

# Pseudo-random-number generators and the square site percolation threshold

Michael J. Lee

*Department of Physics and Astronomy, University of Canterbury, Christchurch, New Zealand*

(Received 3 July 2008; published 25 September 2008)

Selected pseudo-random-number generators are applied to a Monte Carlo study of the two-dimensional square-lattice site percolation model. A generator suitable for high precision calculations is identified from an application specific test of randomness. After extended computation and analysis, an ostensibly reliable value of  $p_c=0.592\,745\,98(4)$  is obtained for the percolation threshold.

DOI: [10.1103/PhysRevE.78.031131](https://doi.org/10.1103/PhysRevE.78.031131)

PACS number(s): 64.60.ah, 02.70.Uu

## I. INTRODUCTION

The square lattice site percolation threshold,  $p_c$ , is a clearly and simply defined mathematical concept [1,2]. Percolation models have been well studied, and are known for their numerous applications [3]. Yet to date, no analytical expression has been found for the numerical value of  $p_c$ . The square site lattice lacks the symmetry that has allowed exact solutions on other topologies [2,4–9]. So long as the problem remains intractable, statistical estimates from Monte Carlo studies can, at least, offer approximate values. Such calculations invariably make extensive use of some form of pseudo-random-number generator (PRNG).

A PRNG is a deterministic algorithm that outputs a sequence of words with properties closely mimicking those of a truly random sequence. Well analyzed generators include the linear congruential, lagged Fibonacci, generalized feedback shift register, and derivatives thereof [10–24]. Because these algorithms are simple, they do not produce output with the complexity of a random sequence [25,26]. The autocorrelation coefficients of a pseudorandom sequence are not identically zero, and these departures from true randomness introduce a sampling bias that leads to systematic error.

Twenty years ago, concern was being given to the demands then made of PRNGs in calculations using  $10^{12}$  pseudorandom numbers generated at MHz rates [27]. Recently, high performance parallel computer systems with thousands, rather than tens or hundreds, of processors have become much more widely available. These enable calculations with  $10^{15}$  pseudorandom numbers generated at GHz rates, and are likely to play a central role in future research. Very high precision can now be achieved through brute force of sampling, but accuracy is another matter. For reliable Monte Carlo estimates at these new higher precision levels, the PRNG(s) chosen must be of sufficient quality. Hence contemporary demands upon PRNGs are, and will continue to become, much greater than in the past.

This study compares several established PRNGs within the context of the square site percolation problem. Following application specific testing, a seemingly reliable generator is identified. This is subsequently used to locate the percolation threshold with, in principle, both accuracy and precision.

## II. GENERATORS

Throughout this study, the computational word length,  $w$ , shall be fixed at 32. All arithmetical operations taking place

within any PRNG are performed in modulo  $2^w$ . All PRNG arithmetical operands, and products thereof, are members of  $\{0:2^w-1\}$ , where  $\{a:b\}$  denotes the set of all integers not less than  $a$  and not greater than  $b$ . Consequently the words of any PRNG output sequence also belong to  $\{0:2^w-1\}$ . The  $i$ th word of an output sequence shall be denoted by  $x_i$ . With one noted exception, no output sequence is decimated in any way. The first million words of each sequence are discarded prior to beginning any Monte Carlo sampling procedure.

Some PRNGs make use of bitwise operations within their internal algorithms. The notation adopted here is  $\oplus$  for bitwise Boolean logical exclusive or, and  $\triangleright m$  for shift  $m$  bits to the right (where  $m$  is a positive integer). Whenever these bitwise operations are performed, the operands are decomposed into their respective standard binary representations, most-significant bit (leftmost) to least-significant bit (rightmost). Arithmetic being constrained to a subset of the integers, any bits shifted to a position right of the decimal point are lost. Hence, within this study, the operation of  $\triangleright m$  is equivalent to integer division by  $2^m$ .

The specific PRNGs considered within this exercise are defined as follows.

TT is the two-tap additive lagged Fibonacci generator  $x_i = x_{i-418} + x_{i-1279}$ . This generator has previously been used for high-precision percolation threshold measurement by Newman and Ziff [28,29].

TTT combines the output from a pair of two-tap generalized feedback shift-register generators,  $u_i = u_{i-471} \oplus u_{i-9689}$  and  $v_i = v_{i-30} \oplus v_{i-127}$ , to return a single word  $x_i = u_i \oplus v_i$ . This is the generator most likely used for two- and three-dimensional percolation by Deng and Blöte [30,31].

SWB is a Marsaglia and Zaman subtract with borrow generator,  $x_i = x_{i-222} - x_{i-237} - \beta_{i-1}$ , where the borrow,  $\beta_i$ , is equal to 1 if  $x_{i-222} < x_{i-237} + \beta_{i-1}$ , and is otherwise equal to zero [16,20].

QTA is the quad-tap generalized feedback shift-register generator  $x_i = x_{i-157} \oplus x_{i-314} \oplus x_{i-471} \oplus x_{i-9689}$ , as used by Ziff and Stell [32,33] (see [17]).

QTB is the quad-tap generalized feedback shift-register generator  $x_i = x_{i-471} \oplus x_{i-1586} \oplus x_{i-6988} \oplus x_{i-9689}$ . This generator has been used by Newman and Ziff, and has been found to produce threshold estimates consistent with those of the TT generator [28,29,34].

XG is Brent's xor4096 generalized Marsaglia xorshift generator [21,23]. Specifically, the implementation xor4096i, from Brent's C language xorgens304 distribution, was that

used here. This generator has performed well in randomness tests conducted by L'Ecuyer and Simard [24].

MT is Matsumoto and Nishimura's MT19937 Mersenne twister generator [18]. Specifically, their MT19937ar C language distribution was the implementation used here. The MT19937 algorithm has been used for computing integrals in semirigorous work by Balister, Bollobás, Walters, and Riordan [35,36], and for Monte Carlo sampling by Lee [37].

DMT is a pair of MT generators operated entirely independently of one another. The output sequence from each of these generators is decimated, with only every fourth word used. Lattice sites are then selected by means of their Cartesian coordinates, using one number from each generator. This scheme has previously been used by Lee [37].

Let  $L$  be the number of sites lying on each edge of an  $L \times L$  square lattice. Every site on that lattice is typically given a unique index or label,  $j \in \{0:L^2-1\}$ . In a microcanonical ensemble Monte Carlo calculation [28,29,34], such as those performed here, PRNG output words,  $x_i$ , are used to pseudorandomly select sites,  $s_j$ , for occupation. Now, consider a transformation  $T(x, N) \equiv x \gg (w - \log_2 N)$ , where  $N$  is a positive-integer power of 2. This is a distribution preserving many-to-one surjective map from the integers  $x \in \{0:2^w-1\}$  to the integers  $T(x, N) \in \{0:N-1\}$ . Further consider a bijection  $H$  that maps the integers  $\{0:N-1\}$  onto site labels. With the usual choice of site labels also being the integers  $\{0:N-1\}$ ,  $H$  is conventionally taken to be the identity map. In the single generator systems defined above, PRNG output word  $x_i$  is associated with site  $s_{j(x_i)}$  via  $j(x_i) = H[T(x_i, L^2)]$ . For the DMT, a pair of PRNG output words,  $u_i$  and  $v_i$  (one from each of the output decimated MT generators), is mapped to site  $s_{j(u_i, v_i)}$  via  $j(u_i, v_i) = H[T(u_i, L) + LT(v_i, L)]$ . This halves the number of bits actually used from each output word (the most significant bits being those retained).

Each of these generators must be provided with an initial finite sequence of words from which to begin calculating an infinite pseudorandom sequence. In the case of the TT generator for instance, a list of some 1279 initial words is required. These initial lists were constructed by one of four simpler generators, here denoted LCGa, LCGb, LCGm, and WMx. LCGa is the linear congruential generator,  $x_i = 69\,069x_{i-1} + 1$ , suggested by Marsaglia [12]. LCGb is a similar linear congruential generator,  $x_i = 69\,069x_{i-1} + 1\,234\,567$ , also due to Marsaglia [20]. LCGm is the modified linear congruential generator,  $x_i = 1\,812\,433\,253[x_{i-1} \oplus (x_{i-1} \gg 30)] + i$ , appearing in Matsumoto and Nishimura's MT19937ar distribution of their Mersenne twister algorithm. WMx is the Weyl modified Marsaglia xorshift generator built into the xor4096i algorithm appearing within Brent's xor-gens304 distribution [21,23]. These initialization generators are themselves seeded from a single word,  $x_0 \in \{0:2^w-1\}$ . The TTT and DMT generators both require two initialization lists, each being derived from one of these four generators, each starting with a distinct independent seed word. Single word initialization does limit the number of different sequences that can be obtained from a given combination of main generator and initialization generator to  $2^w$ . This is not a problem here since the total number of sequences generated was five orders of magnitude less than this, and those were divided between many different generator combinations.

### III. TEST PROCEDURE

The above listed generators were compared, in the context of site percolation on the square lattice, by using each one to make a Monte Carlo estimate of the crossing probability function,  $R_{L,n}$ , at  $L=2048$ , over the domain  $n \in \{2\,474\,000:2\,498\,300\}$ .  $R_{L,n}$  is defined as the probability that a single cluster connects two specified opposing boundary sides of the  $N=L \times L$  square lattice in the microcanonical ensemble when precisely  $n$  random sites are occupied. The value of  $R_{2048,n}$  monotonically increases from around 0.05 at  $n=2\,474\,000$  to around 0.95 at  $n=2\,498\,300$ . Hence, the occupation domain studied encompasses the critical region of the percolative phase transition. The numerous lattice configurations required to accurately determine  $R_{L,n}$  were constructed, from each PRNG output sequence, over the above domain only, by the unbiased algorithm of Lee [37]. The only exception was the XG generator from which samples were obtained over the same domain by the unbiased algorithm of Newman and Ziff [28,29].

The Newman and Ziff binomial convolution

$$R_L(p) = \sum_n \binom{N}{n} p^n (1-p)^{N-n} R_{L,n} \quad (1)$$

then gives the crossing probability,  $R_L(p)$ , in the canonical ensemble where each lattice site is independently randomly occupied at probability  $p$  [28,29,34]. In principle the summation should run over all  $n \in \{0:N\}$ , but as samples were taken only over the restricted domain of  $n$  above, the summation was truncated accordingly. The standard deviation of the binomial distribution in Eq. (1) is given by  $\sigma_{L,p} \approx L\sqrt{p(1-p)}$ . The data analysis here is concerned with values of  $p$  such that the distribution maximum, located at  $n = \text{nint}(pN)$ , lies between 10 and  $12\sigma_{L,p}$  from the nearest end of the sampling region. Consequently, the truncation induced error in  $R_L(p)$  is not more than  $10^{-15}$ . This is completely negligible when compared to statistical sampling uncertainties, which were never less than  $10^{-8}$ .

The canonical crossing probability curve is used to identify a site occupation probability,  $p_f(L)$ , defined such that

$$R_L[p_f(L)] = 1/2 + k/L, \quad (2)$$

where  $k=0.320(1)$ , as determined by Ziff and Newman [34] (their parameter  $b_0$ ). For  $p \approx p_c$ , to first order  $R_L(p) \sim 0.5 + k/L + O[(p-p_c)L^{1/\nu}]$  [33], and hence  $p_f(L)$  provides a reasonable estimate of the critical point  $p_c$ . Ziff and Newman have found that the second-order equation  $R_L(p_c) \approx 0.5 + kL^{-1} - 0.44L^{-2}$  is a better model of the data at small  $L$  [34], however the  $L^{-2}$  term is negligible for  $L \geq 1024$  at the levels of precision considered here. Values for  $p_f(2048)$  were thus obtained from each of the PRNGs described above. These were subsequently compared against each other and against previous  $p_c$  estimates made with the same generators. For large  $L$ ,  $R_L(p)$  rises very steeply in the neighborhood of  $p \approx p_c$ . Consequently,  $p_f(L)$  is relatively insensitive to the exact value of  $k$  provided that  $k/L \ll 1$ . The uncertainty in  $k$  limits the maximum attainable precision in  $p_f(2048)$  to  $\pm 2 \times 10^{-9}$ .

$s_0$	$s_1$	$s_2$	$s_3$	$s_0$	$s_{11}$	$s_6$	$s_1$
$s_4$	$s_5$	$s_6$	$s_7$	$s_{12}$	$s_7$	$s_2$	$s_{13}$
$s_8$	$s_9$	$s_{10}$	$s_{11}$	$s_8$	$s_3$	$s_{14}$	$s_9$
$s_{12}$	$s_{13}$	$s_{14}$	$s_{15}$	$s_4$	$s_{15}$	$s_{10}$	$s_5$

FIG. 1. Standard (left-hand side) and (example) nonstandard (right-hand side) enumerations of the  $L=4$  square lattice.

Combinatorial terms of the binomial distribution in Eq. (1) were calculated by the essentially exact method of Newman and Ziff [29]. For  $p$  near  $p_c$ , use of the Gaussian approximation to the binomial would have introduced an error of order  $10^{-8}$  in  $R_{2048}(p)$ , this corresponding to an error of order  $10^{-10}$  in  $p$  itself. It is sometimes possible to dispense with the convolution altogether and make a microcanonical ensemble approximation of  $R_L(p=n/N) \approx R_{L,n}$ . With  $L=2048$ , and for  $p$  near  $p_c$ , this introduces an error of around  $4 \times 10^{-6}$  in  $R_L(p)$ , which corresponds to an error of around  $2 \times 10^{-8}$  in  $p$ . This approximation is acceptable at low enough precision, has the advantage that only a much narrower domain of sampling need be considered, and has been employed in earlier work by Lee [37]. However, since the induced error [measured as the difference between  $p$  and  $n/N$  such that either  $R_L(p)=R_{L,n}=0.5$  or  $R_L(p)=R_{L,n}=0.5+k/L$ ] was found to scale as only  $L^{-1.5(3)}$ , when a set of measurements are to be taken over a range of lattice sizes, to precisions of order  $1/N$ , the error will become significant at large  $L$ . The approximation was not adopted here.

Correlations inevitably found in the output sequence of any deterministic pseudo-random-number generator will result in correlations within the spatial pattern of occupied sites upon the lattice. As noted by Compagner [38], this in turn will bias the resulting Monte Carlo estimate of the crossing probability function. Consequently, when estimates obtained from two different generators are inconsistent, then at least one of those generators likely suffers from significant correlations in its output sequence, hence rendering it unsuitable for use at the level of precision of the study. Because the true values of  $R_{L,n}$ ,  $R_L(p)$ , and  $p_c$  are unknown, it will be unclear as to which of the generators is deficient.

While there is merit in performing general tests on PRNGs, it is often preferable to have an application specific test such as the sensitive hull walk of Ziff [17]. Here a scheme is used that changes the relation between numerical PRNG output sequences and the spatial patterns of occupied lattice sites, without altering the underlying problem or topology in any way. The standard enumeration of the lattice, as shown in Fig. 1, prescribes a specific relation between patterns in PRNG output and in clusters of occupied sites. By adopting some other (nonstandard) enumeration, as per the example in Fig. 1, some different relation is obtained. An ideal random-number generator will produce results independent of the chosen enumeration. A pseudo-random-number generator, with correlations in its output sequence, will produce results that do depend upon the enumeration. By comparing results from a common generator on two different lattice enumerations, inadequate, outcome biasing, generators may be identified. This simple application specific test does not require knowledge of the percolation threshold or spanning probability curves.

The direct approach to implementing such an enumeration is to allocate each site a set of pointers explicitly identifying

its geometrical neighbors. In the nonstandard enumeration of Fig. 1, for example,  $s_7$  would have pointers to  $s_2, s_3, s_{11}$ , and  $s_{12}$ . Sites are pseudorandomly selected as per normal and the Monte Carlo sampling proceeds just as for the standard enumeration. In practice this results in a dramatic performance decrease of the simulations (more than a factor of 2 was found in this study). The problem is believed to be the cache prefetching of the high performance computer system used, where if, in some linear array,  $s_j$  is being accessed then the hardware assumes that  $s_{j+1}$  (being the next contiguous data element in memory) will be wanted next.

An alternative method is to change the mapping,  $H$ , between scaled PRNG output words,  $y=T(x,N)$ , and site labels,  $j$ . In the standard enumeration of Fig. 1,  $H(y)=y$  is the identity map. In the nonstandard enumeration,  $H(0)=0$ ,  $H(11)=1$ ,  $H(6)=2$ , and so on, with the general relation being  $H(y)=3y \pmod{16}$ . This is analogous to the cluster-label labels of Hoshen and Kopelman [39]. The nonstandard enumeration is that effectively in use, while the standard enumeration is preserved in computer memory, thus avoiding performance problems.

When the hash function,  $H$ , is simple (that is, of similar algebraic complexity to the PRNG), it will not so much hide correlations in the output sequence as manifest those patterns in some other way, giving rise to a different estimate for  $p_f(L)$ . If, on the (systematic) standard enumeration, correlations in PRNG output lead to spatial correlations of occupied sites that in turn bias the estimate, then, on some other (systematic) nonstandard enumeration, those same PRNG correlations will give rise to spatial correlations of a different nature that bias the estimate in some other way. This provides a simple, application specific, test for PRNG biasing of the Monte Carlo samples. If a given PRNG is correlation free, then the estimates derived from it will be independent of the lattice enumeration. If instead, the PRNG output does suffer from correlations, then different enumerations may lead to different results. The test can be tuned, with the hash chosen so as to maximize the observed shift in the test quantity. A simple hash related to the taps or period is bound to highlight intrinsic PRNG shortcomings [38]. Alternatively, a hash much more complex than the generator algorithm could go some way toward hiding output sequence correlation induced bias.

Hence define two further generators, TTH and MTH, as (respectively) the exact same TT and MT generators defined previously, but with a somewhat arbitrary nontrivial mapping  $H(y)=947y \pmod{N}$  between integers  $y \in \{0:N-1\}$  and site labels  $j \in \{0:N-1\}$ . On a lattice of  $L=2048$ , this hash is equivalent to a systematic nonstandard enumeration where the rightward and downward neighbors of site  $s_j$  are (when they exist)  $s_{j+3} \pmod{N}$  and  $s_{j+1} \pmod{N}$ , respectively.

#### IV. TEST RESULTS

The various estimates of  $p_f(2048)$  thus obtained are listed in Table I. Results are separated according to the generator and initialization scheme used. SWBb, for instance, indicates the SWB generator with its initialization list derived from the

TABLE I. Site percolation threshold estimates for the square lattice [ $p_f(2048)$ ] obtained by various pseudo-random-number generators (PRNGs) as described in the text.

PRNG	$p_f(2048)$
TTa	0.592 746 27(11)
TTHa	0.592 745 88(11)
TTTab	0.592 746 28(12)
SWBb	0.592 746 17(17)
QTAA	0.592 745 88(17)
QTAm	0.592 746 03(17)
QTBa	0.592 746 10(17)
QTBb	0.592 746 21(17)
XGx	0.592 745 96(15)
MTa	0.592 745 93(17)
MTm	0.592 745 85(16)
MTHm	0.592 745 98(12)
DMTmm	0.592 745 97(08)

LCGb output sequence. Similarly, XGx is the XG generator initialized from the WMx output sequence. Results shown are based on surveys of order  $10^8$  effectively independent samples per occupation level  $n$ . Each of these sets involved the generation of order  $10^{13}$  to  $10^{14}$  (approaching  $10^{15}$  in the XG case), pseudorandom numbers.

TTa is the TT generator initialized from LCGa. The TTa based  $p_f(2048)$  estimate in Table I is consistent with the results of Newman and Ziff that were also obtained (primarily [40]) from the TT generator (see Table II). TTHa is the hashed TT generator, again initialized with LCGa. The TTa and TTHa based estimates are sufficiently different to indicate the probable existence of statistically significant correlations within the TT generator output sequence. This gives cause for concern about the use of the TT PRNG for this application at this level of precision.

TTTab is the TTT generator with its initial  $u$  and  $v$  lists constructed by LCGa and LCGb, respectively (the two initializing generators being given two different seeds). The TTT-based estimate in Table I is consistent with the Table II results of Deng and Blöte most likely obtained from this generator.

QTAA and QTAm are the QTA generators respectively, initialized from LCGa and LCGm. Since the QTAA and QTAm results are consistent, there is no evidence that estimates from the QTA generator are especially sensitive to initialization. The union of these two data sets gives an overall estimate of  $p_f(2048)=0.592\ 745\ 95(12)$  from the QTA generator. This value is consistent with earlier results, in Table II, obtained by Ziff and Stell with this generator.

QTBa and QTBb are the QTB generator initialized from LCGa and LCGb, respectively. Since the QTBa and QTBb results are consistent, there is no evidence that estimates from the QTB generator are especially sensitive to initialization. The union of these two data sets gives an overall estimate of  $p_f(2048)=0.592\ 746\ 16(12)$  from the QTB generator. This is consistent with the TTa value, and also with the

Newman and Ziff similar observation regarding these two generators [28,29]. Referring to Table II, the value is also consistent with the various threshold estimates obtained by Newman and Ziff using (at least in part) this generator.

MTa is the MT generator initialized from LCGa. MTm is the MT generator initialized from LCGm. Since the MTa and MTm results are consistent, there is no evidence that estimates from the Mersenne twister generator are sensitive to initialization. The union of these two data sets gives an estimate of  $p_f(2048)=0.592\ 745\ 89(12)$ . This is consistent with the MTHm result derived from the hashed generator in Table I, and hence there is no evidence that correlations in the MT output sequence influence the measurement at this level of precision. Hence, the MT generator appears to be an adequate choice for the current application. Further combining the MTHm data into the union gives an overall estimate of  $p_f(2048)=0.592\ 745\ 93(8)$  from the MT generator. This MT result is inconsistent with those of the TTT and (unhashed) TT generators. The difference in results with respect to the QTB generator is no more than could be expected by chance in a data set of this size. The SWB, QTA and XG generator-based estimates are consistent with that of the MT.

DMTmm is the DMT generator with its two initial lists independently constructed, each from one of a pair of seed words, by LCGm. The DMTmm result is consistent with the combined MT result, thereby indicating that any possible correlations between lower order bits in MT output words are insignificant at this level of precision, or at least no worse than correlations in the higher-order bits. This suggests that the single MT generator will be adequate for the purposes of this study. Combining all four Mersenne twister-based data sets; MTa, MTm, MTHm, and DMTmm, produces an estimate of  $p_f(2048)=0.592\ 745\ 95(6)$ . This is consistent with the result of Lee, in Table II, obtained with this same mixture of generators but in the microcanonical approximation  $R_L(n/N) \approx R_{L,n}$ . The combined value does not alter any of the above conclusions regarding the consistency or otherwise of other generators with the Mersenne twister. Regarding the previous estimate of Lee, it was observed here that  $n/N$ , such that  $R_{2048,n}=0.5+k/L$  (interpolating to noninteger  $n$ ), usually exceeds  $p_f(2048)$  by approximately  $2 \times 10^{-8}$ . That being so, a revised estimate of the presented result would be  $p_c=0.592\ 7460(1)$ . This adjustment is much smaller than statistical uncertainties.

Although the procedure used here differs from those of previous works, the results obtained are found to be consistent when the same pseudo-random-number generators are used. However, given the use of a consistent method, it has been shown that the results thus obtained can differ with the choice of generator. The level of PRNG sensitivity will be method dependent. The spread in results seen here is not extreme as only reasonable quality generators have been used.

A great deal of theoretical work has gone into developing these classes of generator, and extensive general tests have been made of them elsewhere. Ziff has performed a sensitive hull generating walk test upon several generalized feedback shift register generators [17]. Two-tap generators performed poorly in this test which concluded that they best be avoided for critical applications. Certain quad-tap generators, particu-

TABLE II. Estimates of the square site percolation threshold presented in the literature. The pseudo-random-number generator(s) used are given where known. Generator T is a Tausworthe generator, while C is a congruential generator. TTT is the generator most likely used by Deng and Blöte. References are provided for both the result and the generator whenever those come from different sources. Uncertainties are quoted as one standard deviation statistical errors, except in the semirigorous results of Balister, Bollobás, and Walters (99.99% confidence bound) and of Riordan and Walters (99.9999% confidence bound). Only those results derived from currently accepted scaling relations are shown from the greater collection in Hu, Chen, and Wu. This table is essentially a continuation of that appearing in Ziff and Sapoval [41], there going back to 1960.

Year	Reference	Author(s)	Method	Generator(s)	Result
1986	[41]	Ziff and Sapoval	Hull gradient	T	0.592 745(2)
1988	[17,32]	Ziff and Stell	Hull gradient	QTA	0.592 746 0(5)
1989	[42]	Yonezawa, Sakamoto, and Hori	Planar crossing		0.593 0(1)
1992	[33]	Ziff	Hull crossing	QTA	0.592 746 0(5)
1994	[43]	Hu	Histogram Monte Carlo		0.592(8)
1995	[44]	Hu	Histogram Monte Carlo		0.592 8(1)
1996	[45]	Hu, Chen, and Wu	Histogram Monte Carlo		0.592 78(2)
1996	[45]	Hu, Chen, and Wu	Histogram Monte Carlo		0.592 83(4)
1996	[45]	Hu, Chen, and Wu	Histogram Monte Carlo		0.592 67(6)
1996	[45]	Hu, Chen, and Wu	Histogram Monte Carlo		0.581 4(30)
1996	[45]	Hu, Chen, and Wu	Histogram Monte Carlo		0.604 1(30)
2000	[28,29]	Newman and Ziff	Toroidal wrapping	TT, QTB	0.592 746 21(13)
2000	[28,29]	Newman and Ziff	Toroidal wrapping	TT, QTB	0.592 746 36(14)
2000	[28,29]	Newman and Ziff	Toroidal wrapping	TT, QTB	0.592 746 06(15)
2000	[28,29]	Newman and Ziff	Toroidal wrapping	TT, QTB	0.592 746 29(20)
2000	[28,40]	Ziff	Hull gradient	QTB	0.592 746 5(2)
2002	[34]	Ziff and Newman	Planar crossing	QTB	0.592 746 4(5)
2003	[46,47]	Martins and Plascak	Toroidal wrapping	C	0.592 7(1)
2003	[46,47]	Martins and Plascak	Toroidal wrapping	C	0.592 9(3)
2005	[30,31]	Deng and Blöte	Cylindrical correlation	TTT	0.592 746 5(4)
2005	[30,31]	Deng and Blöte	Cylindrical correlation	TTT	0.592 746 6(6)
2005	[30,31]	Deng and Blöte	Cylindrical correlation	TTT	0.592 746 6(8)
2005	[30,31]	Deng and Blöte	Cylindrical correlation	TTT	0.592 746 8(10)
2005	[35]	Balister, Bollobás and Walters	Semirigorous	MT	0.592 7(8)
2007	[36]	Riordan and Walters	Semirigorous	MT	0.592 75(25)
2007	[37]	Lee	Planar crossing	MT, DMT	0.592 746 03(9)

larly QTB, performed very well. Analysis indicated that QTB should outperform QTA in principle, although no obvious problems were observed in the latter. Problems with generators in the same class as TT, TTT, and QTA have also been noted by Matsumoto and Kurita [48]. The MT generator has passed tuned collision tests conducted by Tsang, Hui, Chow, Chong, and Tso [22]. The LCGa generator failed those same tests. L'Ecuyer and Simard have recently performed thorough randomness tests upon a large assembly of PRNGs, including SWB, QTB, MT, XG, and LCGa [24]. The XG generator passed all tests, the MT failed in a very limited number of instances, QTB and SWB both failed a small number of times, and LCGa failed badly. TT was not specifically tested, although two-tap generators typically performed poorly. Tezuka, L'Ecuyer, and Couture have shown that generators of the SWB class are essentially equivalent to large prime modulus linear congruential generators and so can be unreliable [49]. New generators named WELL, with

improved output sequence properties over the MT generator, have recently been devised by Panneton, L'Ecuyer, and Matsumoto [50].

As noted by Ferrenberg, Landau, and Wong, it is highly desirable to have algorithm and application specific tests of pseudo-random-number generators, regardless of any general tests that the generator may have passed [51]. Within this exercise, results from the Mersenne twister generator have been consistent under different initialization methods and effective lattice enumerations (hash functions). With the observation that results from the Mersenne twister differ from those of the two-tap lagged Fibonacci generator, in the presence of evidence suggesting that the two-tap suffers from significant output correlations, and in the absence of evidence for any significant correlations in the Mersenne twister output sequence, further Monte Carlo sampling within this exercise shall be performed exclusively with the MT19937 algorithm. Note that results from the SWB, QTA, QTB, and

TABLE III. Site percolation threshold estimators on square lattices of various sizes  $L$ .  $p_m(L)$  is the median- $p$  estimator,  $p_{cc}(L)$  is the cell-to-cell estimator,  $p_h(L)$  is the linear combination estimator, and  $p_r(L)$  is the fixed point estimator. Results were obtained with the Mersenne twister pseudorandom number generator.

$L$	$p_m(L)$	$p_{cc}(L)$	$p_h(L)$	$p_r(L)$
128	0.592 661 08(21)			0.595 983 52(23)
256	0.592 720 62(18)	0.592 808 5(4)	0.592 746 0(5)	0.594 674 66(18)
512	0.592 738 60(15)	0.592 765 1(4)	0.592 746 2(4)	0.593 892 58(15)
1024	0.592 743 77(14)	0.592 751 4(4)	0.592 746 0(4)	0.593 426 99(15)
2048	0.592 745 28(06)	0.592 747 5(2)	0.592 745 9(2)	0.593 150 51(06)
4096	0.592 745 73(10)	0.592 746 4(3)	0.592 745 9(3)	0.592 986 26(10)

XG generators are consistent with those of the MT. Also, it might be interesting to conduct future work with the high quality XG [23] or WELL [50] generators.

### V. THRESHOLD DETERMINATION

Having identified the Mersenne twister as a suitable PRNG for the problem, a more precise determination of the square site percolation threshold can now be made. This will be based upon Monte Carlo estimates of the microcanonical  $R_{L,n}$  curves for  $128 \leq L \leq 4096$  (a span of some three orders of magnitude in  $N$ ).

Data for  $L \leq 1024$  was obtained exclusively from the MT generator initialized by LCGm, and Monte Carlo sampling was conducted with the algorithm of Lee [37]. Sampling domains were  $n \in \{8900:10\,500\}$  on the  $L=128$  lattice,  $n \in \{37\,300:40\,400\}$  on the  $L=256$  lattice,  $n \in \{152\,300:158\,500\}$  on the  $L=512$  lattice, and  $n \in \{615\,500:627\,600\}$  on the  $L=1024$  lattice. The data at  $L=2048$  is the combined MTa, MTm, MTHm, and DMTmm data from Table I. As noted, that data was obtained with the same algorithm over the site occupation domain  $n \in \{2\,474\,000:2\,498\,300\}$ . Due to hardware constraints, the  $L=4096$  data was obtained with the more memory efficient algorithm of Newman and Ziff [28,29]. For this algorithm, the entire domain,  $n \in \{0:N\}$ , is sampled, however observations were made only for  $n \in \{9\,920\,000:9\,969\,000\}$ . Once again, the LCGm initialized MT generator was used. Lattices of  $L$  much more than 4096 could not be accommodated by the computer system used without substantial decreases in performance. Estimates at each  $L$  are based upon between  $1 \times 10^8$  (at  $L=4096$ ) and  $4 \times 10^9$  (at  $L=128$ ) independent samples per occupation level,  $n$ . These required the generation of between  $10^{13}$  (at  $L=128$ ) and  $10^{15}$  (at  $L=4096$ ) pseudorandom numbers.

As before, these microcanonical ensemble crossing probability curves,  $R_{L,n}$ , were transformed into canonical ensemble crossing probability functions,  $R_L(p)$ , by the convolution of Eq. (1). Because the various microcanonical sampling domains all encompass  $\pm 12\sigma_{L,p}$  of the convolution region about the critical point, the domain restriction induced error in  $R_L(p)$  is completely negligible for the values of  $p$  considered here.

Several statistics were calculated from each  $R_L(p)$  curve. These were Ziff's median- $p$  critical point estimator [33],  $p_m(L)$ , defined such that

$$R_L[p_m(L)] = 1/2, \quad (3)$$

the Reynolds, Stanley, and Klein real-space renormalization group cell-to-cell estimator [52,53],  $p_{cc}(L)$ , defined such that

$$R_L[p_{cc}(L)] = R_{L/2}[p_{cc}(L)], \quad (4)$$

the Ziff and Newman linear combination estimator [34],  $p_h(L)$ , defined as

$$p_h(L) \equiv [p_m(L) + \alpha p_{cc}(L)] / (1 + \alpha), \quad (5)$$

where  $\alpha \equiv 1 - 2^{-1/\nu}$ , and the real-space renormalization group cell-to-site fixed point estimator of Reynolds, Klein, and Stanley [54],  $p_r(L)$ , defined such that

$$R_L[p_r(L)] = p_r(L). \quad (6)$$

Numerical estimates for these quantities are shown in Table III.

The estimators  $p_m$  and  $p_{cc}$  are believed to approach their limiting values on the infinite lattice as  $L^{-1-1/\nu}$ , where  $\nu = 4/3$  [33,34]. The estimator  $p_h$  is believed to converge to its limit at a faster rate of  $L^{-1-\omega-1/\nu}$ , where Ziff and Newman have determined a value of  $\omega=0.90(2)$  [34] for the scaling exponent proposed by Aharony and Hovi [55,56]. The estimator  $p_r$  is believed to approach its limit as  $L^{-1/\nu}$ , a much slower rate of convergence than for the other estimators [34]. Although each of these four limits is numerically equivalent to the percolation threshold  $p_c$ , it will be useful to adopt a general notation indicating the origin of any threshold estimates.

To second order, the finite size scaling relation for the median- $p$  estimator is

$$p_m(L) \approx p_m^* - aL^{-1-1/\nu} + bL^{-1-\omega-1/\nu}. \quad (7)$$

A parametrized fit of Eq. (7) to the data of Table III produces  $p_m^* = 0.592\,745\,95(4)$ ,  $a = 0.413(5)$ , and  $b = 0.0(4)$ . That the coefficient  $b$  is indistinguishable from zero suggests that a first-order model [Eq. (7) with  $b$  constrained to zero] is appropriate for the data. As such, a precise value for  $\omega$  is unimportant. In this first-order case, the coefficients are evaluated as  $p_m^* = 0.592\,745\,96(3)$ , and  $a = 0.4135(7)$ . The very good agreement between model and experiment is shown in Fig. 2. An empirical power-law fit of the form  $p_m(L) \sim p_m^* - aL^z$  yields  $p_m^* = 0.592\,745\,95(5)$ ,  $a = 0.42(2)$ , and  $z = -1.75(8)$ . That the value of  $z$  is indistinguishable from the

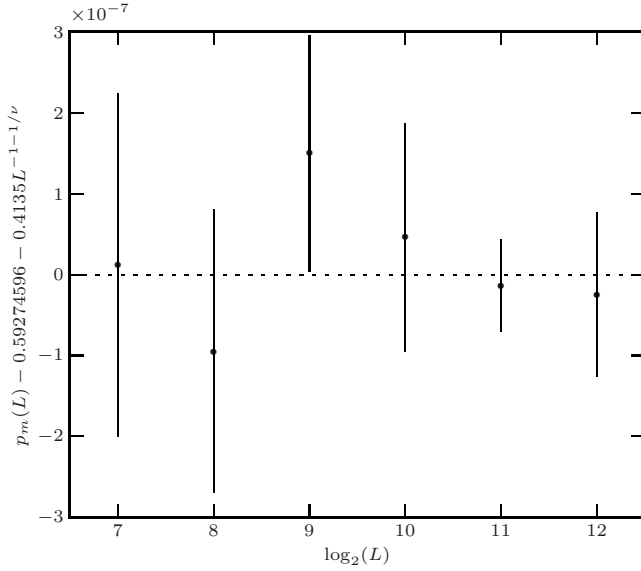


FIG. 2. Parametrized fit of first-order scaling theory [Eq. (7) with  $b=0$ ] to experimental data [ $p_m(L)$  of Table III] for the median- $p$  critical point estimator.

assumed exponent of  $-1-1/\nu$ , further supports scaling of the form  $L^{-1-1/\nu}$  as being the appropriate model for the data at this level of precision. All three  $p_m^*$  estimates are in good agreement with one another.

The second-order scaling relation for the cell-to-cell estimator is given by

$$p_{cc}(L) \approx p_{cc}^* + \frac{a}{\alpha} L^{-1-1/\nu} + cL^{-1-\omega-1/\nu} \quad (8)$$

[34]. The quality of the cell-to-cell data is lower than that of the median- $p$  data, as each point is obtained from the intercept of two lines, with statistical uncertainties, at a shallow angle, and as  $p_{cc}(L)$  and  $p_{cc}(L/2)$  are not entirely independent. A parametrized fit of Eq. (8) to the data of Table III produces  $p_{cc}^* = 0.5927458(2)$ ,  $a = 0.441(5)$ , and  $c = -9(8)$ . Coefficient  $c$  is not inconsistent with zero, and a first-order model [Eq. (8) with  $c$  constrained to zero] does fit the data, as shown in Fig. 3, with coefficients of  $p_{cc}^* = 0.5927459(2)$  and  $a = 0.417(4)$ , in good agreement with the median- $p$  estimator results. An empirical power-law fit of the form  $p_{cc}(L) \sim p_{cc}^* - (a/\alpha)L^z$  yields  $p_{cc}^* = 0.5927458(2)$ ,  $a = 0.34(8)$ , and  $z = -1.71(4)$ , consistent with the assumed  $L^{-1-1/\nu}$  scaling relation. All three  $p_{cc}^*$  estimates are consistent with each other and with the estimates for  $p_m^*$ , although the precision is significantly lower.

The linear combination estimator of Eq. (5) was constructed by Ziff and Newman [34] so as to cancel the first-order terms of Eqs. (3) and (4), leaving a faster approach to the percolation threshold,

$$p_h(L) \sim p_h^* + \frac{b + \alpha c}{1 + \alpha} L^{-1-\omega-1/\nu} \quad (9)$$

(to first order). A parametrized fit of this expression to the data of Table III is shown in Fig. 4 and produces  $p_h^*$

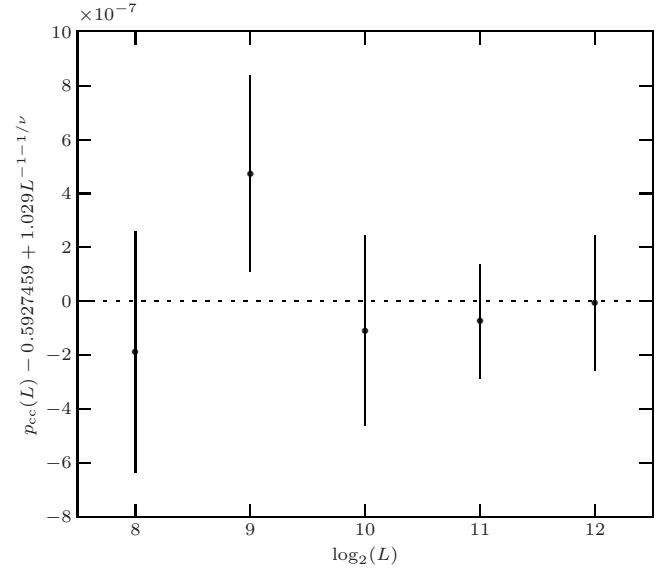


FIG. 3. Parametrized fit of first-order scaling theory [Eq. (8) with  $c=0$ ] to experimental data [ $p_{cc}(L)$  of Table III] for the cell-to-cell critical point estimator.

$= 0.59274596(7)$ , and  $(b + \alpha c) = 0.3(9)$ . The threshold result is in good agreement with those obtained from the median- $p$  estimator data. The value of  $(b + \alpha c)$  is also in agreement, although this is not saying much given the large uncertainties. That this value is essentially indistinguishable from zero is a reflection of the rapid rate of convergence of the  $p_h(L)$  estimator with  $L$ , as suggested by Eq. (9), and the relative lack of precision in the  $p_h(L)$  data. This is unsurprising given that no higher-order terms were apparent in either the  $p_m(L)$  or  $p_{cc}(L)$  data sets. As such, the data was inadequate to empirically test the assumed scaling exponent and is even consistent with  $p_h(L) = \text{constant} = p_h^*$ , for which fitting the weighted mean gives  $p_h^* = 0.5927460(1)$ .

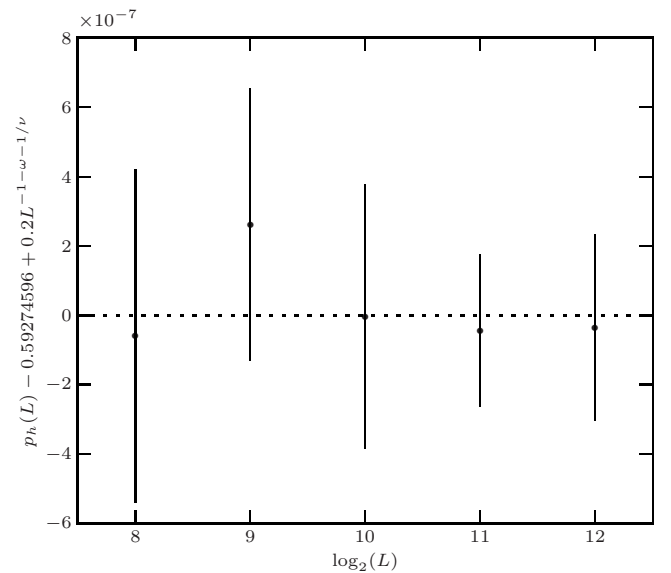


FIG. 4. Parametrized fit of scaling theory [Eq. (9)] to experimental data [ $p_h(L)$  of Table III] for the linear combination critical point estimator.

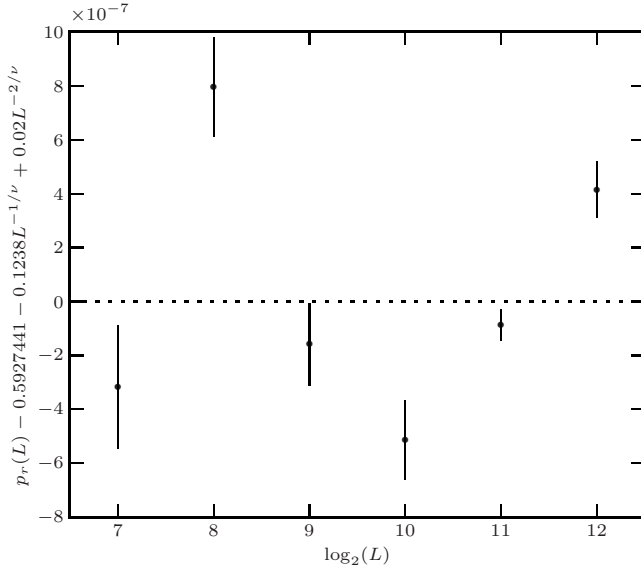


FIG. 5. Parametrized fit of scaling theory [Eq. (10)] to experimental data [ $p_r(L)$  of Table III] for the renormalization group fixed point percolation threshold estimator.

The second-order scaling relation for the fixed point renormalization group estimator is given by

$$p_r(L) \sim p_r^* + rL^{-1/\nu} + sL^{-2/\nu} \quad (10)$$

[34], and so has a slower rate of convergence to its infinite lattice limit than any of the other estimators above. A fit to the data of Table III yields well-defined numerical values for the coefficients;  $p_r^* = 0.5927441(7)$ ,  $r = 0.1238(2)$ , and  $s = -0.021(6)$ . However, as shown in Fig. 5, the model of Eq. (10) is but a loose match to the data at best, with higher-order terms evidently remaining significant. As such, the stated uncertainty in  $p_r^*$  is misleading and will be addressed within the next section. An empirical power-law fit of the form  $p_r(L) \sim p_r^* - rL^z$  yields  $p_r^* = 0.592743(2)$ ,  $r = 0.1219(8)$ , and  $z = -0.748(2)$ , consistent with the assumed first-order exponent of  $-1/\nu$ . The first-order model [Eq. (9) with  $s$  constrained to zero] returns  $p_r^* = 0.5927456(8)$  and  $r = 0.1233(1)$ . Of course, neither of these two functions describe the data any better than does the second-order model.

## VI. ROBUSTNESS

Several estimates have now been made for the square site percolation threshold,  $p_c$ , using all the data of Table III and with varying degrees of precision. Of these, the most precise is  $p_c = p_m^* = 0.59274596(3)$ , obtained from the median- $p$  estimator data using the first-order scaling model  $p_m(L) = p_m^* + aL^{-1/\nu}$ . It is prudent to establish the robustness of the results with respect to variations in the data and in the assumed model, over what domains the various models are valid, and how the domain and any fixed model parameters influence the estimate of  $p_c$ . There is a trade off between fitting to as much data over as great a domain as possible, so as to reduce statistical sampling fluctuations and hence to

refine the result, and fitting to only data from large lattices where finite size effects are smaller and the scaling theories better describe the data. The results in Table I are consistent, where they overlap at  $L = 128$  and  $L = 256$ , with those of Ziff and Newman [34]. Hence their data was used to extend the domain down to  $L = 8$  as necessary.

The fixed point renormalization group estimator is possessed of good quality data, but has a slow rate of convergence to its limit  $p_r^*$ . The  $p_r(L)$  data of Table III can be reasonably well fit with the addition of an  $L^{-3/\nu}$  term to the model, however the coefficient of  $L^{-4/\nu}$ , in an even higher-order model, is not zero. Values of the coefficients fluctuate with the order of the model, suggesting that even higher-order terms remain significant. Empirical power-law fits are consistent with the leading order exponent being  $-1/\nu$ , however a purely first-order model does not fit the data well until the domain is truncated to  $L \geq 256$ . Results for  $p_r^*$  are sensitive to the presence or absence of individual data points, the  $L = 4096$  point altering the result by  $\pm 1 \times 10^{-6}$ . Under different models and data ranges, threshold estimates range from 0.592744 to 0.592746. The difference is much larger than the uncertainty in the individual estimates and so not much weight should be given to those. Consequently, although the raw data at a given  $L$  is relatively precise, the slow rate of convergence of the fixed point renormalization group estimator leads to only a very rough figure of  $p_r^* = 0.592745(1)$ .

The linear combination estimator suffers from relatively large statistical uncertainties in the data, and points are not entirely independent of one another. However, the estimator does claim a very rapid rate of convergence to its limit,  $p_h^*$ . The model of Eq. (9) fits the data well for  $L \geq 32$ . Results thus obtained range from  $p_h^* = 0.59274594(5)$  to  $p_h^* = 0.59274603(8)$ , with the presence or absence of individual data points making differences of as much as  $\pm 4 \times 10^{-8}$  in  $p_h^*$ . Allowing for alternative values of the parameter  $\omega$ , between 0.85 and 0.95, the estimate changes by no more than  $\pm 1 \times 10^{-8}$ . The data is not precise enough to either support or falsify the assumed scaling relation, and is not inconsistent with  $p_h(L) = \text{constant}$ . Even so, all estimates for  $p_h^*$  were consistent with one another and with the  $128 \leq L \leq 4096 p_h(L)$  data mean of 0.5927460(1). Hence the linear combination estimator appears to be robust, and the mean value, which covers the entire range of results, should be a more than safe estimate for  $p_h^*$ . The value of  $p_h^* = 0.59274596(7)$ , obtained from all the  $p_h(L)$  data of this study, should be reliable.

With similarly low data quality, nonindependent points, and a slower rate of convergence, the cell-to-cell renormalization group estimator should not be expected to provide any refinement in  $p_c$  over the linear combination approach. Over domains where the various models fit the data, cell-to-cell results for  $p_{cc}^*$  range from 0.5927458(2) to 0.5927461(2). Sensitivity to the presence or absence of individual data points is as for the linear combination results, but here this is much smaller than statistical uncertainties. The estimate  $p_{cc}^* = 0.5927459(2)$ , obtained earlier from fitting the first-order scaling model to all the  $p_{cc}$  data of Table III, is in agreement with the entire range of cell-to-cell results above, and so is robust, if imprecise.



TABLE IV. Infinite lattice limit estimates for the percolation threshold. Results are shown, by estimator, for the Mersenne twister only data (MT, MTH, DMT), and also for the combined generators data (MT, MTH, DMT, SWB, QTA, QTB, XG).

Limit	Mersenne	Combined
$p_r^*$	0.592 745(1)	0.592 745(1)
$p_{cc}^*$	0.592 745 9(2)	0.592 745 9(2)
$p_h^*$	0.592 745 96(7)	0.592 745 98(6)
$p_m^*$	0.592 745 96(4)	0.592 745 98(3)

The median- $p$  based estimates have the same rate of convergence as the cell-to-cell estimates, but with independent data points of much higher quality. The median- $p$  estimates are less sensitive to the presence or absence of any one particular data point, this making a difference of at most  $2 \times 10^{-8}$ , and typically of less than  $1 \times 10^{-8}$ , in the result for  $p_m^*$ . The first-order model fits the data for  $L \geq 128$ , with results lying in the range  $p_m^* = 0.592 745 94(3)$  to  $p_m^* = 0.592 745 96(4)$ . The second-order model fits the data for  $L \geq 16$ , with results lying between  $p_m^* = 0.592 745 91(8)$  and  $p_m^* = 0.592 746 00(5)$ . The empirical power-law model makes a good fit for  $L \geq 64$ , with estimates of  $p_m^*$  running from 0.592 745 89(2) up to 0.592 746 03(2), and scaling exponents in the range  $-1.729(7)$  to  $-1.79(2)$ . As noted in the preceding section, the first-order fit matches the data of Table III very well, the empirical fit agrees with the assumed exponent of  $-1-1/\nu$ , and coefficients of higher-order terms were insignificant. This indicates that the first-order model does indeed provide an accurate description for the finite-size scaling behavior of the median- $p$  estimator. The estimate thus obtained, of  $p_m^* = 0.592 745 96(3)$ , does not quite encompass the entire range of results above. Allowing for an extreme scenario, where even the model and scaling exponent may not be quite right, a more conservative figure of  $p_m^* = 0.592 745 96(4)$  does cover all of the above results. Hence this final value of the median- $p$  estimate for  $p_c$  should be quite dependable. Incidentally, a standard error of  $4 \times 10^{-8}$  in  $p_c$  is approximately what would be expected from the total amount of data sampled in this study [as listed in the  $p_m(L)$  column of Table III]. Parameter  $a$  of Eq. (7) shows much more sensitivity to the data domain and model than does  $p_m^*$ . The fitted value given in the preceding section was the most precise obtained. An overall result of  $a = 0.415(5)$  is more reasonable in light of the other estimates.

The four estimators have now produced equally many robust estimates for the two-dimensional square site percolation threshold  $p_c$ . As summarized in Table IV, these are  $p_r^* = 0.592 745(1)$ ,  $p_{cc}^* = 0.592 7459(2)$ ,  $p_h^* = 0.592 745 96(7)$ , and  $p_m^* = 0.592 745 96(4)$ , in good mutual agreement. Taking  $p_c = 0.592 745 96$ , and returning to the canonical spanning probability curves, a good match between the data of  $128 \leq L \leq 4096$  and the theory of  $R_L(p_c) \approx 0.5 + kL^{-1} + O(L^{-2})$  was had for  $k = 0.317(1)$ . No higher-order terms were seen, with the coefficient of  $L^{-2}$  being indistinguishable from zero. The value of  $k$  found here is a little lower than those of Ziff,  $k = 0.319(1)$  [33], and Newman and Ziff,  $k = 0.320(1)$  [29].

The difference in  $p_r(2048)$  resulting from using  $k = 0.317$ , as opposed to  $k = 0.320$ , in Eq. (2) is around  $6 \times 10^{-9}$ . This is much less than the statistical uncertainties in the results of Table I, upholding the claimed insensitivity of those estimates to  $k$ . Hence those results remain reasonable (PRNG biased) estimates for  $p_c$ , and the direct comparison with earlier  $p_c$  estimates is valid. The estimate  $a = 0.415(5)$  found above is consistent with the results of Ziff, Newman, Hovi, and Aharony [33,34,56] ( $a$  here equates to their ratio  $b_0/a_1$ ). Since  $k$  equates to  $b_0$ , it follows that  $a_1 = 0.76(1)$  from the data obtained within this exercise. This estimate is also consistent with those of previous works [33,34,56].

The above results are based on data acquired solely from the Mersenne twister, that generator having been determined as suitable for this problem. In Sec. IV, results obtained from the SWB, QTA, QTB, and XG generators were found to be consistent with results obtained from the Mersenne twister. Although those four generators were not tested to the same extent as Mersenne twister, there is no objective reason to discount them entirely. Incorporating the data obtained from these generators earlier leads to revised values of  $p_m(2048) = 0.592 745 32(4)$ ,  $p_r(2048) = 0.593 150 55(4)$ ,  $p_{cc}(2048) = 0.592 7476(1)$ ,  $p_{cc}(4096) = 0.592 7463(2)$ ,  $p_h(2048) = 0.592 7459(1)$ , and  $p_h(4096) = 0.592 7459(3)$  for the various estimators of Table III. Note that the majority of the data remains Mersenne twister based.

Use of these revised values does not alter either the fixed point limit,  $p_r^*$ , or the cell-to-cell limit,  $p_{cc}^*$ . The linear combination limit is raised to  $p_h^* = 0.592 745 98(6)$ , an adjustment of rather less than its statistical uncertainty.

A parametrized fit of Eq. (7) to the revised median- $p$  data yields  $p_m^* = 0.592 745 98(3)$ ,  $a = 0.415(7)$ , and  $b = 0.1(5)$ . As before, the coefficient of the higher-order term is indistinguishable from zero. A first-order fit [of Eq. (7) with  $b$  constrained to zero] produces  $p_m^* = 0.592 745 98(3)$ , and  $a = 0.414(1)$ . An empirical power-law fit of the form  $p_m(L) \sim p_m^* - aL^z$  finds  $p_m^* = 0.592 745 98(4)$ ,  $a = 0.41(2)$ , and  $z = -1.75(1)$ . The excellent agreement between this model and the experimental data is shown in Fig. 6. The fitted value of  $z$  is indistinguishable from the assumed scaling exponent of  $-1-1/\nu$  (with  $\nu = 4/3$ ). All fitted parameters are consistent across the three models. Note that if only the QTB and XG data were combined with that from the Mersenne twister (the SWB and QTA generators being suspect on general grounds [48,49], although no obvious problems were seen here), then the revised median- $p$  estimator would be  $p_m(2048) = 0.592 745 31(5)$ , leading to  $p_m^* = 0.592 745 98(3)$  from the first-order fit and  $p_m^* = 0.592 745 98(4)$  from the empirical fit. These two values are identical to those obtained with the inclusion of SWB and QTA based data.

Performing robustness checks as before, the first-order model fits the data for  $L \geq 128$ , with results lying within a worst case range of  $p_m^* = 0.592 745 98(6)$  to  $p_m^* = 0.592 745 99(8)$ , and much more typically within  $p_m^* = 0.592 745 98(2)$  to  $p_m^* = 0.592 745 99(4)$ . The second-order model fits the data for  $L \geq 16$ , with results lying between  $p_m^* = 0.592 745 96(3)$  and  $p_m^* = 0.592 746 02(4)$ . The empirical power-law model makes a good fit for  $L \geq 128$ , with estimates of  $p_m^*$  running from 0.592 745 98(4) up to

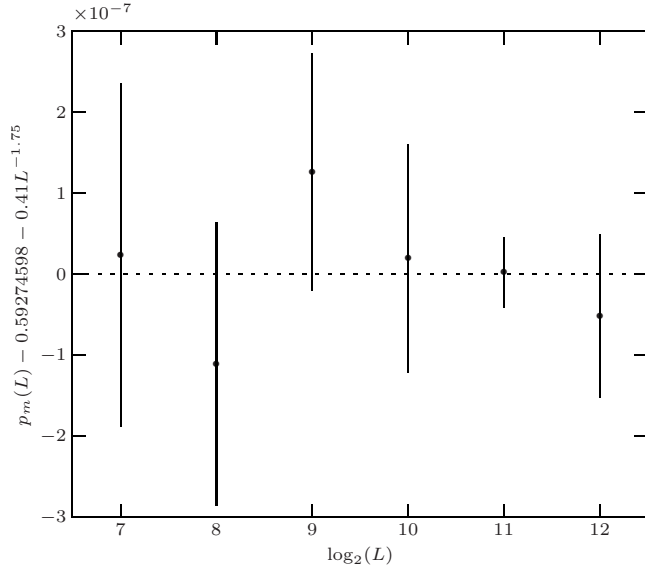


FIG. 6. Parametrized fit of empirical power-law model  $p_m(L) = p_m^* - aL^z$  to combined experimental data from the Mersenne twister, subtract with borrow, generalized xorshift, and both quad-tap generators.

0.592 745 99(8), and scaling exponents,  $z$ , in the range  $-1.74(1)$  to  $-1.77(1)$ . Hence, the data supports the validity of the first-order model with the assumed scaling exponent, and a standard error of  $3 \times 10^{-8}$  in  $p_m^*$  appears fully justified. The various threshold estimates are summarized in Table IV. Using the revised data, and  $p_c = 0.592 745 98$ , the estimate of the finite size correction parameter remains unchanged at  $k = 0.317(1)$ . Nor is any significant change seen in parameter  $a$ .

Assuming the suitability of the Mersenne twister PRNG for this particular Monte Carlo application, and also assuming that the median- $p$  estimator approaches the critical point as  $p_m(L) - p_c \propto L^{-1-1/\nu}$ , where  $\nu = 4/3$ , as supported by the data, then a robust estimate for the square site percolation threshold is  $p_c = 0.592 745 96(4)$ . A value for  $\omega$  is not required. Further assuming the suitability of the generalized xorshift and QTB generators, the additional data adjusts this estimate to  $p_c = 0.592 745 98(3)$  (this value does not change if the subtract with borrow and QTA generators are additionally assumed to be suitable). Continuing to assume the reliability of those generators, while dropping the assumed scaling exponent and requiring only that  $p_m(L) - p_c \propto L^z$ , for some  $z$ , the estimate becomes  $p_c = 0.592 745 98(4)$ . These three estimates are mutually consistent to well within statistical uncertainties. The most precise of them has a standard error of  $3 \times 10^{-8}$ , however a degree of caution is warranted in that none of the generators were tested to that level of precision. That being the case, this study's final estimate for the square site percolation threshold is

$$p_c = 0.592 745 98(4). \quad (11)$$

This, primarily Mersenne twister based, estimate is consistent with almost all previous results in Table II. In particular, it is in good agreement with the Mersenne twister derived

estimate of Lee. Taken collectively however, those results, excluding that of Lee, would suggest a higher value for  $p_c$ , in the vicinity of 0.592 7463(1). Although the value obtained here lies well outside of that range, the difference could be attributable to the various pseudo-random-number generators used. While it is not impossible that the result obtained here may reflect some detectable influence of the chosen generators, precautions against this were taken and no evidence of bias was found.

## VII. CONCLUSIONS

Increasing availability of highly parallel computer facilities now makes it practical to obtain significant quantities of Monte Carlo data from large lattices. This allows for greater precision in derived statistics, but requires very good quality pseudo-random-number generators as it is well established that inadequate generators lead to erroneous results.

Tests were performed upon several generators and it was found that use of simple two-tap generators should probably be avoided for this application. The MT19937 generator appeared to be suitable and was adopted for the majority of the Monte Carlo sampling conducted within this study. No dependence was found upon the (reasonable) choice of generator initialization.

Percolation threshold estimates subsequently made from various crossing probability statistics were found to be in good mutual agreement. The most precise of these was obtained from the median- $p$  estimator. Data quality was such that precise results could be obtained without the need to assume a particular scaling exponent. Even so, results were in good agreement with a leading exponent of  $-1-1/\nu$  and no higher-order term was found. The square site percolation threshold was subsequently determined to be  $p_c = 0.592 745 98(4)$ .

This estimate is consistent with the majority of earlier results on an individual basis, but not with those same results combined. Evidence suggests, however, that at least some of those earlier results have been influenced by the pseudo-random-number generators used. The generators used here appear to be of adequate quality, and the main generator, MT19937, passed an application specific test of randomness. Furthermore, efforts were made to ensure the reliability of the error bounds in that final estimate, which should, then, be accurate.

## ACKNOWLEDGMENTS

The author would like to thank R. M. Ziff for raising the need to address the percolation problem with different generators, for helpful advice and discussions, and for comments on the paper. Thanks are also extended to the referees for their comments, to H. W. J. Blöte, R. P. Brent, and P. H. L. Martins for providing details on their respective pseudo-random-number generators, and to A. J. E. Dale, G. L. Evans, and C. J. McMurtrie for assistance with the University of Canterbury's IBM Blue Gene supercomputing facility, upon which the Monte Carlo sampling was conducted.

- [1] D. Stauffer and A. Aharony, *Introduction to Percolation Theory*, 2nd ed. (Taylor and Francis, London, 1994).
- [2] B. Bollobás and O. Riordan, *Percolation* (Cambridge University Press, Cambridge, 2006).
- [3] M. Sahimi, *Applications of Percolation Theory* (Taylor and Francis, New York, 1994).
- [4] T. Harris, Proc. Cambridge Philos. Soc. **56**, 13 (1960).
- [5] H. Kesten, Commun. Math. Phys. **74**, 41 (1980).
- [6] M. Sykes and J. Essam, Phys. Rev. Lett. **10**, 3 (1963).
- [7] M. Sykes and J. Essam, J. Math. Phys. **5**, 1117 (1964).
- [8] J. C. Wierman, J. Phys. A **17**, 1525 (1984).
- [9] R. M. Ziff and C. R. Scullard, J. Phys. A **39**, 15083 (2006).
- [10] D. E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 2nd ed. (Addison-Wesley, Reading, MA, 1981).
- [11] D. Lehmer, *Proceedings of the 2nd Symposium Large-Scale Digital Calculating Machines* (Harvard University Press, Cambridge, MA, 1951), pp. 141–146.
- [12] G. Marsaglia, in *Applications of Number Theory to Numerical Analysis*, edited by S. K. Zaremba (Academic, New York, 1972), pp. 249–285.
- [13] R. C. Tausworthe, Math. Comput. **19**, 201 (1965).
- [14] T. G. Lewis and W. H. Payne, J. ACM **20**, 456 (1973).
- [15] S. Kirkpatrick and E. P. Stoll, J. Comput. Phys. **40**, 517 (1981).
- [16] G. Marsaglia and A. Zaman, Ann. Appl. Probab. **1**, 462 (1991).
- [17] R. M. Ziff, Comput. Phys. **12**, 385 (1998).
- [18] M. Matsumoto and T. Nishimura, ACM Trans. Model. Comput. Simul. **8**, 3 (1998).
- [19] P. L'Ecuyer, Math. Comput. **68**, 249 (1999).
- [20] G. Marsaglia, Random numbers for C: End, at last, Sci. Stat. Math Web discussion, 1999.
- [21] G. Marsaglia, J. Stat. Software **8**, 1 (2003).
- [22] W. W. Tsang, L. C. K. Hui, K. P. Chow, C. F. Chong, and C. W. Tso, Proceedings of the 27th Australian Conference on Computer Science, 2004, Vol. 26, p. 23.
- [23] R. P. Brent, ANZIAM J. **48**, C188 (2007).
- [24] P. L'Ecuyer and R. Simard, ACM Trans. Math. Softw. **33**, 22 (2007).
- [25] A. Kolmogorov, Russ. Math. Surveys **38**, 29 (1983).
- [26] G. Chaitin, *Information, Randomness and Incompleteness* (World Scientific, Singapore, 1987).
- [27] A. Compagner and A. Hoogland, J. Comput. Phys. **71**, 391 (1987).
- [28] M. E. J. Newman and R. M. Ziff, Phys. Rev. Lett. **85**, 4104 (2000).
- [29] M. E. J. Newman and R. M. Ziff, Phys. Rev. E **64**, 016706 (2001).
- [30] Y. Deng and H. W. J. Blöte, Phys. Rev. E **72**, 016126 (2005).
- [31] H. W. J. Blöte (private communication).
- [32] R. M. Ziff and G. Stell, Laboratory for Scientific Computing, University of Michigan. Report No., 88-4, 1988, footnote 26.
- [33] R. M. Ziff, Phys. Rev. Lett. **69**, 2670 (1992).
- [34] R. M. Ziff and M. E. J. Newman, Phys. Rev. E **66**, 016129 (2002).
- [35] P. Balister, B. Bollobás, and M. Walters, Random Struct. Algorithms **26**, 392 (2005).
- [36] O. Riordan and M. Walters, Phys. Rev. E **76**, 011110 (2007).
- [37] M. J. Lee, Phys. Rev. E **76**, 027702 (2007).
- [38] A. Compagner, Phys. Rev. E **52**, 5634 (1995).
- [39] J. Hoshen and R. Kopelman, Phys. Rev. B **14**, 3438 (1976).
- [40] R. M. Ziff (private communication).
- [41] R. M. Ziff and B. Sapoval, J. Phys. A **19**, L1169 (1986).
- [42] F. Yonezawa, S. Sakamoto, and M. Hori, Phys. Rev. B **40**, 636 (1989).
- [43] C.-K. Hu, Chin. J. Phys. (Taipei) **32**, 519 (1994).
- [44] C.-K. Hu, Phys. Rev. B **51**, 3922 (1995).
- [45] C.-K. Hu, C.-N. Chen, and F. Y. Wu, J. Stat. Phys. **82**, 1199 (1996).
- [46] P. H. L. Martins and J. A. Plascak, Phys. Rev. E **67**, 046119 (2003).
- [47] P. H. L. Martins (private communication).
- [48] M. Matsumoto and Y. Kurita, ACM Trans. Model. Comput. Simul. **6**, 99 (1996).
- [49] S. Tezuka, P. L'Ecuyer, and R. Couture, ACM Trans. Model. Comput. Simul. **3**, 315 (1993).
- [50] F. Panneton, P. L'Ecuyer, and M. Matsumoto, ACM Trans. Math. Softw. **32**, 1 (2006).
- [51] A. M. Ferrenberg, D. P. Landau, and Y. J. Wong, Phys. Rev. Lett. **69**, 3382 (1992).
- [52] P. Reynolds, H. Stanley, and W. Klein, J. Phys. A **11**, L199 (1978).
- [53] P. J. Reynolds, H. E. Stanley, and W. Klein, Phys. Rev. B **21**, 1223 (1980).
- [54] P. J. Reynolds, W. Klein, and H. E. Stanley, J. Phys. C **10**, L167 (1977).
- [55] A. Aharony and J.-P. Hovi, Phys. Rev. Lett. **72**, 1941 (1994).
- [56] J.-P. Hovi and A. Aharony, Phys. Rev. E **53**, 235 (1996).