

# Percolation thresholds of two-dimensional continuum systems of rectangles

Jiantong Li\* and Mikael Östling

*KTH Royal Institute of Technology, School of Information and Communication Technology, Electrum 229, SE-164 40 Kista, Sweden*

(Received 1 November 2012; revised manuscript received 25 April 2013; published 3 July 2013)

The present paper introduces an efficient Monte Carlo algorithm for continuum percolation composed of randomly oriented rectangles. By conducting extensive simulations, we report high-precision percolation thresholds for a variety of homogeneous systems with different rectangle aspect ratios. This paper verifies and extends the excluded area theory. It is confirmed that percolation thresholds are dominated by the average excluded areas for both homogeneous and heterogeneous rectangle systems (except for some special heterogeneous systems where the rectangle lengths differ too much from one another). In terms of the excluded areas, generalized formulas are proposed to effectively predict precise percolation thresholds for all these rectangle systems. This paper is, therefore, helpful for both practical applications and theoretical studies concerning relevant systems.

DOI: [10.1103/PhysRevE.88.012101](https://doi.org/10.1103/PhysRevE.88.012101)

PACS number(s): 64.60.ah

## I. INTRODUCTION

Over the past decade, layered two-dimensional (2D) nanomaterials, such as graphene, MoS<sub>2</sub>, WS<sub>2</sub>, and boron nitride [1–5], have attracted great interest in the fields of material science, electronics, medicine, biology, and so forth. Many applications require these 2D materials to be integrated into percolating systems to acquire certain functions [2,5–7]. In general, a simple and representative model for these systems should be a percolation system comprising randomly oriented rectangles [8] (an example is shown in Fig. 1). As a matter of fact, the interest in rectangle percolation is not limited to practical applications for these emerging 2D nanomaterial systems. It can date back to the 1980s for theoretical studies concerning the excluded area (volume) [9,10].

Unfortunately, rectangle systems suffer from some fundamental problems not yet addressed. For example, the percolation thresholds are not known even for the simplest 2D homogeneous rectangle systems, despite the already known thresholds in high precision for similar 2D continuum systems composed of random disks [11,12], squares [12,13], and sticks [12,14]. The unknown percolation threshold for rectangle systems has also actually caused difficulty in verifying, in these systems, the excluded area theory [9,15], which assumes that the percolation threshold  $N_c$  is inversely proportional to the average excluded area  $A_r$ ,

$$N_c \propto A_r^{-1}, \quad (1)$$

where  $N_c$  is the critical number density of rectangles at percolation. The excluded area (volume) is defined as the minimum area (volume) around an object into which the center of another similar object cannot enter in order to avoid the overlapping of the two objects [9]. In homogeneous systems,  $A_r$  for randomly oriented rectangles of length  $l$  and width  $w$  (and then the aspect ratio  $r = l/w$ ) is [9]

$$\begin{aligned} A_r &= 2lw(1 + 4\pi^{-2}) + 2(l^2 + w^2)\pi^{-1} \\ &= 2l^2[(1 + 4\pi^{-2})r^{-1} + \pi^{-1}(1 + r^{-2})]. \end{aligned} \quad (2)$$

In addition, practical applications may show more interest in heterogeneous rectangle systems which comprise various

types of rectangles of different  $r$ 's and/or different  $l$ 's. A typical example is the composites of 2D layered materials (low- $r$  rectangles) and carbon nanotubes (high- $r$  rectangles) [3,16,17]. These systems have not even been considered in the traditional excluded area theory [9].

In order to meet the requirements for practical applications, to verify, and to extend the excluded area theory, in this paper, we perform Monte Carlo simulations to explore percolation thresholds for a variety of 2D rectangle systems.

## II. SIMULATION ALGORITHM

Previously, by combining the fast Newman-Ziff algorithm [18] with the subcell concept [19], we have developed an efficient algorithm for stick percolation [14]. In brief, during the simulations, each stick is registered into a subcell where its center lies so that it is only necessary to check the connectivity between two sticks belonging to the same and eight neighboring subcells. And each cluster, which comprises connecting sticks, is stored by a tree structure [18] so that the status of a cluster can readily be updated when a new stick is added into the system. This algorithm provides comparable efficiency to those for lattice percolation [14,18]. In fact, it can readily be generalized to any continuum percolation system, including, certainly, the rectangle systems here. Like a stick [14], a rectangle can also be stored simply through a point (its center) and an angle (its orientation). Therefore, the transfer from stick systems to homogeneous rectangle systems only needs two main modifications. One is that the subcell length should be set as the diagonal length of the rectangles, that is,  $\sqrt{l^2 + w^2}$  or  $l\sqrt{1 + r^{-2}}$  as shown in Fig. 1. The other is that a different technique should be employed to check the connectivity between two rectangles in the same and neighboring subcells [14]. In this paper, we adopt the Cohen-Sutherland (CS) algorithm, a famous line clipping algorithm in computer graphics [20]. It can efficiently determine whether a line segment is visible (or partially visible) in a viewport (a horizontally oriented rectangular window). In detail, the CS algorithm divides the 2D space into nine regions in terms of the viewport. Each region has a 4-bit code, and one can readily determine the regions of a line segment's end points by bitwise operations. A line segment is "trivially accepted" (completely

\*jiantong@kth.se

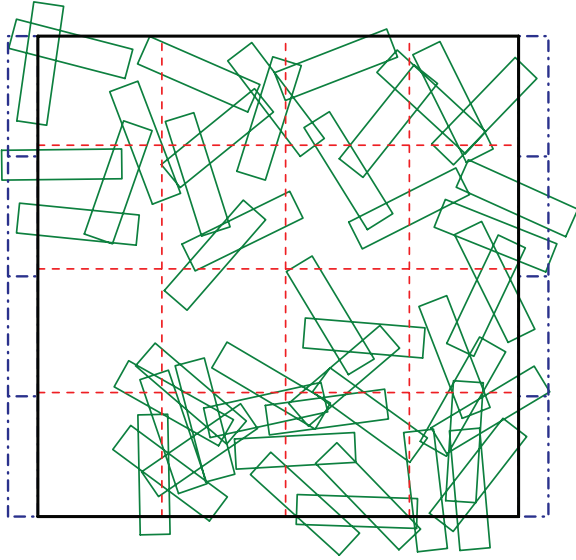


FIG. 1. (Color online) An example of a homogeneous FBC rectangle percolation (produced by our simulation program). Here, the rectangles (olive) are of length  $l = 1$ , width  $w = 0.25$ , and aspect ratio  $r = 4$ . The system (of size  $L = 4$ ) is defined by the four bold (black) boundaries. For convenience in employing the Newman-Ziff algorithm, each of the two opposite (left and right) boundaries is represented by  $L (= 4$  here) vertically oriented intersecting auxiliary rectangles [dashed dotted (blue) rectangles]. The system spans (percolates) if two opposite auxiliary rectangles are connected by the edges of normal rectangles, such as the case in this figure. The whole systems are virtually divided by a number of subcells as shown by the dashed (red) lines. All subcells are of the same side length  $\sqrt{l^2 + w^2}$ , except those in the topmost row or rightmost column. Each rectangle belongs to a subcell in which its center lies. Each auxiliary rectangle belongs to its nearest subcell. Note, in general, subcell side length is larger than the rectangle length.

visible) if both of its end points are inside the viewport region or is “trivially rejected” (completely invisible) if the two end points are on the same side of the viewport. Otherwise, one end point, which is outside the viewport, is replaced with the intersection point between its nearby (extended) viewport edge and the line segment. The process is repeated until the line segment is trivially accepted or trivially rejected. Note the CS algorithm [20] attempts to determine the visible portion of a line segment, whereas rectangle percolations are only interested in its visibility. Therefore, we can further simplify the CS algorithm by relieving the trivially accepted conditions: A line segment is trivially accepted if, at least, one of its end points is inside the viewport *or* one end point is over and the other is under the viewport.

In our simulations, to check the connectivity between two rectangles, we set one as the reference rectangle and the other as the test rectangle and translate and rotate the space so that the reference rectangle becomes the viewport (a horizontally oriented rectangle centered at the origin). Since the target for the reference rectangle is always the viewport, the translation and rotation are only necessary for the test rectangle. Then, for each side of the test rectangle, which is a line segment, we use the CS algorithm to check whether it is visible in the viewport (reference rectangle). Once one side is found visible, one can

conclude that the two rectangles connect (intersect); otherwise, if none of the four sides are visible, the two rectangles do not connect. In order to further improve efficiency, prior to the CS process, we carry out a screening: If the center distance of two rectangles is larger than the sum of their half diagonal lengths, they are impossible to intersect, and the CS process is not needed.

For heterogeneous systems, which may comprise rectangles of different aspect ratios  $r$  and/or different lengths  $l$ , the algorithm works in a similar way. In principle, only three steps need modification. First, the subcell length should be set as the largest diagonal length among all the rectangles. Second, during checking the connectivity between two rectangles, the reference rectangle should be the one of larger  $l$  or of larger area  $S = lw$ . Otherwise, if a large test rectangle encloses a small reference rectangle (viewport), it might be trivially rejected by mistake by the CS algorithm. Third, for every newly generated rectangle, an additional random number  $x$  ( $0 \leq x \leq 1$ ) is allocated to determine its dimensions. A rectangle has length  $l_i$  and aspect ratio  $r_i$  if its  $x$  satisfies  $\sum_{j=1}^{i-1} x_j < x \leq \sum_{j=1}^i x_j$ , supposing the heterogeneous system comprises  $n$  types of rectangles, and the  $j$ th type of rectangles have length  $l_j$ , aspect ratio  $r_j$ , and number fraction  $x_j$  ( $1 \leq j \leq n$ ,  $\sum_{j=1}^n x_j = 1$ ).

Monte Carlo simulations (see Refs. [14,18] for the detailed Newman-Ziff procedure), based on the algorithm above, can produce the spanning probability for any system (either homogeneous or heterogeneous) as a function of the number (integers) of rectangles. Its convolution with the Poisson distribution [14] gives the spanning probability  $R(N, L)$  for a system of size  $L$  at *any arbitrary* rectangle number density  $N$ . In this paper, all the systems are with free boundary conditions (FBCs), and the system size  $L$  is measured in units of  $l_{\max}$ , the largest rectangle length. For simplicity, we set  $l_{\max} \equiv 1$ . From the known universal finite-size scaling [ $R(N_c, \infty) = 0.5$ ] and  $1/L$  corrections for the spanning probability of an FBC percolation system with square boundary [14,21,22], one can expect

$$N_{0.5}(L) - N_c \sim L^{-1-1/v}, \quad (3)$$

where  $N_{0.5}(L)$  is a percolation threshold estimate [14,21] and is defined through  $R[N_{0.5}(L), L] = 0.5$ , and  $v = 4/3$  is the critical correlation-length exponent. From Eq. (3), a high-precision threshold  $N_c$ , i.e., the thermodynamic limit ( $L \rightarrow \infty$ ) value, can often be extracted for any percolation system [14,21,23,24].

### III. RESULTS AND DISCUSSION

In this section, first, the performance of our algorithm is compared with other algorithms (Sec. III A). Then, simulation results and verification of the excluded area theory are presented and discussed for homogeneous rectangle systems (Sec. III B). Finally, the traditional excluded area theory is extended to heterogeneous rectangle systems (Sec. III C).

#### A. Performance of the algorithm

In order to demonstrate the efficiency of the present algorithm (especially the integration of the CS algorithm), we compare its performance with two other algorithms. Both

TABLE I. Average CPU time in hours consumed for each of the three algorithms discussed in the main text for identical homogeneous rectangle systems of  $L = 128$ . The average is over 20 independent batches of simulations, and the uncertainty is the standard deviation.

Algorithm	$r = 1, 10^5$ runs	$r = 10, 10^4$ runs
CS	0.796(14)	0.695(43)
Edge-traverse	0.976(12)	0.948(42)
LB	0.855(13)	0.788(38)

of these algorithms are still based on the combination of the subcell algorithm with the Newman-Ziff procedure but adopt different algorithms to check the connectivity between two rectangles: (i) The first one adopts the “edge-traverse” algorithm, which tests whether any edge of the test rectangle intersects, at least, one edge of the reference rectangle [13]. The connectivity between two rectangle edges (actually line segments) is checked by the Pike and Seager bonding criterion of two sticks (also line segments) [25], i.e., two sticks overlap if and only if, for every stick, the distance between its center and the intersection point (of the corresponding lines for the two sticks) is no longer than its half-length. (ii) The second one employs another famous line clipping algorithm, the Liang-Barsky (LB) algorithm [20,26].

For clarity, hereafter, the above three algorithms are denoted as CS, edge traverse, and LB, respectively. Their performance has been tested via C++ codes on a server equipped with an Intel(R) Xeon(R) CPU E5430 (2.66 GHz). For each algorithm, we conducted 20 batches of Monte Carlo simulations for homogeneous rectangle systems of  $L = 128$  for  $r = 1$  and  $r = 10$ , respectively. Each batch has one independent random seed and contains  $10^5$  (for  $r = 1$ ) or  $10^4$  (for  $r = 10$ ) simulation runs. The whole set of random seeds is identical for all three algorithms. As a result, the three algorithms produce identical simulation results (spanning probability functions) but consume significantly different CPU times. Table I lists the average CPU time consumed for one batch. It is clear that the line clipping (CS and LB) algorithms are evidently faster than the edge-traverse algorithm. In particular, the CS algorithm is  $\sim 20\%$  faster than the edge-traverse algorithm for  $r = 1$ . With increasing  $r$  to 10, the CS algorithm is even more efficient ( $\sim 30\%$  faster than the edge-traverse algorithm). Since the edge-traverse algorithm has already been optimized in this paper, the 20%–30% efficiency improvement of the CS algorithm should be thought of as considerable progress.

It is interesting to note that, in the general case of line clipping, the LB algorithm has proven better efficiency than the CS algorithm [26]. For rectangle percolation, however, the alleviation of conditions for trivial acceptance increases the probability of a rectangle edge to be trivially accepted at the first test so that the CS algorithm becomes more efficient [20].

## B. Homogeneous rectangle systems

The CS algorithm is, thereby, employed to explore precise percolation thresholds for homogeneous rectangle systems. For each class of rectangle systems with the same  $r$ , we determine  $N_{0.5}(L)$  for different sizes  $L = 48, 64, 72, 88, 128$ ,

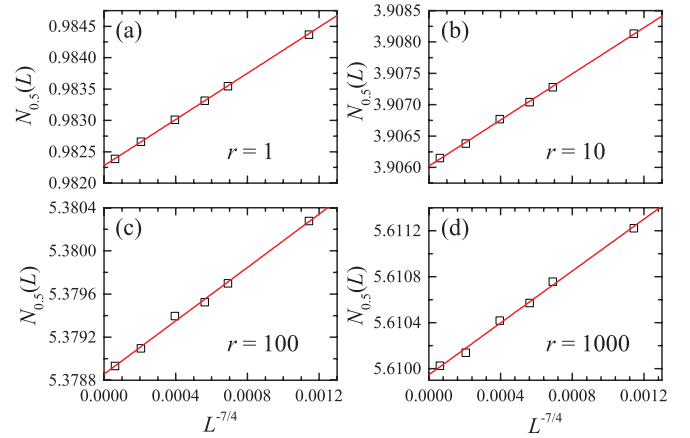


FIG. 2. (Color online) Plots of  $N_{0.5}(L)$  against  $L^{-1-1/\nu}$  for homogeneous rectangle systems with (a)  $r = 1$ , (b)  $r = 10$ , (c)  $r = 100$ , and (d)  $r = 1000$ .

and 256, each of which is based on  $> 10^7$  Monte Carlo samples. It is found that, for all  $r$  studied in this paper,  $N_{0.5}(L)$  follows Eq. (3) quite well (adjusted  $R^2 > 0.99$  for all the fittings; see Ref. [27] for a definition of adjusted  $R^2$ ). Some examples are shown in Fig. 2. Then, from linear fitting by Eq. (3) to these  $N_{0.5}(L)$ , we obtain high-precision values [23] for  $N_c$  for a variety of  $r$ 's. As listed in Table II, the uncertainty for all  $r$ 's is less than 0.0001, which is given by the half-width of the 95% confidence interval from linear fitting (regression) by Eq. (3). Note, for  $r = 1$ , the rectangle systems actually turn to the square systems, and the value of  $N_c$  is consistent with other papers [12,13]. In contrast, for  $r = \infty$ , they degrade to

TABLE II. Percolation thresholds  $N_c$  obtained from Monte Carlo simulations, their products with the excluded areas  $N_c A_r$ , and the corrected product  $(N_c - N_{c0})A_r$  for rectangle systems with different  $r$ 's.  $N_{c0} = 0.20$ . For  $r = \infty$ ,  $N_c$  is taken as the value of the stick systems [14].

$r$	$N_c$	$N_c A_r$	$(N_c - N_{c0})A_r$
1	0.982 278(14)	4.011	3.195
1.5	1.425 745(29)	3.982	3.424
2	1.786 294(26)	3.932	3.492
3	2.333 491(22)	3.837	3.508
4	2.731 318(30)	3.767	3.491
5	3.036 130(28)	3.717	3.472
6	3.278 680(19)	3.681	3.457
7	3.477 211(52)	3.655	3.445
8	3.643 137(24)	3.635	3.436
9	3.784 321(41)	3.621	3.429
10	3.906 022(37)	3.609	3.425
12	4.105 670(38)	3.594	3.418
15	4.329 848(28)	3.580	3.415
18	4.495 767(41)	3.573	3.414
20	4.584 535(54)	3.570	3.414
30	4.878 091(59)	3.566	3.420
50	5.149 008(20)	3.569	3.430
100	5.378 856(60)	3.576	3.443
1000	5.609 947(60)	3.587	3.459
$\infty$	5.637 263(11)	3.589	3.461

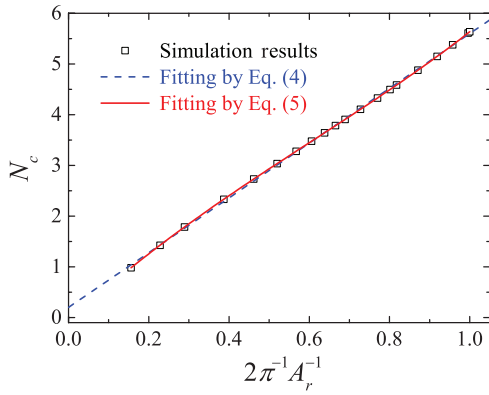


FIG. 3. (Color online) Plots of  $N_c$  against  $2\pi^{-1}A_r^{-1}$  for homogeneous rectangle systems with  $r$  ranging from 1 to  $\infty$ . Note all the systems are measured by their rectangle length, i.e.,  $l = 1$ . The simulation data (open squares) are also listed in Table II. The dashed (blue) and solid (red) curves are fittings by Eqs. (4) and (5), respectively.

stick systems, and the value of  $N_c$  is taken from our previous paper [14].

The product  $N_c A_r$  is also listed in Table II. For all  $r$ 's ranging from 1 to  $\infty$ , this product retains within a narrow region between 3.5 and 4.0. The region  $3.5 < N_c A_r \leq 4.0$  is within, and much smaller than, the limits of  $3.2 \leq N_c A_r \leq 4.5$  predicted earlier [10]. It can, thereby, roughly be verified that the excluded area theory [9] applies to rectangle systems.

For a precise analysis, we plot all  $N_c$ 's against the reciprocal of the normalized excluded area  $A_r^{-1}/A_\infty^{-1}$  or  $2\pi^{-1}A_r^{-1}$  [note from Eq. (2),  $A_\infty = 2\pi^{-1} \approx 0.6366$ ]. As shown in Fig. 3, the plot exhibits excellent linearity. However, the intercept is not at the origin, suggesting that the percolation threshold is not inversely proportional to the excluded area but is apparently linear with its reciprocal, that is,

$$N_c - N_{c0} \propto A_r^{-1}, \quad (4)$$

where  $N_{c0}$  is a corrected factor (constant). From linear fitting (adjusted  $R^2 = 0.9995$ ) by Eq. (4) to all  $N_c$ 's, we obtain  $N_{c0} = 0.20(4)$ . Listing the corrected product  $(N_c - N_{c0})A_r$  in Table II, we find it almost retains constant at 3.4 or 3.5 for all  $r > 1$ . The maximum uncertainty for the prediction of  $N_c$  by Eq. (4) is  $\sim 0.04$ . This uncertainty is given by the half-width of 99% confidence interval of the predictions [28].

In order to gain a more precise analysis, we propose a high-order polynomial [Eq. (5)] to fit the data in Fig. 3,

$$N_c = f(s) = \sum_{i=1}^n c_i s^i = \sum_{i=1}^n c_i (2\pi^{-1}A_r^{-1})^i, \quad (5)$$

where  $s = 2\pi^{-1}A_r^{-1}$  and  $c_i$  are constants. With  $n = 9$ , a perfect fitting (adjusted  $R^2 > 1 - 3 \times 10^{-10}$ ) can be obtained from nonlinear regression [29]. The fitted coefficients  $c_i$  are listed in Table III. As an important outcome, Eq. (5) can then predict high-precision  $N_c$  for any  $r$ . Here,  $n = 9$  is the minimum order number required to ensure small uncertainty of  $< 7 \times 10^{-5}$  for all predictions, a comparable level to those of the simulation results in Table II.

To validate Eq. (5), we investigate several more homogeneous rectangle systems with  $r$  other than those in Fig. 3 or

TABLE III. Coefficients  $c_i$  for nonlinear regression of Eq. (5) to the simulation data in Table II. The uncertainty is the half-width of the 95% confidence interval. In this table, all  $c_i$ 's are rounded to five decimal places, which may be necessary for the prediction of precise  $N_c$  by Eq. (5), although their uncertainty does not really support such a precision.

$i$	$c_i$	Uncertainty
1	5.829 30	0.0243
2	7.949 92	0.453
3	-47.482 82	3.44
4	132.306 51	14.0
5	-231.155 15	33.7
6	264.351 27	49.7
7	-190.880 90	44.0
8	78.922 23	21.4
9	-14.203 10	4.43

Table II. The predicted results [by Eq. (5)] are compared with the simulation results in Table IV. Their difference is really less than the predicted uncertainty  $7 \times 10^{-5}$ . It is concluded that Eq. (5) and the maximum uncertainty  $7 \times 10^{-5}$  can provide high-precision percolation thresholds for rectangle systems with any aspect ratio  $r$ . Although Eq. (5) is merely based on our simulation results rather than physical deductions, it makes it unnecessary for most future studies to explore the percolation thresholds once more and should facilitate many studies concerning continuum percolation of rectangles or the layered 2D nanomaterials.

It is worth mentioning that Eqs. (4) and (5) only provide *mathematic* fittings to the simulated  $N_c$  in Fig. 3 where the average excluded area  $A_r$  is limited within  $[A_\infty, A_1]$ . One may not extrapolate them for  $A_r$  outside this region. For example, as  $A_r \rightarrow \infty$ , it is reasonable to anticipate  $N_c \rightarrow 0$ . However, Eq. (4) expects that  $N_c$  converges to a finite value  $N_{c0}$ . Note that, in heterogeneous systems, the effective average excluded areas may go beyond the region  $[A_\infty, A_1]$ , although it is not possible for homogeneous systems. Relevant issues are discussed below in Sec. III C2.

In addition, Eq. (5) should not be the exclusive form to describe the thresholds in rectangle percolation. Other forms may also apply as long as their prediction uncertainty is sufficiently small. Interestingly, Xia and Thorpe [30] proposed a simple interpolation formula to describe the dependence of critical remaining area fraction  $p_c$  on the aspect ratios for

TABLE IV. Comparison between percolation thresholds obtained from simulations and calculations through Eq. (5) for homogeneous rectangle systems with  $r$ 's other than those in Table II.  $\Delta N_c$  is the absolute value of the percolation threshold difference between simulations and calculations.

$r$	$N_c$ (simulation)	$N_c$ (calculation)	$\Delta N_c$
1.1	1.078 532(21)	1.078 495(58)	$3.7 \times 10^{-5}$
1.25	1.215 636(18)	1.215 581(66)	$5.5 \times 10^{-5}$
2.5	2.083 711(26)	2.083 720(60)	$9.0 \times 10^{-6}$
40	5.043 120(52)	5.043 080(47)	$4.0 \times 10^{-5}$
200	5.504 099(69)	5.504 131(56)	$3.2 \times 10^{-5}$

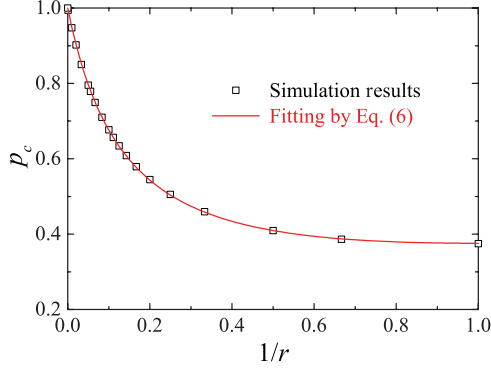


FIG. 4. (Color online) The critical remaining area fraction  $p_c$  for homogeneous rectangle systems of various aspect ratios (listed in Table II).

random ellipse systems as  $p_c = (1 + 4y)/(19 + 4y)$ , where  $y = b/a + a/b$  with  $a$  and  $b$  being the major and minor semiaxes, respectively, and  $p_c = \exp(-N_c S)$  with  $S$  being the ellipse area. We find the form of this interpolation formula also applies to the rectangle systems in this paper, despite the different numerical values for the coefficients, that is,

$$p_c = \exp(-N_c S) = (1.28 + y)/(6.73 + y), \quad (6)$$

where  $y = l/w + w/l = r + 1/r$  and for rectangles,  $S = lw = l^2/r$ . As shown in Fig. 4, Eq. (6) fits the simulation data very well (the maximum prediction uncertainty is about 0.005). However, in this paper, we mainly discuss the relation between  $N_c$  and  $A_r$ , i.e., Eq. (5), which is more important for heterogeneous systems as discussed below.

### C. Heterogeneous rectangle systems

In this section, we explore the relations between percolation threshold and excluded area for heterogeneous rectangle systems. For simplicity, we denote a rectangle of length  $l$  and width  $w$  as  $(l, w)$ . A heterogeneous rectangle system is denoted as  $[(l_1, w_1), x_1; (l_2, w_2), x_2; \dots; (l_n, w_n), x_n]$  if it contains  $n$  types of rectangles and the number fraction of the  $i$ th type of rectangles  $(l_i, w_i)$  is  $x_i$  ( $i = 1, 2, \dots, n$  and  $\sum_{i=1}^n x_i = 1$ ). Accordingly, a group of rectangle systems is denoted as  $\{(l_1, w_1); (l_2, w_2); \dots; (l_n, w_n)\}$ , which includes all systems comprising the  $n$  types of rectangles with any combinations of the number fractions.

Some examples for the heterogeneous rectangle systems are shown in Fig. 5. The Monte Carlo simulation results verify that Eq. (3) still applies to these systems [Figs. 5(b) and 5(d)], and high-precision percolation thresholds can also be extracted. In this section, however, we normally use a threshold estimate  $N_{0.5}(L)$  (typically  $L \geq 64$  and based on  $> 10^5$  Monte Carlo simulations) as a rough estimation of  $N_c$ , which, notwithstanding, has already provided sufficient precision (to about three reliable significant figures) for most discussions and has enabled us to efficiently establish the percolation threshold-excluded area relation for the heterogeneous systems.

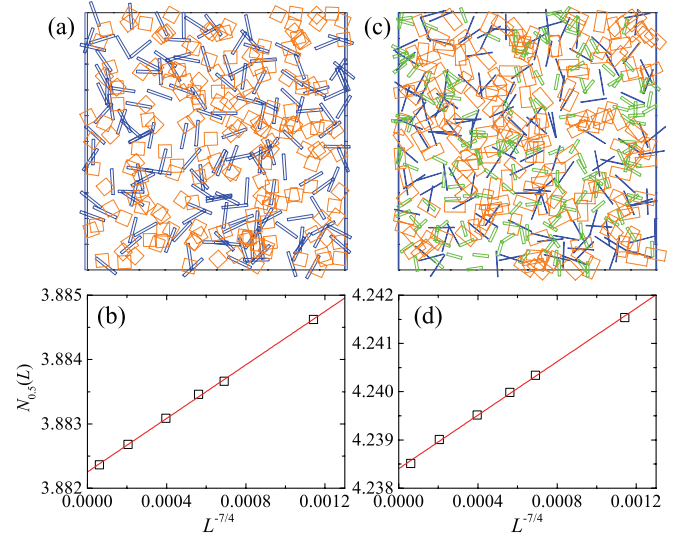


FIG. 5. (Color online) Examples of two heterogeneous systems. (a) A percolating realization ( $L = 10$ ) for the binary system of  $[(1,0.1),0.5;(0.5,0.5),0.5]$ . (b)  $N_{0.5}(L)$  against  $L^{-1-1/v}$  for the binary system in (a). (c) A percolating realization ( $L = 10$ ) for the ternary system of  $[(1,0.02),0.3;(0.8,0.4),0.4;(0.6,0.1),0.3]$ . (d)  $N_{0.5}(L)$  against  $L^{-1-1/v}$  for the ternary system in (c).

#### 1. Generalization of the excluded area theory

First of all, the average excluded area formula [Eq. (2)] should be generalized to heterogeneous rectangle systems. Referring to Fig. 2 of Ref. [9], one can readily find the excluded area between two different rectangles  $(l_i, w_i)$  and  $(l_j, w_j)$  is

$$A_{\theta}^{ij}(l_i, w_i; l_j, w_j) = l_i w_i + l_j w_j + (l_i l_j + w_i w_j) \sin \theta + (l_i w_j + w_i l_j) \cos \theta, \quad (7)$$

where  $\theta$  is the angle between the orientations of the two rectangles. If the orientations of the rectangles are uniformly distributed within the region  $[-\theta_{\mu}, \theta_{\mu}]$ , the average excluded area between the different two rectangles is [see Eqs. (6), (7), and (19) in Ref. [9] for more details]

$$A^{ij}(l_i, w_i; l_j, w_j) = l_i w_i + l_j w_j + (l_i l_j + w_i w_j)(2\theta_{\mu} - \sin 2\theta_{\mu})/2\theta_{\mu}^2 + (l_i w_j + w_i l_j)(1 - \cos 2\theta_{\mu})/2\theta_{\mu}^2. \quad (8)$$

For isotropic ( $\theta_{\mu} = \pi/2$ ) systems, Eq. (8) becomes

$$A^{ij}(l_i, w_i; l_j, w_j) = l_i w_i + l_j w_j + 2\pi^{-1}(l_i l_j + w_i w_j) + 4\pi^{-2}(l_i w_j + w_i l_j). \quad (9)$$

Equation (9) suggests  $A^{ij} = A^{ji}$ , and it degrades to Eq. (2) when  $i = j$ . For a system  $[(l_1, w_1), x_1; (l_2, w_2), x_2; \dots; (l_n, w_n), x_n]$ , the effective average excluded area  $A_e$  is

$$A_e = \sum_{i=1}^n \left( x_i \sum_{j=1}^n x_j A^{ij} \right). \quad (10)$$

For simplicity, in this paper, we only explore the  $N_c - A_e$  relations for heterogeneous systems with the number of rectangle types  $n \leq 3$ , i.e., the binary ( $n = 2$ ) and ternary ( $n = 3$ ) systems. Nevertheless, the generalization into systems

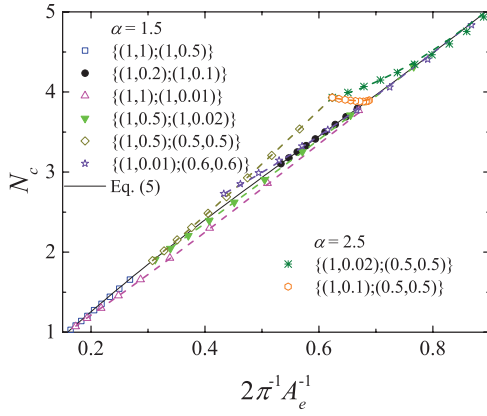


FIG. 6. (Color online) Percolation thresholds obtained from simulations (symbols) and calculations (dashed curves) from Eq. (15) for weakly and strongly correlated binary systems. For each group of systems, the number fractions  $x_i$  vary between 0 and 1 at the step of 0.1. For calculations, only the systems  $\{(1,0.1);(0.5,0.5)\}$  and  $\{(1,0.02);(0.5,0.5)\}$  are with  $\alpha = 2.5$  in Eq. (15) and are classified as strongly correlated systems. All the others are weakly correlated systems with  $\alpha = 1.5$ . Note, for efficiency, not all  $N_c$ 's in this plot are given by the thermodynamic limit ( $L \rightarrow \infty$ ) values as extrapolated from Eq. (3). For  $\{(1,1);(1,0.01)\}$  and  $\{(1,0.5);(1,0.02)\}$ ,  $N_c$  is approximated by  $N_{0.5}$  ( $L = 64$ ), whereas, for  $\{(1,0.5);(0.5,0.5)\}$  and  $\{(1,0.01);(0.6,0.6)\}$ ,  $N_c$  is approximated by  $N_{0.5}$  ( $L = 128$ ). Actually, however, with respect to the discussions of heterogeneous systems in this paper, the difference between  $N_{0.5}$  ( $L \geq 64$ ) and the corresponding thermodynamic limit value is negligible [see Figs. 2, 5(b), and 5(d)].

comprising more rectangle types ( $n > 3$ ) is anticipated to be straightforward.

## 2. Binary systems

We have investigated a series of binary systems as shown in Fig. 6 where the average excluded areas are calculated from Eq. (10). In Fig. 6, it is clear, when the excluded areas are comparable, the systems, no matter whether heterogeneous or homogeneous, have comparable percolation thresholds. It can, thereby, be confirmed that percolation thresholds are dominated by the average excluded areas. Rigorously, however, a slight deviation can be seen between heterogeneous and homogeneous systems. Fortunately, for most binary systems, we have found a general law to describe their precise  $N_c - A_e$  relations on the basis of Eq. (5).

First, one should note that Eq. (5) is correct only when the rectangle systems are measured in units of the rectangle length  $l$ . Certainly, it is possible that a rectangle system is measured by another unit  $l'$  ( $l' \neq l$ ). Under the new measurement, it can be established that the values of the percolation threshold and excluded area, denoted as  $N_{c,l'}$  and  $A_r'$ , respectively, relate to those of the old systems as

$$l'^{-2}N_{c,l'} = l^{-2}N_{c,l} = l^{-2}N_c, \quad (11)$$

and

$$l'^2A_r' = l^2A_r. \quad (12)$$

Then, still based on Eq. (5), one can predict  $N_{c,l'}$  through

$$\begin{aligned} N_{c,l'} &= (l/l')^{-2} f(2\pi^{-1}A_r^{-1}) \\ &= (l/l')^{-2} f[2\pi^{-1}A_r^{-1}(l/l')^2]. \end{aligned} \quad (13a)$$

In particular, if a homogeneous system comprising rectangles  $(l_i, w_i)$  is measured by  $l' = 1$ , but  $l_i \neq l'$ , Eq. (13a) turns out to be

$$N_{c,l'} = l_i^{-2} f(2\pi^{-1}A_r^{-1}) = l_i^{-2} f(2\pi^{-1}A_r'^{-1}l_i^2). \quad (13b)$$

Equation (13b) is very useful for our discussion below about heterogeneous systems comprising rectangles of different lengths.

Now, we consider a binary system  $[(l_1, w_1), x_1; (l_2, w_2), x_2]$ , which is measured by  $l'$  (the larger one between  $l_1$  and  $l_2$ ). With respect to one type of rectangle  $(l_i, w_i)$  ( $i = 1$  or  $2$ ), the effective excluded area of the binary system  $A_e$  can be regarded as a deviation from that of the homogeneous system  $A_{r_i}$  (still measured by  $l'$ ). Note  $N_c$  is roughly linear with  $1/A_r$  as described by Eq. (4). Then, one can use the linear approximation around  $2\pi^{-1}A_{r_i}^{-1}$  as one estimate for the percolation threshold of the binary system, that is,

$$\begin{aligned} N_c^{(i)} &\approx l_i^{-2} f(2\pi^{-1}A_{r_i}^{-1}l_i^2) \\ &\quad + 2\pi^{-1} f'(2\pi^{-1}A_{r_i}^{-1}l_i^2)(A_e^{-1} - A_{r_i}^{-1}), \end{aligned} \quad (14)$$

where  $f(s)$  is given by Eq. (5) and  $f'(s)$  is its first derivative. Note that  $N_c^{(1)}$  may differ significantly from  $N_c^{(2)}$ . Then, the actual  $N_c$  should result from the correlation between  $N_c^{(1)}$  and  $N_c^{(2)}$ . Obviously, both  $N_c^{(1)}$  and  $N_c^{(2)}$  should contribute to  $N_c$ , but their contributions may have unequal weight. It is reasonable to put more weight on the dominant rectangles, which, in essence, contribute more to the system percolation (spanning) when the effect of the fractions ( $x_i$ ) is not considered. In general, the dominant rectangles are those of larger length or of larger width in the case of equal lengths. For convenient discussion, we suppose the dominant rectangles in a binary system are always  $(l_1, w_1)$ , that is to say, either of the two conditions should be satisfied: (1)  $l_1 > l_2$  or (2)  $l_1 = l_2$  and  $w_1 > w_2$ . It is found that  $N_c$  of the binary system can be described as

$$N_c = N_c^{(1)}(1 - x_2^\alpha) + N_c^{(2)}x_2^\alpha, \quad (15)$$

where  $\alpha > 1$  is the correlation exponent.

In terms of the applicability of Eq. (15), we roughly classify the binary systems into three categories: weakly correlated systems (roughly  $l_1 < 2l_2$ , Fig. 6), strongly correlated systems ( $l_1 \approx 2l_2$  and usually  $r_1 \geq 10$ , Fig. 6), and ultrastrongly correlated systems (roughly  $l_1 > 2l_2$  and  $r_1 \geq 10$ , Fig. 7). As shown in Fig. 6,  $N_c$  of the weakly and strongly correlated systems agrees excellently with Eq. (15). The deviation between the simulations and the fittings by Eq. (5) is far less than 0.1. More interestingly, all weakly correlated systems share the common correlation exponent  $\alpha = 1.5$ , whereas, the strongly correlated systems have the common  $\alpha = 2.5$ . The common exponents  $\alpha$  make Eq. (15) nontrivial and allow the direct calculation of percolation thresholds for any binary systems in these two categories.

For ultrastrongly correlated systems, however, Eq. (15) cannot provide good fitting to the simulations, and evident deviation can be observed (Fig. 7). One important common feature of weakly and strongly correlated systems is

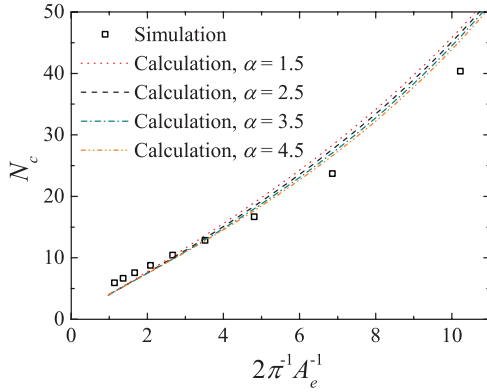


FIG. 7. (Color online) Percolation thresholds obtained from simulations (open squares) and calculations (curves) from Eq. (15) for a group of ultrastrongly correlated binary systems  $\{(1,0.01);(0.1,0.1)\}$ . For the open squares, from left to right, the number fraction  $x_1$  increases from 0.1 to 0.9 at a step of 0.1. Note, for these systems,  $A_e \ll A_\infty$ , and the predictions by Eq. (15) with all  $\alpha$ 's deviate significantly from the simulations. All  $N_c$ 's are approximated by  $N_{0.5}$  ( $L = 64$ ).

that their  $A_e$ 's (measured by  $l' = 1$ ) are still within the region  $[A_\infty, A_1]$ . However,  $A_e$  in the ultrastrongly correlated systems has gone beyond this region. For example, for the system  $[(1,0.01),0.1;(0.1,0.1),0.9]$ ,  $A_e = 0.0623 \ll A_\infty = 0.6366$ . Then, the employment of Eq. (15) requires the extrapolation of Eq. (5) far beyond  $[A_\infty, A_1]$ . This is a risk as discussed in Sec. III B. Consequently, Eq. (15) may become invalid for ultrastrongly correlated systems, especially when  $A_e$  goes far beyond the region.

These ultrastrongly correlated systems are actually very interesting for both theoretical studies and practical applications. In Table V, we compare some of these systems with the homogenous systems which have comparable average excluded areas. Surprisingly, these ultrastrongly correlated systems have significantly lower  $N_c$ 's as well as significantly lower critical coverage area  $N_c S$  than the homogeneous systems. On one hand, this can be regarded as a breakdown of the excluded area theory which, in essence, expects comparable excluded areas lead to comparable percolation thresholds. On the other hand, the significantly lower critical coverage area for ultrastrongly correlated systems may explain the experimental observation [16] that the composites of graphene and carbon nanotubes often produce improved performance (over pure graphene films) for transparent conductors which

TABLE V. Percolation thresholds  $N_c$ , effective average excluded area  $A_e$ , and the critical coverage area  $N_c S$  for some ultrastrongly correlated binary systems and homogeneous systems. For ultrastrongly binary systems,  $N_c$  is approximated by  $N_{0.5}$  ( $L = 64$ ) from simulations, and  $S = \sum_{i=1}^n x_i l_i w_i$ . For homogeneous systems,  $N_c$  and  $A_e$  are calculated from Eqs. (11) and (12), respectively.

System	$N_c$	$A_e$	$N_c S$
$[(1,0.01),0.05;(0.01,0.01),0.95]$	107.5	0.0040	0.064
Homogeneous (0.05,0.0167)	933.4	0.0041	0.78
$[(1,0.01),0.1;(0.01,0.01),0.9]$	53.77	0.0107	0.059
Homogeneous (0.05,0.05)	392.9	0.0102	0.98

prefer low coverage area (to gain more transmittance) upon percolation. Unfortunately, however, the full understanding of these systems may have to rely on further extensive research. We may discuss this topic separately in the future.

### 3. Ternary systems

We now extend our discussion to the ternary systems without ultrastrong correlations. It is found that Eq. (15) may be generalized to be

$$N_c = N_c^{(3)} x_3^{\alpha'} + (1 - x_3^{\alpha'}) [N_c^{(2)} \hat{x}_2^\alpha + (1 - \hat{x}_2^\alpha) N_c^{(1)}], \quad (16)$$

where  $\hat{x}_2 = x_2/(x_1 + x_2)$ ,  $\alpha$  and  $\alpha'$  are appropriate correlation exponents, and  $N_c^{(i)}$ 's are still defined by Eq. (14). Note, in Eq. (16), we suppose the rectangles have already been sorted so that if  $i < j$ ,  $l_i > l_j$  or  $w_i > w_j$  when  $l_i = l_j$ . It might be difficult to determine the exact values for  $\alpha$  and  $\alpha'$  in ternary systems because the correlations are among three

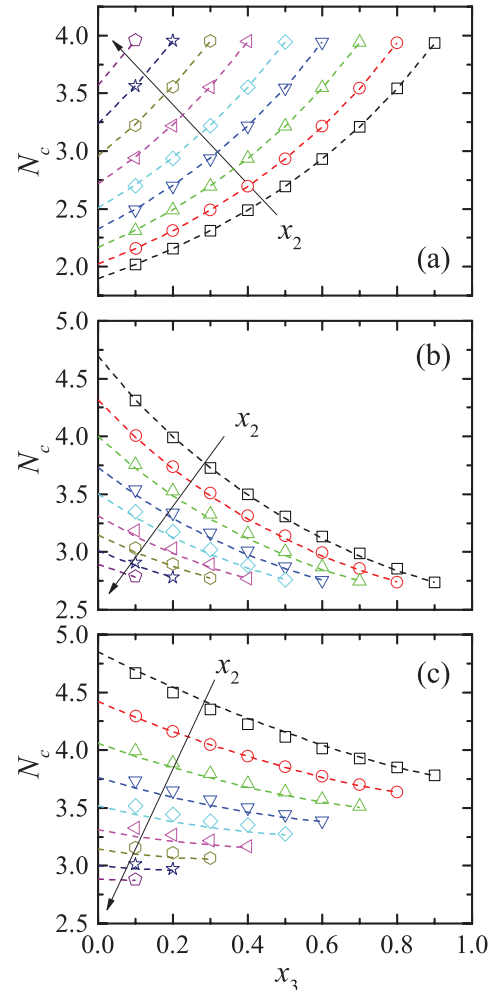


FIG. 8. (Color online) Percolation thresholds obtained from simulations (symbols) and calculations (dashed curves) for three groups of ternary systems: (a)  $\{(1,0.5);(0.6,0.4);(0.5,0.5)\}$ , (b)  $\{(1,0.02);(0.8,0.4);(0.6,0.6)\}$ , and (c)  $\{(1,0.01);(0.8,0.4);(0.5,0.5)\}$ . All calculations are from Eq. (16) with  $\alpha = \alpha' = 1.5$ . In all the plots, along the direction of the arrows, the number fraction  $x_2$  increases from 0.1 to 0.9 at a step of 0.1. Note, in ternary systems,  $x_1 = 1 - x_2 - x_3$ . All  $N_c$ 's are approximated by  $N_{0.5}$  ( $L = 64$ ).

types of rectangles. In most cases, however, one may simply take  $\alpha = \alpha' = 1.5$  at the expense of, possibly, a little higher deviation. Figure 8 compares the simulation and calculation results for three groups of ternary rectangle systems,  $\{(1,0.5);(0.6,0.4);(0.5,0.5)\}$ ,  $\{(1,0.02);(0.8,0.4);(0.6,0.6)\}$ , and  $\{(1,0.01);(0.8,0.4);(0.5,0.5)\}$ . Again, calculations are in excellent agreement with simulations. The systems  $\{(1,0.5);(0.6,0.4);(0.5,0.5)\}$  and  $\{(1,0.02);(0.8,0.4);(0.6,0.6)\}$  only involve weak correlations, and ideal predictions (maximum deviation  $<0.05$ ) are achieved by Eq. (16) [Figs. 8(a) and 8(b)], whereas, the systems  $\{(1,0.01);(0.8,0.4);(0.5,0.5)\}$  involve strong correlations [possibly between the rectangles (1,0.01) and (0.5,0.5)], and the predictions have a little higher deviation (maximum deviation  $\sim 0.07$ ) from the simulations [Fig. 8(c)].

#### IV. CONCLUSIONS

To summarize, integrating the Cohen-Sutherland algorithm into our previously introduced high-efficiency algorithm for continuum percolation, we have developed an efficient Monte Carlo algorithm for both homogeneous and heterogeneous rectangle systems. This algorithm enables us to conduct extensive simulations and to report high-precision percolation thresholds for a variety of rectangle systems. For homogeneous

systems, in terms of the excluded area reciprocals, this paper produces a high-order polynomial [Eq. (5)] which can predict high-precision percolation thresholds (uncertainty  $<7 \times 10^{-5}$ ) for any rectangle aspect ratios. On the basis of this polynomial, a general and simple relation [Eqs. (15) and (16)] is discovered, which can, thereby, be used to predict precise percolation thresholds for heterogeneous rectangle systems without ultrastrong correlations. Therefore, this paper verifies the traditional excluded area theory and extends it into heterogeneous systems. We expect our algorithm and the attained results are useful for many practical applications and theoretical studies concerning 2D layered nanomaterials and various rectangle continuum percolations.

#### ACKNOWLEDGMENTS

Most simulations were performed on the computer cluster “Ferlin” at the PDC Center in KTH, and some were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at C3SE. We acknowledge the financial support by the Advanced Investigator Grant (OSIRIS, Grant No. 228229) from the European Research Council and the 7th Framework Program (NANOFUNCTION, Grant NoE 257375) from the European Commission.

- 
- [1] S. D. Sarma, S. Adam, E. H. Hwang, and E. Rossi, *Rev. Mod. Phys.* **83**, 407 (2011).
- [2] K.-G. Zhou, N.-N. Mao, H.-X. Wang, Y. Peng, and H.-L. Zhang, *Angew. Chem. Int. Ed. Engl.* **50**, 10839 (2011).
- [3] J. N. Coleman *et al.*, *Science* **331**, 568 (2011).
- [4] D. Golberg, Y. Bando, Y. Huang, T. Terao, M. Mitome, C. Tang and C. Zhi, *ACS Nano* **4**, 2979 (2010).
- [5] G. Eda and M. Chhowalla, *Nano Lett.* **9**, 814 (2009).
- [6] M. Hempel, D. Nezych, J. Kong, and M. Hofmann, *Nano Lett.* **12**, 5714 (2012).
- [7] J. Li, F. Ye, S. Vaziri, M. Muhammed, M. C. Lemme, and M. Östling, *Adv. Mater.*, doi: 10.1002/adma.201300361 (2013).
- [8] J. Hicks, A. Behnam, and A. Ural, *Appl. Phys. Lett.* **95**, 213103 (2009).
- [9] I. Balberg, C. H. Anderson, S. Alexander, and N. Wagner, *Phys. Rev. B* **30**, 3933 (1984).
- [10] I. Balberg, *Phys. Rev. B* **31**, 4053(R) (1985).
- [11] J. A. Quintanilla and R. M. Ziff, *Phys. Rev. E* **76**, 051115 (2007).
- [12] S. Mertens and C. Moore, *Phys. Rev. E* **86**, 061109 (2012).
- [13] D. R. Baker, G. Paul, S. Sreenivasan, and H. E. Stanley, *Phys. Rev. E* **66**, 046136 (2002).
- [14] J. Li and S.-L. Zhang, *Phys. Rev. E* **80**, 040104(R) (2009).
- [15] Z. Néda, R. Florian, and Y. Brechet, *Phys. Rev. E* **59**, 3717 (1999).
- [16] V. C. Tung, L.-M. Chen, M. J. Allen, J. K. Wassei, K. Nelson, R. B. Kaner, and Y. Yang, *Nano Lett.* **9**, 1949 (2009).
- [17] G. Cunningham, M. Lotya, N. McEvoy, G. S. Duesberg, P. van der Schoot, and J. N. Coleman, *Nanoscale* **4**, 6260 (2012).
- [18] M. E. J. Newman and R. M. Ziff, *Phys. Rev. Lett.* **85**, 4104 (2000); *Phys. Rev. E* **64**, 016706 (2001).
- [19] T. Vicsek and J. Kertész, *J. Phys. A* **14**, L31 (1981).
- [20] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, 2nd ed. (Addison-Wesley, Reading, 1996), p. 113; [http://en.wikipedia.org/wiki/Line\\_clipping](http://en.wikipedia.org/wiki/Line_clipping).
- [21] R. M. Ziff and M. E. J. Newman, *Phys. Rev. E* **66**, 016129 (2002).
- [22] J. Li and M. Östling, *Phys. Rev. E* **86**, 040105(R) (2012).
- [23] Reference [12] employed almost the same algorithm (combination of Newman-Ziff and subcell) but applied to systems with periodic boundary conditions in both horizontal and vertical directions and, accordingly, used threshold estimates based on wrapping probability [18], instead of spanning probability for FBCs in this paper, to extract more precise values for percolation thresholds in some other continuum systems. The technique may also apply to rectangle systems in this paper. Nevertheless, our attained percolation thresholds should already be sufficiently precise for most practical and theoretical applications.
- [24] J. Li, B. Ray, M. A. Alam, and M. Östling, *Phys. Rev. E* **85**, 021109 (2012).
- [25] G. E. Pike and C. H. Seager, *Phys. Rev. B* **10**, 1421 (1974).
- [26] Y.-D. Liang and B. A. Barsky, *ACM Trans. Graphics* **3**, 1 (1984).
- [27] [http://en.wikipedia.org/wiki/Coefficient\\_of\\_determination](http://en.wikipedia.org/wiki/Coefficient_of_determination); R. J. Freund, W. J. Wilson, and D. L. Mohr, *Statistical Methods*, 3rd ed. (Academic Press, Burlington, MA, 2010), p. 409.
- [28] The prediction may be performed through the `nlpredci` function in Matlab; see <http://www.mathworks.se/help/stats/nlpredci.html>.
- [29] The nonlinear regression may be performed through the `nlifit` function in Matlab; see <http://www.mathworks.se/help/stats/nlifit.html>.
- [30] W. Xia and M. F. Thorpe, *Phys. Rev. A* **38**, 2650 (1988).