Cellular automata approach to non-equilibrium diffusion and gradient percolation

1989 J. Phys. A: Math. Gen. 22 1609

(http://iopscience.iop.org/0305-4470/22/10/016)

J. Phys. A: Math. Gen. 22 (1989) 1609-1619. Printed in the UK

Cellular automata approach to non-equilibrium diffusion and gradient percolation

Bastien Chopard†, Michel Droz†§ and Max Kolb‡

† Department of Theoretical Physics, University of Geneva, 1211 Geneva 4, Switzerland
‡ Laboratoire de Physique de la Matière Condensée, Ecole Polytechnique, 91128, Palaiseau, France

Received 12 October 1988

Abstract. We propose a deterministic approach to lattice diffusion in two dimensions. This method is implemented on a cellular automaton special purpose computer in order to study the properties of the interface of particles diffusing from a source to a sink. Fractal properties of the diffusion front of this non-equilibrium process are compared with results from percolation theory. The observed agreement indicates that diffusion fronts and gradient percolation coincide asymptotically and that the cellular automata method is a viable alternative to standard simulations for this class of problems.

1. Introduction

Most solid interfaces observed in nature are irregular or rough. The geometrical properties of such interfaces are of great importance to materials science. One useful way to generate interfaces that are rough on all length scales is by diffusion processes, describing for example the situation when two solid interfaces are brought into contact. Practical examples where interfaces play an important role are solid-solid chemical reactions, alloys formed out of the melt and of course many processes in solid state technology [1, 2]. Other fields where such interfacial properties are directly relevant are catalysis, corrosion and crystal growth [3] and the physics of superionic conduction [4 and references therein]. A different domain where fractal interfaces occur naturally is invasion percolation of a porous medium [5].

This motivates us to study the out-of-equilibrium properties of diffusion and in particular the fractal interfaces generated by this process. There is one specific experiment that is particularly closely related to the present work: silver is deposited on a polyamide film in such a way that the average silver concentration varies in one direction [6]. The silver deposits have been analysed for their fractal structure. It was found that the hull of the connected silver patches is fractal with a fractal dimension that is in good agreement with theoretical predictions.

There is a second, more formal reason for studying non-equilibrium diffusion: a close relation exists between diffusion in a concentration gradient for small gradients and standard site percolation [7]. Therefore, diffusing particles in a gradient provide an ideal means to compute percolation properties, notably the percolation threshold p_c and the fractal dimension characterising large clusters [8]. The principal advantage

§ Supported by the Swiss National Science Foundation.

0305-4470/89/101609+11$02.50

of using gradients for percolation problems is that one does not have to know or guess the p_c in advance; the gradient method determines it self-consistently [7-10].

Actual diffusion of particles is a time-consuming process. For purely static properties, such as percolation, one can short-cut the diffusion process by simply distributing particles randomly, but inhomogeneously. One expects that the asymptotic results are the same. However, for a finite concentration gradient there are differences due to the hard core exclusion of the particles. This interaction leads to long-range correlations [11]. Furthermore, in realistic diffusion processes there are additional attractive or repulsive interactions that may change the results qualitatively. For example, a sufficiently strong attraction can induce a dynamical phase transition to an ordered state [12].

In the present work we propose a formulation of a random diffusion process with a hard core repulsion in terms of a deterministic cellular automaton (CA). This model is particularly suited to be implemented on a special purpose computer with parallel architecture.

Our reasons for studying the problem of gradient diffusion by a CA are as follows. (1) There are non-trivial exact results that permit us to test the validity of the deterministic approach. (2) The fractal properties of diffusion fronts have not been investigated by a real diffusion process—previous calculations use the above-mentioned short cut. (3) Percolation results suggest that the properties of the front depend on how it is defined [13]. We wish to investigate whether these properties remain valid for a true diffusion process.

Our results show that random diffusion can accurately and efficiently be implemented as a cellular automaton. We have chosen to adapt the algorithm to a CAM-6 [14] special purpose computer, but it is straightforward to transpose such an algorithm to another machine. The statistics of most of our results obtained by this method is comparable to a standard serial algorithm. Furthermore even for moderately small gradients there is good convergence toward the asymptotic zero gradient limit which indicates that the correlations do disappear sufficiently rapidly in order not to change the expected percolation results. Finally we observe that diffusion fronts also have the same fractal features as percolation: the different definitions of the external perimeter or hull yield different fractal exponents.

The paper is organised as follows. In § 2, we briefly review the known theoretical results for two-dimensional diffusion on a lattice. Section 3 is devoted to the cellular automata implementation. The validity of this CA approach is tested against known theoretical results. Then, the properties of the diffusion fronts are studied in § 4. Conclusions are drawn in § 5.

2. Some properties of non-equilibrium diffusion models

In this work we consider lattice diffusion on a two-dimensional square lattice. It turns out that some properties measured, such as the scaling of the concentration profile, generalise to other dimensions [15] while others, such as the fractal aspects of the diffusion fronts, are specially two dimensional. Indeed, because of the different topology the conclusions are known to be different in three dimensions [16].

The diffusion process studied here is the nearest-neighbour Kawasaki spin exchange model at infinite temperature in lattice gas language [17]: particles sit on lattice sites and jump randomly and with equal probability to any of their unoccupied neighbouring

sites. No two particles can ever occupy the same site. The jump rates of the different particles in the system are totally uncorrelated. In the present calculation neither attractive nor repulsive additional forces are included.

In order to study the out-of-equilibrium dynamics, we consider a system with a linear perfect source at $z=0$ which replaces immediately the particles diffusing into the system and a linear sink at $z=L_z$ which removes the arriving particles. In the perpendicular direction, the system has a size L_y with periodic boundary conditions. At time $t=0$ the system is completely empty. Although the diffusion process does conserve the number of particles inside the system, the total number still changes because of the source and the sink.

From the master equation governing this diffusion process an evolution equation for the average profile $P(z, t)$ can be derived in the continuum limit. Here $P(z, t)$ is defined as the average concentration at z and at time t, namely $P=1/L_y\langle\Sigma_y n(y, z)\rangle$ where $n(y, z)=0, 1$ is the occupation number on site (y, z). The average $\langle \ldots \rangle$ is a statistical average over realisations. We only consider systems that are homogeneous in the y direction. For the general case of interacting particles the evolution equation for P is coupled to the equations for the higher-order correlation functions [18]. The set of equations uncouples for a pure hard core repulsion and P obeys a simple diffusion equation [19]. Given the boundary and the initial conditions, this equation can be solved exactly, yielding the solution

$$P(z, t)=\frac{L_z-z}{L_z}-\frac{2}{\pi}\sum_{n=1}^{\infty}\frac{1}{n}\sin\left(n\pi\frac{z}{L_z}\right)\exp\left(-n^2\pi^2\frac{tD}{L_z^2}\right) \tag{1}$$

where D is the diffusion constant. In particular, the stationary density profile is given by the infinite-time limit of the above equation, and is thus obviously linear. In the limit $L_z\to\infty$, the above solution reduces to

$$P(z, t)=1-\frac{2}{\sqrt{\pi}}\int_0^{z/2\sqrt{(Dt)}} \mathrm{d}u\,\exp(-u^2). \tag{2}$$

This solution has an interesting scaling property. In terms of the average position $\bar{z}(t)$ defined by

$$\bar{z}(t)=\frac{\int_0^{\infty} zP(z, t)\,\mathrm{d}z}{\int_0^{\infty}P(z, t)\,\mathrm{d}z} \tag{3}$$

one has

$$P(z, t)=\hat{P}\left(\frac{z}{\bar{z}(t)}\right) \tag{4}$$

with

$$\hat{P}(z)=\mathrm{erfc}\left(\frac{\sqrt{\pi}z}{4}\right). \tag{5}$$

Thus, the time dependence enters only through the diffusion length $l_D\equiv 2\sqrt{Dt}=(4/\sqrt{\pi})\bar{z}$. In the finite geometry of our system this solution is valid for $1\ll l_D\ll L_y, L_z$.

Let us now introduce the diffusion front. The connectivity of the particles is defined as in site percolation: any two particles sitting on nearest-neighbour sites belong to the same cluster. For percolation a threshold p_c separates a region where only finite clusters are present from the region where an infinite cluster occupies a finite fraction of the system. In our model, p varies from $p=0$ to $p=1$ across the sample such that

the finite cluster and the infinite cluster regimes are present simultaneously. Close to the source $p > p_c$ and one cluster spans the system. Close to the sink $p < p_c$ and only finite clusters are present. In a weak gradient large finite clusters have the structure of percolation clusters and are located near p_c. This provides a way to calculate p_c. However, it is more efficient to obtain p_c from the front of the infinite cluster [10].

More precisely, the diffusion front, also called external perimeter or hull is defined as follows.

(1) Determine the infinite cluster ('the land') as all the particles which are connected to the source by nearest-neighbour particle–particle bonds.

(2) The complementary infinite cluster ('the ocean') consists of those empty sites which are connected by nearest- or next-to-nearest-neighbour empty–empty bounds to the sink.

(3) One then finds the land point farthest from the source. Starting from this point, the hull H1 ('the shore') is constructed as the line of first-neighbour particle–particle bonds such that each particle of the hull has a least one 'ocean' point as a first neighbour. This is illustrated in figure 1(a).

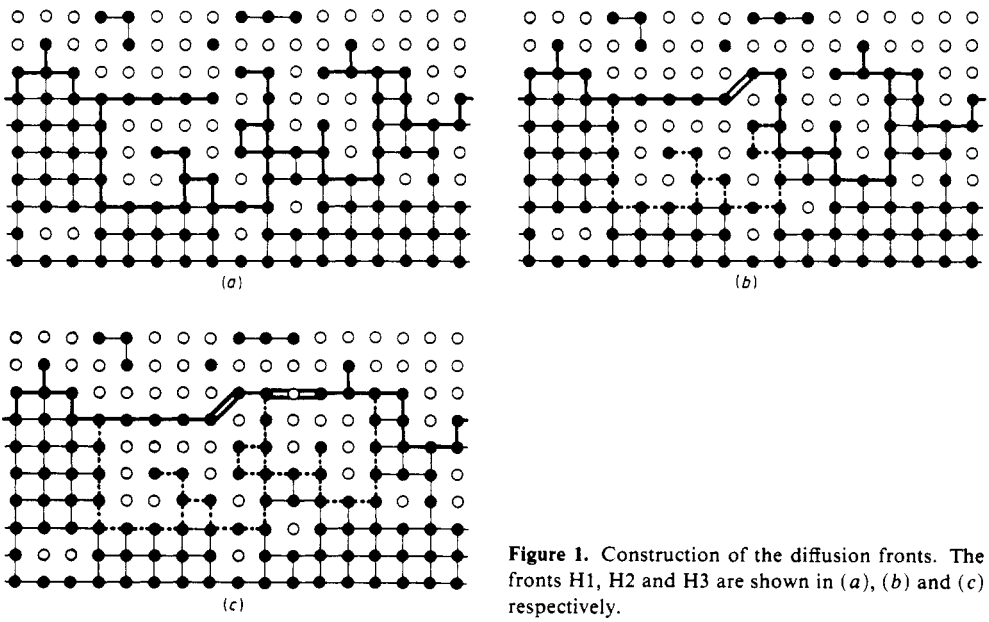

Figure 1. Construction of the diffusion fronts. The fronts H1, H2 and H3 are shown in (a), (b) and (c) respectively.

Different hulls can be defined by changing rule (3) above [12]; here two other hulls are defined by starting again from the farthest particle, and excluding those bays which are connected to the ocean only by a narrow channel.

(3a) The hull H2 excludes the bays which are bridged by second-nearest-neighbours, i.e. a step of length $\sqrt{2}$. This is illustrated in figure 1(b).

(3b) The hull H3 excludes the bays which are bridged by steps of length 2. This is illustrated in figure 1(c).

As the diffusion progresses the gradient of the average concentration decreases. More precisely, at fixed p, the gradient scales like $\nabla P = \mathrm{d}P/\mathrm{d}z = (\mathrm{d}\hat{P}/\mathrm{d}z)/\bar{z}$. Correspondingly, the diffusion fronts increase their widths. Their average position z_f then determines p_c through $p_c = P(z_f(t), t)$. The relation between the number N_f of particles

in the front and its width σ defines the hull dimension D_H. The relevant definitions are: $P_f(z, t)$ is the average density of *hull* points at z at time t, analogous to the density of all points $P(z, t)$ defined above. From P_f, the total number of front points N_f is given by

$$N_f(t) = \int_0^\infty P_f(z, t)\, dz. \tag{6}$$

The average front position z_f is defined by

$$\bar{z}_f(t) = \frac{\int_0^\infty zP_f(z, t)\, dz}{\int_0^\infty P_f(z, t)\, dz} \tag{7}$$

and the width of the front σ is given by

$$\sigma^2 = \int_0^\infty (z - z_f)^2 P_f(z, t)\, dz \left(\int_0^\infty P_f(z, t)\, dz \right)^{-1}. \tag{8}$$

The value of p_c for finite t is expected to extrapolate to the exact p_c as $t \to \infty$. The scaling laws $N_f \sim (\nabla P)^{-\alpha_N}$ and $\sigma \sim (\nabla P)^{-\alpha_\sigma}$ define the exponents α_N and α_σ. The observation $\alpha_N + \alpha_\sigma = 1$ for the hull, defined by (3) combined with the observation that the width σ of the hull scales like the correlation length $\xi \sim (p_c - p)^{-\nu}$ leads to $\alpha_\sigma = \nu/(1+\nu)$ and to

$$D_H = \frac{(1+\nu)}{\nu} \tag{9}$$

in $N_f \sim \sigma^{D_H}(L_y/\sigma)$. With the accepted value of $\nu = \frac{4}{3}$, this gives $D_H = \frac{7}{4}$. This result has been confirmed numerically both for standard percolation and for gradient percolation [12, 18] and is presumably exact [20]. It does not hold for hulls (3*a*) and (3*b*) above as $\alpha_N + \alpha_\sigma = 1$ fails in this case. In this case, D_H is closer to $\frac{4}{3}$ rather than $\frac{7}{4}$. We now wish to calculate these exponents by way of the CA approach.

3. A cellular automaton model for diffusing particles

In this section, we shall propose a new approach to the problem of simulating a diffusion process. The main idea is to find a cellular-automaton-type algorithm which reproduces the simultaneous independent random walks of many particles. For simplicity and reasons of implementation, we shall restrict ourselves to the two-dimensional case. However, there are in principle no difficulties in generalising this algorithm to higher dimensions. The interest of a CA approach resides in the efficiency of the algorithm which takes advantages of the full parallelism of the dedicated automata machines. However, due to this parallel updating, the CA rules have to be constructed carefully in order to treat properly the fact that the particles cannot experience collisions or overlap. These conditions are naturally taken into account in the following very simple algorithm.

Let us consider a two-dimensional square lattice on which the particles diffuse. There is at most one particle per site or cell. This lattice is divided into adjacent blocks, each containing 2×2 cells (see figure 2). Those blocks are called the Margolus

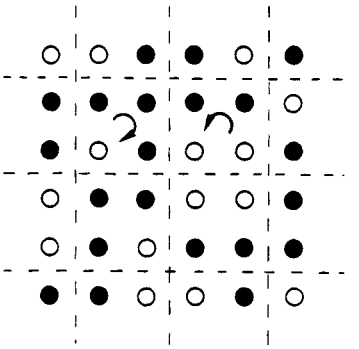

Figure 2. Margolus neighbourhood as used in the algorithm. The arrows indicate two typical motions of the blocks which generate a basic diffusion step. In the next iteration, the blocks are shifted by one elementary step along the diagonal.

neighbourhood [14]. The diffusion process is realised in two steps. First, the particles in a block are moved using three different operations conserving the number of particles in each block. The operations are the $\pm\pi/2$ rotations and the identity transformation. For each block on the lattice, an operation is chosen according to a process that mimics randomness and which will be described below. Secondly, the block sublattice is shifted by one lattice constant in both directions (see figure 2). This step allows the particles to move from one block to an adjacent one. The whole process is then repeated and thus the particles can propagate through the system. Although the motions of the particles belonging to the same block are strongly correlated, the second step of the algorithm allows that these correlations disappear rapidly over long distances.

The problem then is to find a reliable way for producing simultaneously an independent random number for each block. The idea is to use another CA rule as a random number generator. Our special purpose CA computer consists of the superposition of four two-level arrays which can be coupled locally [14]. The basic idea is to use one array for the diffusion particles and another one for the random number generator. At each step of the evolution, the gas of diffusing particles is updated according to the rules described above, where the choice of the symmetry operation depends on the configuration of the adjacent cell in the random generator plane. Obviously, the random number generator plane is updated simultaneously.

A simple way to construct a random number generator consists of using a lattice-gas model which has been tested for its ergodic properties. Thus, our random generator will be constructed with the help of the well-known HPP gas [19], which is here implemented inside a Margolus block [14]. Our HPP gas is made of particles moving along the diagonals of the lattice. In the Margolus implementation, the motion of a HPP particle depends on its location inside a block. For instance, a particle at the upper left corner of a block travels in the south-east direction, while a particle located at lower right corner moves in the opposite direction. In the same way, an upper right particle travels south-west, and so on. As long as a particle does not meet another one, it continues in a straight line. However, when exactly two particles with opposite velocities are simultaneously present in the same block, they undergo a collision and bounce at right angles. This is the collision mechanism which make a lattice gas suitable for simulating our local random number generator. The HPP gas has periodic boundary conditions in both directions.

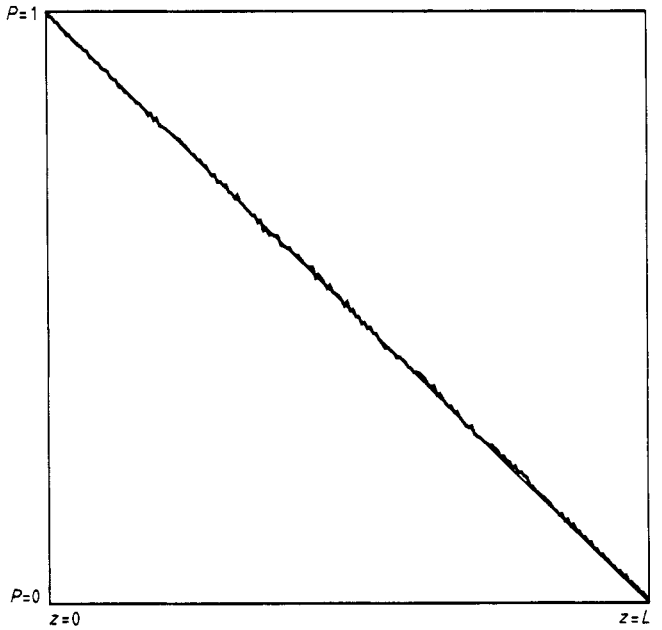

Figure 3. Stationary diffusion profile comparing the simulation result with the straight line plot of the theoretical solution of the diffusion equation.

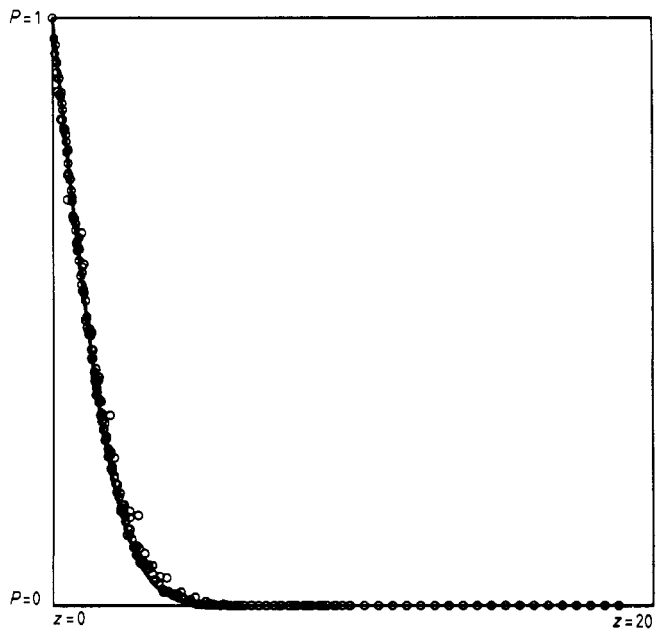

Figure 4. Scaling function $\hat{P}(z/\bar{z}(t))$. The circles are the results of the simulation and the continuous curve is the theoretical prediction. For details see the text.

At each time, an integer number n between 0 and 15 is associated with each block. This number indexes the sixteen possible configurations of the HPP particles in a block. More precisely, they are constructed in terms of the occupation numbers of each cell b_{yz}, which takes the value 1 if the cell is occupied and zero otherwise. They read

$$n = b_{ul}2^3 + b_{ur}2^2 + b_{ll}2^1 + b_{lr}2^0 \tag{10}$$

where the subscript of b indicates the location of the cell inside a block (upper left, upper right, lower left and lower right).

For the HPP gas, every configuration having a given number of particles in a block is equiprobable. Thus, among these configurations, an equal number should give rise to a clockwise or counterclockwise rotation of the block of diffusive particles. For the binary numbers between 0 and 15, there is an even number of block configurations with exactly one bit equal to 1. Therefore, we associate a left rotation with half of them and a right rotation with the other half. The same is true for the numbers with two and three bits equal to 1. Therefore, one associates a $+\pi/2$ rotation with $n = 1$ and 4, and a $-\pi/2$ rotation with 2 and 8. Similarly 3, 6, 10 and 11, 14 produce a clockwise rotation, while 5, 9, 12 and 7, 13 give a counterclockwise rotation. Finally, we choose the identity operation for 0 and 15.

Note that by construction of the algorithm, the process is independent of the number of diffusing particles. On the other hand, the concentration of the HPP gas influences the mobility of the diffusion. Simulations with concentrations of 10, 25 and 50% have been performed. The final results do not depend on this parameter.

In order to test the quality of this algorithm, we have simulated two typical diffusion situations for which exact analytical results exist. The first one consists of studying the stationary density profile produced by particles diffusing in a plane between a source and a sink. The presence of a source and a sink is implemented on the CA as follows. The top row of the lattice of diffusing particles is, at all times, completely filled with particles. In contrast, any particle reaching the bottom row is immediately absorbed.

As shown in § 2, the analytical solution of the diffusion equation leads to a linear stationary density profile. This prediction can now be compared with the results of the simulation of our CA algorithm. The simulation has been performed on a special purpose computer CAM-6 [14] updating 256×256 cells simultaneously 60 times a second. The source consists of a horizontal line of length 256, at the position $z = 0$ while the sink is a parallel line at $z = L$. The density at z is given by the total number of particles on the corresponding horizontal line, divided by 256. Moreover, an average has been made over 100 samples.

The results are given in figure 3. One sees that the agreement with the theoretical predictions is excellent.

The second test consists of studying the diffusion in the presence of the source only (at $z = 0$).

The scaling prediction derived in § 2 can be compared with the results of the simulation. The profile $P(z, t)$ has been measured after 16, 32, 64, 128, 256 and 512 iterations steps. An average over ten samples has been made. The corresponding scaling function is plotted in figure 4. The circles are the results of the simulation while the continuous curve is the theoretical prediction. Again the agreement is excellent.

The above results show that our CA algorithm well describes the diffusion of independent particles. It will be used in the next section to study the properties of the diffusion fronts.

4. Properties of the diffusion fronts

An advantage of the CA approach for studying the diffusion fronts is the fact that they can be constructed by using an efficient CA algorithm. Explicitly, one proceeds in three steps.

(i) The 'land' is constructed by considering the set of diffusing particles being either a nearest neighbour of the source or a nearest neighbour of a point of 'land' already existing. With this rule, the land is emerging from the source under iteration. The implementation in terms of CA is obvious.

(ii) The 'ocean' is constructed as follows. One first removes all the particles which do not belong to the land. Then, a point which is not land becomes ocean if it is a neighbour of the sink or a neighbour of an already existing ocean point. The resulting ocean emerges under iterations from the sink.

(iii) The fronts H1, H2 and H3 are constructed as follows.

H1: The ocean is made of nearest- and next-nearest neighbours. The front is made of the land points, which are first- or second-nearest neighbours of the ocean.

H2: The ocean is made of nearest neighbours. The front is made of the land points which are first- or second-nearest neighbours of the ocean.

H3: The ocean is made of nearest neighbours with the additional restriction that the ocean particles cannot enter a channel whose width is one lattice spacing. The front is made of the land points which are first-nearest neighbours of the ocean.

These definitions were guided by some implementation restrictions. They are essentially equivalent to the definitions in § 2.

As seen in § 2, the number of particles N_f composing the diffusion fronts depends on their average widths σ, according to

$$N_f = A\sigma^{D_H - 1} \tag{11}$$

where A is some amplitude and D_H is the fractal dimension of the front.

Our interest was to determine the values of D_H for the three different constructions of the fronts. The values of N_f and σ were measured for different fronts obtained after letting the particles diffuse during 194, 388, 776, 1512 and 3024 evolution steps. Moreover, for each of the five times, averages over ten samples were performed. Note that for all these times, the sink is still far away from the front. The three types of fronts are shown in figure 5, for the same diffusion experiment.

The value of σ was obtained using equation (8). During the time evolution, the number N_f of particles in the front and its width σ increase. Thus, using the data obtained for different times, one can extract the value of D_H from equation (11). It turns out that the plot of $\log N_f$ against $\log \sigma$ satisfies a linear relation quite well.

The measurements for the three different cases discussed above give the following values of D_H:

(i) $D_{H1} = 1.743 \pm 0.012$

(ii) $D_{H2} = 1.420 \pm 0.020$

(iii) $D_{H3} = 1.329 \pm 0.020$.

The first value agrees with the theoretical prediction [7] relating D_{H1} to the correlation length exponent for percolation problem, i.e. $D_{H1}^{th} = 1.75$. The third value corresponds quite well also with other numerical simulations [12] and theoretical conjectures [20], i.e. $D_{H3}^{th} = \frac{4}{3}$. As far as the second case is concerned, the situation is not so clear. Indeed, according to some conjectures, the value of D_{H2} should be the same as D_{H3}. However, this statement is true only in the limit of zero gradient. For

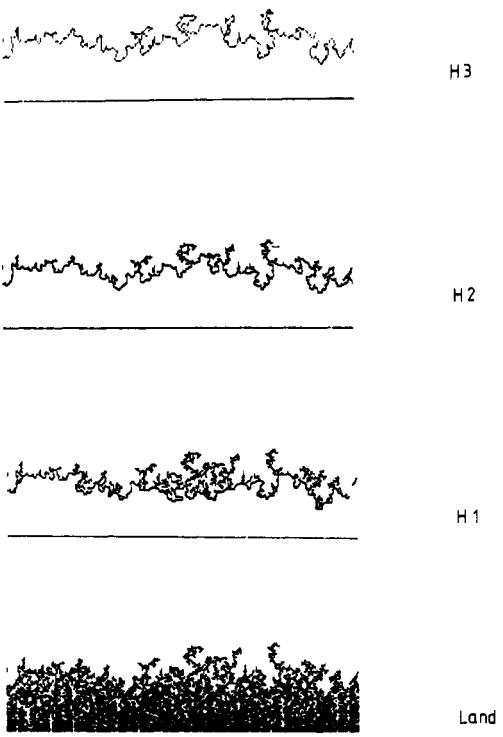

Figure 5. The three different diffusion fronts for the same diffusion experiment. The different fractals show up explicitly in these diagrams.

a stationary state, it corresponds to an infinite distance between the sink and the source. This is obviously not the case in our experiment and thus one should expect corrections to scaling due to finite-size effects.

Another property of the front which has been investigated, is the density p of diffusing particles at the mean position of the front z_f. It has been verified [7] that for the first type of front, $p(z_f)$ is equal to the percolation threshold p_c of the two-dimensional site percolation problem, namely $p_c = 0.5928$ [10]. The value of $p(z_f)$ has been measured in the stationary state for ten samples and two different concentration gradients. For a sink-source separation of 208 lattice constants, we found $p(z_f) = 0.6013$, while, for a separation of 256, we obtained $p(z_f) = 0.5945$, in the stationary limit. These results should be compared with the best available value $p_c = 0.592\,745$ [9]. Although the general tendency is correct, our statistics is not good enough to extract the infinite-separation limit.

5. Conclusion

In the present study the feasibility of implementing inhomogeneous diffusion by a cellular automata algorithm on a special purpose computer has been tested. The parallel updating of the diffusion process has been tailored for the CAM-6 parallel processing machine by modifying the elementary diffusion steps. The results indicate that neither the parallelism nor the modified elementary diffusion steps influence the

results. Both the out-of-equilibrium diffusion profile and the percolation characteristics of the diffusion front agree with exact results and with numerical results obtained from homogeneous percolation. The only sign of a difference in comparison with serial Monte Carlo simulations is the occasional appearance of transients close to the steady state. The detailed investigation of this effect and the possible origins of it would be an interesting topic for further work.

Acknowledgement

One of the authors (MK) has benefited from numerous discussions with J-F Gouyet, M Rosso and B Sapoval.

References

[1] Nowick A S and Burton J J 1978 *Diffusion in Solids* (New York: Academic)
[2] Philibert J 1985 *Diffusion et Transport de Matière dans les Solides* (Les Ulis: Editions de Physique)
[3] Naumovets A G 1985 *Faraday Discuss. Chem. Soc.* **80** 90
[4] Dieterich W, Fulde P and Peschel I 1980 *Adv. Phys.* **29** 527
[5] Hulin J P, Clément E, Baudet C, Gouyet J-F and Rosso M 1988 *Phys. Rev. Lett.* **61** 333
[6] Mazur S and Reich S 1989 *J. Chem. Phys.* in press
Wool R P 1988 *Fundamentals of Adhesion* (New York: Plenum)
Wool R P and Long J M 1989 to be published
[7] Sapoval B, Rosso M and Gouyet J-F 1985 *J. Physique Lett.* **46** 146
[8] Sapoval B, Rosso M, Gouyet J-F and Colonna J F 1986 *Solid State Ionics* **18/19** 21
Ziff R and Sapoval B 1986 *J. Phys. A: Math. Gen.* **19** 1169
[9] Ziff R 1982 *J. Stat. Phys.* **28** 838
[10] Rosso M, Gouyet J-F and Sapoval B 1985 *Phys. Rev.* B **32** 6053
[11] Kolb M, Gouyet J-F and Sapoval B 1987 *Europhys. Lett.* **3** 33
[12] Grossman T and Aharony A 1986 *J. Phys. A: Math. Gen.* **19** L745; 1987 *J. Phys. A: Math. Gen.* **20** 1193
Bunde A and Gouyet J-F 1985 *J. Phys. A: Math. Gen.* **18** L285
Grassberger P 1986 *J. Phys. A: Math. Gen.* **19** 2675
[13] Spohn H 1983 *J. Stat. Phys.* **16** 4275
Katz S, Lebowitz J L and Spohn H 1984 *J. Stat. Phys.* **34** 497
[14] Toffoli T and Margolus N 1987 *Cellular Automata Machine: A New Environment for Modeling* (Cambridge, MA: MIT Press)
[15] Rosso M, Gouyet J-F and Sapoval B *Phys. Rev. Lett.* **57** 3195
[16] Gouyet J-F, Rosso M and Sapoval B 1988 *Phys. Rev.* **37** 1832
[17] Kawasaki K *Phase Transitions and Critical Phenomena* ed C Domb and M S Green (New York: Academic) vol 2, p 443; vol 5, p 165
[18] Rosso M 1988 *Preprint* Orsay
[19] Hardy J, Pomeau Y and de Pazzis O 1976 *Phys. Rev.* A **13** 1949
Frisch U, Hasslacher B and Pomeau Y 1986 *Phys. Rev. Lett.* **56** 1505
[20] Saleur H and Duplantier B 1988 *Phys. Rev. Lett.* **58** 2325