Parallel computation of cluster properties: application to 2D percolation



1987 J. Phys. A: Math. Gen. 20 985

(http://iopscience.iop.org/0305-4470/20/4/032)

J. Phys. A: Math. Gen. **20** (1987) 985-993. Printed in the UK

# Parallel computation of cluster properties: application to 2D percolation

R Dewar† and C K Harris

Department of Physics, University of Edinburgh, King's Buildings, Mayfield Road,- Edinburgh EH9 3JZ, UK

Received 28 May 1986

**Abstract.** We discuss various parallel algorithms for the enumeration of two-dimensional cluster properties in such problems as percolation and the Ising model. As an application, in a Monte Carlo simulation performed on the ICL distributed array processor (DAP), we re-examine the numerical evidence for a recent conjecture by Jug concerning percolation singularities in two dimensions.

## 1. Introduction

The graph problem of analysing clusters has relevance to many areas of physics. In studies of percolation (for reviews see Stauffer 1979, Essam 1980), Ising systems (Gunton *et al* 1983) and models of kinetic growth (Family and Landau 1984), for example, it is natural to express quantities of interest in terms of properties of clusters of connected sites or bonds on a lattice. These may include the total number of clusters, cluster size distribution, cluster surface geometry, etc.

Many authors have developed and applied computer programs to analyse clusters (e.g. Hoshen and Kopelman 1976, Redner 1982). Typically, the algorithms proceed in a serial fashion; sites are analysed one by one for their connectivity to other sites, a labelling scheme is introduced to identify sites on the same cluster and if two candidate clusters are found to be connected their labels are merged. In all such programs there is a trade-off between the time required to solve the problem and the computer space made available to store the graph.

One way to increase efficiency is to analyse more than one site at a time. Parallel algorithms have been written (Hambrusch 1983) to do this on custom-built VLSI chips in which the processing elements operate simultaneously. However, with the advent of general-purpose parallel computers (Hockney and Jesshope 1981), it is of interest to develop algorithms that exploit the parallel architecture of existing machines.

In § 2, we introduce a parallel 'burning' algorithm that counts the total number of clusters of connected sites or bonds on various types of 2D lattice. We also discuss a parallelised 'ants in the labyrinth' algorithm that analyses the cluster size distribution. Both algorithms are very simple. We have implemented them on the ICL distributed array processor (DAP). They are very fast in practice because of their parallelism and because, in identifying the connectivity of clusters, only Boolean variables and operations are used; no site-labelling integers and label merges are required.

The 'ants' algorithm has been used on the Edinburgh DAP machines in a recent study of the equilibrium cluster size distribution of the 2D Ising model (Toral and Wall

† Present address: Department of Theoretical Physics, University of Oxford, 1 Keble Road, Oxford OX1 3NP, UK.

0305-4470/87/040985+09$02.50  

1986). We learn that it has also been written independently for the DAP at Queen Mary College in a study of orientational domain clusters in a planar quadrupole model (Allen 1986). In § 3, we report on a Monte Carlo simulation of 2D percolation that uses the 'burning' algorithm and re-examine the numerical evidence for a recent conjecture (Jug 1984, 1985, 1986) predicting novel singularities at the 2D percolation threshold.

## 2. Parallel algorithms for analysing clusters

Consider a configuration of clusters of nearest-neighbour-connected occupied sites on a 2D lattice (including one-site clusters perhaps). These occupied sites may represent open nodes in the percolation problem, up-spins in the Ising model or infected sites in an epidemic model for example. The following parallelised 'ants in the labyrinth' algorithm solves the problem of calculating the cluster sizes.

On a parallel computer such as the DAP, processing elements operate simultaneously at each lattice site. The configuration of occupied sites is stored as a logical array across the lattice (TRUE = occupied, FALSE = unoccupied). Now an ant is put on some occupied site. At the next time-step this ant places offspring on each of the nearest-neighbour occupied sites. The offspring then proceed to multiply likewise until that entire cluster is populated. The parallel feature is that at each time-step the whole of the frontier of the ant population on the cluster advances at once. The ant population at any time is stored as a second logical array (TRUE = ant present, FALSE = no ant present). The cluster thus identified is a set of logicals (the ants) whose size is measured by a logical SUM function and is then removed from the configuration. The procedure is repeated until all clusters are removed.

Of course, this algorithm is not fully parallel because clusters are identified one by one. (We remark as an aside however that for the problem of directed percolation, because there is no 'back-tracking' of paths connecting pairs of sites, it has been possible to write an efficient parallel algorithm for cluster-connectedness properties (Williams and McKenzie 1984).)

If the problem is restricted to calculating only the total number of clusters in the configuration, the following 'burning' algorithm is very efficient. To illustrate it, consider the typical configuration of figure 1($a$) containing three clusters on a $4\times4$ square lattice with planar boundary conditions. Iteration of the central steps of the

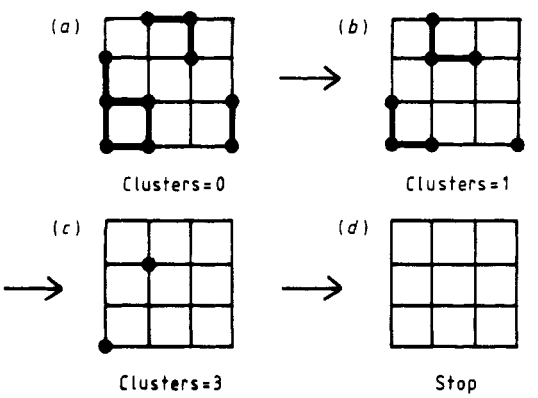


**Figure 1.** Illustrating the 'burning' algorithm for counting clusters.

algorithm systematically reduces all clusters to one-site clusters, each of which augments the cluster total by one and is then removed. The reduction, using Boolean operations only, is performed in parallel across the entire lattice and is depicted in figures 1($b$)-($d$).

The steps are as follows.

(i) Identify all occupied sites having no occupied nearest neighbours to the north and east. There will be four types, shown in figures 2($a$)-($d$). This step identifies the north and east 'coasts' of all clusters.

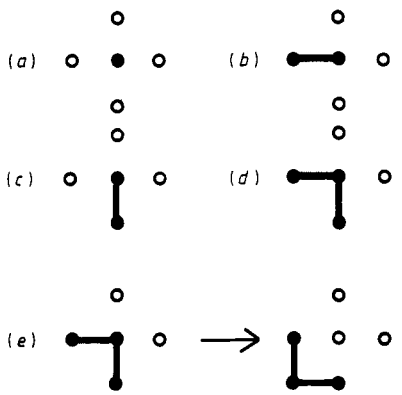


**Figure 2.** ($a$)-($d$), sites with no connections to the north and east; ($e$) indenting the north-east corners to create new dangling ends.

(ii) Then for each ($a$)-type site, count one cluster and remove the site. This step accumulates the cluster total.

(iii) Now remove all sites of types ($b$) and ($c$). This step 'burns off' the north- and east-dangling ends.

(iv) Finally replace the ($d$)-type sites with new occupied sites lying to the south-west (if not already occupied), as shown in figure 2($e$). This step indents the north-east corners to create new dangling ends. Also, since operations are done in parallel, no two clusters are inadvertantly joined at this step, as seen from figures 1($a$) and ($b$).

Repeat steps (i)-(iv) until no sites remain.

We note that for cyclic boundary conditions, some (cyclicly-spanning) clusters only get reduced to minimal loops wrapped round the lattice, so that the algorithm will not terminate. These remaining clusters can be counted by the 'ants' algorithm described above. The algorithm is easily modified to count both site- and bond-occupied clusters on other regular lattices.

For both of the above algorithms the entire configuration is stored in the computer, which limits the analyses to relatively small lattices. For a lattice of linear size $L$, the time to analyse a configuration increases as $L^2$. In practice, on the DAP we enhance the efficiency for lattice sizes $L<64$ by storing and analysing $(64/L)^2$ configurations simultaneously. On the DAP, the 'ants' algorithm analyses 400 000 configurations per hour for the site percolation problem at $p_c=0.5927$ on a square lattice of size $L=32$ with cyclic boundary conditions. The 'burning' algorithm analyses $1.6\times10^6$ configurations per hour for the bond percolation problem at $p_c=0.5$ on the same lattice. Both these figures include the time to generate configurations. The latter figure is to be compared with 70 000 configurations per hour for the same problem on a $L=30$ lattice using a serial algorithm on the Cray 1 machine (Jug 1985).

### 3. 2D percolation singularities

Recently Jug has presented evidence from numerical work and series expansion studies (Jug 1985, 1986) for the result (Jug 1984) that the two-dimensional percolation 'free energy', the mean number of clusters per site $K(p)$ at concentration $p$, has a singular part near the threshold $p_c$ of the form

$$K_s(p) = D(p-p_c)^2 \ln|\ln|p-p_c||\tag{1}$$

instead of the currently accepted form

$$K_s(p) = D(p-p_c)^{2-\alpha}\tag{2}$$

where it is believed that $\alpha = -\frac{2}{3}$ (Stauffer 1979).

Form (1) was derived for the bond percolation problem from a Grassmann path integral (GPI) treatment of the associated bond-diluted Ising-model free energy $f(T, p)$, for which dilute Ising-type critical behaviour was found everywhere along the critical curve $T_c(p)$, including the percolation threshold $T=0$, $p=p_c$.

Numerical evidence for (1) was provided by computing $K'''(p_c, L) \equiv d^3K(p, L)/dp^3|p=p_c$ from the fluctuation formula

$$\begin{aligned}K'''(p, L) = {} & C_1\{\langle N_o n_c\rangle - \langle N_o\rangle\langle n_c\rangle\} + C_2\{\langle N_o^2 n_c\rangle - 2\langle N_o\rangle\langle N_o n_c\rangle + 2\langle N_o\rangle^2\langle n_c\rangle - \langle N_o^2\rangle\langle n_c\rangle\}\\ & + C_3\{\langle N_o^3 n_c\rangle - 3\langle N_o\rangle\langle N_o^2 n_c\rangle + 6\langle N_o\rangle^2\langle N_o n_c\rangle\\ & - 3\langle N_o^2\rangle\langle N_o n_c\rangle - 6\langle N_o\rangle^3\langle n_c\rangle + 6\langle N_o\rangle\langle N_o^2\rangle\langle n_c\rangle - \langle N_o^3\rangle\langle n_c\rangle\}\end{aligned}\tag{3}$$

where $\langle\ \rangle$ is an average over a large number of configurations. $N_o$ and $n_c$ are the number of occupied bonds (or sites) and the number of clusters per site in each configuration, respectively, and the constants $C_1$, $C_2$ and $C_3$ are given by

$$\begin{aligned}C_1 &= 2(p^{-3}+q^{-3})\\ C_2 &= -3(p^{-3}+p^{-2}q^{-1}-p^{-1}q^{-2}-q^{-3})\\ C_3 &= (p^{-1}+q^{-1})^3\end{aligned}\tag{4}$$

where $q = 1-p$. Jug's results supported the finite-size scaling form

$$K'''(p_c, L) = A + B(L) \quad L\to\infty\tag{5}$$

($B(L)$ contains logarithmic corrections) consistent with form (1) and hyperscaling, in preference to

$$K'''(p_c, L) = A + BL^{1/4} \to \infty\tag{6}$$

consistent with the conventional form (2) and hyperscaling.

We present results of a similar Monte Carlo study, using the 'burning' algorithm, which we initiated (Dewar and Harris 1985) on learning of the above work. Kesten (1986) has since noted that form (1) *cannot* be correct since he has shown (Kesten 1983) that for bond or site percolation on the square lattice $K_s(p)$ is twice-continuously differentiable for all $p$ in $[0, 1]$, including $p_c$. Nevertheless, in the absence of a demonstration of *how* the GPI theory fails at $p_c$ and in view of the supporting numerical evidence, it remains of interest to re-examine the question raised by the conjecture (1).

Our results for bond percolation on square lattices (BSQ) and site percolation on triangular lattices (ST) of linear size $L = 8$, 12, 16, 24 and 32 are shown in figure 3($a$), where they are compared with those of Jug (1985) (BSQ, lattice sizes 8, 12, 16, 24 and

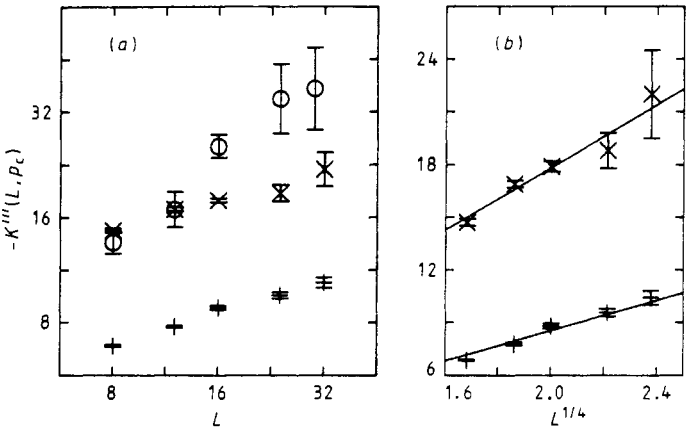


**Figure 3.** $L$ dependence of $-K'''(p_c, L)$ defined in the text. ($a$), log-log plot and ($b$), fit to $BL^{1/4}$ (× = BSQ, + = ST in present work: ○ = BSQ from Jug (1985)).

30). Our data are consistent with the straight line of slope $\frac{1}{4}$ in the log-log plot expected from the conventional result (6). The fit to the form $K'''(p_c, L) = BL^{1/4}$ is shown in figure 3($b$) and has $\chi^2 = 5.57$ (BSQ) and 6.3 (ST, last four points).

In our analysis we have adopted the following approach to reduce the considerable fluctuation in the value of $K'''(p_c, L)$ as given by (3). $K(p)$ contains an analytic contribution

$$K_a(p) = a + b(p - p_c) + c(p - p_c)^2 + \ldots \tag{7}$$

where constants $a$, $b$ and $c$ are either known analytically, estimated from series (Domb and Pearce 1976), or can be estimated from the numerical simulation itself as a useful check on the correctness of the algorithm. When $K'''(p_c, L)$ is averaged over a large but finite number of sampled configurations, the $b$ and $c$ terms in (7) contribute negligibly to its mean but substantially to its fluctuation. Therefore in the fluctuation formula (3) we substitute for $n_c$ the decomposition

$$n_c = n_c^* + b(N_o - Np_c)/N + c(N_o - Np_c)^2/N^2 \tag{8}$$

where $N$ is the total number of nodes (edges) in the lattice for the site (bond) problem, and then subtract off the contribution arising from the $b$ and $c$ terms. The effect of subtracting off this noise is shown graphically in figure 4. In the ST simulation we subtract an additional cubic term in (8) with coefficient $d = 1$, in order to remove the constant $A$ in (6), the bulk value of which is 6.

In addition we recall that, as remarked in § 2, the 'burning' algorithm does not completely reduce certain spanning clusters on lattices with cyclic boundary conditions. Such remaining clusters may be counted by the 'ants' algorithm, although in practice remaining sites were taken to constitute a single spanning cluster contributing one to the cluster total. We have verified that the effect of the relatively rare extra spanning clusters on the value of $K'''(p_c, L)$ is negligible, thus enabling an additional gain in speed of the algorithm to $2 \times 10^6$ configurations analysed per hour for the BSQ problem on a $L = 32$ lattice. In this way error bars have been reduced sufficiently to indicate a clear discrepancy between the two sets of BSQ data in figure 3($a$). Following numerous program checks, we are unable to explain the reason for this discrepancy.

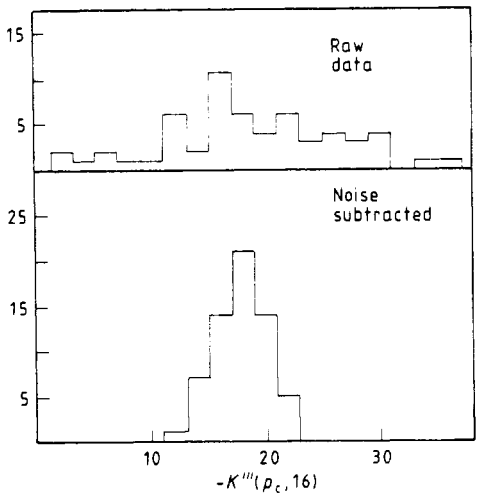


**Figure 4.** Effect of noise reduction (described in the text) for $K'''(p_c, 16)$, bond percolation. 1 box = average over 1 638 400 configurations, total histogram = 62 boxes.

Further evidence in support of the conventional theory of 2D percolation is obtained on comparing the measured value of the amplitude ratio $r_{\text{meas}} \equiv B_s/B_t$ (where subscripts s and t refer to BSQ and ST respectively) with the following value obtained (see the appendix) using finite-size scaling and two-scale-factor universality:

$$r = (\rho_s/\rho_t)^{-1}(D_s\rho_s/D_t\rho_t)^{3/d\nu}. \tag{9}$$

$\rho$ here is the number of lattice sites per unit volume and $D$ is the constant appearing in (2). Using values of $D$ calculated by Domb and Pearce (1976)† we obtain $r = 2.08\ (2)$ while $r_{\text{meas}} = 2.09\ (2)$. Such a good agreement is probably fortuitous as (9) holds only for lattices of the same shape while ours were not (they were squares of side $L$ for BSQ and 60° rhombuses of side $L$ for ST). Taking this into account yields (see the appendix) the bounds

$$1.97\ (2) \leqslant r_{\text{meas}} \leqslant 2.26\ (2). \tag{10}$$

The fact that $r_{\text{meas}}$ lies comfortably within this range strongly suggests that we have reached the asymptotic scaling regime with the lattice sizes that we used, and, together with the observed $L^{1/4}$ behaviour, provides convincing evidence for the validity of the conventional picture of 2D percolation.

## Acknowledgments

We thank G Jug for discussing his work with us prior to publication and for bringing our attention to the work of Domb and Pearce. We also thank the SERC for financial support.

† Note that the $k_L(p)$ of this reference for BSQ is related to $K(p)$ by $K(p) = 2k_L(p) + (1-p)^4$.

## Appendix. Critical amplitudes in the percolation 'free energy' $K(p, L)$ and its derivatives

Let $\varepsilon = (p - p_c^x)/p_c^x$ where $x$ refers to the lattice type. From eqution (2) and hyperscaling, the singular part of the bulk 'free energy' per site has the form

$$K_s(p, \infty) = D_x(p_c^x)^{d\nu}(K_0^x \xi_x)^{-d} \tag{A1}$$

where $\xi_x^{-1} = K_0^x \varepsilon^\nu$ is the inverse correlation length. If $\rho_x$ is the density of lattice sites we may also write

$$K_s(p, \infty) = F\rho_x^{-1}\xi_x^{-d} \tag{A2}$$

where two-scale-factor universality (Stauffer *et al* 1972) tells us that $F$, the (singular) free energy of a volume $\xi_x^d$, is a universal constant independent of lattice type. (A1) and (A2) give

$$(K_0^x/K_0^y)^d = (\rho_x D_x/\rho_y D_y)(p_c^x/p_c^y)^{d\nu}. \tag{A3}$$

Now invoking finite-size scaling when $L, \xi_x \to \infty$ with $L/\xi_x$ arbitrary and fixed, $K_s'''(p, L)$ will have the scaling form

$$K_s'''(p, L) = C_x \xi_x^{(3-d\nu)/\nu} g(L/\xi_x) \tag{A4}$$

where $g$ is a universal function, while from (A1) we find that the bulk $K_s'''(p, \infty)$ diverges as

$$K_s'''(p, \infty) = d\nu(d\nu - 1)(d\nu - 2)D_x(p_c^x)^{d\nu-3}(K_0^x \xi_x)^{(3-d\nu)/\nu}. \tag{A5}$$

By demanding that (A5) be recovered from (A4) when $L \to \infty$ with $\xi_x$ fixed we find

$$C_x/C_y = (D_x/D_y)(p_c^x/p_c^y)^{d\nu-3}(K_0^x/K_0^y)^{(3-d\nu)/\nu}. \tag{A6}$$

Substituting (A3) into (A6) to eliminate $K_0^x/K_0^y$ we obtain the result

$$C_x/C_y = (\rho_x/\rho_y)^{-1}(\rho_x D_x/\rho_y D_y)^{3/d\nu}. \tag{A7}$$

For lattice blocks of the same shape, this will be the ratio of critical amplitudes $r = B_x/B_y$ where (in the conventional theory) $K_s'''(p_c, L) = BL^{1/4}$. This gives equation (9). In the present case, $d = 2$ and $\nu = \frac{4}{3}$ (conventionally), for lattice types $x =$ square and $y =$ triangular, $\rho_x = 1$ and $\rho_y = 2/\sqrt{3}$, and $D_x = -8.48\,(3)$, $D_y = -4.370\,(15)$ (Domb and Pearce 1976) giving

$$r = 2.08\,(2). \tag{A8}$$

To take into account the difference in shapes of the two lattice blocks in our simulations, we place bounds on the measured amplitude ratio as follows.

Consider a critical quantity $F(L, x, s)$ (such as $K_s'''(p_c, L)$) where $L$ is the size of the lattice, equal to $(\text{volume})^{1/d}$, $x$ is the lattice type and $s$ is the shape of lattice block. For large $L$,

$$F(L, x, s) = B(x, s)L^\mu \tag{A9}$$

where $\mu$ is the critical exponent (we take $\mu > 0$ here). If a different block $L', s'$ can be placed entirely within the first block $L, s$ then clearly

$$L' \leqslant L$$

and we assume

$$F(L', x, s') \leqslant F(L, x, s)$$

i.e.

$$B(x, s')L'^{\mu} \leqslant B(x, s)L^{\mu}.$$

Suppose an $s'$ block which *just* fits inside the $L, s$ block has

$$L' = \alpha L \tag{A10}$$

for some $\alpha$ in $[0, 1]$. Then we have

$$B(x, s')\alpha^{\mu} \leqslant B(x, s). \tag{A11}$$

For lattice types $x$ and $y$ we can calculate the ratio

$$r = B(x, s)/B(y, s) = B(x, s')/B(y, s') \tag{A12}$$

as in (A8), for blocks of the same shape. For blocks of different shape the ratio will be

$$r_{\mathrm{d}} = B(x, s)/B(y, s'). \tag{A13}$$

Thus from (A11)-(A13)

$$r \leqslant \alpha^{-\mu} r_{\mathrm{d}}. \tag{A14}$$

Now suppose the block $L, s$ just fits inside another block $L'', s'$ so that

$$L = \beta L'' \tag{A15}$$

for some $\beta$ in $[0, 1]$. Then by a similar argument we have

$$r \geqslant \beta^{\mu} r_{\mathrm{d}}. \tag{A16}$$

(A14) and (A16) then place bounds

$$\alpha^{\mu} r \leqslant r_{\mathrm{d}} \leqslant \beta^{-\mu} r \tag{A17}$$

on the ratio of amplitudes for blocks of different shape.

In our simulations, the blocks are: $s$ = square, size $L$; $s' = 60°$ rhombus, side $L$, so size $L' = (3^{1/4}/2^{1/2})L$. With $\mu = \frac{1}{4}$ (conventional value), the ratio $r_{\mathrm{meas}}$ we actually measure is therefore

$$r_{\mathrm{meas}} = (2^{1/2}/3^{1/4})^{1/4} r_{\mathrm{d}}. \tag{A18}$$

From the geometry in figure 5, we find the parameters $\alpha = \beta = 3^{-1/4}$; putting this, (A18) and (A8) into (A17) we find

$$1.97\ (2) \leqslant r_{\mathrm{meas}} \leqslant 2.26\ (2).$$

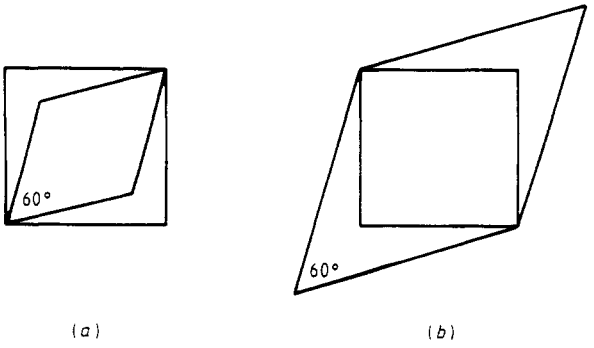


**Figure 5.** Configurations of blocks that place bounds on $r_{\mathrm{d}}$. (*a*) gives $\alpha = 3^{-1/4}$, (*b*) gives $\beta = 3^{-1/4}$, defined in the text.

## References

Allen M P 1986 Private communication
Dewar R and Harris C K 1985 *Proc. 6th Int. Conf. on Fractals in Physics* ed L Pietronero and E Tosatti (Amsterdam: North-Holland) p 145
Domb C and Pearce C J 1976 *J. Phys. A: Math. Gen.* **9** L137
Essam J W 1980 *Rep. Prog. Phys.* **43** 833
Family F and Landau D P (ed) 1984 *Kinetics of Aggregation and Gelation* (Amsterdam: North-Holland)
Gunton J D, San Miguel M and Sahni P S 1983 *Phase Transitions and Critical Phenomena* vol 8, ed C Domb and J L Lebowitz (New York: Academic)
Hambrusch S E 1983 *SIAM J. Comput.* **12** 354
Hockney R W and Jesshope C R 1981 *Parallel Computers* (Bristol: Adam Hilger)
Hoshen J and Kopelman R 1976 *Phys. Rev.* B **14** 3438
Jug G 1984 *Phys. Rev. Lett.* **53** 9
—— 1985 *Phys. Rev. Lett.* **55** 1343
—— 1986 *J. Phys. A: Math. Gen.* **19** 1459
Kesten H 1983 *Percolation Theory For Mathematicians* (Boston, MA: Birkhauser)
—— 1986 *Phys. Rev. Lett.* **56** 1210
Redner S 1982 *J. Stat. Phys.* **29** 309
Stauffer D 1979 *Phys. Rep.* **54** 1
Stauffer D, Ferrer M and Wortis M 1972 *Phys. Rev. Lett.* **29** 345
Toral R and Wall C 1986 *Edinburgh preprint*
Williams J K and McKenzie N D 1984 *J. Phys. A: Math. Gen.* **17** 3343