Low-density series expansions for directed percolation: III. Some two-dimensional lattices

2004 J. Phys. A: Math. Gen. 37 6899

(http://iopscience.iop.org/0305-4470/37/27/003)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **37** (2004) 6899–6915 PII: S0305-4470(04)79231-8

Low-density series expansions for directed percolation: III. Some two-dimensional lattices

Iwan Jensen

ARC Centre of Excellence for Mathematics and Statistics of Complex Systems,
Department of Mathematics and Statistics, The University of Melbourne, Victoria 3010, Australia

E-mail: I.Jensen@ms.unimelb.edu.au

Received 15 April 2004
Published 22 June 2004
Online at stacks.iop.org/JPhysA/37/6899
doi:10.1088/0305-4470/37/27/003

Abstract

We use very efficient algorithms to calculate low-density series for bond and site percolation on the directed triangular, honeycomb, kagomé and (4.8^2) lattices. Analysis of the series yields accurate estimates of the critical point p_c and various critical exponents. The exponent estimates differ only in the 5th digit, thus providing strong numerical evidence for the expected universality of the critical exponents for directed percolation problems. In addition we also study the non-physical singularities of the series.

PACS numbers: 05.50.+q, 05.70.Jk, 05.70.Ln, 05.10.−a

1. Introduction

Percolation is one of the fundamental problems in statistical mechanics [4, 12], and is of great theoretical interest in its own right as well as being applicable to a wide variety of problems in physics, biology and many other areas of science. Percolation is commonly formulated as a problem on a lattice in which the edges and/or vertices are occupied (vacant) with probability $p\,(1-p)$. Throughout this paper we shall refer to *occupied* edges and vertices as bonds and sites, respectively, while edges and vertices refer to the the underlying lattice. Nearest neighbour bonds (sites) are said to be connected and clusters are sets of connected bonds (sites). Directed percolation (DP) is a specialization to problems in which connections are allowed only along a preferred direction given by an orientation of the edges of the lattice.

We use generalizations of a recently devised and very efficient algorithm [10] to calculate long low-density series for the average cluster size and other properties of directed percolation on various two-dimensional lattices. In this paper we study bond and site percolation on the directed triangular, honeycomb, kagomé and (4.8^2) lattices. In figure 1 we show a part of these lattices and the orientation of the edges. The (4.8^2) lattice is slightly different from the other cases since some edges are bi-directional. In the electrical network language some of

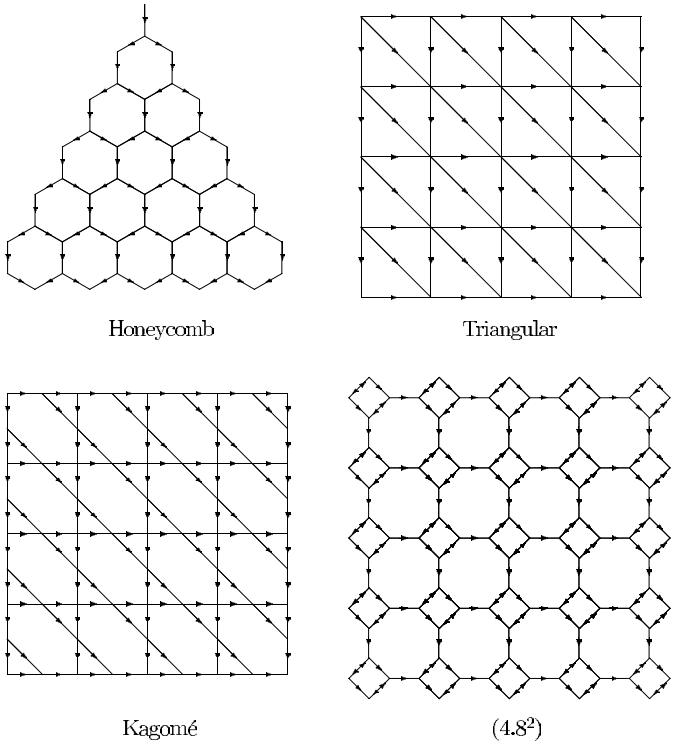

Figure 1. The directed honeycomb, triangular, kagomé and (4.8^2) lattices.

the edges are occupied by resistors, others are occupied by diodes and the remaining edges are insulators (empty). The bi-directional edges are required in order to make the problem symmetric. Due to the orientation of the remaining edges the lattice has an overall preferred direction and the problem is still in the DP universality class.

In the following section we first describe the general graph theoretical basis of the low-density series expansion technique and then give details of how the algorithms have been implemented for each of the lattices studied in this paper. In section 3 we present the results from our analysis of the series, including accurate estimates of the critical point p_c and critical exponents and estimates for the location of various non-physical singularities, as well as the corresponding exponents.

2. Calculation of low-density series

In the low-density phase ($p < p_c$) many quantities of interest can be derived from the pair-connectedness $C_{t,x}(p)$, which is the probability that the vertex at position x is occupied at time t given that the origin was occupied at $t = 0$. The coordinate t measures the distance from the origin along an axis parallel to the preferred direction, while the coordinate x measures the distance in the transverse direction. Of particular interest are moments of the pair-connectedness

$$\mu_{m,n}(p) = \sum_t \sum_x t^m x^n C_{t,x}(p) \tag{1}$$

since they enable us to obtain estimates of the critical point p_c as well as the critical exponents γ, $\nu_\parallel$ and $\nu_\perp$. In particular the average cluster size $S(p) = \mu_{0,0}(p)$. Due to symmetry, moments involving odd powers of x are identically zero. The remaining moments diverge as p approaches p_c from below

$$\mu_{m,n}(p) \propto (p_c - p)^{-(\gamma + m\nu_\parallel + n\nu_\perp)} \qquad p \to p_c^-. \tag{2}$$

It has been shown [1] that the pair-connectedness can be expressed as a sum over all graphs g, which are coverable by a union of directed paths connecting the origin to the vertex at (t, x),

$$C_{t,x}(p) = \sum_g d(g) p^{|g|} \tag{3}$$

where $|g|$ is the number of bonds in g. The weight $d(g) = (-1)^{c(g)}$, where $c(g)$ is the cyclomatic number of the graph g. The restriction to coverable graphs is very strong and leads to a huge reduction in the number of graphs that need be counted in order to calculate the pair-connectedness. One immediate consequence is that graphs with dangling parts make no contribution to $C_{t,x}$ and any contributing graph terminates exactly at (t, x). Another way of stating the restriction is that any vertex with an incoming bond *must* have an outgoing bond unless it is the terminal vertex (t, x).

Any directed path to a site whose parallel distance from the origin is t contains at least $k(t)$ bonds, where $k(t)$ is lattice dependent. From this it immediately follows that if $C_{t,x}$ has been calculated for $t \leqslant t_{\max}$ then one can determine the moments to order $k(t_{\max})$. One can however do much better as shown in the work by Essam *et al* [6]. They used a so-called non-nodal graph expansion, based on work by Bhatti and Essam [2], to extend the series to order $2k(t_{\max}) + 1$ on the square and triangular lattices. A graph g is nodal if it has a vertex (other than the terminal vertex) through which all paths pass. It is clear that each such nodal point effectively works as a new origin for the cluster growth, and we can obviously obtain any coverable graph by concatenating non-nodal graphs. More precisely we concatenate two coverable graphs g' and g'' by placing the original vertex of one graph on top of the terminal vertex of the other graph. If g' terminates at (x', t') and g'' terminates at (x'', t'') then

$$\begin{aligned} &g = g'g'' \text{ terminates at } (x' + x'', t' + t'') \\ &|g| = |g'| + |g''| \\ &c(g) = c(g') + c(g''). \end{aligned} \tag{4}$$

Note that the graph consisting of a single bond is non-nodal so all linear graphs can be obtained by repeated concatenations. This is the essential idea behind the non-nodal graph expansion, which proceeds in two principal steps. First we calculate the contribution $C^*_{t,x}$ of non-nodal graphs to the pair-connectedness. Next we use repeated concatenation operations of $C^*_{t,x}$ to calculate the pair-connectedness $C_{t,x}$ and from this we finally calculate various moments $\mu_{m,n}(p)$. One is mainly interested in moments involving only t or x (that is cases with $n = 0$ and/or $m = 0$), and in practice we calculate only the first two nonzero moments. The exclusion of 'cross' moments has the advantage that we need only calculate two-parameter generating functions $C_t(p)$ and $C_x(p)$. The savings in time obtained by the elimination of one variable more than compensates for the minor complication of having to calculate these two functions separately.

The calculation of $C^*_{t,x}$ is done efficiently using transfer-matrix techniques. This involves drawing a boundary line across a finite slice of the lattice and then moving the boundary line such that one adds row after row with each row built up one lattice cell at a time. The sum over all contributing graphs is calculated as the lattice is constructed. At any given stage

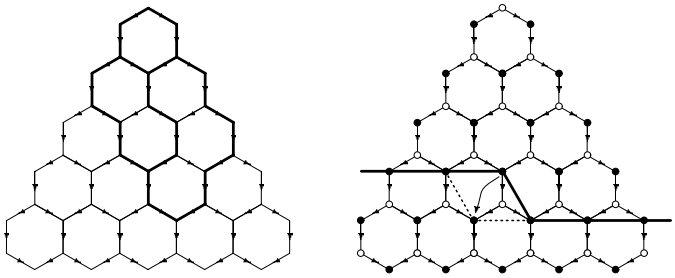

Figure 2. A piece of the directed honeycomb lattice. In the left panel we show an example of a non-nodal graph with 24 bonds and 3 cycles. In the right panel we show a snapshot of the boundary line (thick solid) during the transfer-matrix calculation and indicate how it is moved to a new position (dotted line) while adding an extra 'cell' with two vertices and three edges to the lattice.

the boundary line cuts through a number of, say j, vertices. There are two possible states (0 or 1) per vertex leading to a total of 2^j possible boundary configurations. In site percolation the different states correspond to the vertex being empty (0) or occupied (1), while in bond percolation we distinguish between vertices with (1) and without (0) incoming occupied edges. The weight of each configuration is given by a polynomial in p truncated at order N.

The maximal order N to which the series can be calculated is primarily limited by available computer memory. While the maximal number of boundary configurations is 2^j it must be emphasized that not all of these are required because they represent graphs contributing at an order exceeding N. Savings in memory use can be achieved as follows. Firstly, for each boundary configuration we keep track of the current minimum number of bonds $N_{\min}$ that have already been inserted. Secondly, we calculate the minimum number of addition bonds N_{add} required to produce a valid non-nodal graph. If $N_{\min} + N_{\text{add}} > N$ we can discard that configuration because it will not make a contribution to the pair-connectedness up to the maximal order we are trying to obtain. N, as well as $N_{\min}$ and N_{add}, depend on the given lattice and on the specific implementation of the transfer-matrix algorithm and one naturally tries to find an implementation which is simple yet tends to maximize N. In the following subsections we give some details of the transfer-matrix algorithms and concatenation operations for the case of directed *bond* percolation on the four lattices studied in this paper. The extension to site percolation is outlined briefly.

2.1. Honeycomb lattice

Percolation on the directed honeycomb lattice is a simple generalization of the directed square lattice problem. The various moments for site percolation can be obtained from the square series by the simple substitution $p \to p^2$ [3, 5], and we have thus not investigated this problem further. Bond percolation on the other hand is closely related to the problem of site-bond percolation on the square lattice in which both sites and bonds are present with probability p.

2.1.1. The transfer-matrix algorithm. The algorithm used to calculate the non-nodal contributions C^* to the pair-connectedness is a simple generalization of the square lattice algorithm [10]. In figure 2 we show a snapshot of the boundary line (solid) during the transfer-matrix calculation and indicate how it is moved to a new position (dotted line) while adding an extra 'cell' to the lattice. The updating of the weights, W, associated with the boundary

configurations depend only on the states of the topmost vertex and the neighbours to the left and right in the row below. The only difference from the square lattice rules is that when the top vertex is in state '1' we have to insert the extra vertical bond leading to the middle vertex (indicated by open circles) since only graphs with no dangling ends make a contribution. Thus we obtain the honeycomb lattice updating rules from those on the square lattice simply by inserting an extra factor p on all weights of configuration with the topmost vertex in state '1':

$$\begin{aligned} W(\overline{S}_{1,1}) &= p^2 W(S_{1,0}) + p^2 W(S_{1,1}) - p^3 W(S_{1,1}) \\ W(\overline{S}_{0,1}) &= W(S_{0,1}) + p^2 W(S_{1,0}) - p^2 W(S_{1,1}) \\ W(\overline{S}_{1,0}) &= p^2 W(S_{1,0}) \\ W(\overline{S}_{0,0}) &= W(S_{0,0}) \end{aligned} \tag{5}$$

where $S_{i,j}$ is a boundary configuration *before* the move with the topmost vertex in state i and the rightmost vertex in state j, while similarly $\overline{S}_{i,j}$ is a boundary configuration *after* the move with the leftmost vertex in state i and the rightmost vertex in state j. Other aspects of the algorithm are as in [10], except that the minimum order to which a configuration can make a contribution to $C^*_{t,x}$ from row t' is

$$N_{\mathrm{cont}} = 2N_{\mathrm{min}} + N_1 + 2t - 4t' \tag{6}$$

N_1 is the number of vertices in state '1' and this factor reflects the fact that for each vertex in state '1' we have to insert an extra vertical bond. The calculation was carried out for values of t up to 75 and the series were thus obtained to order 301.

2.1.2. The concatenation operations. Any graph contributing to $C_{t,x}$ is either a linear graph or can be broken into non-nodal components connected by linear pieces and possibly ending with a linear piece. Using standard techniques of generating functions we have

$$C_{t,x} = E + LC^*E + LC^*LC^*E + \cdots = \frac{E}{1 - LC^*}. \tag{7}$$

The generating function, L, for the linear pieces is easily obtained by noting that a linear graph connecting two non-nodal pieces starts at a vertex in the left panel of figure 2 indicated by a closed circle and terminates at a vertex indicated by an open circle. Thus a linear piece can consist of a single step down (this adds a bond and increases t by one so it is represented by a factor pt) or a down step followed by repeated instances of either a step to the left or a step to the right and a step down

$$L = pt + p^3t^3(x + x^{-1}) + p^5t^5(x + x^{-1})^2 + \cdots = \frac{pt}{1 - p^2t^2(x + x^{-1})} \tag{8}$$

where we put in a weight x (x^{-1}) for each step to the right (left). Similar arguments demonstrate that the generating function, E, for the end piece is

$$E = \frac{1 + pt}{1 - p^2t^2(x + x^{-1})}. \tag{9}$$

Note that the variables x and t appearing in these generating functions are not the same as those appearing in the definition of the moments in equation (1). The relationship is simply that *the power* of x and t in the generating functions are the x and t values which should be used in calculating the moments.

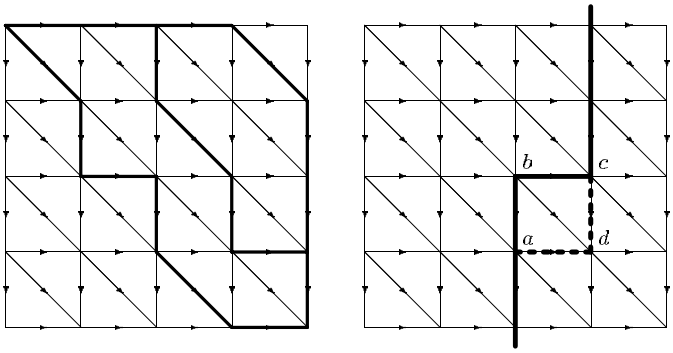

Figure 3. A piece of the directed triangular lattice. The left panel contains an example of a non-nodal graph while the right panel illustrates how the boundary line is moved during the transfer-matrix calculation in order to build up a rectangular slice of the lattice.

2.2. Triangular lattice

On the triangular lattice we calculate the weight of non-nodal graphs starting in the top left corner and terminating in the bottom right corner of rectangles of size $w \times l$, as illustrated in the left panel of figure 3. Such graphs contribute to $C^*_{t,x}$, where $t = w + l$ and $x = l - w$, and contain at least $t + \max(x, 1)$ bonds. Due to symmetry we need only consider rectangles with $l \geqslant w$. So if we want to calculate the series to order N we must sum the contributions from all rectangles with widths up to $N/2$ and lengths from w to $N/2$.

2.2.1. The transfer-matrix algorithm The right panel of figure 3 shows how the boundary line is moved in order to add an extra cell to the triangular lattice. As the boundary line is moved we insert bonds emanating from the vertex in the top left corner of the kink (in figure 3 this vertex is marked b), so only if this vertex is in state '1' are any bonds added. Note that bonds have been inserted only to the left of the boundary line while those *on* and to the right may be inserted in subsequent moves. Depending on which bonds are added the states of the vertices marked a and c can change while the 'new' vertex d is assigned a state. We use the notation $\overline{S}_{i,j,k}$ ($S_{i,j,k}$) to indicate a boundary configuration after (before) the move with vertices a, d and c (a, b and c) in states i, j and k, respectively. The updating rules show how to obtain the weight $W(\overline{S}_{i,j,k})$ of a given 'target' configuration from the weights of the 'source' configurations. Considerations similar to those for the square lattice case [10] yield

$$\begin{aligned}
W(\overline{S}_{1,1,1}) &= (p - 2p^2 + p^3)W(S_{1,1,1}) + (p^2 - p^3)W(S_{0,1,1}) \\
&\quad + (p^2 - p^3)W(S_{1,1,0}) + p^3 W(S_{0,1,0}) \\
W(\overline{S}_{0,1,1}) &= (p - p^2)W(S_{0,1,1}) + p^2 W(S_{0,1,0}) \\
W(\overline{S}_{1,0,1}) &= (-2p + p^2)W(S_{1,1,1}) + (p - p^2)W(S_{0,1,1}) \\
&\quad + W(S_{1,0,1}) + (p - p^2)W(S_{1,1,0}) + p^2 W(S_{0,1,0}) \\
W(\overline{S}_{0,0,1}) &= -pW(S_{0,1,1}) + W(S_{0,0,1}) + pW(S_{0,1,0}) \\
W(\overline{S}_{1,1,0}) &= (p - p^2)W(S_{1,1,0}) + p^2 W(S_{0,1,0}) \\
W(\overline{S}_{0,1,0}) &= pW(S_{0,1,0}) \\
W(\overline{S}_{1,0,0}) &= -pW(S_{1,1,0}) + pW(S_{0,1,0}) + W(S_{1,0,0}) \\
W(\overline{S}_{0,0,0}) &= W(S_{0,0,0}).
\end{aligned} \tag{10}$$

We shall only briefly indicate how these equations are derived by looking at the updating rule for $\overline{S}_{1,1,1}$. The contribution from $S_{0,1,0}$ is the simplest. Since only the vertex b has an incoming bond all three bonds have to be inserted in order to yield the target $\overline{S}_{1,1,1}$ and we thus get the contribution $p^3 W(S_{0,1,0})$. The contribution from $S_{1,1,0}$ arise as follows. Since vertex c is in state '0' we have to insert the two bonds from b to c and d. The bond from b to a can be either absent or present. If the bond is present, we note that since vertex a already has an incoming bond the additional bond results in the formation of an extra cycle and we have to weight this case with a factor $-p^3$. Thus the total contribution is $(p^2 - p^3)W(S_{1,1,0})$. Due to symmetry an identical weight is assigned to the contribution from $S_{0,1,1}$. Finally, the contribution from $S_{1,1,1}$ is derived by noting that the diagonal bond from b to d must be inserted while the remaining two bonds can be either absent or present. If only one of these bonds is inserted we add in total two extra bonds and form one extra cycle. When both these bonds are inserted we form two extra cycles. All in all we get the contribution $(p - 2p^2 + p^3)W(S_{1,1,1})$. The remaining updating rules can be derived via similar lines of reasoning.

In this case N_{add} has to be calculated using a simple algorithm. There are essentially only two contributions. Firstly there is the number of extra bonds required to connect the top and bottommost sites with incoming bonds to the first permissible vertex on the bottom row. Since we have to span at least a $w \times w$ rectangle the first column at which the two paths can join is $l = \max(w, l' + \delta)$, where l' is the current column position of the boundary and δ is a small number depending on the position of top and bottommost sites with incoming bonds with respect to a possible kink in the boundary line. δ is 2 if there is a kink in the boundary *and* the bottommost site is *above* the kink, 1 if there is a kink in the boundary *and* the topmost site is *above* the kink, 0 otherwise the connection of the two sites has to be done in such a way as to ensure there are no nodal vertices, so we are just dealing with two non-intersecting directed walks joining at (l, w). We shall refer to these two walks as the outer perimeter. Secondly, some additional bonds are added in order to connect any intermediate sites to the outer perimeter. We have to choose the shape of the outer perimeter so as to minimize the number of extra bonds. In practice this is quite a simple minimization problem and it is easy to write the required algorithm.

We were able to calculate C^* up to width 44 and consequently obtain the series to order 90. This is a very substantial improvement on the previous best of 57 terms [9].

2.2.2. The concatenation operations. The concatenation operations are very similar to the honeycomb case, except that the end piece and the linear connecting pieces are identical. So in this case we have

$$C_{t,x} = L + LC^*L + LC * LC^*L + \cdots = \frac{L}{1 - LC^*}. \tag{11}$$

The generating function for a linear piece is

$$L = \frac{1}{1 - ptx - ptx^{-1} - pt^2} \tag{12}$$

which represents repeated single steps either to the right, down or along the diagonal.

2.2.3. The site problem. The transfer-matrix algorithm for the site problem is very similar to that of the bond case, but of course the weights used in the updating rules are quite different. The updating rules for the site case are however easy to derive from the bond rules. All we have to do is change the weights in equation (10) so that we count the number of additional occupied vertices rather than edges. This results in many cancellations and leads to the following site

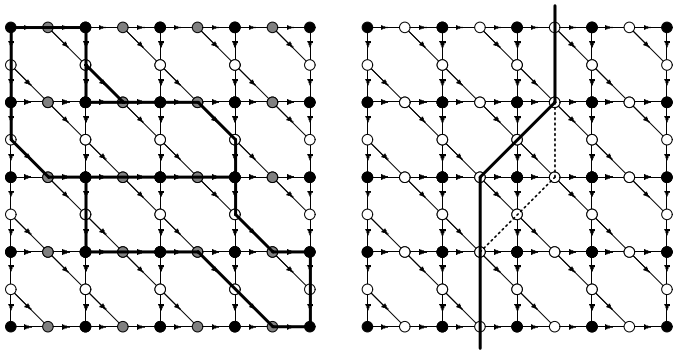

Figure 4. A piece of the directed kagomé lattice. Indicated in the left panel are the three different vertex types used in the calculation of the pair-connectedness and an example of a non-nodal graph contributing to $C^*_{a,a}$. The right panel illustrates how the boundary line is moved during the transfer-matrix calculation.

updating rules:

$$\begin{aligned}
W(\overline{S}_{1,1,1}) &= p^3 W(S_{0,1,0}) \\
W(\overline{S}_{0,1,1}) &= p^2 W(S_{0,1,0}) \\
W(\overline{S}_{1,0,1}) &= -pW(S_{1,1,1}) + W(S_{1,0,1}) + p^2 W(S_{0,1,0}) \\
W(\overline{S}_{0,0,1}) &= -pW(S_{0,1,1}) + W(S_{0,0,1}) + pW(S_{0,1,0}) \\
W(\overline{S}_{1,1,0}) &= p^2 W(S_{0,1,0}) \\
W(\overline{S}_{0,1,0}) &= pW(S_{0,1,0}) \\
W(\overline{S}_{1,0,0}) &= -pW(S_{1,1,0}) + pW(S_{0,1,0}) + W(S_{1,0,0}) \\
W(\overline{S}_{0,0,0}) &= W(S_{0,0,0}).
\end{aligned} \tag{13}$$

We calculated C^* up to width 40 and obtained the series to order 82 as compared to the previous best of 57 terms [9].

2.3. Kagomé lattice

In order to facilitate our calculation of $C_{t,x}$ we divide the vertices of the kagomé lattice into three subsets as indicated in the left panel of figure 4. In the following we shall refer to vertices indicated by black, shaded or open circles as being of type a, b or c, respectively. The transfer-matrix algorithm is designed to calculate $C^*_{a,a}$, that is the contribution to the pair-connectedness from non-nodal graphs starting and terminating on vertices of type a. From this we use concatenation operations to calculate the full pair-connectedness $C_{t,x}$, starting at a vertex of type a and terminating on a vertex of any type.

2.3.1. The transfer-matrix algorithm. In this case we calculate the contribution of non-nodal graphs starting in the top left corner and terminating in the bottom right corner of $w \times l$ rectangles, as illustrated in the left panel of figure 4. Due to symmetry we need only consider rectangles with $l \geqslant w$. These graphs contribute to $C^*_{t,x}$, where $t = 2(l+w)$ and $x = 2(l-w)$, and they contain at least $t + x + 2 = 4l + 2$ bonds. So in a calculation to order N we have to calculate the contribution from rectangles with width up to $N/4$ and lengths from w to $N/4$.

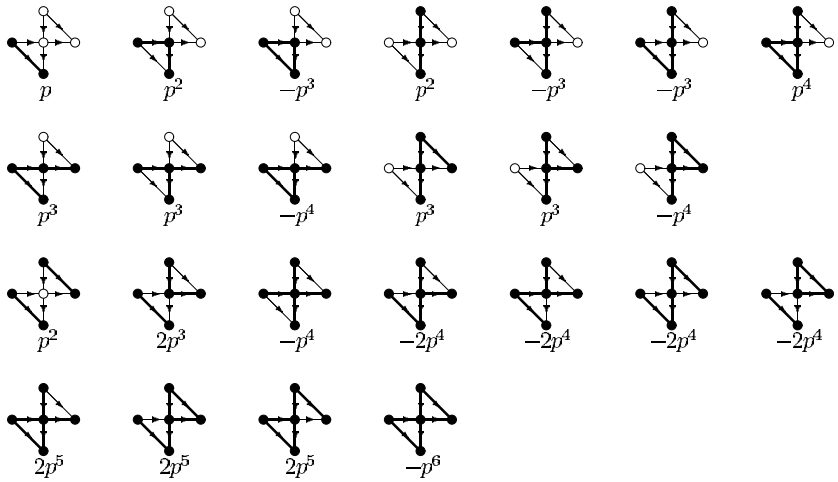

Figure 5. The allowed configurations of occupied edges (thick lines) and their associated weights generated as the boundary line is moved. The top line shows the configurations resulting in the target state $\overline{S}_{1,0}$, while the remaining three lines show the configurations leading to $\overline{S}_{1,1}$.

The right panel of figure 4 illustrates how the kink in the boundary line is moved in order to add an extra cell to the kagomé lattice. The updating rules are derived by considering all possible bond configurations and discarding those that do not conform to the basic rules 'vertices with incoming bonds must have outgoing bonds' and 'vertices without incoming bonds have no outgoing bonds'. The weight of each allowed configuration is simply calculated by multiplying by a factor p for each inserted bond and a factor -1 for each new cycle. In figure 5 we show the allowed configurations (and their associated weights) occurring in the derivation of the updating rules for the target boundary configurations $\overline{S}_{1,0}$ (top line) and $\overline{S}_{1,1}$ (remaining lines). The updating rule for the target $\overline{S}_{0,1}$ follows from symmetry. Summing over all contributions we get the following set of equations:

$$\begin{aligned}
W(\overline{S}_{1,1}) &= (2p^3 - p^4)[W(S_{1,0}) + W(S_{0,1})] + (p^2 + 2p^3 - 9p^4 + 6p^5 - p^6)W(S_{1,1}) \\
W(\overline{S}_{0,1}) &= p^2 W(S_{1,0}) + (p + p^2 - p^3)W(S_{0,1}) - (2p^3 - p^4)W(S_{1,1}) \\
W(\overline{S}_{1,0}) &= (p + p^2 - p^3)W(S_{1,0}) + p^2 W(S_{0,1}) - (2p^3 - p^4)W(S_{1,1}) \\
W(\overline{S}_{0,0}) &= W(S_{0,0}).
\end{aligned} \tag{14}$$

While these equations hold in the general case there is one special case to consider. When all vertices except those in the kink are in state '0' the updating is different because we have to carefully avoid forming nodal points. Thus the only input and output configuration with nonzero weight is the one in which both vertices in the kink are in state '1'. In this special case the third configuration in the third row of figure 5 is forbidden because the central vertex would be a nodal point and we thus get the updating rule

$$W(\overline{S}_{1,1}) = (p^2 + 2p^3 - 8p^4 + 6p^5 - p^6)W(S_{1,1}).$$

As for the triangular case we have to use a small algorithm to calculate N_{add}. While the details are slightly different, the general considerations are the same in the two cases and we will therefore refrain from further elaboration. We were able to calculate $C^*_{a,a}$ on rectangles up to width 42 and obtain the series to order 173.

2.3.2. The concatenation operations. As indicated above the pair-connectedness is obtained by adding the contributions C_i from graphs terminating at a vertex of type i. C_i is in turn obtained by repeated concatenations of non-nodal graphs. The sequence of concatenations can formally be expressed as

$$C_i^{(k)} = \sum_j C_j^{(k-1)} C_{j,i}^* \tag{15}$$

where we start with the initial conditions $C_a^{(0)} = 1, C_b^{(0)} = 0$ and $C_c^{(0)} = 0$. $C_i^{(k)}$ is the contribution after exactly k iterations of equation (15), $C_{j,i}^*$ is the non-nodal pair-connectedness from a vertex of type j to one of type i, and C_i is finally obtained by summation over k. As shown above $C_{a,a}^*$ is calculated directly by the transfer-matrix algorithm while the general cases of $C_{j,i}^*$ are derived from $C_{a,a}^*$ as follows:

$$\begin{aligned}
C_{a,b}^* &= tx[p + (p - p^2)C_{a,a}^*] \\
C_{a,c}^* &= tx^{-1}[p + (p - p^2)C_{a,a}^*] \\
C_{b,a}^* &= C_{a,b}^* \\
C_{b,b}^* &= t^2x^2(p^2 - 2p^3 + p^4)C_{a,a}^* \\
C_{b,c}^* &= t^2[p - p^3 + (p^2 - 2p^3 + p^4)C_{a,a}^*] \\
C_{c,a}^* &= C_{a,c}^* \\
C_{c,b}^* &= C_{b,c}^* \\
C_{c,c}^* &= t^2x^{-2}(p^2 - 2p^3 + p^4)C_{a,a}^*.
\end{aligned} \tag{16}$$

In these expressions there are generally two terms representing either 'elementary' steps or more complex non-nodal graphs derived by simple 'decorations' of graphs contributing to $C_{a,a}^*$. As an example we look at $C_{a,b}^*$, that is non-nodal graphs from black to shaded vertices in figure 4. The first term, txp, is just a single horizontal bond, while the second term, $tx(p - p^2)C_{a,a}^*$, arises as a decoration of a non-nodal graph from black-to-black vertices by appending a little triangle to the bottom right corner of the original non-nodal graph so as to extend the graph to the nearest shaded vertex to the right. We have to insert the horizontal bond from the black to the shaded vertex and the diagonal bond from the white to the shaded vertex. However, we can now either delete the existing vertical bond from the white to the black vertex, (yielding an overall additional factor p) or leave the bond in place (yielding the additional factor $-p^2$ since an additional cycle was formed). The other expressions involving vertices of type a are derived in the same manner and correspond to different ways of placing the elementary steps and the decorating triangle. $C_{b,b}^*$ and $C_{c,c}^*$ involve decorating a non-nodal graph contributing to $C_{a,a}^*$ by two little triangles, either at the top pointing left and at the bottom pointing right or at the top pointing up and at the bottom pointing down. Finally, $C_{b,c}^*$ contains the two elementary contributions of a diagonal bond, t^2p, or a little triangle, $-t^2p^3$. Note that the simple linear graph consisting of a horizontal and vertical bond is not non-nodal and its contribution is in fact derived from the concatenation $C_{b,a}^* C_{a,c}^*$. The remainder of $C_{b,c}^*$ is obtained by decorating graphs in $C_{a,a}^*$ by appending two little triangles one on the top pointing left and one at the bottom pointing down.

We could chose not to distinguish between sites of type b and c, which would lead to fewer equations but also make the concatenations less transparent.

2.3.3. The site problem. Again the updating rules for the site case are derived from the bond rules by changing the weights in equation (14) so as to count the number of additional

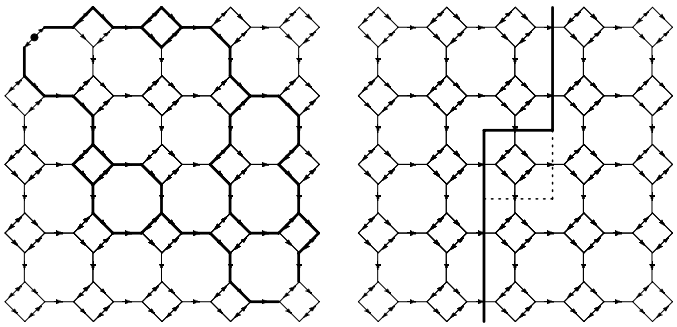

Figure 6. A piece of the directed (4.8^2) lattice. Shown in the left panel is an example of a non-nodal graph. The right panel illustrates how the boundary line is moved during the transfer-matrix calculation.

occupied vertices rather than edges. This leads to the site updating rules:

$$\begin{aligned} W(\overline{S}_{1,1}) &= p^3[W(S_{1,0}) + W(S_{0,1})] + (p^2 - 2p^3)W(S_{1,1}) \\ W(\overline{S}_{0,1}) &= p^2 W(S_{1,0}) + pW(S_{0,1}) - p^2 W(S_{1,1}) \\ W(\overline{S}_{1,0}) &= pW(S_{1,0}) + p^2 W(S_{0,1}) - p^2 W(S_{1,1}) \\ W(\overline{S}_{0,0}) &= W(S_{0,0}). \end{aligned} \tag{17}$$

The concatenation is performed as in the bond case. However, as we changed the updating rules so we have to change $C^*_{i,j}$. A little calculation will show that the terms $C^*_{i,j}$ in the site case are very simple and are derived from equation (16) by deleting all factors involving $C^*_{a,a}$, leaving just the elementary steps from a to b and so on. We calculated $C^*_{a,a}$ on rectangles up to width 41 and thus obtained the series to order 169.

2.4. (4.8^2) lattice

In the case of the (4.8^2) lattice we define a slightly restricted pair-connectedness, which in essence corresponds to a special choice for the initial vertex and a restriction on the possible terminal bonds. The details of our choice of initial vertex should be unimportant as long as we preserve the symmetry among the horizontal and vertical directions. We have chosen to introduce a special initial vertex as indicated in the left panel of figure 6. This vertex is, in the parlance of electrical networks, the point at which current is injected into the network and the current may flow along either one or both of the adjoining horizontal and vertical edges. Bonds along the short edges connecting the initial vertex to the network are not counted in $|g|$. Finally, for our convenience, we have chosen to specify that we always terminate the flow in either a horizontal or vertical bond. In other words $C_{t,x}$ is the probability of finding a path from the initial vertex to either a horizontal or vertical edge.

2.4.1. The transfer-matrix algorithm. The transfer-matrix algorithm is designed to calculate the contribution from non-nodal graphs starting and terminating on adjoining horizontal and vertical edges, as illustrated in the left panel of figure 6. Again we calculate $C^*_{t,x}$ for graphs on rectangles of size $w \times l$, and as usual we need only consider rectangles with $l \geqslant w$. These graphs contribute to $C^*_{t,x}$ for $t = l + w$ and $x = l - w$, and contain at least $2w + 6l = 4t + 2x$ bonds. Thus, in a calculation or order N, we must calculate the contributions from rectangles of widths up to $N/8$ and lengths from w to $(N - 8w)/6$.

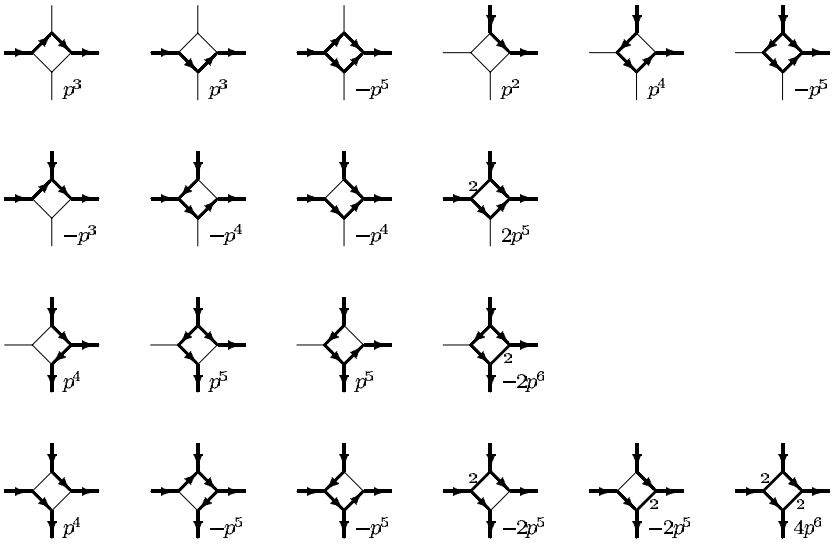

Figure 7. The allowed configurations of occupied edges (thick lines) and their associated weights generated as the boundary line is moved. The top two lines show the configurations resulting in the target state $\overline{S}_{0,1}$, while the remaining two lines show the configurations leading to $\overline{S}_{1,1}$. In each case we show the direction of the flow. Along the thick edges with no arrows the flow can progress in either direction, and hence we have marked these edges with a '2' indicating that there are two distinct flow patterns.

The right panel of figure 6 shows how the boundary line is moved in order to add an extra cell to the (4.8^2) lattice. The updating rules are derived in a similar fashion to those for the kagomé lattice. The only major difference is that along some of the diagonal edges the current can flow in either direction, so in some cases depending on the configuration of incoming and outgoing bonds we may have several distinct flow patterns associated with a given bond configuration. In figure 7 we show the configurations and the associated weights occurring in the derivation of the updating rules for the target configurations $\overline{S}_{0,1}$ (top two lines) and $\overline{S}_{1,1}$ (remaining two lines). This leads to the following set of equations:

$$\begin{aligned}
W(\overline{S}_{1,1}) &= (p^4 + 2p^5 - 2p^6)[W(S_{1,0}) + W(S_{0,1})] + (p^4 - 6p^5 + 4p^6)W(S_{1,1}) \\
W(\overline{S}_{0,1}) &= (2p^3 - p^5)W(S_{1,0}) + (p^2 + p^4 - p^5)W(S_{0,1}) - (p^3 + 2p^4 - 2p^5)W(S_{1,1}) \\
W(\overline{S}_{1,0}) &= (p^2 + p^4 - p^5)W(S_{1,0}) + (2p^3 - p^5)W(S_{0,1}) - (p^3 + 2p^4 - 2p^5)W(S_{1,1}) \\
W(\overline{S}_{0,0}) &= W(S_{0,0}).
\end{aligned} \tag{18}$$

As for the previous cases we wrote an algorithm to calculate N_{add} and again the same general considerations apply. In this case we were able to calculate $C^*_{t,x}$ on rectangles up to width 31 and consequently calculate the series to order 255.

2.4.2. The concatenation operations. The restricted pair-connectedness is calculated by adding the contributions C_h and C_v from graphs terminating with a horizontal or vertical bond, respectively. As in the previous cases these contributions are obtained from repeated concatenations

$$C_h^{(k)} = C_h^{(k-1)}C^*_{h,h} + C_v^{(k-1)}C^*_{v,h} \qquad C_v^{(k)} = C_h^{(k-1)}C^*_{h,v} + C_v^{(k-1)}C^*_{v,v} \tag{19}$$

where we start with the initial condition

$$\begin{aligned} C_h^{(0)} &= txp + tx(-p^3 - 2p^4 + 2p^5)C^* \\ C_v^{(0)} &= tx^{-1}p + tx^{-1}(-p^3 - 2p^4 + 2p^5)C^* \end{aligned} \tag{20}$$

and $C^*_{i,j}$ is the non-nodal pair-connectedness from an edge of type i to one of type j. As described above C^* is calculated using the transfer-matrix algorithm, while $C^*_{i,j}$ are easy to derive from the updating rules in equation (18)

$$\begin{aligned} C^*_{h,h} &= tx[2p^3 - p^5 - p^5(1 + 2p - 2p^2)^2C^*] \\ C^*_{h,v} &= tx^{-1}[p^2 + p^4 - p^5 - p^5(1 + 2p - 2p^2)^2C^*] \\ C^*_{v,h} &= tx[p^2 + p^4 - p^5 - p^5(1 + 2p - 2p^2)^2C^*] \\ C^*_{v,v} &= tx^{-1}[2p^3 - p^5 - p^5(1 + 2p - 2p^2)^2C^*]. \end{aligned} \tag{21}$$

The pre-factor multiplying C^* arise as the product of two contributions namely going from a single occupied bond to two occupied bonds and the reverse. The weights of each of these contributions can be extracted from equation (18). The first contribution is derived from the updating rule for $W(\overline{S}_{1,1})$ as generated from either $W(S_{1,0})$ or $W(S_{0,1})$. Noting that two of the bonds inserted in this updating operation are already counted as part of the non-nodal graph we find the weight $p^2(1 + 2p - 2p^2)$. The second contribution is derived from the updating rule for $W(\overline{S}_{1,0})$ or $W(\overline{S}_{0,1})$ as generated from $W(S_{1,1})$ and thus carries the weight $-p^3(1 + 2p - 2p^2)$, thus giving us the total pre-factor $-p^5(1 + 2p - 2p^2)^2$. In addition we have to add the weights corresponding to simple graphs not involving non-nodal pieces. These contributions arise from the graphs shown on the first line in figure 7. Note that strictly speaking these graphs are not non-nodal but they are the simplest graphs not counted in C^* and they are non-nodal because we have chosen only to allow graphs to terminate at horizontal or vertical edges.

2.4.3. The site problem. Changing the weights in equation (18) by counting the number of additional occupied vertices rather than edges leads to the site following updating rules:

$$\begin{aligned} W(\overline{S}_{1,1}) &= p^3[W(S_{1,0}) + W(S_{0,1})] - p^4W(S_{1,1}) \\ W(\overline{S}_{0,1}) &= (2p^3 - p^4)W(S_{1,0}) + p^2W(S_{0,1}) - p^3W(S_{1,1}) \\ W(\overline{S}_{1,0}) &= p^2W(S_{1,0}) + (2p^3 - p^4)W(S_{0,1}) - p^3W(S_{1,1}) \\ W(\overline{S}_{0,0}) &= W(S_{0,0}). \end{aligned} \tag{22}$$

The concatenation operations (19) are identical but the initial conditions are

$$C_h^{(0)} = txp - txp^2C^* \qquad C_v^{(0)} = tx^{-1}p - tx^{-1}p^2C^* \tag{23}$$

while

$$\begin{aligned} C^*_{h,h} &= tx[2p^3 - p^4 - p^4C^*] & C^*_{h,v} &= tx^{-1}[p^2 - p^4C^*] \\ C^*_{v,h} &= tx[p^2 - p^4C^*] & C^*_{v,v} &= tx^{-1}[2p^3 - p^4 - p^4C^*]. \end{aligned} \tag{24}$$

In this case we calculated $C^*_{t,x}$ on rectangles up to width 29 and consequently the series to order 239.

3. Analysis of the series

The various series were analysed using inhomogeneous differential approximants [7]. Our use of differential approximants for series analysis has been detailed in previous papers

Table 1. Estimates for p_c and critical exponents for various directed percolation problems.

Problem	p_c	γ	$\gamma + \nu_\parallel$	$\gamma + 2\nu_\perp$
Square bond	0.644 700 185(5)	2.277 730(5)	4.011 577(11)	4.471 438(13)
Square site	0.705 485 15(20)	2.277 65(6)	4.011 35(15)	4.471 30(8)
Honeycomb bond	0.822 856 80(6)	2.277 65(5)	4.011 3(1)	4.471 35(5)
Triangular bond	0.478 025 25(5)	2.277 83(6)	4.011 76(8)	4.471 56(6)
Triangular site	0.595 646 75(10)	2.277 71(2)	4.011 45(5)	4.471 44(2)
Kagomé bond	0.658 969 10(8)	2.277 90(6)	4.011 86(8)	4.471 67(9)
Kagomé site	0.736 931 82(4)	2.277 78(3)	4.011 60(5)	4.471 50(4)
(4.8^2) bond	0.767 651 16(7)	2.277 65(4)	4.011 44(3)	4.471 28(6)
(4.8^2) site	0.816 742 20(4)	2.277 70(6)	4.011 45(4)	4.471 27(4)

[10, 9] and the interested reader can refer to these papers and the comprehensive review [7] for further details. Suffice to say that this technique of series analysis usually enables one to obtain accurate estimates for the critical point p_c, the associated critical exponents, and possible non-physical singularities. In the following sections we shall briefly describe the results obtained from the series analysis.

3.1. Critical points and exponents

In table 1 we have listed the estimates for the critical points and exponents for the problems studied in this paper as obtained by analysing the series for the average cluster size and the first nonzero parallel and perpendicular moments of the pair-connectedness. The estimates for p_c and the critical exponents are based on results of series analysis using both second- and third-order approximants with varying degrees of the inhomogeneous polynomial. The estimates were obtained by averaging values obtained from second- and third-order differential approximants. For each order L of the inhomogeneous polynomial we averaged over those approximants to the series which used at least the first 90% of the terms in the series. The error quoted for these estimates reflects the spread among the approximants. Note that these error bounds should *not* be viewed as a measure of the true error as they cannot include possible systematic sources of error. For completeness and comparison we have also listed the results for the square lattice bond and site problems from [10]. Note that in this case the exponent estimates are not necessary obtained by direct analysis of the corresponding series.

In all cases we have obtained accurate estimates of the critical point p_c and the critical exponents, with the exponent estimates differing only in the 5th digit. In most cases we note that the error bounds are an order of magnitude larger than for the square bond case. Standard arguments from renormalization group theory says that the exponents should be universal. This expectation is clearly supported by the present analysis. Obviously, the exponent estimates for some of the problems do not quite agree with the square bond case within the quoted error bounds. However, the differences are quite small and any discrepancy is likely to be due to the difficulty of obtaining reliable error bounds. Certainly the discrepancies are not of such an order as to challenge the strong belief in universality. In particular we note that the exponent estimates scatter around the square bond value, with some being higher (triangular bond, kagomé bond and site) others being lower (square site, honeycomb bond, (4.8^2) bond) and yet others being in complete agreement (triangular site, (4.8^2) site). Furthermore, we note that in many cases (e.g. honeycomb bond) if we look at other exponent estimates, say the one we can obtain of $2\nu_\perp$, there is a close agreement with the square bond case. The square and

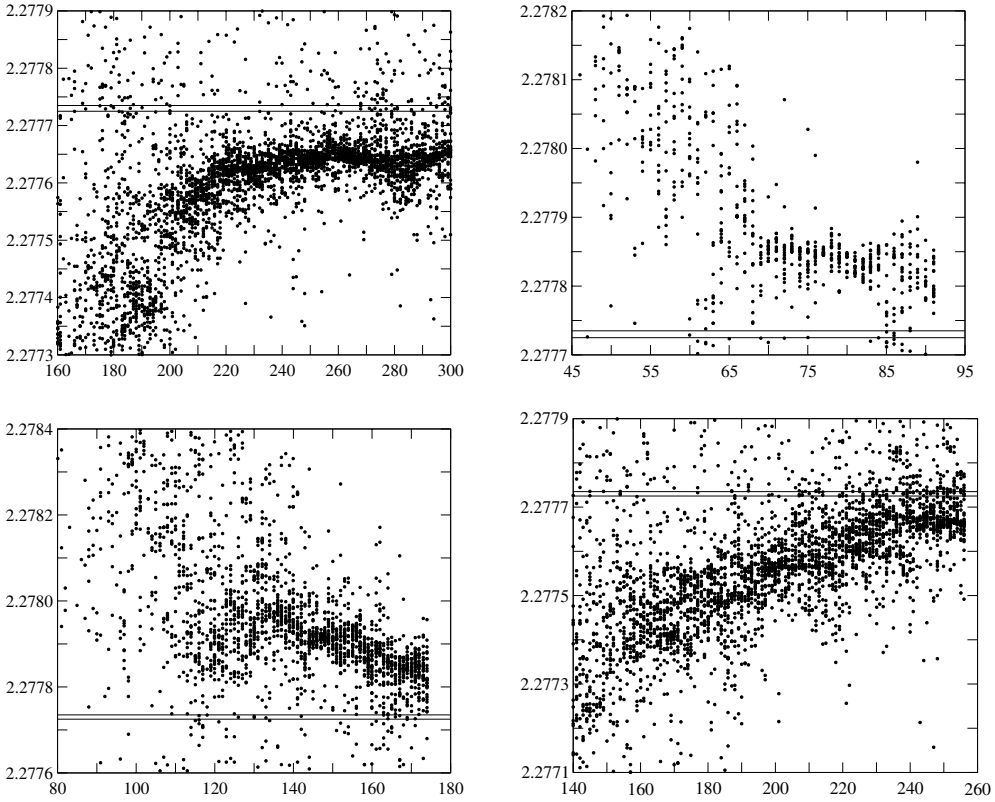

Figure 8. Estimates of the critical exponent γ versus the number of terms used by third-order differential approximants to the cluster-size series $S(p)$ for bond percolation on the directed honeycomb (top left), triangular (top right), kagomé (bottom left) and (4.8^2) (bottom right) lattices.

honeycomb bond cases yield $2\nu_\perp = 2.193\,708(18)$ and $2\nu_\perp = 2.193\,70(11)$, respectively. The worst case scenario therefore would be that we would have to adopt a more conservative error estimate on the exponents. For reasons explained in detail in [10] we have a great deal of confidence in the square bond estimates and are reluctant to increase the error bounds. More cautious readers may choose to take a different view.

As we have already mentioned above, one of the major unresolved problems of series analysis is the calculation of reliable error estimates. So in trying to confirm, as we are here, the universality of the critical exponents, it is often useful to plot the behaviour of the exponent estimates versus the number of terms used by the differential approximants. In this way it often possible to gauge more clearly whether or not the high-order approximants have settled down to the limiting value of the true exponent. In figure 8 we carry out such an analysis of the cluster size series for the directed bond percolation problems. Each point in the four panels corresponds to an estimate of γ, obtained from a third-order differential approximant, plotted against the number of terms used by the differential approximant. The straight lines indicate the error bounds on the very accurate estimate of γ obtained from the analysis of the square bond series. From these plots we can see that the estimates of γ obtained from the triangular (top right panel) and kagomeé lattices (bottom left panel) exhibit a pronounced downwards drift as the number of terms is increased, while the estimates from the (4.8^2) lattice (bottom right panel) display an upward drift. So the estimates of γ have not yet settled at their

limiting value, but there can be no doubt that the indicated value of γ is fully consistent with the estimates. The only possible disagreement comes from the honeycomb problem (top left panel).

3.2. Non-physical singularities

Non-physical singularities are of interest for several reasons. Firstly, one expects that the presence of non-physical singularities (particularly if they are closer to the origin than p_c) can have a dramatic influence on the precision of the estimates of p_c and the critical exponents. Secondly, knowledge of the position and associated exponents of non-physical singularities may help in the search for exact solutions and in some cases one may gain a better understanding of the problem by studying the behaviour of various physical quantities as analytic functions of complex variables. Many of the series have a radius of convergence smaller than p_c due to singularities in the complex p-plane closer to the origin than the physical critical point. Since all the coefficients in the expansion are real, complex singularities always come in conjugate pairs. That such non-physical singularities must be present is evident from the fact that the coefficients in the various series change sign. If only the physical singularity was present all coefficients would have the same sign. In the following we shall briefly outline our findings. A more detailed description of the method can be found in [10].

3.2.1. Honeycomb lattice problem. The series have a pronounced and well-defined singularity at $0.022\,734(2) \pm 0.708\,909(2)$i. The exponents estimates are consistent with the exact values $1/2$, $-1/2$, $-3/2$ and $-1/2$ for the series $S(p)$, $\mu_{1,0}$, $\mu_{2,0}$ and $\mu_{0,2}$, respectively. A second singularity occurs at $0.347\,15(2) \pm 0.588\,13(3)$i with exponent estimates consistent with the values 3, 2, 1 and 2 for the series $S(p)$, $\mu_{1,0}$, $\mu_{2,0}$ and $\mu_{0,2}$, respectively. In addition there are other less well-defined singularities for which no meaningful exponent estimates can be obtained. We estimate the positions of these singularities to be $-0.6183(2)$ and $-0.180(5) \pm 0.725(5)$i.

3.2.2. Triangular lattice problems. The series do not appear to have any well-defined non-physical singularities. There is some evidence of singularities for the site problem at $-0.323(5) \pm 0.39(1)$i.

3.2.3. Kagomé lattice problems. The series for the bond problem shows no evidence of clearly defined non-physical singularities. The series for the site problem have a singularity on the negative axis at $-0.618\,04(2)$ with exponents $-0.344(3)$, $-1.34(1)$, $-2.34(1)$ and $-1.656(5)$ for the series $S(p)$, $\mu_{1,0}$, $\mu_{2,0}$ and $\mu_{0,2}$, respectively. The is evidence for further singularities at $-0.53(1) \pm 0.375(5)$i.

3.2.4. (4.8^2) lattice problems. The series for bond problem have a singularity at $-0.618\,04(2)$ with exponents $-0.505(5)$, $-1.505(5)$, $-2.49(1)$ and $-1.555(10)$ for the series $S(p)$, $\mu_{1,0}$, $\mu_{2,0}$ and $\mu_{0,2}$, respectively. Further singularities occur at $0.428(4) \pm 0.458(4)$i, $0.215(5) \pm 0.590(5)$i and $-0.528(5) \pm 0.346(5)$i. The series for the site problem have a singularity at $-0.618\,034(3)$ and also at $0.185\,36(2) \pm 0.659\,39(3)$i. In these cases the exponents are consistent with the exact values $1/2$, $-1/2$, $-3/2$ and $-1/2$ for the series $S(p)$, $\mu_{1,0}$, $\mu_{2,0}$ and $\mu_{0,2}$, respectively. Singularities also occur at $-0.3000(5) \pm 0.6247(2)$ and $0.428(2) \pm 0.536(3)$.

4. Summary and discussion

We have presented new algorithms for the calculation of low-density series expansions for directed percolation on several two-dimensional lattices. Analysis of the series yielded accurate estimates of the critical point p_c and various critical exponents, including the exponent γ governing the divergence of the cluster size at p_c. The estimates of the critical exponents confirm, to a high degree of accuracy and confidence, the universality of the exponents for directed percolation. This conclusion is in full agreement with earlier studies on the triangular [6, 9], honeycomb [8] and kagomé lattices [11] which also confirm universality although with lower precision.

To some extent the exponent estimates in table 1 do not agree with the very accurate square bond estimates. This is largely explained by the plots in figure 8, which show that in many cases the exponent estimates exhibit a pronounced drift and thus have not yet settled down to their limiting value. A further complicating factor is the presence of several non-physical singularities. They appear to be particularly prominent in the series for the honeycomb bond problem, which might explain the slight discrepancy between the exponent estimates for this problem and those for the square bond problem.

4.1. E-mail or WWW retrieval of series

The series for the directed percolation problems studied in this paper can be obtained via e-mail by sending a request to I.Jensen@ms.unimelb.edu.au or via the world-wide web on the URL http://www.ms.unimelb.edu.au/~iwan/ by following the relevant links.

Acknowledgments

The calculations presented in this paper were in part performed on the facilities of the Australian Partnership for Advanced Computing (APAC) and the Victorian Partnership for Advanced Computing (VPAC). We gratefully acknowledge financial support from the Australian Research Council.

References

[1] Arrowsmith D K and Essam J W 1977 Percolation theory on directed graphs *J. Math. Phys.* **18** 235–8
[2] Bhatti F M and Essam J W 1984 The resistive susceptibility of random diode-insulator networks *J. Phys. A: Math. Gen.* **17** L67–L73
[3] Dhar D, Phani M K and Barma M 1982 Enumeration of directed site animals on two-dimensional lattices *J. Phys. A: Math. Gen.* **15** L279–L284
[4] Essam J W 1980 Percolation theory *Rep. Prog. Phys.* **43** 833–912
[5] Essam J W and De'Bell K 1982 The pair connectedness for directed percolation on the honeycomb and diamond lattices *J. Phys. A: Math. Gen.* **15** L601–L604
[6] Essam J W, Guttmann A G and De'Bell K 1988 On two-dimensional directed percolation *J. Phys. A: Math. Gen.* **21** 3815–32
[7] Guttmann A J 1989 Asymptotic analysis of power-series expansions *Phase Transitions and Critical Phenomena* vol 13 ed C Domb and J L Lebowitz (New York: Academic) pp 1–234
[8] Jensen I and Guttmann A J 1996 Series expansions of the percolation probability on the directed triangular lattice *J. Phys. A: Math. Gen.* **29** 497–517
[9] Jensen I 1996 Low-density series expansions for directed percolation on square and triangular lattices *J. Phys. A: Math. Gen.* **29** 7013–40
[10] Jensen I 1999 Low-density series expansions for directed percolation: I. A new efficient algorithm with applications to the square lattice *J. Phys. A: Math. Gen.* **32** 5233–49
[11] Rushin H J and Cadilhe A 1991 Percolation on directed Archimedean nets *J. Non-Cryst. Solids* **127** 114–20
[12] Stauffer D and Aharony A 1992 *Introduction to Percolation Theory* 2nd edn (London: Taylor and Francis)