On the hulls of directed percolation clusters

1997 J. Phys. A: Math. Gen. 30 6679

(http://iopscience.iop.org/0305-4470/30/19/011)

J. Phys. A: Math. Gen. **30** (1997) 6679–6691. Printed in the UK PII: S0305-4470(97)82835-1

On the hulls of directed percolation clusters

A L Owczarek†, A Rechnitzer, R Brak and A J Guttmann
Department of Mathematics, University of Melbourne, Parkville, 3052, Australia

Received 24 March 1997

Abstract. The properties of the hulls of directed percolation clusters are studied. The scaling and finite-size scaling of many quantities around the percolation threshold p_c are derived and a novel Monte Carlo algorithm, which is more than twice as fast as the standard algorithm at p_c, has been formulated to study these properties. Simulations have been conducted that enable an estimation of all the exponents involved. In particular, the central exponent, x, relating the average hull length of clusters to their mass, has been estimated to be 0.773(4) at the percolation threshold. This same exponent is estimated to be 0.905(5) for $p < p_c$. Thus, this second value should hold for directed animals.

1. Introduction

Directed percolation in two dimensions [1, 2], unsolved on any regular lattice, remains an intriguing example of a model with critical behaviour that is simultaneously not conformally invariant and yet apparently complex. One question that has made directed percolation to be an object of study by several authors [3–5] has been the rationality or otherwise of its critical exponents. Exponent rationality holds for all conformally invariant two-dimensional models (see for example [6]). Also, a varied collection of physical problems, such as fluid flow in porous media, epidemics, forest fires, Reggeon field theory, various chemical reactions, and population dynamics have kept directed percolation as a canonical model of study for the past 20 years.

Isotropic, or standard, percolation in two dimensions [7, 8] has been extensively studied for many years and its critical behaviour is now well understood (even in the absence of a mathematically rigorous solution). Critical exponents associated with a whole menagerie of physical quantities have been calculated. Included in this list is the set of exponents associated with the scaling of the standard properties, such as size and number of clusters with the length of their external perimeters (h). Depending on its precise definition the external perimeter is known as the hull. This scaling is in contrast to the more usual one in terms of the mass, s (being number of sites), of the clusters. Central to this collection is the exponent, x, that connects, via scaling relations, the exponents derived using hull lengths and those using the mass as the basic scaling variable. It is defined by the relationship between the average hull length of a given mass and that mass:

$$\langle h \rangle \sim s^x \qquad \text{as } s \to \infty \tag{1.1}$$

and it is clear that $\frac{1}{2} \leqslant x \leqslant 1$. In fact, it attains three different values depending on whether the concentration, p, is above, at, or below, the critical concentration p_c.

† E-mail address: aleks@maths.mu.oz.au

0305-4470/97/196679+13$19.50

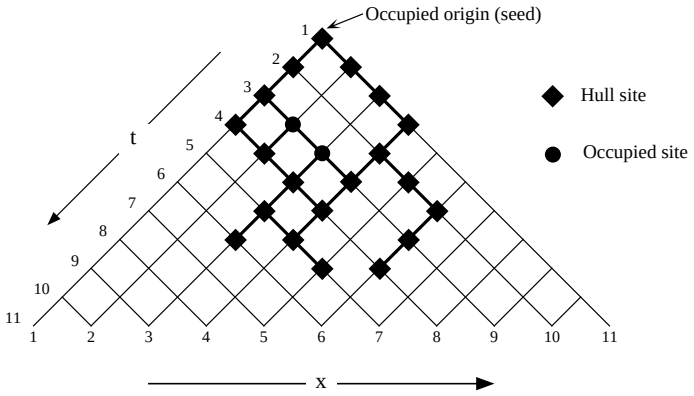

Figure 1. A directed percolation cluster of mass $s = 22$, hull $h = 20$, length $v = 9$ and width $w = 4$.

In this paper we study the hull scaling exponents for directed percolation in two dimensions. In particular we have estimated the value of the exponent x to be 0.773(4) at $p = p_c$. We note that this is close to the appealing rational $\frac{7}{9}$. However, for directed problems, for which conformal invariance does not hold, there is no reason to expect rational exponents, and past experience [9] has shown that these appealing rational fractions are often just good approximations. In the course of our calculations we have introduced several novel algorithms for simulating directed percolation clusters. These have a speed advantage over the traditional method of simulation. Our results are summarized in tables 3 and 4.

This paper is organized as follows. In section 2 we define the properties to be calculated and review their associated scaling theory. In section 3 we introduce the algorithms utilized for our simulations. We have set out our results and commented on their accuracy and precision in section 4, with a brief concluding summary following in section 5.

2. Definitions and scaling theory

We have considered directed site percolation on a quadrant of the square lattice seeded from a corner as shown in figure 1. Starting from the origin one can produce a cluster by occupying the site of the lattice at (x, t) with probability p provided either $(x - 1, t - 1)$ or $(x, t - 1)$ is occupied. This produces a cluster where the origin is connected to each occupied site by a *directed* path of occupied sites. The number of occupied sites in the cluster is denoted by s (and called the mass). If an occupied site of the cluster is a perimeter site (one adjacent to an unoccupied site) and if further it can be connected by a path of unoccupied sites (via nearest and next-nearest neighbours) to the edge of the lattice then it is deemed to be a hull site of the cluster. Hence, the hull is the external perimeter of the cluster itself (whereas often in percolation theory the 'external perimeter' denotes the unoccupied sites adjacent to the hull). The number of hull sites is denoted h.

We define the *normalized cluster number* as

$$n_s(s, p) = \frac{\Pr(\text{origin} \in s\text{-cluster})}{s} \tag{2.1}$$

which is identical to the definition used in isotropic percolation. Of course, in a simulation of this directed percolation problem one would naturally estimate $N_s = sn_s$ as the average

proportion of clusters produced that are of mass (size) s. The *normalized hull number* $n_h(h, p)$ is defined in an analogous way. Note that in our set-up the probability of obtaining a cluster is 1. Hence, the probability that the origin belongs to a infinite cluster $P(p)$ satisfies

$$P(p) + \sum_{s=1}^{\infty} sn_s = P(p) + \sum_{h=1}^{\infty} hn_h = 1. \tag{2.2}$$

Canonical functions of interest are the *mean cluster size*, $S(p)$ and *mean hull size*, $H(p)$ which are defined as

$$S(p) = \frac{\sum_{s=1}^{\infty} s^2 n_s}{\sum_{s=1}^{\infty} sn_s} \tag{2.3}$$

and similarly

$$H(p) = \frac{\sum_{h=1}^{\infty} h^2 n_h}{\sum_{h=1}^{\infty} hn_h}. \tag{2.4}$$

The *geometric* size of the clusters are also of interest. One wants to calculate some measure of the horizontal and vertical extent of the clusters, such as the radius of gyration. For ease of calculation we have simply chosen to use as a measure of the vertical size, v, the vertical length (or 'calliper' length) of the cluster in the variable t (that is, the value of t for which there is at least one occupied site such that at $t + 1$ there are no occupied sites). For the horizontal size we have chosen the maximum width, w, of the rows of the cluster, over all the rows of the cluster ('calliper width'). We shall use these same symbols to denote the averages over all clusters of mass s, that is $v(s, p)$ and $w(s, p)$, and with primes for averages over all clusters of particular hull lengths. Averages over all clusters weighted by the the probability of obtaining clusters of mass s or hull h are given as

$$V(p) = \frac{\sum_{s=1}^{\infty} vsn_s}{\sum_{s=1}^{\infty} sn_s} \tag{2.5}$$

and

$$V'(p) = \frac{\sum_{h=1}^{\infty} v'hn_h}{\sum_{h=1}^{\infty} hn_h} \tag{2.6}$$

respectively for the vertical size. Similar equations define $W(p)$ in terms of $w(s, p)$ and $W'(p)$ in terms of $w'(s, p)$. The calliper length and width should also be of direct physical interest when modelling a situation where the maximum extent of some physical process, such as an epidemic, is important.

Of most interest is the scaling in the vicinity of the critical point. The single variable scaling assumption for the normalized cluster numbers (see [8] and references therein) is central to our current understanding of the behaviour of the system near the percolation threshold and is given by

$$n_s(s, p) \eqsim s^{-\tau} f((p - p_c)s^{\sigma}). \tag{2.7}$$

This involves the two basic exponents τ and σ. For an exact definition of the 'scales as' symbol, $\eqsim$, see Brak and Owczarek [10]. The behaviour of the function $f(z)$ for large argument must match the scaling of n_s for p away from p_c. The function $f(z)$ is non-zero at the origin. An analogous assumption can be written down for the normalized hull numbers as

$$n_h(h, p) \eqsim h^{-\tau'} f'((p - p_c)h^{\sigma'}) \tag{2.8}$$

with the same conditions on f'.

To connect these two forms and more precisely their exponents we use equation (1.1) and the assumption that the number of clusters, N_s, counted by size is related to the number of clusters, N_h, counted by hull length with the following equation:

$$\sum_{s=1}^{s_0} N_s \approx \sum_{h=1}^{h_0} N_h \tag{2.9}$$

where h_0 is the average hull size of clusters of mass s_0. That is, the number of clusters of size $\leqslant s_0 \approx$ number of clusters of hull $\leqslant h_0$. Scaling arguments [11, 12] can then be used to show that

$$\frac{2-\tau}{2-\tau'} = \frac{\sigma}{\sigma'} = x. \tag{2.10}$$

It is worth noting that this scaling law is *different* to the one that holds in the case of isotropic percolation. This is simply because in directed percolation a single realization of the system produces a single cluster whereas in isotropic percolation it produces a distribution of clusters. In isotropic percolation one has $(1-\tau)/(1-\tau') = \sigma/\sigma' = x$ which has been verified using Monte Carlo simulations [13].

The exponents associated with the other quantities we defined above are

$$P(p) \sim |p - p_c|^{\beta} \qquad \text{as } p \to p_c^+ \tag{2.11}$$

$$S(p) \sim |p - p_c|^{\gamma} \qquad \text{as } p \to p_c \tag{2.12}$$

$$H(p) \sim |p - p_c|^{\gamma'} \qquad \text{as } p \to p_c \tag{2.13}$$

$$V(p) \sim |p - p_c|^{\nu_\parallel - \beta} \qquad \text{as } p \to p_c \tag{2.14}$$

$$V'(p) \sim |p - p_c|^{\nu'_\parallel - \beta} \qquad \text{as } p \to p_c \tag{2.15}$$

$$W(p) \sim |p - p_c|^{\nu_\perp - \beta} \qquad \text{as } p \to p_c \tag{2.16}$$

$$W'(p) \sim |p - p_c|^{\nu'_\perp - \beta} \qquad \text{as } p \to p_c \tag{2.17}$$

as functions of p, and

$$v(s, p_c) \sim s^{\overline{\nu}_\parallel} \qquad \text{as } s \to \infty \tag{2.18}$$

$$v'(h, p_c) \sim h^{\overline{\nu}'_\parallel} \qquad \text{as } h \to \infty \tag{2.19}$$

$$w(s, p_c) \sim s^{\overline{\nu}_\perp} \qquad \text{as } s \to \infty \tag{2.20}$$

$$w'(h, p_c) \sim h^{\overline{\nu}'_\perp} \qquad \text{as } h \to \infty \tag{2.21}$$

as functions of the two sizes.

These exponents are not independent and should satisfy the following relations:

$$\nu_\parallel = \nu'_\parallel \tag{2.22}$$

with

$$\overline{\nu}_\parallel = \sigma \nu_\parallel \qquad \text{and} \qquad \overline{\nu}'_\parallel = \sigma' \nu_\parallel \tag{2.23}$$

and hence

$$\overline{\nu}_\parallel = x \overline{\nu}'_\parallel \tag{2.24}$$

with an analogous set for $\nu_\perp$ and its relatives. Also, other scaling arguments using (2.7) and (2.8) give

$$\gamma = \frac{3-\tau}{\sigma} \qquad \text{and} \qquad \gamma' = \frac{3-\tau'}{\sigma'} \tag{2.25}$$

and

$$\beta = \frac{\tau - 2}{\sigma} = \frac{\tau' - 2}{\sigma'}. \tag{2.26}$$

2.1. Finite-size scaling

In order to extract values for the exponents defined via scaling in the concentration p, such as (2.11), it may be advantageous to perform a finite-size scaling analysis, and scale against a cut-off instead. We have found it convenient and a good use of the data for the ordinary scaling analysis to fix a cut-off in the mass, or hull length, of the clusters generated, so that $s < s_{\max}$, or $h < h_{\max}$ respectively. We used the same data for a finite-size scaling analysis.

One can argue that for a finite system of maximum size $s_{\max}$, or $h_{\max}$, that the geometric size measures should scale with the cut-offs as

$$V \propto {s_{\max}}^{(\nu_\parallel-\beta)\sigma} g((p-p_c){s_{\max}}^{\sigma}) \tag{2.27}$$

$$V' \propto {h_{\max}}^{(\nu_\parallel-\beta)\sigma'} g'((p-p_c){h_{\max}}^{\sigma'}) \tag{2.28}$$

where g and g' are scaling functions, and are expected to be unimodal: for large enough 'maximum sizes' plots of V and V' are unimodal with peaks at p_{peak} (note that p_{peak} is a function of the maximum size $s_{\max}$ or $h_{\max}$). Hence, at $p = p_c$, or at $p = p_{\text{peak}}$, we expect

$$V \sim {s_{\max}}^{(\nu_\parallel-\beta)\sigma} \tag{2.29}$$

$$V' \sim {h_{\max}}^{(\nu_\parallel-\beta)\sigma'}. \tag{2.30}$$

Again there are analogous equations for W and W' with $\nu_\perp$ substituted for $\nu_\parallel$. These equations can then be used to analyse the data produced in simulations for the exponents associated with the mean cluster length and width.

3. Generation algorithms

We have used two algorithms to generate data. The first is the canonical Markov algorithm which treats two-dimensional directed site percolation as a one-dimensional branching Markov process. This has been the main method used in the past, and has proved to be simple to encode and fast on execution. To calculate the hull length, whole clusters need to be stored and so there are more stringent memory limitations than with the usual implementation of this algorithm. Using this algorithm and calculating the hull we were able to simulate clusters of size $s = 2^{17}$ on a DEC Alpha 250/4/266 using approximately 44 MB RAM.

However, we have developed algorithms that generate the external hull of a cluster, and as little of the internal structure as is necessary, making them even faster though more difficult to code. The exponent x relating mass to hull length scaling (1.1) cannot be calculated from these simulations alone.

3.1. Hull algorithms

An algorithm for isotropic percolation that iteratively generates the external hull of a cluster was formulated some time ago [14, 13]. We shall refer to this as the ZCS algorithm. Because the number of hull sites scales as $h \sim s^x$, where $\frac{1}{2} \leqslant x < 1$, one expects that in general $s \gg h$, and hence that the time taken to generate a cluster hull is much less than that taken to generate a full cluster. The ZCS algorithm has been used to accurately estimate p_c for isotropic site percolation [13], and confirm the values of hull exponents predicted by the (isotropic) hull scaling law. It was natural to consider adapting the idea of this algorithm to the case of directed percolation. However, as one can see, below this application is not straightforward. With the addition of some extra parts though the algorithm can be adapted to the case of directed site percolation on the square lattice.

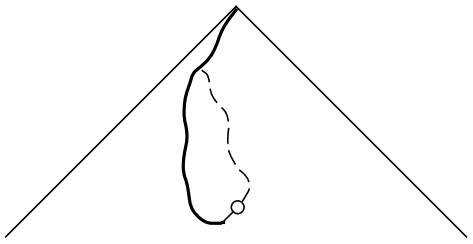

Figure 2. Before the walker can move to the new site it must first check to see if a support exists (broken curve).

3.1.1. A single-walker algorithm for directed percolation hulls. We first describe the single-walker algorithm, as its features and limitations are reflected in the two-walker algorithm we have developed for our study. This *single-walker* algorithm describes the movement of a single walker starting from the origin, moving anticlockwise laying down the hull of the cluster.

Adding anisotropy to the hull walker of ZCS is itself not difficult. However, it is complicated by two features. First, the directions that the walker is allowed to move from the current site depend upon the absolute direction just moved (rather than on only the relative direction). Secondly, and more significantly is the fact that the hull of a directed percolation cluster is not necessarily a directed percolation cluster itself. That is to say, in isotropic percolation the hull is itself a cluster so the internal structure of the cluster does not matter. Hence, the hull is independent of the internal structure so that each realization of the hull can be achieved with the correct probability by constructing the hull alone. The directed percolation hull structure depends upon the internal structure since all sites must be *supported* by a directed path of occupied sites from the origin to that site. To obtain each hull with the correct probability in a simulation one must exclude internal configurations that do not satisfy the definition of directed percolation. Practically, this means that when the walker is about to step in certain directions (as described below), we must first test to see if the site is *supported* (see figure 2). The support of the site is checked for, and generated, by a subroutine called *orphan.*

The orphan subroutine generates a tree of pseudo-occupied (not included in the cluster definition) sites in an attempt to construct a directed path back to part of the cluster already occupied (such as the hull). It does so by always trying to find the 'left-most' such path. If it succeeds that single-directed path becomes a designated part of the directed percolation cluster. The rest of the generated tree is inaccessible to the continuing algorithm, since it is to the left of the path. If one were to include in the definition of the cluster the whole generated tree this subroutine would produce non-directed percolation clusters. The pseudosites generated are, however, required to obtain the correct probability for a given hull. As such, this feature is both peculiar and novel but nevertheless true. A full proof of the equality of the probabilities of generation by our algorithm, and by the definition of directed percolation, is long and tedious. The essence of the proof utilizes the fact that any site not visited by either algorithm can be arbitrarily designated as occupied or not so long as the probability of doing so adds up to 1.

The single-walker algorithm generates a two-dimensional structure and naively requires the whole square lattice on which to work, though memory management procedures such as data blocking can be employed to alleviate this constriction. The algorithm requires four possible site states: *blank, vacant, occupied* and *hull*. A blank site is one that has not yet been visited by the algorithm and so could be either vacant or occupied. The orphan routine sets occupied sites as occupied, and the single-walker sets occupied sites as hull. For simplicity, directions from (x, t) to (x', t') will be abbreviated as in table 1.

Table 1. Directions, their abbreviations and the translations they represent on the directed square lattice.

Abbreviation	Direction	Δx	Δt
dl	down-left	$x' = x$	$t' = t + 1$
dr	down-right	$x' = x + 1$	$t' = t + 1$
ul	up-left	$x' = x - 1$	$t' = t - 1$
ur	up-right	$x' = x$	$t' = t - 1$

Table 2. Movement table for the single-walker algorithm giving the movement preferences (see figure 3 for an example) and the required tests for each possible previous movements (see table 1 for abbreviations). The tests are denoted in parentheses by (b), (s) and (n). The test (b) is whether or not the site (in that direction) is blank and then a random number generated between 0 and 1 is less than p; and the test (s) is whether or not the site is supported, which is tested by the orphan subroutine. The case when a test is unnecessary is denoted by (n): true is automatically returned.

Direction moved	First	Second	Third	Fourth
down-left	dl(b)	dr(b)	ur(n)	○
down-right	dl(b)	dr(b)	ur(s)	ul(n)
up-left	ur(s)	ul(n)	○	○
up-right	dr(b)	ur(s)	ul(n)	○

Starting from the origin the algorithm proceeds as follows.

(I) Set the current site as the hull.

(II) Remembering the direction just moved, make the next move according to table 2. Use the row appropriate to the direction just moved and:

(a) check to see if the first test is true (that is, the one in the first column);

(b) if the test is true proceed to (III);

(c) if otherwise set the tested site as vacant and repeat (II)(a) with the next preference (next column).

(III) If the walker tries to move in an upward direction from the origin the algorithm terminates, otherwise execute the move and return to (I).

Initially it is assumed that the walker has moved down-left to the origin.

On long runs, this algorithm proved to be slower than the Markov (plus hull walker) near p_c. Closer examination of generated clusters showed that the orphan routine (see figure 4) was 'overflowing' when it was searching for supports. It was searching a very wide area for supporting sites and this area was frequently found to be outside the final cluster hull and at times it was approximately the same size as the (virtual) cluster itself. This problem was overcome by using two walkers in parallel.

3.1.2. The dual-walker algorithm In the dual-walker algorithm two walkers move in tandem, with one always waiting for the other to catch up. Thus, before another step downwards is taken both are on the same row. This avoids the overflow problem that the single walker faced (see figure 5), as now the only area that will be searched by orphan routines is strictly between the paths the two walkers have set, i.e. within the cluster.

Both walkers move in much the same way as the single walker described above, but now one walker is left biased (as above), and the other is right biased. Similarly there are two orphan routines, one left biased and the other right biased.

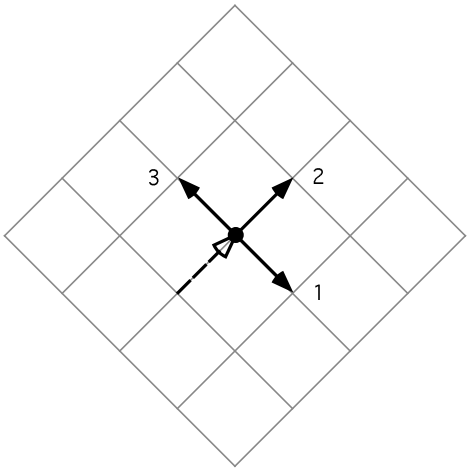

Figure 3. The order preference of moves after an up-right step for a left-biased walker.

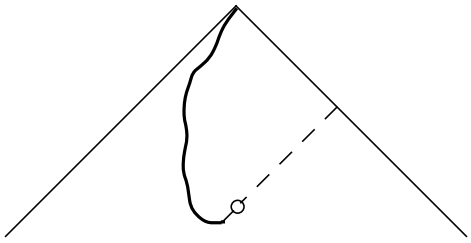

Figure 4. The orphan routine can possibly search anywhere between the currently defined hull and the right-hand boundary of the lattice.

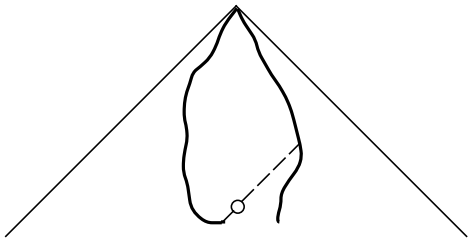

Figure 5. When there are two walkers in parallel, the orphan routine can only search strictly inside the cluster.

Also, because there is the possibility that both walkers may be at the same site at the same time there is a *together* walker subroutine, to account for this:

(1) set the current site as the hull;

(2a) if there is a blank site dl, occupy it with probability p (set as hull) or else set it as vacant;

(2b) if there is a blank site dr, occupy it with probability p (set as hull) or else set it as vacant;

(3a) if the sites at both dl and dr have now been occupied, then return to the main part of the dual-walker algorithm with two separate walkers;

(3b) if only one site has been occupied then move there and repeat from 1;

(3c) otherwise, there are no unoccupied sites below, and so the whole algorithm terminates.

The dual-walker algorithm proved difficult to encode, but was faster than both the single-walker and the Markov (plus hull walker) algorithms when simulating at p_c. It was combined with an appropriate memory management code (data blocking [13]) and used to generate the hull-specific data which we analysed.

4. Results and discussion

The Markov and dual-walker algorithms were used to generate clusters at various values of p and system size. One may be tempted to use data from clusters of length less than

some cut-off L to compute exponents. However, there is a crucial loss of data; if clusters are limited by a length L then the cluster numbers n_s, where $s > L$, lose the contribution of clusters with length greater than L. This leads to inaccurate distributions for $s > L$, and the use of ordinary scaling for analysis is then inappropriate. Instead we set the system size in terms of the number of sites, with maximum cluster size set at $s_{\max}$ for the Markov simulations, and maximum hull size $h_{\max}$ for the dual-walker simulations.

The comparison of the speed of the two algorithms needs careful analysis. Since we are looking for distributions of lengths and widths as well as hulls, we need to compare the Markov and dual walker when they are generating the same distributions. The critical factor in this comparison is the hull scaling exponent x. The further the value of x is from 1, the smaller the hull of an average cluster of particular mass is, and hence the greater the time saved by the dual-walker algorithm is. It was found that when the cut-offs were set as $h_{\max} \approx 25\,000$ and $s_{\max} \approx 131\,072$, the algorithms generated approximately the same distributions, but the dual-walker algorithm was about 2.5 times as fast as the Markov (plus hull finding) algorithm.

Depending on the algorithm, for each cluster the following quantities were calculated:

- size s, the total number of occupied sites in the cluster;
- hull h, the total number of occupied sites in the external hull as defined previously;
- 'calliper' length v, the maximum length of the cluster;
- 'calliper' width w, the maximum width of the rows of the cluster.

Using the Markov algorithm and a simple walk-around-the-hull, clusters were first generated in full and then their hulls determined. This gave distributions of each of s, h, v and w. A total of 1.8×10^6 clusters were generated at p_c using this method up to the cut-off of $s_{\max} = 131\,072$, of which 1.3×10^6 had mass less that the maximum. By calculating the average hull length as a function of mass the hull scaling exponent x for $p < p_c$, $p = p_c$ and $p > p_c$ was estimated. The unprimed length scale exponents $\overline{\nu}_{\parallel}$ and $\overline{\nu}_{\perp}$ were also calculated. With the exponent τ found directly from the distribution of mass, these provided a check on our analyses by allowing a comparison with recent series estimates [5]. The dual-walker algorithm was used to generate h, v and w distributions in a wide range of p, with special attention paid to p_c. A total of 4.5×10^6 clusters were generated at p_c using the dual-walker algorithm up to the cut-off of $h_{\max} = 32\,768$ of which 3.3×10^6 had hull less that the cut-off. For the sake of general comparison the simulation of 10^5 cluster hulls at $p = p_c$ with a cut-off of $h_{\max} = 32\,768$ took 2.5 CPU hours on a DEC Alpha 250/4/266.

We have performed several different analyses of the data. Ordinary scaling and finite-size scaling relations allow us to calculate exponents by examining the behaviour of the following distributions:

- quantity versus hull or cluster size at $p = p_c$, using ordinary scaling such as (2.18);
- quantity peak height versus system size, using finite-size scaling such as (2.29).

In the ordinary scaling analyses, the value of p_c used was that provided by the precise series estimate of Jensen [5], that is $p_c = 0.705\,4853$ (which is more precise than our simulations could achieve).

4.1. Ordinary scaling analysis

From the Markov simulations data we extracted estimates of the exponents x at p_c and x below p_c, $\sigma\nu_{\parallel}$, $\sigma\nu_{\perp}$, and τ using ordinary scaling assumptions. From the dual-walker simulation we calculated estimates of $\sigma'\nu_{\parallel}$, $\sigma'\nu_{\perp}$ and τ'. To do this we analysed the data in several different ways. Generically we calculated local exponent estimates from (weighted)

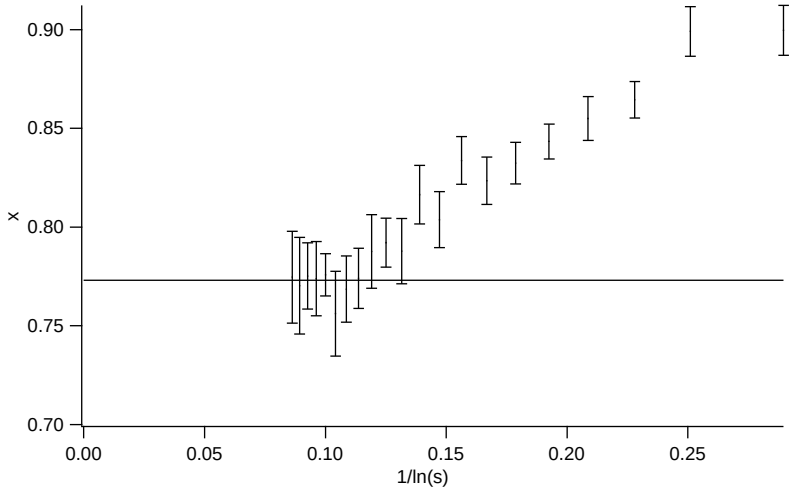

Figure 6. A plot of local estimates of the exponent x against $1/\log(s)$, with (95% confidence interval) error-bars. The horizontal line represents our final estimate.

linear regressions on small sections of the data and then plotted these against the reciprocal of the logarithm of the mean position of the local regression (for visual purposes only). Fortunately our estimates seemed to converge within the range simulated and so we took a final estimate from the (weighted) average of the last few local exponent estimates. We did not attempt any further extrapolation. Important to this process was the determination of the range over which the local exponent estimates were essentially constant. In figure 6 we plot the local estimates for the exponent x obtained by first calculating the average hull length for each value of mass from our simulations, then binning those results in logarithmically equal bins of length $\Delta \ln(s) = 0.02$, and extracting local slopes from a weighted linear regression of disjoint contiguous sets of 20 points. In this case we used the last five points to provide an estimate of x. We varied the binning size and range over which the linear regressions were calculated to test the robustness of our estimate. We utilized bin sizes of 0.02, 0.05 and 0.4 as well as simple linear bins of size 1, varying also the number of bins used in the linear regressions so as to make the error bars reasonable. We are therefore confident that our estimates and associated error bars represent a best estimate from the largest end of our data. We however do not provide an independent estimate of the systematic error but given the trend of the data this should be less than the statistical error quoted. The exponent estimates obtained in the above way are listed in table 3.

The accuracy of our method can be gauged by considering the estimates of the mass-based exponents $\sigma\nu_{\parallel}$, $\sigma\nu_{\perp}$ and τ, noting that these were calculated from the Markov algorithm simulations. The internal consistency of these estimates can also be gauged by using scaling relations. One can obtain an independent estimate of the exponent x by using relation (2.10) and the pairs of estimates for $\sigma\nu_{\parallel}$ and $\sigma'\nu_{\parallel}$, and $\sigma\nu_{\perp}$ and $\sigma'\nu_{\perp}$, and finally τ and τ'. These give 0.774(9), 0.774(10) and 0.772(14) respectively, which are clearly consistent with our direct estimate of 0.773(4).

Below p_c, at $p = 0.65$, the exponent x was also estimated (see table 3). This value should be the value of x that holds for directed animals since the large s (and so h) behaviour of clusters are dominated by directed animals (analogous assumptions are true for percolation: see [8]). The exponent x was also estimated above p_c at $p = 0.715$, and

Table 3. Our best estimates for the exponents obtained via a standard scaling analysis of our data.

Quantity	Exponent	Our estimate	Series results
$\langle h\rangle(s)$ for $p < p_c$	x	0.905(5)	uncalculated
$\langle h\rangle(s)$ at p_c	x	0.773(4)	uncalculated
$v(s, p_c)$	$\sigma\nu_{\parallel}$	0.680(5)	0.678 818(22)
$w(s, p_c)$	$\sigma\nu_{\perp}$	0.431(8)	0.429 431(14)
$n_s(s, p_c)$	τ	2.1077(13)	2.108 25(8)
$v'(h, p_c)$	$\sigma'\nu_{\parallel}$	0.879(4)	uncalculated
$w'(h, p_c)$	$\sigma'\nu_{\perp}$	0.557(5)	uncalculated
$n_h(h, p_c)$	τ'	2.1395(8)	uncalculated

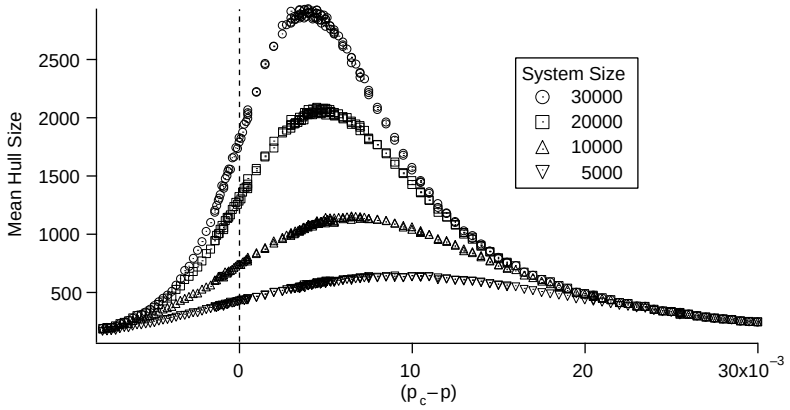

Figure 7. Plots of mean hull size, H, versus $p_c - p$ for various system sizes.

was found to be steadily decreasing as s became larger. Given this lack of convergence we did not attempt to estimate the value of the exponent. However, above p_c, clusters are expected to scale as two-dimensional objects. Consequently we expect the hull to scale as the surface of a two-dimensional object. So we predict that x will converge to the surface value of $\frac{1}{2}$ in the large s limit.

4.2. Finite-size scaling analysis

For the three quantities mean height V', mean width W' and average hull H calculated from the dual-walker algorithm, we utilized their peak values to obtain exponent estimates. Simulations were conducted in the range $p = 0.66$–0.7134 at values of p spaced in intervals that varied from 0.0001 to 0.0005 depending on whether the simulations were in the peak region or in the shoulders. At each point 5×10^5 clusters were generated.

We first plotted each quantity for various values of cut-off $h_{\max}$ against p (see, for example, figure 7). Using a weighted quadratic fit near the peak we estimated the peak position and value. We note here that estimates at different values of $h_{\max}$, but the same values of p, were correlated in this analysis (as we used the same data runs to produce the estimates). However, our peak values were not so correlated given that the peak positions were distinct and as simulations at different values of p were independent.

The peak heights were calculated for $h_{\max} = (\sqrt{2})^m$ for $m = 20, \ldots, 30$ and weighted

Table 4. Our estimates for the exponents obtained via finite-size scaling (FSS) analysis of the data. These can be compared with estimates calculated from scaling relations and our estimates from the ordinary scaling analysis in section 4.1.

Quantity	Exponent	FSS estimate	Scaling estimate
$V'(h_{\max})$	$(\nu_\parallel - \beta)\sigma'$	0.72(2)	0.740(6)
$W'(h_{\max})$	$(\nu_\perp - \beta)\sigma'$	0.408(12)	0.418(5)
$H(h_{\max})$	$\gamma'\sigma'$	0.85(2)	0.8605(8)

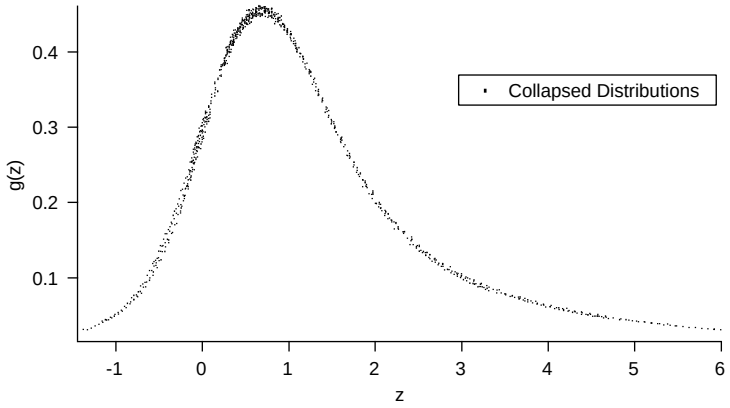

Figure 8. A plot of the scaling function $g(z)$ associated with the mean hull $H(p; h_{\max})$ against the variable $z = h_{\max}^{\sigma'}(p - p_c)$. The values of $h_{\max}$ used were 5000, 10 000, 20 000, 30 000. Note that the maximum values of $|p - p_c|$, rather than z, for each plot were the same.

linear regression was performed over those points and subsets fixed at the large end of the data range. There seemed to still be significant systematic trends in the data and the statistical errors were large. Our finite-size scaling estimates, and estimates calculated from our ordinary scaling analysis above and appropriate scaling relations, are given in table 4.

While these estimates are not as precise or stable as those obtained from the ordinary scaling analysis, they are nevertheless consistent with them. Using the peak height exponent estimates, we have illustrated the goodness-of-fit produced by plotting the scaling function $g(z)$ of the mean hull H (see figure 8) using data from different cut-offs.

5. Conclusions

This study has estimated values for the exponents associated with the scaling of the standard properties of directed percolation cluster hulls by means of Monte Carlo simulations. We have found an internally consistent set of values, that are also consistent with series estimates of the mass scaling exponents: the comparison made possible by scaling relations and an estimate of the connecting exponent, x between the two sets of exponents. Furthermore, this connecting exponent x is close to the rational $\frac{7}{9}$, though we expect further analysis of this problem to exclude this value. This is intriguing from the point of view of a possible exact solution: its form must be unusual if the exponents are not rational. It tallies though with other exponent values for this problem which seem also to have defied rational (fractional) conjecture in the past.

Acknowledgments

Financial support from the Australian Research Council is gratefully acknowledged by the authors.

References

[1] Kinzel W 1983 *Percolation Structures and Processes (Annals of the Israel Physical Society 5)* ed G Deutscher, R Zallen and J Alder ch 18, pp 425–45
[2] Bunde A and Havlin S 1991 *Fractals and Disordered Systems* ed A Bunde and S Havlin (Heidelberg: Springer) ch 2, pp 81–2
[3] Jensen I and Guttmann A J 1995 *J. Phys. A: Math. Gen.* **28** 4813
[4] Jensen I and Guttmann A J 1996 *J. Phys. A: Math. Gen.* **29** 497
[5] Jensen I 1996 *J. Phys. A: Math. Gen.* **29** 7013
[6] Cardy J L 1996 *Scaling and Renormalization in Statistical Physics* (Cambridge: Cambridge University Press)
[7] Bunde A and Havlin S 1991 *Fractals and Disordered Systems* ed A Bunde and S Havlin (Heidelberg: Springer) chs 2, 3, pp 51–149
[8] Stauffer D and Aharony A 1992 *An Introduction to Percolation Theory* 2nd edn (London: Taylor and Francis)
[9] Conway A and Guttmann A J 1994 *J. Phys. A: Math. Gen.* **27** 7007
[10] Brak R and Owczarek A L 1995 *J. Phys. A: Math. Gen.* **28** 4709
[11] Leath P L and Reich G R 1978 *J. Phys. C: Solid State Phys.* **11** 4017
[12] Weinrib A and Trugman S A 1985 *Phys. Rev.* B **31** 2993
[13] Ziff R M 1986 *Phys. Rev. Lett.* **56** 545
[14] Ziff R M, Cummings P T and Stell G 1984 *J. Phys. A: Math. Gen.* **17** 3009