

This is the accepted manuscript made available via CHORUS. The article has been published as:

Percolation critical polynomial as a graph invariant

Christian R. Scullard

Phys. Rev. E **86**, 041131 — Published 18 October 2012

DOI: [10.1103/PhysRevE.86.041131](https://doi.org/10.1103/PhysRevE.86.041131)

The percolation critical polynomial as a graph invariant

Christian R. Scullard*

Lawrence Livermore National Laboratory, Livermore CA 94550, USA

Every lattice for which the bond percolation critical probability can be found exactly possesses a critical polynomial, with the root in $[0, 1]$ providing the threshold. Recent work has demonstrated that this polynomial may be generalized through a definition that can be applied on any periodic lattice. The polynomial depends on the lattice and on its decomposition into identical finite subgraphs, but once these are specified, the polynomial is essentially unique. On lattices for which the exact percolation threshold is unknown, the polynomials provide approximations for the critical probability with the estimates appearing to converge to the exact answer with increasing subgraph size. In this paper, I show how this generalized critical polynomial can be viewed as a graph invariant, similar to the Tutte polynomial. In particular, the critical polynomial is computed on a finite graph and may be found using the recursive deletion-contraction algorithm. This allows calculation on a computer, and I present such results for the kagome lattice using subgraphs of up to 36 bonds. For one of these, I find the prediction $p_c = 0.52440572\dots$, which differs from the numerical value, $p_c = 0.52440503(5)$, by only 6.9×10^{-7} .

INTRODUCTION

Percolation is the study of the formation of random clusters on lattices. Given an infinite lattice, we declare each bond to be open with probability p , and closed with probability $1 - p$. The resulting random clusters grow in average size with p until we reach the critical point, p_c , above which an infinite cluster appears. The determination of p_c is an unsolved problem except in one dimension and on two-dimensional lattices that are self-dual 3-uniform hypergraphs [1]. An example is shown in Figure 1a, where the shaded triangle, excluding corner vertices, can represent any configuration of bonds, sites and correlations. Critical thresholds on these lattices are given by the Ziff criterion [2],

$$P(\bar{A}, \bar{B}, \bar{C}) - P(A, B, C) = 0 \quad (1)$$

where $P(A, B, C)$ is the probability that all three vertices are connected through open paths within the triangle, and $P(\bar{A}, \bar{B}, \bar{C})$ is the probability that none are connected. Application of (1) to find a bond threshold results in a polynomial in p , with order at most equal to the number of bonds in the unit triangle. We may consider each bond of an n -bond triangle to have a different probability, giving a critical surface of the generic form

$$f(p_1, \dots, p_n) = 0 \quad (2)$$

where f is at most first order in any of its arguments, a property referred to as multi-linearity. Examples of such critical surfaces include the square lattice,

$$S(p_1, p_2) = 1 - p_1 - p_2, \quad (3)$$

the honeycomb lattice,

$$H(p_1, p_2, p_3) = 1 - p_1 p_2 - p_1 p_3 - p_2 p_3 + p_1 p_2 p_3. \quad (4)$$

and the triangular lattice,

$$T(p_1, p_2, p_3) = 1 - p_1 - p_2 - p_3 + p_1 p_2 p_3. \quad (5)$$

The critical probability for the lattice is then found by setting all probabilities equal. For the square lattice, for example, this gives the polynomial $S(p, p) = 1 - 2p = 0$, and the threshold $p_c = 1/2$.

In this paper, I will show how such critical surfaces may be generalized to any lattice, even those which are not in the solvable class, by employing a deletion-contraction algorithm. Such surfaces may not be exact, but we will see that they give approximations that, in principle, can be made arbitrarily precise. I begin by describing the deletion-contraction property of critical surfaces derived from (1). Then I show how this can be used to define the critical surfaces and polynomials for unsolved problems. Although it appears that the definition may not result in a unique critical surface in all cases, I present an argument to show that it is in fact always well-defined. I conclude by reporting generalized critical polynomials for the kagome lattice, giving approximations to the threshold that become increasingly more precise, with the best estimate within only 6.9×10^{-7} of the numerically determined value. Although I use the kagome lattice as an illustrative example, this procedure can be applied on any periodic lattice.

DELETION-CONTRACTION

Consider the martini-A lattice, with the probability assignments shown in Figure 1b. If we set $p_1 = 1$, this bond is contracted and its end vertices merged, the result being the honeycomb lattice with two bonds doubled in parallel. This doubled bond can be replaced by a single effective bond, so we have

$$H(1 - [1 - p_2][1 - p_3], p_4, p_5). \quad (6)$$

By setting $p_1 = 0$, we delete the bond, and the result is the square lattice with bonds doubled in series. Thus, we

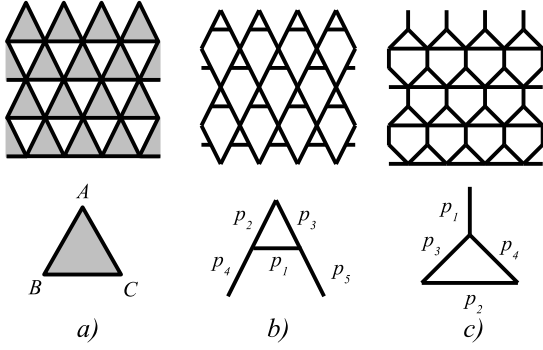


FIG. 1. a) A 3-uniform hypergraph. The shaded region need not be a simple triangle; b) the martini-A lattice; c) the martini-B lattice.

have,

$$S(p_2 p_4, p_3 p_5). \quad (7)$$

The critical surface of the martini-A lattice is given by (1). The only way to reduce to the correct deleted and contracted surfaces and preserve the required multilinearity property is to set

$$\begin{aligned} A(p_1, p_2, p_3, p_4, p_5) = & \\ & p_1 H(1 - [1 - p_2][1 - p_3], p_4, p_5) \\ & + (1 - p_1)S(p_2 p_4, p_3 p_5), \end{aligned} \quad (8)$$

which has the formal appearance of an average of the two special cases. It is a basic property of critical surfaces found using (1) that they satisfy such deletion-contraction formulas. That is, if such a lattice, L , has a triangular unit of n bonds such that deletion and contraction of the p_1 bond results in the lattices L_0 and L_1 , then we have for the critical surface of L ,

$$\begin{aligned} L(p_1, p_2, \dots, p_n) = & p_1 L_1(p_2, p_3, \dots, p_n) \\ & + (1 - p_1)L_0(p_2, p_3, \dots, p_n). \end{aligned} \quad (9)$$

As another example, we find for the martini-B lattice (Figure 1c),

$$\begin{aligned} B(p_1, p_2, p_3, p_4) = & p_4 S(p_1, 1 - [1 - p_2][1 - p_3]) \\ & + (1 - p_4)S(p_1 p_3, p_2). \end{aligned} \quad (10)$$

GENERALIZED CRITICAL SURFACES AND POLYNOMIALS

The deletion-contraction formula can be used to extend the definition of the critical surface to lattices for which the threshold is not known exactly, and in these cases it will be referred to as the *generalized* critical surface. For the kagome lattice of Figure 2a, an unsolved problem, we may write,

$$K(p_1, p_2, p_3, p_4, p_5, p_6) =$$

$$\begin{aligned} & p_4 B(1 - [1 - p_5][1 - p_6], p_1, p_2, p_3) \\ & + (1 - p_4)A(p_1, p_2, p_3, p_5, p_6), \end{aligned} \quad (11)$$

where A and B are given by (8) and (10). Generalized critical polynomials are found by setting all probabilities equal, and in this case $K(p, p, p, p, p, p) = 0$, i.e.,

$$1 - 3p^2 - 6p^3 + 12p^4 - 6p^5 + p^6 = 0 \quad (12)$$

which gives Wu's well-known [3, 4] approximation $p_c = 0.52442971\dots$, compared to the numerical [5] $p_c = 0.52440502(5)$. Thus, a generalized critical polynomial may be defined on any lattice for which known lattices appear upon deletion and contraction of a bond, but there is no guarantee that the resulting prediction will be exact. In fact, the definition extends to any periodic lattice. If L is in the solvable class, the polynomial is given by (1), and otherwise it is defined by (9), with this formula applied recursively if either L_0 or L_1 have unknown threshold. By repeated application, known lattices eventually appear and the full critical surface can be found. That it satisfies such a deletion-contraction formula makes the generalized critical surface similar to other graph invariants, such as the Tutte polynomial and its special cases (e.g., the chromatic polynomial) [6].

In the following, I call the subgraph of a lattice, on which different probabilities are assigned, the “base” of the process, which is then tiled to form the infinite lattice. For example, in Figures 2a, 2b, 3 and 4 are bases for the kagome lattice consisting of one, two, four, and six unit cells. I now enumerate some properties of generalized critical surfaces and polynomials:

1. They are unique once the base and tiling are specified [7, 8]. In particular, critical surfaces are independent of the bonds chosen to delete and contract at each step. To prove this, we proceed by induction. First, we call critical surfaces for which the final result is independent of deletion-contraction bond order *well-defined*. Clearly, any self-dual 3-uniform hypergraph has this property, as the polynomial is defined by (1); deletion-contraction must give this answer, regardless of the bond chosen, because it only imposes the correct boundary values on the critical surface. Similarly, any critical surface that is a special case of a well-defined surface is itself well-defined. Next, we show that the critical surface for a lattice, L , that is not a realization of a self-dual 3-uniform hypergraph, and is thus not solved by (1), is well-defined if every critical surface resulting from deleting and contracting any bond of L is well-defined. The kagome lattice with the 6-bond base is one example of such a system. Assuming that L has this property, consider the generic formula (9). The question is whether this is the same as what we would obtain by choosing the p_2 bond instead,

$$\tilde{L}(p_1, p_2, \dots, p_n) = p_2 \tilde{L}_1(p_1, p_3, \dots, p_n)$$

$$+ (1 - p_2)\tilde{L}_0(p_1, p_3, \dots, p_n), \quad (13)$$

i.e., whether $L = \tilde{L}$. We can apply deletion-contraction on p_2 to the graphs L_0, L_1 , and on p_1 to \tilde{L}_0, \tilde{L}_1 to obtain L_{00}, L_{01} , etc. We then have,

$$\begin{aligned} L &= p_1 p_2 L_{11}(p_3, \dots, p_n) \\ &+ p_1 (1 - p_2) L_{10}(p_3, \dots, p_n) \\ &+ (1 - p_1) p_2 L_{01}(p_3, \dots, p_n) \\ &+ (1 - p_1)(1 - p_2) L_{00}(p_3, \dots, p_n) \end{aligned} \quad (14)$$

and

$$\begin{aligned} \tilde{L} &= p_1 p_2 \tilde{L}_{11}(p_3, \dots, p_n) \\ &+ p_1 (1 - p_2) \tilde{L}_{10}(p_3, \dots, p_n) \\ &+ (1 - p_1) p_2 \tilde{L}_{01}(p_3, \dots, p_n) \\ &+ (1 - p_1)(1 - p_2) \tilde{L}_{00}(p_3, \dots, p_n), \end{aligned} \quad (15)$$

and clearly these two expressions are the same if $L_{11} = \tilde{L}_{11}$, $L_{10} = \tilde{L}_{10}$, etc. By assumption, L_1, L_0, \tilde{L}_1 and \tilde{L}_0 are well-defined. Now, the function \tilde{L}_{11} arises by contracting the bonds p_2 and p_1 in L , which is the same way we arrive at L_{11} . Thus, these two functions must describe the same lattice with a well-defined critical surface, and so we have $\tilde{L}_{11} = L_{11}$. By the same reasoning, $L_{10} = \tilde{L}_{10}$, etc., and we conclude that if every critical surface arising from deleting and contracting any bond of L is well-defined, then so is L . Because each step of the algorithm lowers the number of bonds in the base by (at least) one, well-defined surfaces eventually appear and the general result follows by induction. A slight complication arises due to the existence of certain seemingly pathological cases. Consider, for example, setting $p_1 = 0$ on the square lattice. Now we have created a one-dimensional system, with the trivial critical point $p_2 = 1$, or $1 - p_2 = 0$. This latter form is the correct surface to use whenever the one-dimensional case appears, and is indeed reflected in the formula (3). A less straightforward example appears on the square lattice when we set $p_1 = 1$ in (3). It is not obvious how to interpret the result of contracting this bond since it seems to collapse the lattice. However, in such cases it is better to think of the bond as not contracting to zero length but simply carrying probability 1 and thus introducing infinitely long linear clusters on the lattice. In the square case, the system is supercritical unless we also set $p_2 = 0$, which gives a critical one-dimensional system. The supercritical phase of (3) is represented by $S(p_1, p_2) < 0$, so by the above considerations we would expect the surface for $p_1 = 1$ to read $-p_2 = 0$, which is indeed what we find from (3). Even if the lattice is unknown, by keeping these “supporting” bonds uncontracted when they are used in the algorithm, we can continue until we reach cases on which we may use (1),

and then the supporting bond is just given probability 1 in the formula. Finally, consider the result of setting $p_1 = p_2 = 1$ in the triangular formula (5), thus imposing an infinite two-dimensional cluster. This system is super-critical (there is no infinite cluster at the critical point) regardless of the value of p_3 , and thus we have $T(1, 1, p_3) = -1$, which is a general rule for whenever such a situation arises. This completes the discussion of potentially problematic cases, and thus the proof that the critical surface for a general lattice and base is well-defined. When calculating the polynomials reported below, each one was computed several times using different bond paths to ensure that the code was working properly. For the largest bases of 36 bonds, each calculation was repeated over a dozen times and the same polynomial always resulted.

2. If the single-cell prediction is not exact, it is generally very close, usually within 10^{-5} of the numerically determined threshold [9]. It is not clear why this should be so, but it has been clearly demonstrated for many different systems, including all the Archimedean lattices [7, 9]. In addition, proper choices of larger bases lead to predictions closer to the exact answer, with accuracy increasing with base size. It is this latter conjecture I seek to support here by computing polynomials on the kagome lattice for bases of up to 36 bonds.
3. Conversely, if the critical polynomial gives the exact threshold for a base of a single unit cell, as is the case for self-dual 3-uniform hypergraphs, then any critical polynomial using a larger base makes the same prediction, i.e. the original polynomial always factors out. This is also a conjectured property but it would seem to be necessary for consistency, and I have found no counter-examples. While it would be ideal to have a lattice for which the exact threshold is known and is not a root of a polynomial, so that we can directly check the convergence of these polynomials, no such example is known at present and the conjecture can only be tested against numerically determined values.
4. In many cases, if the single-cell polynomial does not give the exact answer, then it can be shown [7] that no critical polynomial derived from a finite-sized base will give the correct percolation threshold. This is true for the kagome lattice, as discussed below.

Generalized critical polynomials have been found for all the Archimedean lattices [7, 9, 10]. The polynomial (12) was originally found by Wu using his “homogeneity” [3] assumption and recently, he extended the method to include kagome subnets [8], with excellent results [11]. In

fact, the homogeneity approximation gives a prediction for the full q -state Potts critical frontier, but at present it appears limited to the kagome lattice and its subnets as it relies on a transformation from the triangular lattice to these kagome-type graphs. Although I focus on the kagome lattice in this paper, mostly because of the interest it has attracted over the years (e.g., [3, 12, 13]), the method presented here may be applied on any periodic lattice for either site or bond percolation.

KAGOME LATTICE POLYNOMIALS

In previous work, critical polynomials were found “by hand”, with the (4, 6, 12) lattice and its 18-bond unit cell probably representing the limit of what one would be inclined to do this way (see the appendix of [7]). However, the recursive nature of the algorithm makes it an ideal problem for a computer, and here I report the results of using a program to calculate critical polynomials on the kagome lattice for bases containing 12, 24, and 36 bonds. At each step, the program chooses a bond, and finds the two graphs that result from its deletion and contraction. It knows a small number of graphs (e.g., triangular, honeycomb, and square lattices), and it repeats the deletion-contraction algorithm recursively until it recognizes all the lattices it has found. The output is a set of function definitions, like equations (8), (10) and (11) which can be evaluated in a computer algebra package to get the critical surface and then the critical polynomial. Of course, there are many issues to overcome in programming this scheme, and a full account is given elsewhere [14].

Base of 2 unit cells

The first extension beyond a single unit cell base is that shown in Figure 2b, in which we employ a base using two unit cells which are indicated by different colors, with the tiling shown in Figure 2c. The polynomial can be written in the factored form

$$-(1 - 3p^2 - 6p^3 + 12p^4 - 6p^5 + p^6) \times (-1 - p^2 - 2p^3 + 10p^4 - 10p^5 + 3p^6). \quad (16)$$

We recognize the first term in brackets as the polynomial for the 6-bond base. The second term has no real roots, and thus the prediction here is the same as the one we found previously, $p_c = 0.52442971\dots$. There is no other way to tile this base to give a different result.

Bases of 4 unit cells

Using the base consisting of 4 unit cells shown in Figure 3a, we may tile it in two different ways as indicated in

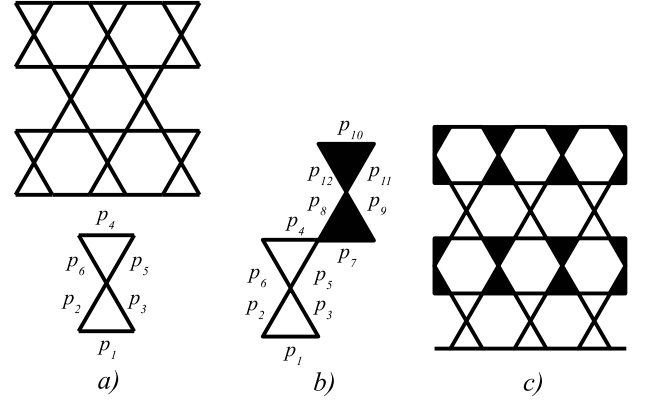


FIG. 2. a) The kagome lattice and its single unit cell 6-bond base; b) a 12-bond base for the kagome lattice; c) a tiling for the base in b).

Figures 3b and 3c. Starting with Figure 3b, the critical polynomial can be written in the factored form

$$-(1 - 3p^2 - 6p^3 + 12p^4 - 6p^5 + p^6) \times (-1 - p^2 - 2p^3 + 10p^4 - 10p^5 + 3p^6) \times (1 - 2p^2 - 4p^3 + 7p^4 + 24p^5 - 28p^6 - 64p^7 + 172p^8 - 184p^9 + 110p^{10} - 36p^{11} + 5p^{12}) \quad (17)$$

Once again, the first term in brackets is just the 6-bond polynomial (12), and the others have no roots in $[0, 1]$. The prediction is again the same as the 6-bond estimate.

Things finally become more interesting when we use the staggered embedding in Figure 3c. In this case, the polynomial is

$$1 - 6p^4 - 24p^5 - 24p^6 - 24p^7 + 27p^8 + 552p^9 + 1056p^{10} - 1224p^{11} - 8548p^{12} - 4872p^{13} + 68568p^{14} - 50664p^{15} - 226650p^{16} + 643944p^{17} - 843684p^{18} + 684384p^{19} - 368886p^{20} + 133152p^{21} - 31068p^{22} + 4248p^{23} - 259p^{24} = 0, \quad (18)$$

and the solution on $[0, 1]$ is $p_c = 0.52440672\dots$, which differs from the numerical result by 1.7×10^{-6} , a great improvement over the single-cell 6-bond case. Although finding all the bases that give different polynomials is something of an art, especially as the number of bonds becomes large, there does not appear to be one that gives a different prediction using 24 bonds.

Bases of 6 unit cells

There are many options for bases of 36 bonds, and I will discuss only three here. If we take the 6-unit cell version of the base shown in Figure 3a with the embedding analogous to Figure 3b, we once again find a polynomial in which (12) appears as a factor. Thus, for this base

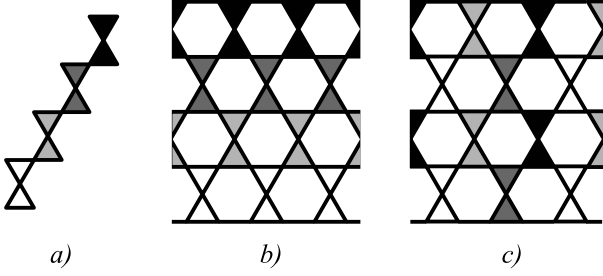


FIG. 3. A 24-bond base for the kagome lattice with two inequivalent embeddings.

and embedding, we get the 6-bond estimate again. It is tempting to assume that this trend continues for larger bases and embeddings of this type.

Turning now to the base shown in Figure 4a, in which the embedding is indicated by the matching shapes on the external vertices, we get the polynomial,

$$\begin{aligned}
 &1 - 3p^4 - 12p^5 - 20p^6 - 60p^7 - 132p^8 + 56p^9 \\
 &+ 684p^{10} + 1440p^{11} + 2108p^{12} + 2052p^{13} \\
 &- 10452p^{14} - 68708p^{15} - 82980p^{16} + 280152p^{17} \\
 &+ 1316026p^{18} - 49980p^{19} - 12878976p^{20} \\
 &+ 5124684p^{21} + 90816816p^{22} - 199458252p^{23} \\
 &- 12979085p^{24} + 816398808p^{25} - 1939348056p^{26} \\
 &+ 2677229528p^{27} - 2575935942p^{28} \\
 &+ 1832168220p^{29} - 984362272p^{30} \\
 &+ 400507236p^{31} - 121897767p^{32} + 26954680p^{33} \\
 &- 4096134p^{34} + 382956p^{35} - 16617p^{36} = 0, \quad (19)
 \end{aligned}$$

which has solution on $[0, 1]$ $p_c = 0.52440607\dots$, differing from the numerical value by 1.1×10^{-6} . A different base and embedding is shown in Figure 4b and has polynomial,

$$\begin{aligned}
 &1 - 6p^4 - 24p^5 - 14p^6 + 36p^7 + 39p^8 - 100p^9 \\
 &- 462p^{10} + 780p^{11} + 4583p^{12} + 4812p^{13} \\
 &- 9276p^{14} - 71600p^{15} - 85626p^{16} \\
 &+ 312336p^{17} + 1091146p^{18} - 509340p^{19} \\
 &- 9675936p^{20} + 5297340p^{21} + 66607704p^{22} \\
 &- 151097304p^{23} - 5319734p^{24} + 610494828p^{25} \\
 &- 1461237180p^{26} + 2022998000p^{27} \\
 &- 1949295060p^{28} + 1387593528p^{29} \\
 &- 745850356p^{30} + 303533928p^{31} \\
 &- 92388675p^{32} + 20427736p^{33} - 3103578p^{34} \\
 &+ 290052p^{35} - 12579p^{36} = 0, \quad (20)
 \end{aligned}$$

with solution in $[0, 1]$, $p_c = 0.52440572\dots$, slightly better than the previous estimate and within 6.9×10^{-7} of the numerical value.

DISCUSSION

Clearly, we are observing the predictions converging to the exact answer with increasing base size. The results

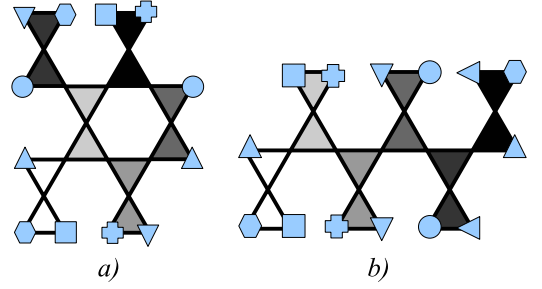


FIG. 4. 36-bond bases for the kagome lattices. Matching shapes on the external vertices indicate how each base is tiled to create the lattice.

suggest that, if a generic base can be described as containing $n \times m$ unit cells, the exact answer is approached only as both n and m go to ∞ since the $1 \times n$ bases appear all to provide the same incorrect prediction. From these few examples it also appears that the kagome estimates are converging from above, as they all seem to be greater than the numerical value. However, I know of no argument that guarantees this trend will continue. Also, as the polynomials respect duality, i.e., making the substitution $p \rightarrow 1 - p$ gives a polynomial for the dual graph, the estimates for the dual of the kagome lattice, the dice lattice, would converge from below. Note also that no polynomial for the kagome lattice derived from any finite-size base will give the exact answer. This can be seen as follows. Using the base in Figure 4a, we delete many of the bonds to leave a single unit cell and a few connecting bonds, as in Figure 5a. Now, we may contract the connecting bonds to recover the kagome system in which the base is a single cell, as in Figure 5b. By uniqueness, the prediction for this case must be equation (12), which contradicts the prediction found by setting all bonds equal in the full critical surface, i.e. equation (19). Thus, if the single-cell polynomial does not provide the exact threshold, and for the kagome lattice it does not, then, although we can get arbitrarily close by using ever larger bases, no finite critical polynomial derived in this way will ever solve the problem.

There are many possible directions for further study. Aside from more firmly establishing the various conjectures, it would be interesting to try to quantify the manner of convergence to the exact solution, perhaps through a version of finite-size scaling. Another avenue is the extension to three and higher dimensions. While the polynomials are well-defined by the deletion-contraction algorithm in any dimension, preliminary results seem to indicate that they are not as successful at predicting higher-D critical points. However, this will be the subject of future work.

I have presented critical polynomials for the kagome lattice, up to bases of 36 bonds. This is the limit of the current implementation of the algorithm, as it be-

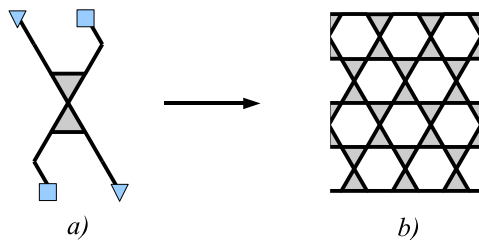


FIG. 5. a) the subgraph and embedding resulting from deleting 22 bonds in Figure 4a, giving a single cell of the kagome lattice plus some connecting bonds; b) contracting the connecting bonds gives the kagome lattice partitioned into unit cells.

comes progressively more difficult to add bonds due to the exponential complexity. It is not uncommon for a large calculation to produce over a million function definitions. Nevertheless, there is room for improvement in the efficiency, as the rule used to choose the bond for deletion and contraction at each step can have a significant effect on the rate of reduction to the known simple cases. I have hardly explored this issue and presently use what amounts to a random bond selection, rejecting only those choices that lead to undue complications (such as the contraction of supporting bonds described above). Moreover, this algorithm is perfect for a parallel implementation as it would require little inter-processor communication. It remains to be determined how much extra performance can be wrought from these considerations, but the problem will hopefully be seen as an interesting

computational challenge.

I thank Robert Ziff for helpful advice on this manuscript as well as the fruitful collaboration that led to this work, and I thank Jesper Jacobsen for informative discussions. I am also grateful to two anonymous referees who provided many useful suggestions. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

* scullard1@llnl.gov

- [1] B. Bollobás and O. Riordan, in *An Irregular Mind*, Bolyai Society Mathematical Studies, Vol. 21 (Springer Berlin Heidelberg, 2010) pp. 131–217, arXiv:1001.4674.
- [2] R. M. Ziff, Phys. Rev. E **73**, 016134 (2006).
- [3] F. Y. Wu, J. Phys. C **12** (1979).
- [4] C. R. Scullard and R. M. Ziff, Phys. Rev. E **73**, 045102(R) (2006).
- [5] X. Feng, Y. Deng, and H. W. J. Blöte, Phys. Rev. E **78**, 031136 (2008).
- [6] B. Bollobás, *Modern Graph Theory* (Springer, 1998).
- [7] C. R. Scullard, J. Stat. Mech. , P09022 (2011).
- [8] F. Y. Wu, Phys. Rev. E **81**, 061110 (2010).
- [9] C. R. Scullard and R. M. Ziff, J. Stat. Mech. , P03021 (2010).
- [10] C. R. Scullard and R. M. Ziff, Phys. Rev. Lett. , 185701 (2008).
- [11] C. Ding, Z. Fu, W. Guo, and F. Y. Wu, Phys. Rev. E **81**, 061111 (2010).
- [12] R. M. Ziff and H. Gu, Phys. Rev. E **79**, 020102 (2009).
- [13] C. Tsallis, J. Phys. C: Solid State Phys. **15**, L757 (1982).
- [14] C. R. Scullard, arXiv:1111.1061 (2012).