Diffusion of pairs in random lattices
O. Päetzold

To cite this version:

HAL Id: jpa-00232329
https://hal.archives-ouvertes.fr/jpa-00232329
Submitted on 1 Jan 1984

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Diffusion of pairs in random lattices

O. Päetzold
Breddestr. 13, D-4390 Gladbeck, F.R.G.

(Reçu le 4 octobre 1983, accepté le 9 janvier 1984)

Résumé. — La marche aléatoire de paires de particules dans un réseau de percolation désordonné est simulée par ordinateur. On s'intéresse particulièrement au comportement de l'exposant $k$ au seuil de percolation. Cet exposant est approximativement le même que dans le cas d'une particule isolée.

Abstract. — The random walk of pairs of particles in a disordered percolation lattice is simulated on a computer. The main emphasis is on the behaviour of the exponent $k$ at the percolation threshold. This exponent is about the same as that for the single particle.

In the « ant in the labyrinth » problem [1], many people [2-10] have investigated the random walk of single particles, i.e., movements, randomly made by particles, which are allowed to move on occupied sites only. Now I examined the problem for the case of pair movements by computer simulation. I was interested in the case of rigidly-coupled pairs because I wanted to know if kinematic effects, not existing in the case of single particles, have an influence on the asymptotic critical exponent. An exact solution of a similar problem, the diffusion of pairs in an ideal lattice gas, and an analytic solution for the walk in one dimension can be found in reference [11].

My inquiry covered the walk in two and three dimensions. In the following text the case of cubic lattices is often mentioned just after the simpler case of the two dimensional walk by using parentheses like the following ones: \{ ... \}.

The pairs must obey the same laws of the random walk as single particles have to. The main rule is that a particle is restricted to move on occupied sites or bonds. These sites or bonds are occupied randomly with the probability $p$, with $p$ ranging from 0 to 1.

If the pair tries to walk in a random direction, it has to inquire whether the destination bond is a prohibited or non-prohibited one. In case of a move, the direction vector is stored and added to the others (cf. later passages in this paper). After every try, no matter whether a move was made or not, the time increases by one unit. The distance $R$ and the time $t$ are related by [4, 12, 13]

$$R \propto t^k \quad (t \to \infty).$$

(1)
In this equation \( R \) is the square root of the averaged square end-to-end distance vector, measured in units of the lattice spacings. This vector connects the centre of the bond at starting position with the centre of the bond at the position after the bond has made \( t \) step attempts. I chose this kind of representation of the walked distance because of its mathematical simplicity. \( R \) is calculated with different starting positions on many different labyrinths at the same probability \( p \). \( t \) is the symbol for the time needed. It is measured in the number of step attempts the particle does and is an equivalent to the computing time.

The most important fact for the pair is that its two members never can be separated. They are bound together forever. So the difference to the single ant problem can be seen easily:

Whereas the single ant has the possibility to walk in every direction it wants, a partner in the pair is always forced to follow its companion in the direction it walks. The shape of the pair looks as follows:

\[
A = == = A.
\]

Between the ants a bond \((====)\) can be seen. Its task is to prevent the ants from wandering into their companions direction. Any other directions, whether \(x-, y-\), or \(z\)-direction of the lattice, are allowed to be taken. So in case of two dimensions every ant is able to walk in three directions, thus together the pair has the possibility to walk in six directions in contrast to the four directions of a single ant. If you look at three dimensions, you can recognize that every ant gets two more directions to walk into. Thus, a pair has the ability to walk into ten directions. The structure of the pair, together with the directions it is able to move into, looks as follows:

2 dimensional

\[
\begin{array}{c}
\text{2 dimensional} \\
\hline
\text{A} = == = A
\end{array}
\]

3 dimensional

\[
\begin{array}{c}
\text{3 dimensional} \\
\hline
\text{A} = == = A
\end{array}
\]

The bond can be oriented along \(x-\) or \(y-\) \{ or \(z\) \}-direction, at which always all six \{ ten \} moves are possible, if the neighbours are occupied. If a direction is chosen, one ant walks one unit distance into this direction. Then its companion leaves its place and moves onto the previous site of the first ant. The following picture illustrates three of the six possibilities in the square lattice:

\[
\begin{array}{c}
\text{\cdot \cdot \cdot \cdot \cdot \cdot} \\
\text{\cdot A---\cdot a \cdot} \quad \Rightarrow \quad \text{\cdot a \cdot or \ A-a \cdot or \ a \cdot etc.}
\end{array}
\]

To execute the walk in two and three dimensions, I used a Monte Carlo simulation of the diffusion on a CDC Cyber 76 computer. The general techniques of a Monte Carlo simulation program are described in reference [14].

There are two possibilities of creating the Fortran program of the pair-movement. The first one covers site percolation. That means that the sites of the lattice are stored as for the case of single ants. Then the program checks, if a site and its neighbour site are occupied. If that is true, a memory with the number of the bond is stored as 1, else as 0, meaning occupied and not occupied. During the walk, only the bond memories are needed. The program inquires if the bond can change its position to a neighbouring one. This is possible only, if the memory shows the value 1.
The other case deals with bond percolation, i.e., a bond is stored randomly without looking at occupied and non-occupied sites. The program decides at once whether a bond is 1 or 0. The advantage of this version is the possibility to build greater lattices than in the other version, because the memories, needed for storing the sites, can be used now for storing further bonds. Besides this version needs less time for building the lattices.

As said above the number of steps is an equivalent to the computing time. This computing time amounted 5 to 8 μs per step attempt for both, site and bond percolation. Depending on the number of dimensions, the program needed 12 to 15 μs per place in a site percolation lattice and 2 to 4 μs per place in a bond percolation lattice, respectively, to initially build up the lattice.

Table I. — *The table shows how many ants, each making one million steps, were set in how many lattices of certain sizes.*

<table>
<thead>
<tr>
<th>$d$</th>
<th>Kind of percolation</th>
<th>Number of lattices</th>
<th>Lattice size</th>
<th>Number of ants</th>
<th>Signs in the figures</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Bond</td>
<td>10</td>
<td>$1950^d$</td>
<td>50</td>
<td>○</td>
</tr>
<tr>
<td>2</td>
<td>Site</td>
<td>10</td>
<td>$1500^d$</td>
<td>75</td>
<td>△</td>
</tr>
<tr>
<td>3</td>
<td>Bond</td>
<td>10</td>
<td>$120^d$</td>
<td>50</td>
<td>●</td>
</tr>
<tr>
<td>3</td>
<td>Site</td>
<td>10</td>
<td>$120^d$</td>
<td>90</td>
<td>×</td>
</tr>
</tbody>
</table>
In my inquiry, both possibilities were investigated. The main emphasis was laid on the behaviour of the exponent \( k \) at the percolation threshold \( p_c \). This concentration \( p_c \) is well known for the cases of my work. In case of site percolation its value is 0.59277 ± 0.3117; examining bond percolation, I had to take 0.5000 ± 0.2493 [15, 16]. Thus I was able to inquire the behaviour of the exponent \( k \) systematically.

The evaluation of the data yields the following results. The number of lattices, the lattice sizes and the number of ants having been used can be looked up in the table. To determine the values, I had to extrapolate the data for \( R \to \infty \) (cf. Fig. 2). It must be taken into account that the number of Monte Carlo steps was not as large as in references [12, 14], so that the data show great fluctuations. To yield better results, it would be desirable to use greater lattices and more ants, making more step attempts or to find a more efficient algorithm for this kind of the random walk simulation.

In case of a two-dimensional lattice, I determined 0.33 ± 0.02 as the value of the exponent \( k \) in bond percolation. For the site percolation lattice \( k \) has the value 0.32 ± 0.02. These values are consistent with the value \( k = 1/3 \) in the case of single particle, determined previously [12-14]. In a cubic bond percolation lattice \( k \) gets the value 0.21 ± 0.03, whereas \( k \) has the value 0.23 ± 0.03 in a three-dimensional site percolation lattice. These exponents are also in agreement with the single-particle exponent \( k = 0.2 ± 0.01 \).

So the conclusion can be drawn that the behaviour of a double particle does not differ from the behaviour of a single one. As far as the critical exponent \( k \) is concerned, it makes also no difference whether the pair moves in a site-percolation or in a bond-percolation lattice: The result, i.e., the exponent \( k \), is the same, though the probability \( p \) of the percolation threshold depends on the kind of the percolation, i.e., site and bond percolation, respectively.

This work has confirmed the universality hypothesis. The critical exponent \( k \) does not change if one replaces single sites by pairs as the diffusing particles. Such a universality result, though not unexpected, is not trivial. For example, in the case of kinetic gelation, when clusters were grown as a function of time, these kinetic effects seemed to change the universality class away from random percolation [17]. Fortunately, no such complication occurred here. The result may be relevant apart from this basic universality question, also for the diffusion of elongated particles in porous media.

Fig. 2. — The exponent \( k \) versus \( 1/R \) to extrapolate the result to the asymptotic limit \( R \to \infty \), again plotted for all four examined types of percolation.
Acknowledgments.

This work was suggested by C. Allain (Paris) and performed at the Institute of Theoretical Physics, Cologne University, with partial support from Sonderforschungsbereich 125. Thanks are due to D. Stauffer for the introduction into the problem of random walks and for his helpful advice during my inquiry.

References