
| 20 | 1000 | 0.5559 | 0.0599 |
| 30 | 1000 | 0.5830 | 0.0463 |
| 40 | 1000 | 0.5986 | 0.0397 |
| 50 | 1000 | 0.6097 | 0.0323 |
| 60 | 1000 | 0.6160 | 0.0289 |
| 70 | 1000 | 0.6230 | 0.0267 |
| 80 | 1000 | 0.6270 | 0.0237 |
| 90 | 1000 | 0.6317 | 0.0211 |
| 100 | 1000 | 0.6364 | 0.0209 |
| 200 | 1000 | 0.6568 | 0.0120 |
| 300 | 1000 | 0.6651 | 0.0098 |
| 400 | 1000 | 0.6719 | 0.0082 |
| 500 | 1000 | 0.6784 | 0.0064 |
| 600 | 1000 | 0.6775 | 0.0048 |
| 700 | 1000 | 0.6818 | 0.0037 |
| 800 | 500 | 0.6829 | 0.0060 |
| 900 | 500 | 0.6808 | 0.0047 |
| 1000 | 500 | 0.6843 | 0.0054 |

A similar use of the 'Boolean transport' introduced in the first part of this letter also proves to be an efficient way to compute geometric critical exponents such as $\beta$ for directed percolation [9].

We would like to thank E Guyon, D Stauffer and J Vannimenus for many useful discussions on this subject. We would also like to thank T Jøssang and M Novotny for their assistance in obtaining computer time at Bergen Scientific Center and D

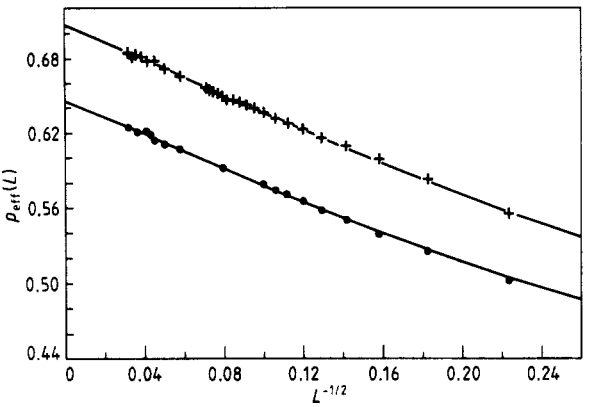


**Figure 1.** $p_{eff}(L)$ as a function of $L$ for the site (+) and bond (●) directed percolation problem. The full curves are the best fits for the two data sets.

Stauffer for his hospitality at the University of Cologne where further computations were done. Most of the computations were done at the Ecole Normale Supérieure on an FPS-164 computer supported by GRECO 70 (Expértimentation Numérique). AH acknowledges support from CEN-Saclay through a Joliot-Curie Fellowship, and SR is supported by the Ecole Nationale de Ponts et Chaussées.

## References

[1] D Stauffer 1985 *Introduction to Percolation Theory* (London: Taylor and Francis)
[2] Wilkinson D and Barsony M 1984 *J. Phys. A: Math. Gen.* **17** L129
[3] Kinzel W 1982 *Percolation Structures and Processes* ed G Deutcher, R Zallen and J Adler (Bristol: Adam Hilger)
[4] De'Bell K and Essam J W 1983 *J. Phys. A: Math. Gen.* **16** 385
[5] Essam J W, De'Bell K, Adler J and Bhatti F M 1986 *Phys. Rev.* B **33** 1982
[6] Day A R, Tremblay R R and Tremblay A-M S 1986 *Phys. Rev. Lett.* **56** 2501
[7] Redner S 1982 *Phys. Rev.* B **25** 3242
[8] Vannimenus J and Nadal J-P 1984 *Phys. Rep.* **103** 47
[9] Roux S 1987 *Eur. J. Phys.* **8** 186