

Finite-Size Effects in a Cellular Automaton for Diffusion

K. Froböse^{1,2}

Received October 5, 1988; revision received January 8, 1989

The question whether diffusion in the hard-square lattice gas is blocked in the thermodynamic limit is mapped to the problem whether percolation occurs in the time evolution of a cellular automaton. The final states of the cellular automaton are investigated for varying lattice sizes from 6×6 up to $20,035 \times 20,032$. The results seem to indicate that there is a percolation threshold, i.e., a range of concentrations for which diffusion is blocked. However, since this cannot be true for the infinite system, as proven rigorously, it is concluded that finite-size effects persist for this system up to very large sizes.

KEY WORDS: Percolation threshold; diffusion; cellular automata; supercomputers; multi-spin-coding; hard-square lattice gas.

1. INTRODUCTION

Recently Ertel *et al.*⁽¹⁾ studied the diffusion in a hard-square lattice gas by Monte Carlo simulation. They found that at sufficiently high concentrations the diffusion becomes blocked due to the hard cores of the particles. However, the critical concentration above which blocking occurs depends on the size of the lattice and increases when the system becomes larger. Therefore, one may ask whether in the thermodynamic limit of an infinite system the critical concentration approaches the maximum value for the hard-square lattice gas or whether it converges to some fixed value c^* below this maximum. The second case would imply a dynamical phase transition in diffusion.

Ertel *et al.*⁽¹⁾ related this question to a modified percolation problem. When the corresponding lattice is not percolating, the original diffusive system is blocked. They proved that at any concentration near the maxi-

¹ HLRZ c/o KFA Jülich, D-5170 Jülich, West Germany.

² Permanent address: Fakultät für Physik, Universität Konstanz, D-7750 Konstanz, West Germany.

imum concentration the system percolates with probability 1 in the thermodynamic limit, and consequently in this limit blocking should not occur. The percolation problem under study is related to bootstrap percolation on the square lattice for $m=3$.⁽²⁾ The nonexistence of a finite percolation threshold in the thermodynamic limit is common to both types of percolation problems. With a slight modification, van Enter's proof⁽³⁾ for the case of bootstrap percolation is also applicable to our problem.⁽⁴⁾ However, the size dependence of the percolation probability for finite systems is very different for both percolation problems. This difference is the subject of the present paper.

For our problem the asymptotic relation between the critical concentration and the size of the system is not known. Therefore, we study the percolation problem for square lattices numerically in this present paper. It turns out that it can be mapped to the time evolution of a two-dimensional cellular automaton. The relation to the original diffusion problem is described in Section 2.

The linear dimensions L of the lattices were varied over more than three orders of magnitude from 6 to 20,032. Including lattices with side lengths up to 20,032 requires a considerable amount of computing power. The calculations could only be achieved using both multi-spin-coding⁽⁵⁾ and the vector facilities of a CRAY X-MP/48 computer. The algorithm made it possible to obtain nearly 500 millions lattice sites swept per second. The details of the implementation are given in Section 3.

For a series of runs of the computer program at a fixed concentration we determine the fraction of samples for which percolation occurs. This quantity is called the percolation probability. We investigate how the concentration must be changed when the lattice size is varied so that the percolation probability remains fixed at a given value. In order to save computing time, a low percolation probability of 10% for all lattices and a higher one of 50% for only the smaller lattices were selected.

It turns out that the data found for the percolation probability of 10% can be represented by a simple function. Plotting the data for the particle concentration versus $1/\ln(L)$ results in a straight line for $L \geq 10$. The extrapolation of this line to the thermodynamic limit, however, does not yield an intercept of 0 for $1/\ln(L) \rightarrow 0$ as is expected from ref. 1, but one obtains an intercept of 0.035 instead. Thus the results indicate a finite percolation threshold even for the infinite lattice.

2. THE CELLULAR AUTOMATON FOR DIFFUSION

In the model discussed in ref. 1 the occupation of lattice sites is governed by two rules: (i) a lattice site can be occupied by at most one particle,

(ii) the simultaneous occupation of nearest-neighbor pairs is not allowed. The second rule simulates a square-shaped hard core of the particles. As a consequence the maximum concentration is $c_{\max} = 0.5$. At this concentration all particles are located at one of the two sublattices and the system looks like a checkerboard.

The particles can only move to a nearest-neighbor site when the above rules are also fulfilled for the destination site. Thus, a jump to a nearest-neighbor site is only possible when not only the final site itself, but also three of its nearest-neighbor sites are empty. (One of the nearest-neighbor sites is occupied by the particle under consideration.) We call such particles movable. If particles happen to be aligned on a diagonal line of the lattice, it is only at the end of the line that they can escape from the diagonal, whereas within the chain they are kept fixed by the other particles of just the same chain. Moreover, if such diagonal lines are arranged in such a way that they form closed rectangles and do not have free ends, these rectangles cannot alter their shape under the diffusion processes. Consequently, all particles inside such rectangular cages remain captured all the time.

Blocking of diffusion occurs when it is possible at a certain instant of time to enclose all movable particles by such rectangular cages, since then the movement of the particles remains always restricted to their cages. Thus, it can be decided from an inspection of the actual configuration whether diffusion is blocked or not.

We are interested in the determination of such cages. But before giving rules for detecting them, we will introduce a simplification to the problem which was also made in ref. 1. Consider a configuration for which diffusion is blocked. Since the blocking status will not be changed under diffusion, the particles may be rearranged within their cages by diffusion processes. We assume that it would be possible to shift all the particles to the predominantly occupied sublattice. In the following we only consider this sublattice. It is again a square lattice, but turned by 45 deg. The occupation of the lattice is characterized by the probability p of finding holes in this sublattice. It is related to the particle concentration c of the total lattice by $p = 1 - 2c$.

A movable particle as defined above is recognized in the selected sublattice by having two adjacent nearest-neighbor sites and additionally their common nearest-neighbor site all empty. This may be expressed differently by saying that a particle is movable when a square of four sites can be found in such a way that this particle is the only one in this square.

In a first step for the detection of cages we remove all movable particles from the system. By taking away these particles, other ones, previously blocked by the removed particle, may become movable. In a

second sweep through the lattice they are also removed. This procedure is repeated until there has been no change in the configuration of the lattice after a sweep.

The removing procedure can be understood as a rule for the time evolution of a cellular automaton on the square lattice. During a sweep those particles are taken away for which at least two adjacent nearest-neighbor sites and also the enclosed corner site are empty. Since such a configuration can occur for four different orientations, the state of a lattice site after a sweep not only depends on the occupation of its nearest-neighbor sites, but also on the four next-nearest-neighbor sites lying between the nearest-neighbor sites.

There are two possible results: Either there were cages in the system, in which case their walls never become movable, and all particles belonging to the walls and outside of the cages are left over; or there were no cages at all, and the procedure does not come to an end until the lattice is completely empty. In this latter case the system has percolated. If there is no percolation, the diffusion is blocked.

3. IMPLEMENTATION OF THE LATTICE IN THE COMPUTER

We consider a rectangular lattice composed of L_2 lines of length L_1 . The lattice sites can be considered as a one-dimensional ordered set by enumerating at first the first line from left to right, then the second line, and so on, down to the last site in the bottom line. Since we are using helical boundary conditions, for a site I which is not located in the first or the last line, its left-hand and its right-hand neighbors are labeled by $I-1$ and $I+1$, whereas its upper and lower nearest neighbors are given by $I-L_1$ and $I+L_1$ and its next-nearest neighbors, which we also need in our analysis, are found as $I-L_1-1$, $I-L_1+1$, $I+L_1-1$, and $I+L_1+1$, respectively.

The last line of the lattice is considered to represent the upper neighbors of the first line, and the first line the lower neighbors of the last line. We do not need to consider how the relations among the labels of a lattice site and its neighbors must be modified, because they emerge quite naturally by our coding technique.

We use multi-spin coding as described by Herrmann,⁽⁵⁾ i.e., the state of each lattice site is stored in just one bit of a computer word. An occupied lattice site is represented by 0, an empty one by 1. If W is the width of a computer word, then the whole lattice is stored in an array of $N=L_1 * L_2/W$ words. (Of course $L_1 * L_2$ must factorize with respect to W .) The states of the first N lattice sites are stored in their given order in the first bits of the array. The next portion of N lattice sites is stored in

the second bits of the array, etc. By this representation we achieve that all sites which neither belong to the first $L1 + 1$ nor to the last $L1 + 1$ words of the array are coded in the same bit positions as the neighbors which must be taken into account for the time evolution. We can get rid off this restriction by storing the last $L1 + 1$ words of the array in their given order just before the first element and adding the first $L1 + 1$ words also in the correct order to the end of the array. (The extended array now contains $N + 2 * L1 + 2$ elements.) To get things correctly matched, however, it is necessary that the words inserted at the beginning of the array are circular-shifted by one bit to the right, whereas the words pushed to the end are circular-shifted by one bit to the left. In this way for all sites of the lattice, represented in the original range of the array, the neighbors which determine the evolution appear at the same bit position in the array.

This kind of storage is a little bit different from that suggested by Hermann⁽⁵⁾ and uses slightly more memory, but it has the advantage that all lattice sites can be treated by the same programming code without additional programming effort for properly taking into account the boundary conditions.

The logic of the program is as follows: In a first step we fill each lattice site with probability p with holes, denoted by a 1 bit. Then sweeps through the lattice are performed until changes in the configuration no longer occur. Then the program is stopped and it is stated whether the system had percolated or not.

Since we succeeded in getting all the lattice sites, which must be considered for changing the state of a certain site, to the same bit position, we can treat W ($W = 64$ for the CRAY) lattice sites in parallel by a combination of logical bit manipulations in which all neighbors of the site are involved.

The loop sweeping the lattice contains the following statements:

```
SUM = 0
DO 2 I = L1 + 2, L1 + 1 + N
  LB(I) = LA(I)
  &      .OR. (LA(I - L1) .AND. LA(I - 1) .AND. LA(I - L1 - 1))
  &      .OR. (LA(I - L1) .AND. LA(I + 1) .AND. LA(I - L1 + 1))
  &      .OR. (LA(I + L1) .AND. LA(I - 1) .AND. LA(I + L1 - 1))
  &      .OR. (LA(I + L1) .AND. LA(I + 1) .AND. LA(I + L1 + 1))
  SUM = SUM + POPCNT(LB(I))
2 CONTINUE
```

Here some simplifications reducing the computational effort have been omitted for the sake of clarity. The array LA represents the lattice. The sweep concerns only the inner part of LA, not the extensions made for taking account of the boundary conditions. Each result is temporarily

stored in the array LB, so that LA is not affected during the sweep. After completion of the sweep, the contents of LB are transferred back to LA. The intrinsic CRAY function POPCNT counts the number of 1-bits. The variable SUM monitors the number of holes created in the system. If it has not changed with respect to the previous sweep, the system has reached its fixpoint and the program can be stopped. Whether the system has percolated can be easily checked from SUM. In the case of percolation the value of SUM is equal to $L1 * L2$.

This loop and also the other parts of the program fully exploit the vectorization facilities of the CRAY computer. By this combination of vectorization and true parallel computing we achieve a testing rate of nearly 500 millions spins per second and per processor.

The data were consistent with those of ref. 1 for smaller lattices obtained through a different program.

4. RESULTS AND DISCUSSION

The program was run for square lattices of varying sizes from 6×6 up to $20,035 \times 20,032$. For the larger lattices we chose slightly different lengths of the edges in order to facilitate memory access for the computer. For each lattice size we varied the probability for holes p until the portion of samples which percolated was 10% (50%) of the total number of samples. With the exception of the largest system with $L = 20,032$, where only ten (two) samples were used for a probability of 10% (50%) to percolate, the number of samples was at least 100 for each tested concentration for the large lattices and was increased to 10,000 for the small ones. In Table I the obtained probabilities p are listed for the two probabilities to percolate of 10% and 50%, respectively. With the exception of the values for the 6×6 and the 8×8 lattice, all the data for the probability to percolate of 10% are located on a straight line. The values for the probability to percolate of 50% do not follow a straight line. With increasing lattice size the curve for 50% seems to approach the line for the 10% probability to percolate.

With this finding one is tempted to extrapolate the line to the case of an infinite system. The intersection point with the p axis can be estimated to be at 0.035 ± 0.003 , from which one might speculate that there is a percolation threshold even in the thermodynamic limit, a clear contradiction to theory,^(1,3) which predicts intercept zero.

A similar study has been conducted in connection with the two-spin facilitated kinetic Ising model for a square lattice^(6,7) by Nakanishi and Takano.⁽⁸⁾ The problem of the ergodicity of this model is equivalent to bootstrap percolation with $m = 3$. The rules of the percolation problem also define a cellular automaton, but with a rule for updating simpler than ours.

Table I. The Probability for Holes p As a Function of the System Size for Probabilities to Percolate of 10% and 50%, Respectively

System size	Probability for holes	
	10%	50%
6×6	0.223	0.339
8×8	0.208	0.299
10×10	0.196	0.270
12×12	0.186	0.249
16×16	0.171	0.219
20×20	0.160	0.201
24×24	0.152	0.189
30×30	0.145	0.176
40×40	0.136	0.163
48×48	0.131	0.155
64×64	0.123	0.146
88×88	0.119	—
131×128	0.113	0.131
259×256	0.102	0.117
451×448	0.098	—
963×960	0.091	—
1987×1984	0.084	—
7363×7360	0.076	—
20035×20032	0.071	0.076

For this model Nakanishi and Takano found a linear relationship between $1/p$ and $\ln(L)$ in the range from $L = 70$ to $L = 1775$. We tried to plot their data in the same way as ours. We also obtained a straight line which does not intersect the p axis at zero, but the intercept is so small that one would not infer a finite percolation threshold from it. In contrast to the findings for the present percolation problem, these numerical results confirm the theoretical predictions for the asymptotic behavior of the probability to percolate in the case of bootstrap percolation.^(3,9)

On the other hand, an equally unexpected percolation threshold was found by Herrmann *et al.*⁽¹⁰⁾ when studying the time evolution of an Ising ferromagnet at fixed energy represented by a Q2R cellular automaton. Different regions of the system show different periods in their time evolution. At a certain energy they observed a kinetic phase transition in that the number of particles having infinite time period grew suddenly when the energy was raised. Regarding their system as a lattice gas, the percolation threshold appears at a concentration of $p \approx 0.03$, which is about the same as ours. However, it should be noted that there was no significant size dependence observed.

The contradiction between the present results and those of ref. 1 may be resolved by the assumption that the present lattices are still too small. The findings of ref. 1 only apply to large systems. But it is not clear how large the system must be chosen for verifying the conclusions. From this work it can be stated that the lattices must be at least larger than $10^4 \times 10^4$. Without the multi-spin algorithm and a high-speed computer one would not have found this result.

ACKNOWLEDGMENTS

I thank Prof. J. Jäckle for suggesting the problem. I am grateful to HLRZ at KFA Jülich for the hospitality and the computing time at the CRAY X-MP/48 and to Prof. D. Stauffer for performing the calculation for the largest sample and for his assistance and advice during the completion of this work. Financial support by the Deutsche Forschungsgemeinschaft (SFB 306) is also acknowledged.

REFERENCES

1. W. Ertel, K. Froböse, and J. Jäckle, *J. Chem. Phys.* **88**:5027 (1988).
2. J. Chalupa, P. L. Leath, and G. R. Reich, *J. Phys. C: Solid State Phys.* **12**:L31 (1979); J. Adler and A. Aharony, *J. Phys. A: Math. Gen.* **21**:1387 (1987).
3. A. C. D. van Enter, *J. Stat. Phys.* **48**:943 (1987).
4. J. L. Lebowitz, private communication.
5. H. J. Herrmann, *J. Stat. Phys.* **45**:145 (1986).
6. G. H. Fredrickson and H. C. Anderson, *Phys. Rev. Lett.* **53**:1244 (1984).
7. G. H. Fredrickson and H. C. Anderson, *J. Chem. Phys.* **83**:5822 (1985).
8. H. Nakanishi and H. Takano, *Phys. Lett. A* **115**:117 (1986).
9. M. Aizenman and J. L. Lebowitz, *J. Phys. A: Math. Gen.* **21**:3801 (1988).
10. H. J. Herrmann, H. O. Carmesin, and D. Stauffer, *J. Phys. A: Math. Gen.* **20**:4939 (1987).