Bond-propagation algorithm for thermodynamic functions in general two-dimensional Ising models

Y. L. Loh and E. W. Carlson

Department of Physics, Purdue University, West Lafayette, Indiana 47907, USA

M. Y. J. Tan

Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

Recently, we developed and implemented the bond-propagation algorithm for calculating the partition function and correlation functions of random-bond Ising models in two dimensions [Y. L. Loh and E. W. Carlson, Phys. Rev. Lett. **97**, 227205 (2006)]. The algorithm is the fastest available for calculating these quantities near the percolation threshold. In this paper, we show how to extend the bond-propagation algorithm to directly calculate thermodynamic functions by applying the algorithm to derivatives of the partition function and we derive explicit expressions for this transformation. We also discuss variations of the original bond-propagation procedure within the larger context of $Y\nabla Y$ reducibility and discuss the relation of this class of algorithm to other algorithms developed for Ising systems. We conclude with a discussion on the outlook for applying similar algorithms to other models.

I. INTRODUCTION

For nearly 80 years, Ising models have given valuable insight into phase transitions and critical phenomena in magnets, alloys, and many other systems. Random-bond Ising models (RBIMs), in particular, are often used to study frustration and spin-glass behavior, and they are closely related to neural networks and information theory.[1] In Ref. 2, we presented a numerically exact[3] algorithm for computing the partition function and correlation functions in a class of two-dimensional (2D) Ising models, which works for any planar network of Ising spins with arbitrary bond strengths but without applied fields in the bulk. Applications include RBIMs (including $\pm K$ disorder, Gaussian disorder, site dilution, and bond dilution) and geometric frustration as in the case of triangular Ising antiferromagnets. Here, we show how to extend the algorithm to directly calculate thermodynamic quantities such as the internal energy.

Since its introduction in 1925, many exact results have been found for regular 2D Ising models (i.e., those with translational symmetry in two directions). There has been continued interest in random-bond Ising models, which have more complexity because of the need to average not only over all spin configurations but also over all configurations of bond strengths. Approximate analytic methods can give misleading or conflicting results, so numerical calculations play an important role in the field.

Numerical methods for 2D RBIMs have developed along several lines. Onsager's original solution using operator techniques generalizes to the fermion-network method of Merz and Chalker.[4] The Ising partition function is related to the number of dimer coverings (perfect matchings) of a modified graph and hence to the problem of computing a determinant or Pfaffian;[5,6] this approach was applied to RBIMs by Saul and Kardar[7,8] and by Galluccio *et al.*[9] The fastest of these algorithms[9] takes of order $O(N^{3/2})$ time for a network of N spins. Algorithms in this class have the disadvantage that one must first map to fermions or dimers before the model can be solved.

A different line of development involves the Y-∇ and ∇-Y transformations for the partition functions of Ising models,[10,11] operating directly in the spin basis, without the need to map to fermion or dimer models. In a Y-∇ (or star-triangle) transformation, three Ising spins connected by bonds to a central spin can be converted into the same three Ising spins but without the central spin, now connected by mutual bonds, in such a way as to preserve the total partition function of the system. The reverse ∇-Y transformation can also be done so as to preserve the partition function. For networks which are $Y\nabla Y$ reducible, one can then compute the partition function Z at a given temperature using a suitable sequence of such transformations. Colbourn *et al.*[12] suggested using the Feo-Provan[13] $Y\nabla Y$ reduction method for general networks, which takes $O(N^2)$ time, where N is the number of nodes in the network (i.e., spins). Frank and Lobb invented the bond-propagation algorithm[14] (a form of $Y\nabla Y$ reduction) for 2D resistor networks which takes $O(N^{3/2})$ time for square lattices (grid graphs).[15] They also suggested an extension to Ising models. Recently, we developed and implemented the Ising bond-propagation algorithm.[2] The algorithm executes in $O(N^{3/2})$ time for most planar networks of interest and in $N \ln N$ time for dilute models near percolation,[14] making it the fastest method for computing the Ising partition function and correlation functions near the percolation threshold. In this paper, we derive the transformations necessary to implement the algorithm on derivatives of the partition function, allowing for fast, direct, and exact calculation of thermodynamic quantities. We also discuss the relation of this class of algorithm to graph theory as well as to other methods for Ising systems.

The outline of this paper is as follows. In Sec. II, we introduce the models to which the algorithm is applicable. In Sec. III, we derive the network transformations, most notably the Y-∇ and ∇-Y transformations, that are the building

blocks of the bond-propagation algorithm (BPA) for Gaussian models (such as resistor networks) as well as Ising models, including new forms of the transformations which have improved numerical stability in the latter case. In Sec. IV, we describe some variations of the algorithm. In Sec. V, we show that the BPA can be used to compute derivatives of the Ising partition function directly, and we derive these transformations for the first derivative with respect to temperature. In Sec. VI, we discuss practical issues such as the strengths and limitations of the BPA, as well as the outlook for application of similar algorithms to other models.

II. MODELS

$Y \nabla Y$ reduction is applicable to both Gaussian models (such as resistor networks) and Ising models. We define a "Gaussian model" in statistical mechanics as one whose action $S(\{\phi\})$ is bilinear in the field variables ϕ, which may be real, complex, or Grassmannian,

$$e^{S(\{\phi\})} = \exp \sum_{ij} K_{ij} \phi_i^* \phi_j. \tag{1}$$

This action also applies to dynamical models that are diagonal in frequency space, such as electrical LCR networks involving inductors, capacitors, and/or resistors where the K_{ij} represent admittances (complex conductances), and noninteracting bosons, fermions, phonons, or magnons on lattices with hopping amplitudes K_{ij}. In the case of a resistor network, ϕ_i is the electrostatic potential at site i.

An Ising model action is a quadratic form in spin variables $\sigma_i = \pm 1$ with a symmetric kernel K_{ij},

$$e^{S(\{\sigma\})} = e^{-\beta \mathcal{H}(\{\sigma\})} = \exp \sum_{ij} K_{ij} \sigma_i \sigma_j, \tag{2}$$

where $K = \beta \widetilde{K}$ are dimensionless Ising couplings, $\widetilde{K}$ are the physical couplings with dimensions of energy, and $\beta = 1/T$ is the inverse temperature. This includes various types of RBIMs such as those with "binary" couplings ($K_{ij} = \pm K_0$), Gaussian-distributed couplings, and zero couplings (dilute Ising models). Such models are often used to study spin glasses and dilute magnets.

In the Ising system, our goal is to calculate the partition function,

$$Z = \sum_{\{\sigma\}} e^{S(\{\sigma\})}, \tag{3}$$

correlation functions such as

$$\langle \sigma_i \sigma_j \rangle = Z^{-1} \sum_{\{\sigma\}} \sigma_i \sigma_j e^{S(\{\sigma\})}, \tag{4}$$

and/or thermodynamic functions such as

$$U = -Z^{-1} \frac{dZ}{d\beta} = Z^{-1} \sum_{\{\sigma\}} \mathcal{H}(\{\sigma\}) e^{S(\{\sigma\})}. \tag{5}$$

For Gaussian models such as resistor networks, we aim instead to calculate the effective resistance R_{ij} between two points in the network.
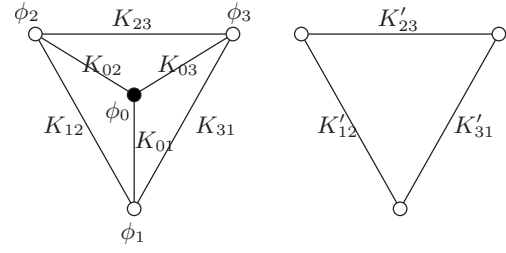

FIG. 1. Elimination of an N-degree node, illustrated for $N=3$. Diagonal couplings K_{11}, K_{22}, K_{33} correspond to loops (edges connecting a node to itself) and have been omitted.

Computing the partition function of a many-body system is usually a very difficult problem that involves summing over a number of configurations that is exponential in system size. Even if this is recast into the form of a high-temperature or low-temperature series, it is still usually the case that one needs to sum over a large number of graphs, or count a large number of graph embeddings, using some amount of brute force. Treatments based on successive elimination of spins are often limited because integrating out spin degrees of freedom tends to generate longer range couplings or multispin interactions such as three-spin couplings. Another approach is real-space decimation, which is approximate because one has to discard most interactions beyond a certain level of complexity.

There are some nice exceptions: for Gaussian models and Ising models on planar networks,[16] it is possible to reduce an entire network by successive Y-∇ and ∇-Y transformations, local moves which never induce multispin interactions. Thus, throughout the whole procedure, the system can be described just by pairwise interactions; therefore, it is useful to adopt the language of graph theory and represent it by a network, with nodes (vertices), edges (bonds), and edge weights (couplings). The next section describes transformations that can be used in the reduction of such a "statistical mechanics network."

III. STAR-MESH AND MESH-STAR TRANSFORMATIONS

A. Gaussian models

For pedagogical reasons, we first review Gaussian models (which include resistor networks as a special case).[17] We show how to eliminate any node of a Gaussian model via a star-mesh transformation regardless of its coordination number. Consider the network in Fig. 1 described by an action $S(\phi_0, \phi_1, \ldots, \phi_N)$ containing bilinear couplings $K_{ij} \phi_i^* \phi_j$ as well as a constant "free energy" term F. The variable ϕ_0 can be eliminated by Gaussian integration, leading to a new effective action $S'(\phi_1, \ldots, \phi_N)$ with parameters F' and K'_{ij}:

$$\int (d\phi_0) \exp S(\phi_0, \phi_1, \ldots, \phi_N)$$

$$= \int (d\phi_0) \exp \left(F + \sum_{i=0}^{N} \sum_{j=0}^{N} K_{ij} \phi_i^* \phi_j \right)$$

$$= \exp \left[F + \zeta \ln \frac{-1}{K_{00}} + \sum_{i=1}^{N} \sum_{j=1}^{N} \left(K_{ij} - \frac{K_{i0} K_{0j}}{K_{00}} \right) \phi_i^* \phi_j \right]$$

N	Star	Mesh	Transformation name
0	●	(empty)	Elimination of free spin
1	J01 ●—○	○	Elimination of dead end, pendant, or dangling bond
2	J01 J02 ○—●—○	J12 ○—○	Series reduction, decoration–iteration transformation
3	J01 ●J03 / J02 (Y shape)	J31 / J12 J23 (triangle)	Y–Delta, Y–nabla, wye–delta, star–triangle transformation, Yang–Baxter equations
4	(X / cross shape)	(square with diagonals)	Star–square, cross–tetrahedron transformation

FIG. 2. Star-mesh transformations for various N. Circles represent nodes and lines represent edges; the filled node is the one being eliminated.

$$= \exp\left(F' + \sum_{i=1}^{N} \sum_{j=1}^{N} K'_{ij} \phi_i^* \phi_j \right) = \exp S'(\phi_1, \ldots, \phi_N), \tag{6}$$

where $\int (d\phi) \equiv \int \frac{d(\Re\phi)d(\Im\phi)}{\pi}$ and $\zeta = 1$ in the case where ϕ are complex variables; similar expressions exist for real ϕ and Grassmannian ϕ. The integration generates additive changes to the free energy, $\delta F = F' - F$, and to the couplings, $\delta K_{ij} = K'_{ij} - K_{ij}$:

$$\delta F = \zeta \ln \frac{-1}{K_{00}}, \tag{7}$$

$$\delta K_{ij} = -\frac{K_{i0} K_{0j}}{K_{00}}, \quad i,j = 1,2,3,\ldots,N. \tag{8}$$

Equation (8) contains only algebraic operations—addition, multiplication, and division—and it is homogeneous, i.e., invariant under multiplication of all K's by the same constant. In fact, Eq. (8) corresponds to row and column subtractions on the matrix K_{ij}; the elimination of ϕ_0 by Gaussian integration is closely related to the procedure of Gauss elimination in linear algebra. The above derivation is valid even if the variables $\phi_1, \ldots, \phi_N$ are coupled to external fields or additional variables; these extra terms simply cancel out on both sides of Eq. (6).

Node elimination corresponds to the "star-mesh transformations" illustrated in Fig. 2 and defined as follows. An *N-degree star-mesh transformation* eliminates a node of coordination number N, introducing $N(N-1)/2$ edges between its neighbors. A mesh-star transformation is the inverse, which does not always exist.

For Gaussian models, the star-mesh formula is

$$K'_{ij} = -K_{i0} K_{0j}/K_{00}, \quad i,j = 1,2,3,\ldots,N. \tag{9}$$

In the special case of a resistor network, the off-diagonal elements of the kernel are equal to the conductances, $K_{ij} = G_{ij}$, and the diagonal elements are given by a sum rule $K_{ii} = -\Sigma_{j=1}^{N} G_{ij}$, so the star-mesh formula becomes $G'_{ij} = G_{i0} G_{0j}/\Sigma_{j=1}^{N} G_{0j}$. For $N=2$, this corresponds to the familiar series reduction of conductances. For $N=3$, it reduces to the Y-∇ transformation,

$$G'_{12} = G_{01} G_{02}/(G_{01} + G_{02} + G_{03}) \quad \text{(cycl.)}, \tag{10}$$

where "cycl." indicates additional equations in which the indices 1,2,3 are cyclically permuted. Equation (10) can be inverted to give the ∇-Y transformation, which is most easily written in terms of the resistances $R = 1/G$:

$$R_{01} = R'_{31} R'_{12}/(R'_{31} + R'_{12} + R'_{23}) \quad \text{(cycl.)}. \tag{11}$$

The equations for node-voltage analysis of a planar electrical network are the same as those for loop-current analysis of the dual network, which is obtained by interchanging node voltages V_i with loop currents I_μ, and conductances G_{ij} joining adjacent nodes with resistances $R_{\mu\nu}$ separating adjacent loops. Thus, the similarity between Eqs. (10) and (11) is no coincidence.

For computer implementation, it may be preferable or necessary to supplement the above formulas by formulas for special cases,[14] involving zero couplings (opens) or infinite couplings (shorts). Assuming that floating-point overflows never occur, the following rules are sufficient:

$$\text{YDelta}(\infty, G_1, G_2) = (0, G_2, G_1), \tag{12}$$

$$\text{DeltaY}(0, G_1, G_2) = (\infty, G_2, G_1), \tag{13}$$

$$\text{DeltaY}(\infty, G_1, G_2) = (G_1 + G_2, \infty, \infty). \tag{14}$$

The G_{ij} can be complex numbers representing admittances at a particular frequency, in which case the transformations involve complex arithmetic.

B. Ising models

For Ising networks, one may attempt to derive a star-mesh transformation in a similar way, integrating out a spin σ_0 to generate an effective action S' with parameters F' and K' to be determined (see Fig. 2):

$$\sum_{\sigma_0 = \pm 1} e^{S(\sigma_0, \sigma_1, \ldots, \sigma_N)} = e^{S'(\sigma_1, \ldots, \sigma_N)}, \quad \forall \, \sigma_1 \ldots \sigma_N, \tag{15}$$

$$\therefore \sum_{\sigma_0 = \pm 1} e^{F + \Sigma_{i=1}^{N} K_{0i} \sigma_0 \sigma_i} = e^{F' + \Sigma_{i=1}^{N} \Sigma_{j=i+1}^{N} K'_{ij} \sigma_i \sigma_j}. \tag{16}$$

Because $\sigma_1, \ldots, \sigma_N$ can each take the values ± 1, there are 2^N equations that need to be satisfied. Due to spin-flip symmetry, only 2^{N-1} of these are independent. For $N \geq 4$, the system of equations is overdetermined, with no solution. For example, for $N=4$, there are eight independent equations in seven unknowns F', K'_{12}, K'_{13}, K'_{14}, K'_{23}, K'_{34}, K'_{34}; the true effective action S' contains a four-spin interaction of the

form $K'_{1234}\sigma_1\sigma_2\sigma_3\sigma_4$ that does not fit within the network formalism. It is futile to keep track of such multispin terms because their number can grow to $O(2^N)$.

For $N=3$, however, there are four independent equations in four unknowns F', K'_{12}, K'_{13}, K'_{23}, so a Y-∇ transformation exists[10] such that
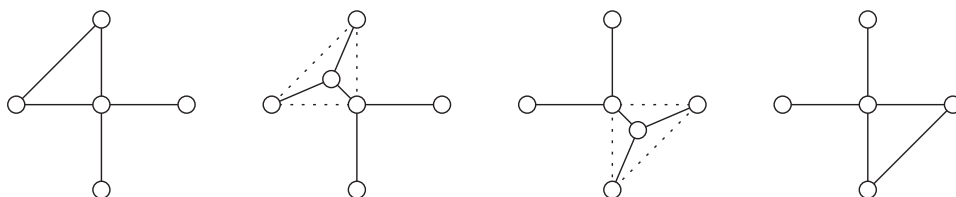
$$\sum_{\sigma_0} e^{F+K_{01}\sigma_0\sigma_1+K_{02}\sigma_0\sigma_2+K_{03}\sigma_0\sigma_3}$$

$$= 2e^F \cosh(K_{01}\sigma_1 + K_{02}\sigma_2 + K_{03}\sigma_3)$$

$$= e^{F'+K'_{23}\sigma_2\sigma_3+K'_{31}\sigma_3\sigma_1+K'_{12}\sigma_1\sigma_2}. \tag{17}$$

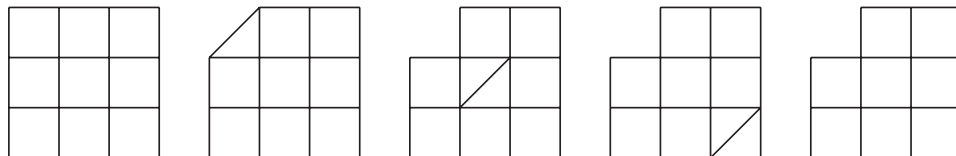Writing $\delta F = F' - F$ and the new couplings K'_{ij} in terms of the old couplings K_{i0},

$$2\cosh(K_{01}\sigma_1 + K_{02}\sigma_2 + K_{03}\sigma_3)$$
$$= e^{\delta F+K'_{23}\sigma_2\sigma_3+K'_{31}\sigma_3\sigma_1+K'_{12}\sigma_1\sigma_2}, \quad \forall \sigma_1, \sigma_2, \sigma_3,$$

$$\therefore \begin{cases} a_0 = 2\cosh(+K_{01}+K_{02}+K_{03}) = e^{\delta F+K'_{23}+K'_{31}+K'_{12}} \\ a_1 = 2\cosh(-K_{01}+K_{02}+K_{03}) = e^{\delta F+K'_{23}-K'_{31}-K'_{12}} \\ a_2 = 2\cosh(+K_{01}-K_{02}+K_{03}) = e^{\delta F-K'_{23}+K'_{31}-K'_{12}} \\ a_3 = 2\cosh(+K_{01}+K_{02}-K_{03}) = e^{\delta F-K'_{23}-K'_{31}+K'_{12}}, \end{cases}$$

$$\therefore \begin{cases} \delta F = \ln a_0^{1/4} a_1^{1/4} a_2^{1/4} a_3^{1/4} \\ K'_{23} = \ln a_0^{1/4} a_1^{1/4} a_2^{-1/4} a_3^{-1/4} \quad \text{(cycl.)}, \end{cases} \tag{18}$$

where a_i are functions of K_{0i} as above.

For computer implementation, it is preferable to write the Ising action as

$$e^{S(\{\sigma\})} = \text{const } z\prod_{\langle ij \rangle} k_{ij}^{(1-\sigma_i\sigma_j)/2}, \tag{19}$$

where $z=e^F$ and $k_{ij}=e^{-2K_{ij}}$. In this representation, the Y-∇ transformation is

$$\text{YDelta}(k_1,k_2,k_3) = (k_{23},k_{31},k_{12};\delta z)$$

where

$$\delta z = 1 + k_1 k_2 k_3,$$

$$z_1 = k_1 + k_2 k_3 \quad \text{(cycl.)},$$

$$b = \sqrt{z_1 z_2 z_3 / \delta z},$$

$$k_{23} = b/z_1 \quad \text{(cycl.)}, \tag{20}$$

where we have written $k_{0i} \equiv k_i$ and $\delta z = e^{\delta F}$ (and omitted primes on k_{23}, etc.). The transcendental functions (exp, log, and cosh) have been replaced by algebraic functions ($+$, $\times$, $\div$, and $\sqrt{}$), which are quicker to evaluate and have fewer complications arising from multivaluedness. Furthermore, infinite ferromagnetic couplings $K=\infty$ ("shorts" in the language of Ref. 14) are readily represented by $k=0$.

The ∇-Y transformation is equivalent to a Y-∇ transformation in the dual representation using Syozi's "cyclic change of lattices:"[11]

$$\text{DeltaY}(k_{23},k_{31},k_{12}) = (k_1,k_2,k_3)$$

where

$$p_1 = \text{dual}(k_{23}) \quad \text{(cycl.)},$$

$$(p_{23},p_{31},p_{12}) = \text{YDelta}(p_1,p_2,p_3),$$

(a) A single bond propagation move

(b) Lattice reduction

FIG. 3. The bond-propagation algorithm in the original form invented by Frank and Lobb. (a) A single bond-propagation move consists of a Y-∇ transformation followed by a ∇-Y transformation. (b) To reduce a finite square lattice, first, a corner node is eliminated by series reduction, producing the first diagonal bond, which is then propagated diagonally down and to the right until it is "absorbed" at the opposite edge. Other corner nodes are eliminated in the same way.
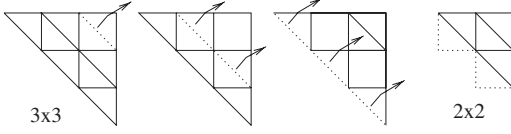
FIG. 4. Reduction of a triangular lattice of side L to one of side $L-1$ using $\frac{1}{2}L^2$ bond propagations and L series reductions.

$$k_1 = \text{dual}(p_{23}) \quad \text{(cycl.)}, \tag{21}$$

where the duality relation is the Möbius transformation $\text{dual}(k) = \frac{1-k}{1+k}$. However, this approach does not give an expression for the free energy change δF,[10] and we have found that a direct implementation of Eq. (21) is susceptible to roundoff error. The scheme below has better numerical performance:

$$x_1 = 2k_{31}k_{12}(1 - k_{23}^2) \quad \text{(cycl.)},$$

$$y_1 = 1 + k_{31}^2 k_{12}^2 - k_{12}^2 k_{23}^2 - k_{23}^2 k_{31}^2 \quad \text{(cycl.)},$$

$$v = \sqrt{y_1^2 - x_1^2},$$

$$k_1 = x_1/(y_1 + v) \quad \text{or} \quad (y_1 - v)/x_1 \quad \text{(cycl.)},$$

$$\delta z = 1 + k_1 k_2 k_3. \tag{22}$$

The two expressions for k_1 are formally equivalent; the choice depends on which one is numerically more stable (i.e., avoids subtraction of similar quantities).

Equations (20) and (22) are preferable to those in Refs. 2 and 12, because they each contain only one square root and have been optimized to reduce roundoff error. Both transformations are one-to-two mappings by virtue of the square roots, whose signs can be chosen arbitrarily.[18] For the ∇-Y transformation, this reflects the Z_2 symmetry of the action: for a given Y, one can obtain an equivalent Y by flipping the signs of σ_0, K_{01}, K_{02}, and K_{03} simultaneously. There is no
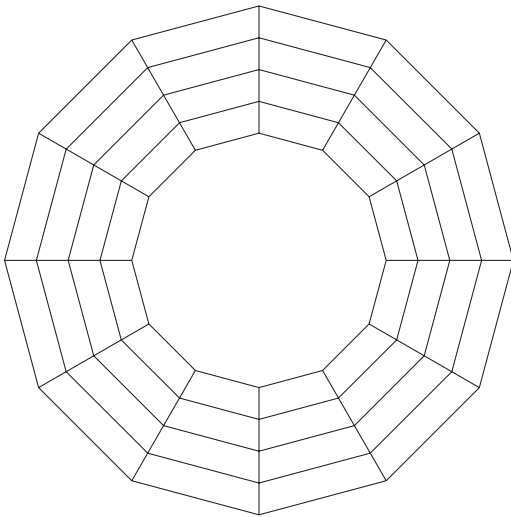

FIG. 5. A cylinder, when viewed in perspective from one end, is equivalent to an annulus, which is planar and hence $Y\nabla Y$ reducible.
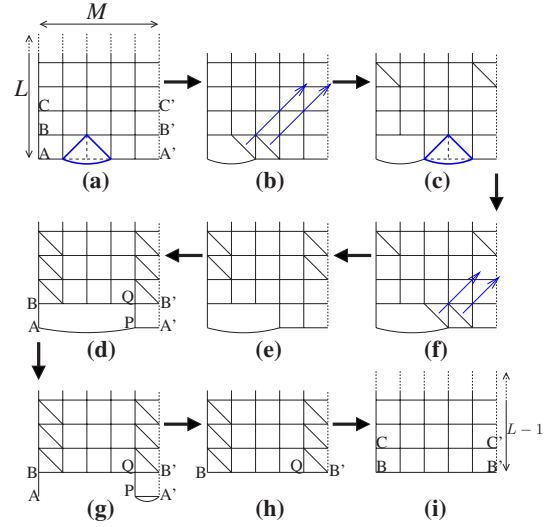

FIG. 6. (Color online) Bond propagation for cylindrical boundary conditions, illustrated for a cylinder of circumference $M=5$ and length $L \gg M$. Points A', B', C' are the periodic images of A, B, C. The procedure starts at (a) with a Y-∇ transformation at the bottom row. The resulting network is topologically equivalent to network (b). Propagating the two diagonal bonds in (b) all the way to the far end of the cylinder would lead to an algorithm requiring $O(L^2M)$ time. For a long cylinder with $L \gg M$, this is inefficient. It is better to propagate the diagonal bonds just far enough (c) so that only two nodes remain in the bottom row, as in (f). Then, since A and A' are equivalent, the bond AP can be combined in parallel with PA'. Subsequent series reductions lead to (h). We can now propagate the diagonal bonds *back* downward until they disappear at the bottom edge of the network, leaving a cylinder of length $L-1$. Repeating this process allows us to eliminate an entire cylinder in $O(LM^2)$ time.

clear physical interpretation of the double valuedness of the Y-∇ transformation.

Setting one or more of the k_i to 1 in Eqs. (20) gives the star-mesh formulas for $N=2,1,0$:

$$N=2: k_{12} = \frac{k_1 + k_2}{1 + k_1 k_2}, \quad z_0 = 1 + k_1 k_2, \tag{23}$$

$$N=1: z_0 = 1 + k_1, \tag{24}$$

$$N=0: z_0 = 2. \tag{25}$$

The $N=0$ case (integrating out a free spin) simply increases the constant free energy term by ln 2, i.e., by 1 bit.

The special cases below are useful for computer implementation. They are equivalent to Eq. (14), with G replaced by K:

$$\text{YDelta}(0, k_1, k_2) = (1, k_2, k_1),$$

$$\text{DeltaY}(1, k_1, k_2) = (0, k_2, k_1),$$

$$\text{DeltaY}(0, k_1, k_2) = (k_1 k_2, 0, 0). \tag{26}$$

IV. BOND-PROPAGATION PROCEDURE

The transformations derived in the previous section can be used to *reduce* a network (i.e., eliminate all or nearly all its nodes) and hence to calculate the partition function or effective coupling(s). Although an inefficient method, Gaussian models can be reduced simply by eliminating the nodes one by one, in any order, using the N-star-mesh transformation, Eq. (9). In contrast, for Ising models, a general star-mesh transformation does not exist. However, certain networks can be reduced using only $N=3,2,1,0$ star-mesh and mesh-star transformations; these are known as $Y\nabla Y$-reducible networks. The bond-propagation algorithm is an efficient example of $Y\nabla Y$ reduction.

A. Original bond-propagation algorithm

Gaussian models and Ising models on square lattices[19] can be reduced by the bond-propagation procedure[2,14] depicted in Fig. 3. A single bond-propagation move is illustrated in Fig. 3(a). First, a ∇-Y transformation is performed on the upper left ∇ of Fig. 3(a). This introduces one new node. The new node is now effectively shifted so as to replace the node to its lower right, a "move" which does not change the topology of the network. Then, a Y-∇ transformation is used to convert the lower right Y into a ∇, removing a node. In this way, any diagonal bond can be "propagated" into a diagonally adjacent plaquette. The prescription originally implemented in Ref. 14 is shown in Fig. 3(b). Starting from the upper left corner in Fig. 3(b), a series reduction is used to convert the corner into a diagonal bond. Then, using successive bond-propagation moves, this diagonal bond can be moved diagonally down and to the right until it annihilates at an edge with open boundary conditions.

For resistor networks, repeated bond-propagation moves reduce the network to a single string of conductors in series, which is easily reducible to one effective conductance. For an Ising model, the effective coupling is related to the correlation function between corner spins: $\langle \sigma_{11}\sigma_{NN}\rangle$ $=\tanh K_{11,NN}$. One may also wish to calculate the partition function in the Ising model; in order to do this, one must collect the contributions to the partition function during every transformation of the network. At the end of the calculation, after the last spin or node is eliminated, one has the free energy; the penultimate step gives the effective Ising coupling or effective resistance.

The number of operations necessary to accomplish this is of the order of $O(L^3)$. Reference 14 showed empirically that by taking advantage of the early termination of bond propagation on dilute lattices, it is possible to achieve $O(L^2 \ln L)$ computational time scaling near the percolation threshold.

B. Useful variants

Figure 4 illustrates a variant of the algorithm for a (finite) triangular lattice. Here, by embedding the triangular lattice in a square lattice, we see that one-third of the bonds can be considered "diagonal" to begin with. If these diagonal bonds are only propagated out when it is necessary to make space for other diagonal bonds, then this procedure reduces the size
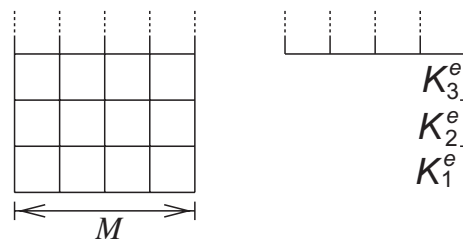

FIG. 7. Bond propagation for strips. From the asymptotic distribution of K_n^e, one can infer quantities such as the asymptotic effective resistance per unit length as a function of strip width M (for a random resistor network) or the correlation length ξ_M for a random-bond Ising model of width M.

of the lattice by one unit in every direction.[20] Alternatively, the diagonal bonds can all be propagated out, and then the BPA can proceed as with any square lattice.

Any planar network can be embedded in a square or triangular lattice (e.g., by inserting "opens" and shorts which do not change the partition function), to which the BPA can then be applied. A network with cylindrical (open-periodic) boundary conditions is equivalent to an annular network, which is planar (see Fig. 5). Figure 6 shows an adaptation of the BPA for reducing a square lattice with cylindrical boundary conditions in $O(LM^2)$ time, where L is the length of the cylinder and M is the circumference. Networks with toroidal (doubly periodic) boundary conditions are nonplanar.

Self-averaging quantities can be calculated efficiently in a strip geometry. For the rectangular strip in Fig. 7, use a series reduction on the lower left corner to produce the first diagonal bond, which is propagated up and to the right until it is removed from the system. This creates two new corners in the lower left, to which the same procedure can be applied. As corners are eliminated, this leaves a string of bonds with couplings K_1^e, K_2^e, and so on. For sufficiently large n, the couplings K_n^e, K_{n+1}^e, etc., come from the same distribution. This facilitates the calculation of self-averaging quantities in $O(LM^2)$ time for a rectangular $L \times M$ strip. Figure 8 shows an example of such a calculation for a RBIM with a Gauss-
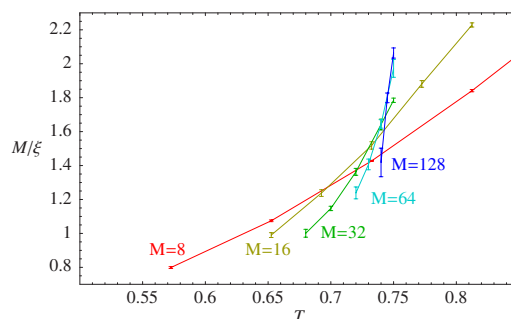

FIG. 8. (Color online) Scaled inverse correlation length $M/\xi_M(T)$ for the square-lattice Gaussian RBIM for $M \times L$ strips at disorder $\sigma_K=0.9$. Critical properties can be extracted by focusing on the intersections of these curves and extrapolating to $M=\infty$ or by the method of scaling collapse (Ref. 3). The widest strips (M $=128$) had length $L \sim 2 \times 10^5$ and took 3300 s to reduce on a 3.20 GHz Pentium 4 CPU.
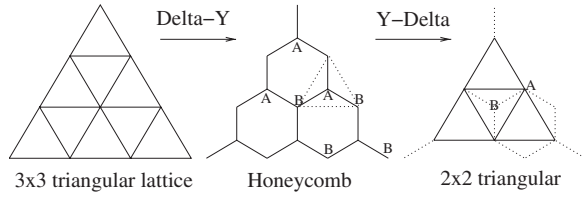
FIG. 9. Highly parallel algorithm for reduction of a triangular lattice. Starting with a triangular lattice of A sites, perform ∇-Y transformations to insert B sites. On the resulting honeycomb lattice, perform Y-∇ transformations to eliminate all the A sites. This gives a triangular lattice that is one unit smaller in every direction.

ian distribution of bond strengths, $P(\widetilde{K}_{ij}) \propto \exp -\frac{1}{2}\left(\frac{\widetilde{K}_{ij}-1}{\sigma_K}\right)^2$, with $\sigma_K = 0.9$. To save memory, the BPA is run incrementally: as the bottom rows are eliminated, new rows of random bonds are added at the top. The first $10M$ effective couplings K_n^e are discarded. For the subsequent couplings, we calculate the duals $R_n = -\frac{1}{2} \ln \tanh K_n^e$ and hence the inverse correlation length $\xi_M^{-1} = 2R_n$. ξ_M is the decay length of the correlations along the strip, i.e., $|\langle \sigma_{1,1}\sigma_{1,L}\rangle| \sim e^{-L/\xi_M}$ for large L. As $M \to \infty$, the quantity $M/\xi_M(T)$ tends to infinity or zero away from criticality, and to a finite value, exactly at criticality. As described in Ref. 3, critical properties can be extracted by finite-size scaling of the intersections of the curves or by collapsing the plots onto a universal curve.

Instead of using the BPA $Y\nabla Y$ reduction procedure, efficient *parallel algorithms* may be developed by direct $Y\nabla Y$ reduction on every plaquette simultaneously. The algorithm shown in Fig. 9 represents the same arithmetic operations as in Fig. 3 but performed in a different order. Starting with a triangular lattice, one performs a ∇-Y transformation on every upward-pointing ∇, thus producing a honeycomb lattice. One then performs a Y-∇ transformation on every downward-pointing Y, producing a triangular lattice one unit smaller than the original.

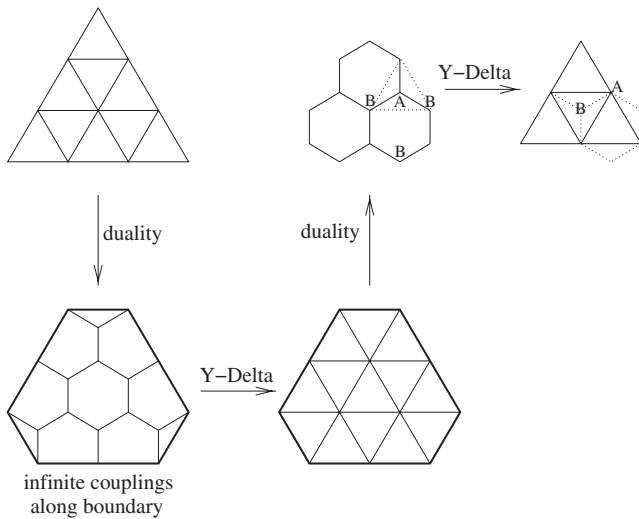Using the cyclic change of lattices introduced by Syozi,[11] it is even possible to devise an algorithm involving only Y-∇

FIG. 10. A variant of Fig. 9 using duality and Y-∇ transformations only. Thick lines represent infinite-strength bonds.
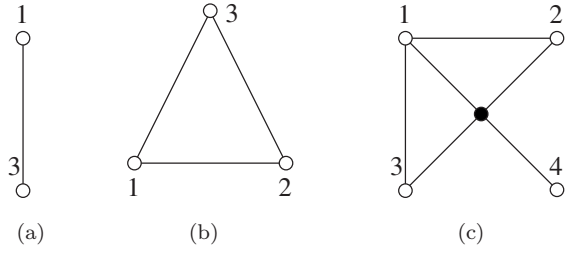

FIG. 11. Irreducible two-, three-, and four-terminal networks.

and duality transformations, as in Fig. 10. The dual of a planar network is derived by drawing a dual bond cutting across each bond of the original network, with a bond strength given by the duality relation $\text{dual}(G) = \frac{1}{G}$ (for resistors) or $\text{dual}(k) = \frac{1-k}{1+k}$ (for Ising bonds).

The latter two algorithms for $Y\nabla Y$ reduction are very suitable for parallelization. In Figs. 9 and 10, the $O(L^3)$ operations can be distributed among L^2 processors so that the
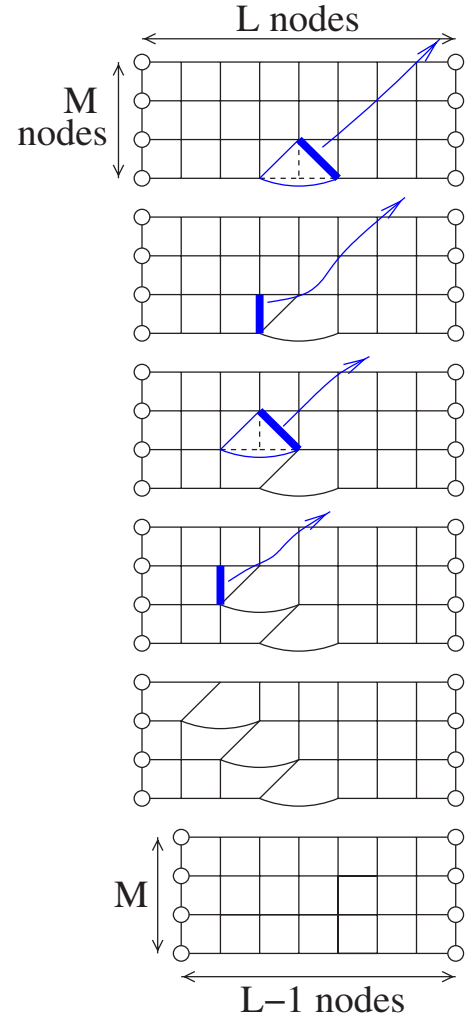

FIG. 12. (Color online) "Contraction" of an $L \times M$ rectangular strip preserving nodes along short edges, where $L > 2M$. Dotted lines denote bonds that are removed by Y-∇ transformations; bold lines indicate bonds that are propagated out. Each curved arrow indicates a sequence of bond propagations.

entire network can be reduced in only $O(L)$ time.

C. Reduction of networks with external terminals

When the BPA is used to compute the effective couplings between an even number of "external" or "terminal" spins (2, 4, 6,…), one must take care *not* to eliminate these terminal spins. Given any two-terminal planar network, it is always possible to integrate out all the "internal" spins, reducing the network to a single bond between the two terminals [Fig. 11(a)]. However, for a general four-terminal planar network, it is not possible to reduce the network beyond the stage illustrated in Fig. 11(c), which has one internal spin. It may be interesting to find irreducible networks with larger numbers of terminals (see, e.g., Ref. 21). A rectangular $L \times M$ strip ($L > 2M$) with terminals all along its short edges can be reduced to a $2M \times M$ strip by successive applications of the procedure in Fig. 12; this may be thought of as a decimation procedure which preserves the form of the Ising Hamiltonian.

For a triangular network with one terminal at each corner, the algorithm in Fig. 9 leads to an irreducible three-terminal network [Fig. 11(b)], from which the effective two-terminal couplings K_{23}, K_{31}, K_{12} may easily be found. For uncorrelated random-bond models, this approach produces effective couplings for three disorder realizations for the price of one.

V. BOND PROPAGATION FOR CALCULATING DERIVATIVES OF THE PARTITION FUNCTION

The algorithm described above (and its variants) allows us to calculate the numerical value of the partition function $Z(\beta)$ at one value of β. To compute other thermodynamic quantities such as the mean energy U, one approach is to calculate Z at closely spaced values of β and perform numerical differentiation. This is not appealing because while the algorithm produces near-machine-precision values for Z, this would yield only approximate values for U. We now show that it is, in fact, possible to extend the bond-propagation algorithm to compute derivatives of Z *directly*, i.e., without numerical differentiation.

The quantity $U = -\frac{d}{d\beta} \ln Z(\beta)$ can be computed from

$$- \beta Z U = \sum_{\{\sigma\}} S(\{\sigma\}) e^{S(\{\sigma\})} = \sum_{\{\sigma\}} \left(\sum_{\langle ij \rangle} K_{ij} \sigma_i \sigma_j \right) \exp \sum_{\langle ij \rangle} K_{ij} \sigma_i \sigma_j. \tag{27}$$

In order to use the BPA to calculate this quantity directly, we develop Y-∇ and ∇-Y transformations which preserve this quantity. First, consider a Y with terminal spins σ_1, σ_2, σ_3 and central spin σ_0. In fact, we can solve a more general expression, in which the bond strengths in the prefactor can differ from those in the exponent, such that the Y is described by eight parameters A, L_{01}, L_{02}, L_{03} and F, K_{01}, K_{02}, K_{03}:

$$C = \sum_{[\sigma_{1,2,3,\ldots}]} \sum_{\sigma_0} (c_{\text{ext}} + A + L_{01}\sigma_1\sigma_0 + L_{02}\sigma_2\sigma_0 + L_{03}\sigma_3\sigma_0) e^{S_{\text{ext}} + F + K_{01}\sigma_1\sigma_0 + K_{02}\sigma_2\sigma_0 + K_{03}\sigma_3\sigma_0}. \tag{28}$$

In the above expression, the quadratic terms in the prefactor and exponent not involving σ_0 have been collected in the functions $c_{\text{ext}}(\sigma_1, \sigma_2, \ldots)$ and $S_{\text{ext}}(\sigma_1, \sigma_2, \ldots)$. Factorizing and evaluating the sum over σ_0,

$$C = \sum_{[\sigma_{1,2,3,\ldots}]} e^{S_{\text{ext}} + F} \left\{ \sum_{\sigma_0} (c_{\text{ext}} + A) e^{K_{01}\sigma_1\sigma_0 + K_{02}\sigma_2\sigma_0 + K_{03}\sigma_3\sigma_0} + \sum_{\sigma_0} (L_{01}\sigma_1\sigma_0 + L_{02}\sigma_2\sigma_0 + L_{03}\sigma_3\sigma_0) e^{K_{01}\sigma_1\sigma_0 + K_{02}\sigma_2\sigma_0 + K_{03}\sigma_3\sigma_0} \right\}$$

$$= \sum_{[\sigma_{1,2,3,\ldots}]} e^{S_{\text{ext}} + F} \{ \underbrace{2(c_{\text{ext}} + A)\cosh(K_{01}\sigma_1 + K_{02}\sigma_2 + K_{03}\sigma_3)}_{\text{1st term}} + \underbrace{2(L_{01}\sigma_1 + L_{02}\sigma_2 + L_{03}\sigma_3)\sinh(K_{01}\sigma_1 + K_{02}\sigma_2 + K_{03}\sigma_3)}_{\text{2nd term}} \}. \tag{29}$$

We wish to find the new parameters A', L'_{23}, L'_{31}, L'_{12} and F', K'_{23}, K'_{31}, K'_{12} of the equivalent ∇ such that

$$C = \sum_{[\sigma_{1,2,3,\ldots}]} (c_{\text{ext}} + A' + L'_{23}\sigma_2\sigma_3 + L'_{31}\sigma_3\sigma_1 + L'_{12}\sigma_1\sigma_2) e^{S_{\text{ext}} + F' + K'_{23}\sigma_2\sigma_3 + K'_{31}\sigma_3\sigma_1 + K'_{12}\sigma_1\sigma_2}$$

$$= \sum_{[\sigma_{1,2,3,\ldots}]} e^{S_{\text{ext}} + F} \{ \underbrace{(c_{\text{ext}} + A) e^{\delta F + K'_{23}\sigma_2\sigma_3 + K'_{31}\sigma_3\sigma_1 + K'_{12}\sigma_1\sigma_2}}_{\text{1st term}} + \underbrace{(\delta A + L'_{23}\sigma_2\sigma_3 + L'_{31}\sigma_3\sigma_1 + L'_{12}\sigma_1\sigma_2) e^{\delta F + K'_{23}\sigma_2\sigma_3 + K'_{31}\sigma_3\sigma_1 + K'_{12}\sigma_1\sigma_2}}_{\text{2nd term}} \}, \tag{30}$$

where $\delta A = A' - A$ and $\delta F = F' - F$. Comparing Eqs. (29) and (30) suggests equating the two terms inside the braces separately. Equating the first term leads to

$$2 \cosh(K_{01}\sigma_1 + K_{02}\sigma_2 + K_{03}\sigma_3)$$

$$= e^{\delta F + K'_{23}\sigma_2\sigma_3 + K'_{31}\sigma_3\sigma_1 + K'_{12}\sigma_1\sigma_2}, \quad \forall \, \sigma_1, \sigma_2, \sigma_3, \tag{31}$$

which is identical to Eq. (18). Equating the second terms in the braces in Eqs. (29) and (30) gives

$$2(L_{01}\sigma_1 + L_{02}\sigma_2 + L_{03}\sigma_3)\sinh(K_{01}\sigma_1 + K_{02}\sigma_2 + K_{03}\sigma_3)$$

$$= (\delta A + L'_{23}\sigma_2\sigma_3 + L'_{31}\sigma_3\sigma_1 + L'_{12}\sigma_1\sigma_2)$$

$$\times e^{\delta F + K'_{23}\sigma_2\sigma_3 + K'_{31}\sigma_3\sigma_1 + K'_{12}\sigma_1\sigma_2}, \quad \forall \, \sigma_1, \sigma_2, \sigma_3. \tag{32}$$

We can now use Eq. (18) to rewrite Eq. (32) as

$$2(L_{01}\sigma_1 + L_{02}\sigma_2 + L_{03}\sigma_3)\sinh(K_{01}\sigma_1 + K_{02}\sigma_2 + K_{03}\sigma_3)$$
$$= (\delta A + L'_{23}\sigma_2\sigma_3 + L'_{31}\sigma_3\sigma_1 + L'_{12}\sigma_1\sigma_2)2\cosh(K_{01}\sigma_1 + K_{02}\sigma_2 + K_{03}\sigma_3), \quad \forall \ \sigma_1,\sigma_2,\sigma_3, \tag{33}$$

$$\therefore (\delta A + L'_{23}\sigma_2\sigma_3 + L'_{31}\sigma_3\sigma_1 + L'_{12}\sigma_1\sigma_2)$$
$$= (L_{01}\sigma_1 + L_{02}\sigma_2 + L_{03}\sigma_3)\tanh(K_{01}\sigma_1 + K_{02}\sigma_2 + K_{03}\sigma_3), \tag{34}$$

$$\therefore \begin{cases} \delta A + L'_{23} + L'_{31} + L'_{12} = u_0 = (+L_{01} + L_{02} + L_{03})\tanh(+K_{01} + K_{02} + K_{03}) \\ \delta A + L'_{23} - L'_{31} - L'_{12} = u_1 = (-L_{01} + L_{02} + L_{03})\tanh(-K_{01} + K_{02} + K_{03}) \quad \text{(cycl.)}, \end{cases} \tag{35}$$

$$\therefore \begin{cases} \delta A = (u_0 + u_1 + u_2 + u_3)/4 \\ L'_{23} = (u_0 + u_1 - u_2 - u_3)/4 \quad \text{(cycl.)}, \end{cases} \tag{36}$$

where u_i are functions of L_{0i} and K_{0i} as defined above.

This Y-∇ transformation can be inverted to give ∇-Y transformation. First, compute K_{01}, K_{02}, K_{03} using Eq. (22). Then, let

$$\delta A = \frac{(c_0 - c_1 + c_2 + c_3)L'_{23} + cycl.}{-c_0 + c_1 + c_2 + c_3},$$

$$L_{01} = \frac{(c_0 - c_1)(c_2 + c_3)L'_{23} + (c_1 c_2 + c_0 c_3)L'_{31} + (c_3 c_1 + c_0 c_2)L'_{12}}{-c_0 + c_1 + c_2 + c_3} \quad \text{(cycl.)}, \tag{37}$$

where

$$c_0 = \coth(K_{01} + K_{02} + K_{03}),$$

$$c_1 = \coth(-K_{01} + K_{02} + K_{03}) \quad \text{(cycl.)}. \tag{38}$$

Equations (20), (22), (36), and (37) can be used in the bond-propagation algorithm (or any other $Y\nabla Y$ reduction scheme) to compute the mean energy U of an Ising network directly, without numerical differentiation. Note that since the transformations which preserve Z are not the same as those which preserve its derivative Z', obtaining $U = -Z'/Z$ requires two separate bond-propagation calculations: one to calculate Z; the other to calculate Z'. It may be possible to calculate the heat capacity and other higher derivatives of Z in a similar manner.

VI. DISCUSSION

We have found that bond propagation generally gives answers to high accuracy (close to machine precision), provided that one uses a careful implementation that avoids overflow and underflow and minimizes roundoff error, especially in the case of large systems, low temperatures, or frustration. We discuss specific cases below.

A. Ferromagnetic models

The algorithm works rather straightforwardly for ferromagnetic Ising models where all the K_{ij} are positive. The

Y-∇ and ∇-Y transformations should be implemented in a way that minimizes roundoff error, as in Eqs. (20) and (22). For large systems in the ferromagnetic phase ($L \gtrsim 10^3$), the partition function Z may cause floating-point overflow, which can easily be dealt with by storing $F = \ln Z$ instead. The couplings themselves k_{ij} may also underflow, in which case, it may be necessary to use an alternative representation to store the couplings and perform the transformations or to change the order of the bond propagations.

B. Dilute models

Applying bond propagation to dilute models containing zero couplings may generate infinite couplings, as pointed out by Frank-Lobb. We have confirmed that this can be successfully dealt with using Eqs. (14) or (26).

C. Frustration

All couplings K_{ij} in a physical Ising model must be real valued. The Y-∇ transformation, Eq. (20), preserves this property, but the ∇-Y transformation, Eq. (22), can generate complex-valued couplings. This happens if the argument of the square root is negative ($v^2 = y_1^2 - x_1^2 < 0$), i.e.,

$$(1 + k_{23}k_{31} + k_{31}k_{12} + k_{12}k_{23})(1 + k_{23}k_{31} - k_{31}k_{12} - k_{12}k_{23})$$
$$\times (1 - k_{23}k_{31} + k_{31}k_{12} - k_{12}k_{23})$$
$$\times (1 - k_{23}k_{31} - k_{31}k_{12} + k_{12}k_{23}) < 0. \tag{39}$$

This "frustration inequality" can be taken as the definition of a "frustrated ∇."[22] Frustration occurs in ∇'s where all three couplings have similar magnitudes and one or three or them are antiferromagnetic ($K_{ij} < 0$, $k_{ij} > 1$). The BPA still works for frustrated systems provided that complex arithmetic is used. Despite the occurrence of complex intermediate couplings, we have found that the value of the partition function emerging from the entire calculation is real, as expected, with a small imaginary part arising from roundoff error.

The models typically studied in the literature are random-bond Ising models with bond strengths picked independently from the same distribution, $P(K)$. The two most common models are the binary ($\pm K_0$) RBIM, with $P(K_{ij}) = (1-p)\delta(K_{ij}-K_0)+p\delta(K_{ij}+K_0)$, where p is the concentration of antiferromagnetic couplings, and the Gaussian RBIM, with $P(K_{ij}) \propto \exp-\frac{1}{2}\left(\frac{K_{ij}-K_0}{\sigma_K}\right)^2$.

The BPA works well for the Gaussian RBIM. For the binary RBIM, certain disorder realizations contain perfectly frustrated plaquettes; these cause the BPA to generate infinite and zero couplings, which then lead to indeterminate results. This is due to singularities in the function represented by the bond-propagation procedure (the composition of all the Y-∇ and ∇-Y transformations). These can be avoided using perturbations away from perfect frustration, including roundoff error itself. Such perturbations may be relevant in a renormalization-group sense, changing the universality class; if the scaling flow is weak, it may still be possible to obtain accurate results this way. Introducing perturbations this way is acceptable as long as the systematic error it produces does not exceed the random statistical error from disorder averaging, which is an inevitable part of numerical calculations on RBIMs. At very low temperatures, roundoff error may become an issue for both types of RBIM.

D. Comparisons with other algorithms

Gaussian models (including resistor networks) possess N-star-mesh transformations for all $N \geq 1$, so there are a large number of algorithms for reducing them. Below is a comparison of the number of multiplications ($\times$) and divisions ($\div$) required to perform various algorithms on Gaussian models (e.g., resistor networks) on an $L \times L$ lattice.

(a) Cramer's rule: $O(L^2!)$ ($\times$), $O(1)$ ($\div$);
(b) full Gauss elimination: $O(L^6)$ ($\times$), $O(L^2)$ ($\div$);
(c) transfer matrices:[23,24] $O(L^4)$ ($\times$), $O(L^3)$ ($\div$);
(d) banded Gauss elimination: $O(L^4)$ ($\times$), $O(L^2)$ ($\div$);
(e) nested dissection:[30,31] $O(L^3)$ ($\times$), $O(L^2)$ ($\div$); and
(f) bond propagation:[14] $O(L^3)$ ($\times$), $O(L^3)$ ($\div$).

The last two methods are the fastest, both taking $O(L^3)$ time overall. The nested dissection algorithm developed by George[30] and by Lipton et al.[31] is a hierarchical "divide-and-conquer" method that exploits prior knowledge about the connectivity of the network. It is quite a general idea and it can be used with doubly periodic boundary conditions, as well as for lattices in dimensions $d > 2$, for which the time requirement is $O(L^{3d-3})$. Its drawback is the amount of bookkeeping required. On the other hand, bond propagation, while slightly more costly in terms of divisions, is simpler to implement and especially to parallelize (see Sec. IV B). For

dilute networks near the percolation threshold, bond propagation is even faster, taking $O(L^2 \ln L)$ time, because many propagating bonds terminate early. In the context of Gaussian models, bond propagation can be viewed as an efficient linear algebra method for reducing certain pentadiagonal matrices while preserving their sparsity.

For Ising models, numerically exact polynomial-time methods only exist in two dimensions.[32] Below are operation counts including addition ($+$), subtraction ($-$), multiplication ($\times$), division ($\div$), and square roots ($\checkmark$) for selected methods.

(a) Transfer matrices in the spin basis:[33] $O(2^L)$ operations ($+\times$) and memory; can be generalized to higher-dimensional lattices;
(b) fermion-network method:[4] $O(L^4)$ operations ($+\times$);
(c) Pfaffian elimination with nested dissection:[9] $O(L^3)$ operations ($+\times\div$);
(d) $Y\nabla Y$ reduction by Feo-Provan algorithm:[12] $O(L^4)$ operations ($+-\times\div\checkmark$) for general graphs; and
(e) $Y\nabla Y$ reduction by bond propagation:[2] $O(L^3)$ operations ($+-\times\div\checkmark$).

Due to the lack of a general N-star-mesh transformation for Ising models, nested dissection cannot be applied directly to the Ising network. Rather, one must first map the Ising problem to a dimer problem on the Kasteleyn "terminal lattice" which has four nodes for each Ising spin. In two dimensions, the dimer problem maps to a Pfaffian elimination problem, to which nested dissection can then be applied. In contrast, the bond-propagation algorithm can be performed directly in the Ising network representation, with no need to map to dimers or fermion models.

E. Limitations

$Y\nabla Y$ reduction techniques (of which the bond-propagation algorithm is one) for statistical mechanics models rely on two criteria: (a) $Y\nabla Y$ reducibility of the graph on which the model is defined and (b) the existence of Y-∇ and ∇-Y transformations that preserve some aspect of the model at hand. We first discuss criterion (a). The Robertson-Seymour theory of forbidden minors and obstruction sets (see Ref. 34 and references therein) implies the following relationships among various sets of graphs:

$$\{\text{Forests}\} \subset \{\text{Outerplanar graphs}\}$$
$$\subset \{\text{Series-parallel-reducible graphs}\} \subset \{\text{Planar graphs}\}$$
$$\subset \{Y\nabla Y\text{-reducible graphs}\}$$
$$\subset \{\text{Linklessly embeddable graphs}\}. \tag{40}$$

In particular, this shows that all planar graphs are $Y\nabla Y$ reducible. However, it is also true that nearly all "sufficiently large" nonplanar graphs with crossing number 2 or greater are not $Y\nabla Y$ reducible. This means that $Y\nabla Y$ approaches such as bond propagation are unable to completely reduce three-dimensional (3D) lattices or even 2D "pyrochlore" lattices (lattices of edge- or corner-sharing tetrahedra). For example, a large enough 3D network with simple cubic symmetry has as a minor the Petersen graph $KG_{5,2}$. Since $KG_{5,2}$

TABLE I. Degrees of freedom of Y and ∇ subnetworks. For all cases except the standard Ising model, there are more equations than ∇ parameters, indicating that the effective action contains three-spin interactions and that exact Y-∇ transformations do not exist.

Type of model	Distinct equalities	Y parameters	∇ parameters
Ising	4	4 $(F, K_{01}, K_{02}, K_{03})$	4 $(F, K_{23}, K_{31}, K_{12})$
Ising with fields	8	8 $(F, K_{01}, K_{02}, K_{03}, h_1, h_2, h_3, h_0)$	7 $(F, K_{23}, K_{31}, K_{12}, h_1, h_2, h_3)$
Potts $(q \geqslant 3)$	5	4	4
Potts (generalized, $q \geqslant 3$)	q^3	$3q^2 - 2$	$3q^2 - 2$

is a forbidden minor for $Y \nabla Y$ reduction, the simple cubic lattice is not amenable to the BPA nor to any algorithm based purely on $Y \nabla Y$ reduction. Since general 3D lattices contain the simple cubic lattice as a minor, they are not $Y \nabla Y$ reducible either. As a further example, although diagonal bonds can be made to propagate on, e.g., a tiling of three space by truncated octahedra, this 3D lattice is not *reducible* by bond propagation. Of course, even for networks that are not *completely* $Y \nabla Y$ reducible, it may be useful to perform *partial* $Y \nabla Y$ reduction before resorting to more expensive computational techniques.

We now turn to criterion (b). In Secs. III and V, we derived Y-∇ and ∇-Y transformations for Gaussian models and Ising models. These models have the special property that three-spin interactions are forbidden by symmetry. For most other models in statistical mechanics, eliminating a three-coordinated spin generates three-spin interactions which cannot be described within the network formalism. For example, consider the Ising model in the presence of magnetic fields h_i (which may be uniform or random) described by the action

$$e^{S(\{\sigma\})} = \exp\left(\sum_{\langle ij \rangle} K_{ij} \sigma_i \sigma_j + \sum_i h_i \sigma_i \right). \quad (41)$$

Let us integrate out the center spin of a Y and find the effective action for the resultant ∇ by equating Boltzmann weights for each of the eight configurations of the outer spins $(\sigma_1, \sigma_2, \sigma_3 = \pm 1)$:

$$\sum_{\sigma_0} e^{S(\sigma_0, \sigma_1, \sigma_2, \sigma_3)} = e^{S'(\sigma_1, \sigma_2, \sigma_3)}. \quad (42)$$

If we restrict ourselves to the form of Eq. (41), there are only seven possible terms: $S' = F + h_1 \sigma_1 + h_2 \sigma_2 + h_3 \sigma_3 + K_{23} \sigma_2 \sigma_3 + K_{31} \sigma_3 \sigma_1 + K_{12} \sigma_1 \sigma_2$. Therefore, S' must contain a three-spin interaction $K_{123} \sigma_1 \sigma_2 \sigma_3$, and it cannot be written in the same analytic form as the original action: a Y-∇ transformation does not exist.

Table I shows a "feasibility study" for Y-∇ transformations for the q-state Potts model, with couplings K_{ij} that tend to align adjacent spins in the same "direction" in spin space,

$$e^{S(\{\sigma\})} = \exp \sum_{\langle ij \rangle} K_{ij} \delta_{\sigma_i \sigma_j}, \quad \sigma_i \in \{1, 2, \ldots, q\} \,\forall\, i, \quad (43)$$

a generalized Potts model (where every bond is described by a matrix $\mathbf{P}_{ij}$ representing the weight factors for every combination of directions of two adjacent spins),

$$e^{S(\{\sigma\})} = \exp \sum_{\langle ij \rangle} P_{ij\sigma_i\sigma_j}, \quad \sigma_i \in \{1, 2, \ldots, q\} \,\forall\, i, \quad (44)$$

and the q-state clock model (where $q \geqslant 3$),

$$e^{S(\{\sigma\})} = \exp \sum_{\langle ij \rangle} K_{ij} \cos \frac{2\pi}{q} (\sigma_i - \sigma_j). \quad (45)$$

For these models there are, again, more equations than parameters, so exact Y-∇ transformations do not exist.

In the limit $q \to \infty$, the clock model becomes the XY model. This suggests that Y-∇ transformations do not exist for XY models nor for other models with continuous spins (e.g., classical Heisenberg models).

This paper has focused on statistical mechanics, but $Y \nabla Y$ reduction and the BPA also have important applications in combinatorics and operations research. Exact Y-∇ and ∇-Y transformations exist for the shortest path problem, for the maximum flow (or minimum cut) problems,[13] for counting spanning trees and perfect matchings,[12] and even for 2D foams in mechanical equilibrium.[35] Such transformations *do not* exist for network reliability problems,[36,37] nor for equilibrium flow problems with arbitrary potential-flow relations[13] (i.e., electrical circuits composed of elements with arbitrary nonlinear current-voltage characteristics). Thus, the former group of problems can be solved on planar networks in polynomial time, whereas the latter group cannot. In Table II, all of the above problems are classified according to the maximum degree of star-mesh transformations that they support.

The above examples have been for systems defined on undirected graphs, i.e., "symmetric" bonds. $Y \nabla Y$ reduction

TABLE II. Maximum degree of star-mesh and mesh-star transformations for various models.

Type of model	$N_{\max}^{\text{star-mesh}}$	$N_{\max}^{\text{mesh-star}}$
Gaussian models, resistor networks, shortest path, max flow	∞	3
Ising model	3	3
Ising with field	2	3
Potts, clock, XY, Heisenberg, network reliability	2	2

(and the BPA) may also be applicable to *directed* graph problems such as dynamical Ising cellular automata.[38] Further examples of models with Y-∇ transformations (or Yang-Baxter equations) can be found in Ref. 39; the existence of such transformations is related to the "integrability" of these models. Note that in systems where *exact* Y-∇ transformations do not exist, *approximate* Y-∇ transformations may still have some use.[36]

VII. CONCLUSION

In conclusion, we have shown that the bond-propagation algorithm (BPA) may be applied not only to the partition function and correlation functions in the Ising model but that it can also be applied to thermodynamic quantities via transformations which preserve the first derivative of the partition function. Similar transformations may exist for higher derivatives as well. Since the BPA is a form of $Y\nabla Y$ reduction, it is limited to networks which are $Y\nabla Y$ reducible. This includes planar graphs, and the BPA may be applied to Ising models on planar networks as well as planar resistor networks. In general, 3D lattices are not $Y\nabla Y$ reducible, and so the BPA is not amenable to 3D resistor networks or Ising models in 3D.

[1] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing* (Oxford University Press, Oxford, 2001).

[2] Y. L. Loh and E. W. Carlson, Phys. Rev. Lett. **97**, 227205 (2006).

[3] That is, its accuracy is limited by the precision of the floating-point numbers used in the computation; it suffers from roundoff error but not statistical error like Monte Carlo methods.

[4] F. Merz and J. T. Chalker, Phys. Rev. B **65**, 054425 (2002).

[5] P. W. Kasteleyn, J. Math. Phys. **4**, 287 (1963).

[6] M. E. Fisher, J. Math. Phys. **7**, 10 (1966).

[7] L. Saul and M. Kardar, Phys. Rev. E **48**, R3221 (1993).

[8] L. Saul and M. Kardar, Nucl. Phys. B **432**, 641 (1994).

[9] A. Galluccio, M. Loebl, and J. Vondrak, Phys. Rev. Lett. **84**, 5924 (2000).

[10] R. M. F. Houtappel, Physica **16**, 425 (1950).

[11] I. Syozi, in *Phase Transitions and Critical Phenomena*, edited by C. Domb and M. S. Green (Academic, New York, 1972), Vol. 1.

[12] C. J. Colbourn, J. S. Provan, and D. Vertigan, Discrete Appl. Math. **60**, 119 (1995).

[13] T. A. Feo and J. S. Provan, Oper. Res. **41**, 572 (1993).

[14] D. J. Frank and C. J. Lobb, Phys. Rev. B **37**, 302 (1988).

[15] The algorithms in Refs. 25–27 also take $O(N^{3/2})$ time.

[16] For Gaussian models such as Eq. (1), the partition function Z is related to the determinant det K, and for planar Ising models, Z is related to the Pfaffian of the "Kasteleyn terminal lattice," both of which may be computed in polynomial time. It is interesting that these same two models are amenable to the bond-propagation algorithm that we describe in this paper.

[17] See the Appendix of Ref. 39 for a similar discussion.

[18] The Ising Y-∇ and ∇-Y transformations of Ref. 12 each contain three square roots, suggesting that there are eight solutions, but only two of these are valid. In general, the couplings may be complex, and choosing principal values for the roots does not guarantee a valid solution: $\delta\beta\gamma=D\sqrt{AC/BD}\sqrt{AB/CD}$ is not necessarily equal to A as required. The transformations in Ref. 2 are written such that principal values give a valid solution.

[19] We use "lattice" to mean a "finite lattice"—a periodic structure that has been truncated at a closed bounding surface.

[20] Hilhorst *et al.* (Refs. 28 and 29) used a similar procedure to derive exact renormalization-group transformations for Ising models without disorder.

[21] D. Archdeacon, C. J. Colbourn, I. Gitler, and J. S. Provan, J. Graph Theory **33**, 83 (2000).

[22] There is an interesting analogy in geometry: the area of a triangle, $A=\frac{1}{4}\sqrt{(a+b+c)(b+c-a)(c+a-b)(a+b-c)}$, must satisfy $A^2\geq0$, which leads to "triangle inequalities"—geometrical constraints on the lengths of the sides of a triangle.

[23] B. Derrida and L. de Seze, J. Phys. (Paris) **43**, 475 (1982).

[24] B. Derrida and J. Vannimenus, J. Phys. A **15**, L557 (1982).

[25] K. Truemper, J. Graph Theory **13**, 141 (1989).

[26] I. Gitler, Ph.D. thesis, University of Waterloo, 1991.

[27] I. Gitler, report, 1991 (unpublished).

[28] H. J. Hilhorst, M. Schick, and J. M. J. van Leeuwen, Phys. Rev. Lett. **40**, 1605 (1978).

[29] H. J. Hilhorst, M. Schick, and J. M. J. van Leeuwen, Phys. Rev. B **19**, 2749 (1979).

[30] A. George, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **10**, 345 (1973).

[31] R. J. Lipton, D. Rose, and R. E. Tarjan, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **16**, 346 (1979).

[32] It is easily seen that the BPA (and other numerically exact methods), when they apply, is far superior to any form of Monte Carlo averaging over spin configurations. Bond propagation gives results to near machine precision (10^{-12}) in $O(L^3)$ time. Given $O(L^3)$ time, the most efficient Monte Carlo methods can generate at most L independent configurations, so the statistical error will be $O(L^{-1/2})$, i.e., about 10^{-2}. Furthermore, Monte Carlo methods have serious problems for frustrated systems.

[33] I. Morgenstern and K. Binder, Phys. Rev. Lett. **43**, 1615 (1979).

[34] N. Robertson and P. D. Seymour, J. Comb. Theory, Ser. B **92**, 325 (2004).

[35] M. Mancini and C. Oguey, Eur. Phys. J. E **22**, 181 (2007).

[36] A. Lehman, J. Soc. Ind. Appl. Math. **11**, 773 (1963).

[37] The network reliability problem is more commonly known in physics as "bond percolation." The BPA can be used to reduce a planar network with a particular configuration of "active" bonds, but it is not possible to automatically average over all bond configurations based purely on the bond probabilities.

[38] G. Odor, Rev. Mod. Phys. **76**, 663 (2004).

[39] J. H. H. Perk and H. Au-Yang, *Encyclopedia of Mathematical Physics*, edited by J.-P. Françoise, G. L. Naber, and S. T. Tsou (Oxford: Elsevier, 2006), Vol. 5, pp. 465–473.