

Random-Cluster Representation for the Blume–Capel Model

M. B. Bouabci¹ and C. E. I. Carneiro²

Received December 14, 1999

We present in this paper a way to perform the mapping of the spin-1 Blume–Capel model into a random-cluster model, and analyze thermodynamic properties of the former model in terms of geometric properties of clusters generated in the random-cluster representation. It is shown that there are two different relevant types of cluster, and that one of them is the exact analogue of the type of cluster generated in the Ising model. We use this result to derive expressions for thermodynamical properties on the second-order transition line which are equivalent to the ones found in the Ising model. The other type of cluster is responsible for the first-order transitions, and we may see the tricritical point as a point where both types of cluster compete on the same footing.

KEY WORDS: Phase-transitions; percolation; tricritical point.

1. INTRODUCTION

The description of thermodynamical phase transitions in terms of percolation models was introduced by Fortuin and Kastelyn in 1969,^(1,2) when it was shown that the q -state Potts model can be mapped into a random cluster model, and the percolation transition associated with the emergence of an infinite cluster is related to the divergence of the correlation length in the second-order phase-transition of the spin model.

Over the years this connection proved very fruitful. FKG inequalities, extremely useful in finding bounds to critical exponents, were first derived

¹ Mathematical Institute-MFG, Oxford University, 24–29 St. Giles, Oxford OX1 3LB, United Kingdom.

² Instituto de Física-FGE, Universidade de São Paulo, P.O. Box 66318, São Paulo, São Paulo 05315-970, Brazil.

using a random cluster representation.⁽³⁾ More recently, this representation has been used to derive exact large deviation bounds^(4,5) and magnetization discontinuity⁽⁶⁾ in some types of Ising models. Pfister and Velenik⁽⁷⁾ generalized the random cluster representation in order to map the Ashkin–Teller model into a percolation model.

Generally speaking, the map provides a unifying view over different models, allowing the description of one model in terms of concepts borrowed from others. An interesting use of this feature is the algorithm devised by Swendsen and Wang⁽⁸⁾ to simulate the Potts model by means of a cluster Monte Carlo simulation. This method generates non-local spin updates, resulting in a smaller dynamical critical exponent and providing a way to overcome the phenomena of critical slowing-down that happens at second-order phase transitions.

In one way or another, all applications described so far build upon the inversion symmetry possessed by Ising variables. The purpose of this paper is to present a random cluster representation of the Blume–Capel model, a model where inversion symmetries do not represent a full description of the different phases present in the phase-diagram. Not only have we a second-order transition line of the Ising type, but also a first-order phase transition between asymmetric phases. These two lines join at a tricritical point, where three different phases become indistinguishable. How is it possible for a cluster description to capture all these physical features?

In this paper we show that there is indeed a way to map the Blume–Capel model into a percolation model. As could be expected from its complex phase-diagram, this representation is richer than the one for the Ising model, in the sense that there now exists two different kinds of clusters, playing different roles for each type of transition. The tricritical point becomes particularly interesting, since there the different types of fluctuations become equivalent, and the two different clusters should perform similar roles. The mapping we present was inspired in a cluster Monte Carlo algorithm proposed by us⁽⁹⁾ in the more general context of the Blume–Emery–Griffiths model.⁽¹⁰⁾ It is by no means unique and in ref. 11 a different formulation is presented. This situation was already present in the Ising model, where there are also different ways of performing the mapping. This issue is addressed in the conclusions, where a physical criterion for selecting the most appropriate way of doing the mapping is presented.

The paper is organized as follows: in Section 2 we describe the Blume–Capel model and a cluster algorithm for its simulation, Section 3 contains the mapping into a percolation model, and Section 4 relates the mapping with the Monte Carlo algorithm. In Section 5 we discuss theoretical aspects of the map, point some directions for future work and draw the conclusions.

2. BLUME–CAPEL MODEL AND CLUSTER ALGORITHM

The Blume–Capel model was introduced independently by M. Blume⁽¹²⁾ and H. Capel⁽¹³⁾ in 1966 to explain first-order phase transitions not driven by magnetic-field effects. Its main importance nowadays stands for its being one of the simplest generalizations of the Ising model to possess a tricritical point. Its Hamiltonian in temperature units is defined as

$$\beta\mathcal{H} = -K \sum_{\langle ij \rangle} S_i S_j + \Delta \sum_i S_i^2, \quad S_i = \pm 1, 0 \quad (1)$$

where $\beta = 1/k_B T$. Throughout this paper we denote spin values $+1, 0, -1$ by $+, -, 0$, respectively. The first sum runs over nearest neighbors and the second one over all sites on a d -dimensional hypercubic lattice. The first term represents an exchange interaction, as in the Ising model, and in the second term Δ may be interpreted as a cristaline field or as a chemical potential. Its phase-diagram is represented schematically in Fig. 1.

The vertical axis is the temperature. Continuous second-order phase transitions are represented by a solid line, while the dotted one represents first-order transitions. These two lines join at point P , called tricritical because in the presence of a magnetic-field coupling two other second-order lines depart from it. The second order line belongs to the Ising universality class, while the first-order line marks the coexistence of three different phases, two ferromagnetic ones with opposite spin signal, like in the Ising model, and a third paramagnetic one, which at low temperatures is constituted mainly by zero spins. At the tricritical point we expect critical exponents to be mean-field like, except for logarithm corrections.⁽¹⁴⁾

Cluster Monte Carlo simulations are successful in Ising models because they correctly pick up the relevant physics of fluctuations in the model. Since the second-order line of the the Blume–Capel model is of the

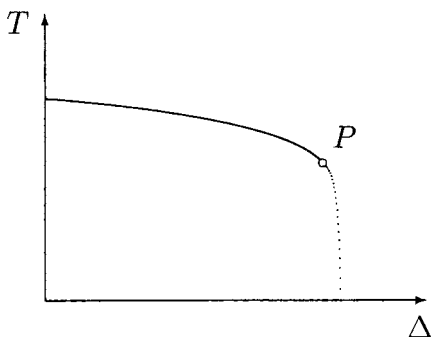


Fig. 1. Phase-diagram of the Blume–Capel model.

Ising type, it is reasonable to expect the same type of cluster to perform well in this line. This is indeed the case, as confirmed by the use of an embedding algorithm.^(15, 11) This kind of algorithm has two different steps, one that makes use of an Ising-like cluster, leaving zero spins unaltered, and one with a Metropolis procedure, to ensure ergodicity.

Although efficient in second-order transitions, this sort of procedure does not address what is happening in the first-order line. There we have a transition between a ferromagnetic phase and a paramagnetic one, where most of the spins are zero. A procedure that leaves the number of zero spins unchanged is clearly unsuitable for this kind of transition, and consequently for simulation near the tricritical point. Furthermore, it is a computational approach, bearing no relationship to a percolation description and being unable to give any of its insights.

In ref. 9 we described a cluster algorithm that can successfully deal with both types of transitions. It is a Wolff type algorithm,⁽¹⁵⁾ in the sense that it generates just one cluster per step, and can be described as follows:

- Choose randomly an initial site for the cluster—the seed.
- Choose with equal probability one of the two spin values different from the seed's one. We will try to change the seed to this new spin value.
- If the transition involved is of the type $\pm \leftrightarrow \mp$, use Wolff's algorithm exactly in the same way as in the Ising model. Connect nearest neighbors spins that are equal to the seed with probability $p = 1 - e^{-2K}$. Repeat this procedure for each new spin connected, and continue until all neighbors to a connected spin have been tested, and no new connection made. Next, identify a cluster with the set of all connected spins, and reverse the sign of all its constituents spins.
- If the transition is of the type $\pm 1 \leftrightarrow 0$ we proceed differently. Two different kinds of spin are allowed in the cluster, spins equal to the seed or spins equal to the value we are trying to change the seed into. For all effects, these two spins are regarded as the same in the linking procedure. The spin value that cannot be linked to the cluster will be called forbidden. The linking procedure is performed in the same way as in the preceding item, but the linking probability now is $p = 1 - e^{-K}$. The fact that we have two different spins values in the cluster only appears when we consider updating the cluster. This is done in the following way: to each spin in the cluster, we add to its spin value the value of the spin forbidden to the cluster. To exemplify, consider that our seed is a + spin, and that we are trying to change it to a 0 spin. The cluster may be made of + and 0 spins, and they will be transformed as $+ \rightarrow 0$ and $0 \rightarrow -$, which amounts to summing $-$, the forbidden spin, to each spin value. The transition is not

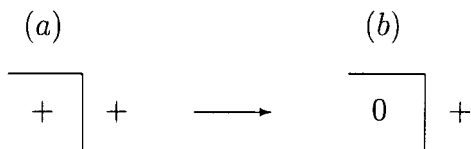


Fig. 2. Possible boundary between a cluster and its surroundings, before (a) and after transition (b).

made automatically, being accepted with probability $P_{\text{flip}} = \exp(-\beta\Delta\mathcal{H}_{\text{bulk}})$ if the bulk energy of the cluster increases, or with probability $P_{\text{flip}} = 1$ if it decreases. $\mathcal{H}_{\text{bulk}}$ is the internal energy of the cluster, calculated disregarding links between spins in the cluster and spins outside it.

A proof that this procedure satisfies detailed balance can be found at ref. 9. For transitions that only change spin signs the proof is exactly the same as in the Wolff algorithm for the Ising Model. An intuitive understanding of why the second step also satisfies detailed balance may be achieved through Fig. 2.

In (a) we represent a possible boundary between a cluster that may contain + and 0 spins (surrounded by a solid line), and in (b) the same boundary after changing the spins that belong to the cluster. It is clear that the ratio between probabilities of not linking the spin with its neighborhood results exactly in the required quantity $\exp(-\beta\Delta\mathcal{H})$. For example, in the (a) configuration the probability of not linking the represented spins is $1 - p = e^{-K}$, while in *b* it is 1, since + is a forbidden spin there. The ratio is exactly what is needed to balance the difference in energy of this particular contour. The flipping probability responds for the energy difference inside the cluster. An important point is that it does not include interactions with spins outside the boundary, whose energy difference after the transition is automatically taken into account when building the cluster. Otherwise, we could get very slow dynamics.

Is this cluster algorithm just a computational approach that enables us to perform an efficient Monte Carlo simulation, or does it reflect a deeper connection between this model and a percolation one? Answering this question amounts to following the inverse track performed in the Ising model, using the cluster algorithm to achieve the random-cluster representation.

As described the algorithm is of the Wolff type, unsuitable for a random-cluster representation. To achieve the latter it is necessary to represent all spins on the lattice into a bond analogue. In the Ising model it is easy to extend the proof of detailed balance from the Wolff algorithm to a Swendsen–Wang one (see ref. 16), since by symmetry when two neighboring

clusters are modified it is as if none of them have changed. Here this symmetry is no longer present, and the proof needs to be carried out in another way. In the next section we show how to perform this task.

3. AN ALGORITHM OF THE SWENDSEN–WANG TYPE

The algorithm just presented has some unusual properties. It is difficult to understand why it is possible to connect zero spins, since there is no linking energy associated with them in the Hamiltonian. Also strange is the fact that in some clusters we may connect two types of spin, but not the third one. This is an asymmetry not present in the Hamiltonian.

Devising a Swendsen–Wang algorithm that generalizes the previous algorithm is also challenging. Some points seem to make such a generalization impossible. For example, the connection between two + spins may be made with two different probabilities, depending on the type of cluster that is being built. But this type is not known in advance, since in a Swendsen–Wang algorithm we first establish connections between spins, and only after we identify clusters. Also, how can we know the two different types of spin allowed in a cluster, before identifying a cluster?

In order to deal with these problems it is convenient to perform the following change of variables:

$$\begin{aligned}\mu_i &= +S_i^2 + S_i - 1 \\ v_i &= -S_i^2 + S_i + 1\end{aligned}\tag{2}$$

which may be represented as in the following table:

S	μ	v
+	+	+
0	-	+
-	-	-

Note that there is not a state such as $\mu = +$, $v = -$. This is a constraint that must be taken into account when writing the partition function and in the development of the cluster algorithm.

From the previous definition we find $S_i = (\mu_i + v_i)/2$, and with this substitution the Hamiltonian becomes

$$\beta\mathcal{H} = -\frac{K}{4} \sum_{\langle ij \rangle} (\mu_i \mu_j + v_i v_j + \mu_i v_j + \mu_j v_i) + \frac{A}{4} \sum_i (\mu_i + v_i)^2\tag{3}$$

$$\begin{array}{cc} \mu_i & \mu_j \\ \nu_i & \nu_j \end{array}$$

Fig. 3. Possible configurations and respective energies.

This is different from the approach followed in ref. 11, where all four different combinations of states $\mu \nu$ are allowed, and the proper statistic weight achieved through a correction term in the partition function. This new approach is necessary in order to re-obtain the previous algorithm, which we believe to be the most appropriate to describe the physical features of the model.

In the absence of cristaline field, it is convenient to distribute the different possible energy configurations as indicated in Fig. 4. We use the representation

$$\begin{array}{cc} \mu_i & \mu_j \\ \nu_i & \nu_j \end{array}$$

for a neighborhood $\langle ij \rangle$, the μ lattice being represented above the ν lattice.

Reading Hamiltonian (3) it may seem natural to define bond probabilities as in the Ising model, for each one of the four linking terms in the Hamiltonian. But this is not possible. If we build clusters that way at the end we would be left with independent clusters in each lattice, what would result in configurations not satisfying the constraint between the

$$\begin{array}{l} u_1 \quad \begin{array}{cc} + & + \\ + & + \end{array} \quad \begin{array}{cc} - & - \\ - & - \end{array} \quad E_1 = -K \\ u_2 \quad \begin{array}{cc} - & - \\ + & + \end{array} \quad E_2 = 0 \\ u_3 \quad \begin{array}{cc} - & - \\ + & - \end{array} \quad \begin{array}{cc} + & - \\ + & + \end{array} \quad E_3 = 0 \\ u_4 \quad \begin{array}{cc} - & - \\ - & + \end{array} \quad \begin{array}{cc} - & + \\ + & + \end{array} \quad E_4 = 0 \\ u_5 \quad \begin{array}{cc} + & - \\ + & - \end{array} \quad \begin{array}{cc} - & + \\ - & + \end{array} \quad E_5 = K \end{array}$$

Fig. 4. Possible configurations and respective energies.

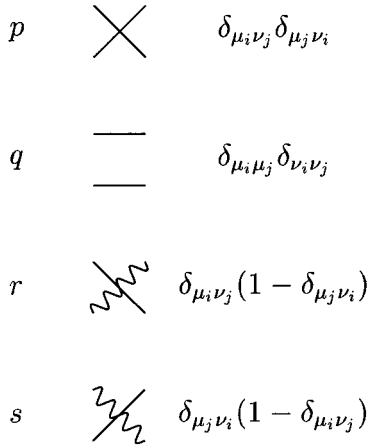
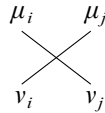


Fig. 5. Possible bond operations.

variables. The correct approach is to define bond probabilities as in Fig. 5. In this figure, a wavy line represents a bond between spins with different signals, while a solid one a bond between equal spins. For example, a p bond between a neighborhood $\langle ij \rangle$ in lattices μ and ν would be represented as









We are left now with the problem of devising a cluster algorithm that uses these bond probabilities and satisfies detailed balance. To find the right value for the bond probabilities we make use of a general scheme devised by Kandel and Domany.⁽¹⁷⁾ It modifies the original Hamiltonian through two different types of operations, called freeze and delete operations, in such a way that the cluster algorithm becomes natural in terms of the new Hamiltonian. These operations are defined in such a way that the final algorithm satisfies detailed balance.

To implement the procedure, it is useful to visualize the different bond probabilities as in Table I. The u_i are the energy configurations represented in Fig. 3, and for each one of them we can perform different bond probabilities. $P_d(u_i)$ represents the probability that no bond is established in a given configuration. The bond operations p , q , r and s are the freezing ones, while the $P_d(u_i)$ are deletion operations.

The point is that in order that the cluster algorithm satisfies the detailed balance condition it is sufficient to require that the sum of probabilities of performing any operation in a given configuration to be

Table I. Freeze and Deletion Probabilities for Different Energy Configurations

						
(u_1)	pq	$p(1-q)$	$q(1-p)$	0	0	$P_d(u_1)$
(u_2)	0	0	q	0	0	$P_d(u_2)$
(u_3)	0	0	0	r	0	$P_d(u_3)$
(u_4)	0	0	0	0	s	$P_d(u_4)$
(u_5)	0	0	0	0	0	$P_d(u_5) = 1$

one (normalization condition) and to link all deletion probabilities through the relation

$$P_d(u_i) = e^{E_i - C_d} \tag{4}$$

where E_i is the energy of a particular configuration u_i and C_d is a constant independent of this particular configuration. The motivation for choosing $P_d(u_5) = 1$ comes from the fact that this is the configuration of highest energy, and we do not want to freeze it. With this choice, $C_d = K$, and all other deletion probabilities are also defined. The value of the freezing probabilities come from the normalization condition, and we find

$$P_d(u_1) = e^{-2K}, \quad P_d(u_2) = P_d(u_3) = P_d(u_4) = e^{-K} \tag{5}$$

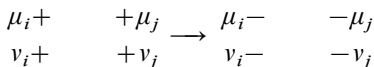
and

$$p = q = r = s = 1 - e^{-K} \tag{6}$$

This procedure obeys detailed balance, as can be easily checked by taking examples of specific transitions. But is this algorithm equivalent to the one devised for the Hamiltonian in terms of spin 1 variables?

At first sight it is not. In the previous algorithm we had two different bond probabilities, $p = 1 - e^{-2K}$ for clusters of the Wolff type and $p = 1 - e^{-K}$ for clusters constituted by two different types of spin. In the new one there is just one bond probability, and no sign of forbidden spins to a specific cluster.

But the algorithms are indeed similar, as can be confirmed by considering each one of the different transition possibilities. In a Wolff transition between a cluster made of two + spins and a cluster of two - spins the probability of linking spins in the cluster is given by $p = 1 - e^{-2K}$. In the new variables μ and ν this transition is represented by



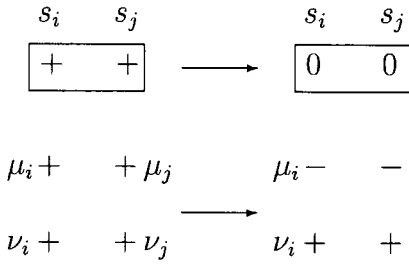


Fig. 6. Transition that changes spins modulo represented in S variables and in μ, ν variables.

and may be done in three different ways,

$$\begin{array}{c} \overline{\times} \\ \underline{\times} \end{array} \text{ or } \begin{array}{c} \times \\ \times \end{array} \text{ or } \begin{array}{c} \text{---} \\ \text{---} \end{array} \tag{7}$$

Consequently, the probability of performing this transition is

$$\mathcal{P} = pq + p(1 - q) + q(1 - p) = 1 - e^{-2K} \tag{8}$$

and we see that probabilities in the two representations are indeed the same.

What about a transition that changes the spin modulo in the cluster? It is represented schematically in Fig. 6.

In the μ, ν representation the way of performing this type of transition is through links of type

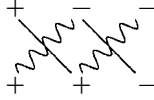
$$\begin{array}{c} \text{---} \\ \text{---} \end{array} \tag{9}$$

In u_1 (see Table I) this probability is given by $P(u_1) = (1 - e^{-K}) e^{-K}$, while in u_2 it is $P(u_2) = (1 - e^{-K})$. The additional e^{-K} term refers to the change in the internal energy of the cluster. Here it is taken into account at each linking step, while in the original algorithm it was considered only through the flipping probability. The way we take into account the difference in the internal energy is just a matter of choice and convenience.

The last issue refers to the presence of forbidden spins. In the new formulation this concept is no longer present, but occurs in an indirect way. Consider the situation when the three different kinds of spin are present in the same cluster, a situation that never happens in the spin S formulation. The only way to connect spins with opposite signals is through one (or more) zero spins, like the following:

$$\begin{array}{|c|c|c|}
 \hline
 S_i & S_j & S_k \\
 \hline
 + & 0 & - \\
 \hline
 \end{array}$$

In the μ, ν representation the bonds for this cluster must be given by



The only spin configuration compatible with this configuration of bonds is the one represented above, and there is no spin transition associated with it. Thus, every time we build a cluster that contains three different types of spins they will not change, and this explains the role of forbidden spins. It is not that we cannot build clusters with them, but only that if we have made that then all spins would have remained unchanged. It is better not to build them at all.

The cristaline field can be easily introduced, because its only relevant contribution will be through the term $\exp((\Delta/2) \mu_i \nu_i)$, representing anti-ferromagnetic interactions between spins μ_i and ν_i . Thus we can link spins $S=0$ throughout a bond probability between opposite spins μ, ν given by $p = 1 - e^{-\Delta}$. Due the the constraint between variables, there is only one configuration compatible with this bond. Again, every time we make such a connection all spins in the cluster remain the same. In order that a transition in the spin modulo to be possible it is necessary that no bond between zero spins is present. The probability for this to happen is $e^{-\Delta N_0}$, where N_0 is the number of zero spins in the cluster. Once more, in the original algorithm this term was considered in the flipping probability, and again the two procedures are the same.

We have concluded the development of a Swendsen–Wang like version of the algorithm previously proposed. The existence of such an algorithm indicates that the model may be mapped into a percolation one, and in the next section we make use of the ideas introduced in this section to derive the mapping.

4. RANDOM-CLUSTER REPRESENTATION

The existence of a Swendsen–Wang like algorithm indicates that it should be possible to find a map into a percolation model. In this type of algorithm new bond variables are added to the original spin configuration. Initial spin values are then disregarded, leaving a configuration where only

bonds are present. New spin values compatible with this given bond configuration are then chosen. The intermediate step where only bond variables are considered shows that it is possible to represent the model only in terms of bonds, which should correspond to a percolation model. In what follows, an explicit formula for the partition function of the percolation model is derived.

In the absence of cristaline field, the partition function of the Blume–Capel model in terms of variables μ , ν is given by

$$\mathcal{Z} = \sum_{\{\mu, \nu\}}^* \prod_{\langle ij \rangle} \exp \left[\frac{K}{4} \mu_i \mu_j + \frac{K}{4} \nu_i \nu_j \right] \exp \left[\frac{K}{4} \mu_i \nu_j + \frac{K}{4} \mu_j \nu_i \right] \quad (10)$$

where the star indicates that the sum must be carried out only over states that obey the constraint between the variables. It may be rewritten as

$$\mathcal{Z} = \sum_{\{\mu, \nu\}}^* \mathcal{A} \mathcal{B} \quad (11)$$

where

$$\begin{aligned} \mathcal{A} = e^{K/2} [& (1 - e^{-K}) \delta_{\mu_i \mu_j} \delta_{\nu_i \nu_j} + (e^{-K/2} - e^{-K}) \delta_{\mu_i \mu_j} (1 - \delta_{\nu_i \nu_j}) \\ & + (e^{-K/2} - e^{-K}) \delta_{\nu_i \nu_j} (1 - \delta_{\mu_i \mu_j}) + e^{-K}] \end{aligned} \quad (12)$$

and

$$\begin{aligned} \mathcal{B} = e^{K/2} [& (1 - e^{-K}) \delta_{\mu_i \nu_j} \delta_{\mu_j \nu_i} + (e^{-K/2} - e^{-K}) \delta_{\mu_i \nu_j} (1 - \delta_{\mu_j \nu_i}) \\ & + (e^{-K/2} - e^{-K}) \delta_{\mu_j \nu_i} (1 - \delta_{\mu_i \nu_j}) + e^{-K}] \end{aligned} \quad (13)$$

Using now symbolic notation from Fig. 5, and introducing the symbols

$$\begin{array}{cc} \overline{\underbrace{\quad}} \delta_{\nu_i \nu_j} (1 - \delta_{\mu_i \mu_j}) & \overline{\underbrace{\quad}} \delta_{\mu_i \mu_j} (1 - \delta_{\nu_i \nu_j}) \\ \underbrace{\quad} & \underbrace{\quad} \end{array}$$

we may rewrite \mathcal{A} as

$$\begin{aligned} \mathcal{A} = e^{K/2} \left[& (1 - e^{-K}) \overline{\underbrace{\quad}} + e^{-K} + e^{-K/2} (1 - e^{-K/2}) \underbrace{\quad} \\ & + e^{-K/2} (1 - e^{-K/2}) \underbrace{\quad} \right] \end{aligned} \quad (14)$$

or

$$\mathcal{A} = e^K \left[\underbrace{(1 - e^{-K}) \text{---} + e^{-K}}_{A1} \right] \left[e^{-K/2} + (1 - e^{-K/2}) \left(\text{---} + \text{wavy} \right) \right] \quad (15)$$

Similarly, \mathcal{B} may be written as

$$\mathcal{B} = e^K \left[\underbrace{(1 - e^{-K}) \text{X} + e^{-K}}_{B1} \right] \left[e^{-K/2} + (1 - e^{-K/2}) \left(\text{wavy} + \text{zigzag} \right) \right] \quad (16)$$

The terms $A1$ and $B1$ are already in a convenient form. Due to the constraint between variables,

$$\text{---} + \text{wavy} = \text{zigzag} + \text{zigzag}$$

meaning that configurations compatible with this set of bonds are the same. We conclude that $A2 = B2$, and rewriting in convenient way,

$$A2 = e^{K/2} \left[e^{-K/2} + (1 - e^{-K/2}) \text{zigzag} \right] \left[e^{-K/2} + (1 - e^{-K/2}) \text{zigzag} \right] \quad (17)$$

We may then rewrite the partition function as

$$\mathcal{Z} = \sum_{\{\mu, \nu\}}^* \prod_{\langle ij \rangle} e^{3K} \left[(1 - e^{-K}) \text{---} + e^{-K} \right] \left[(1 - e^{-K}) \text{X} + e^{-K} \right] \\ \times \left[(1 - e^{-K}) \text{zigzag} + e^{-K} \right] \left[(1 - e^{-K}) \text{zigzag} + e^{-K} \right] \quad (18)$$

Disregarding constant terms, the crystalline field is introduced by multiplying the above partition function by $\prod_i \exp[(-A/2) \mu_i \nu_i]$, which in symbolic notation becomes

$$\prod_i \left[(1 - e^{-A}) \text{zigzag} + e^{-A} \right] \quad (19)$$

where $\text{zigzag} \equiv (1 - \delta_{\mu_i \nu_i})$ was introduced to represent bonds between spins μ_i and ν_i with opposite signals. This type of bond will be indexed by the label t .

The complete partition function, up to constant factors, is given then by

$$\begin{aligned} \mathcal{Z} = & \sum_{\{\mu, \nu\}}^* \prod_{\langle ij \rangle} \left\{ \left[(1 - e^{-K}) \overline{\quad} + e^{-K} \right] \left[(1 - e^{-K}) \times + e^{-K} \right] \right. \\ & \times \left[(1 - e^{-K}) \text{zigzag} + e^{-K} \right] \left. \left[(1 - e^{-K}) \text{zigzag} + e^{-K} \right] \right\} \\ & \times \prod_i \left\{ \left[(1 - e^{-A}) \text{zigzag} + e^{-A} \right] \right\} \end{aligned} \quad (20)$$

It is still written only in terms of spin variables, but bond variables may now be easily introduced. Using the labels p , q , r and s for bond variables defined in Fig. 5, the t label for crystalline field interactions, and considering that each label may assume two different values, 0 for absence of bond and 1 for its presence, we rewrite the partition function as

$$\begin{aligned} \mathcal{Z} = & \sum_{\{\mu, \nu\}}^* \prod_{\langle ij \rangle} \left\{ \sum_{q=0}^1 \left[(1 - e^{-K}) \overline{\quad} \delta_{q1} + e^{-K} \delta_{q0} \right] \right\} \\ & \times \left\{ \sum_{p=0}^1 \left[(1 - e^{-K}) \times \delta_{p1} + e^{-K} \delta_{p0} \right] \right\} \\ & \times \left\{ \sum_{r=0}^1 \left[(1 - e^{-K}) \text{zigzag} \delta_{r1} + e^{-K} \delta_{r0} \right] \right\} \\ & \times \left\{ \sum_{s=0}^1 \left[(1 - e^{-K}) \text{zigzag} \delta_{s1} + e^{-K} \delta_{s0} \right] \right\} \\ & \times \left\{ \prod_i \sum_{t=0}^1 \left[(1 - e^{-A}) \text{zigzag} \delta_{t1} + e^{-A} \delta_{t0} \right] \right\} \end{aligned} \quad (21)$$

or else,

$$\begin{aligned} \mathcal{Z} = & \sum_{\{\mu, \nu\}}^* \sum_{\{q, p, r, s, t\}} \left\{ \prod_{\langle ij \rangle} \left[(1 - e^{-K}) \overline{\quad} \delta_{q1} + e^{-K} \delta_{q0} \right] \right. \\ & \times \left[(1 - e^{-K}) \times \delta_{p1} + e^{-K} \delta_{p0} \right] \left. \left[(1 - e^{-K}) \text{zigzag} \delta_{r1} + e^{-K} \delta_{r0} \right] \right. \\ & \times \left. \left[(1 - e^{-K}) \text{zigzag} \delta_{s1} + e^{-K} \delta_{s0} \right] \right\} \left\{ \prod_i \left[(1 - e^{-A}) \text{zigzag} \delta_{t1} + e^{-A} \delta_{t0} \right] \right\} \end{aligned} \quad (22)$$

where $\sum_{\{q, p, r, s, t\}}$ denotes a sum over all possible bond configurations.

Considering now that each possible bond configuration constitutes a graph G' , and that the set of all possible bonds configurations is G , we rewrite the above expression as

$$\begin{aligned}
 \mathcal{Z} = & \sum_{\{\mu, \nu\}}^* \sum_{G' \subseteq G} \left[\prod_{\langle ij \rangle: q=1} (1 - e^{-K}) \text{---} \right] \left[\prod_{\langle ij \rangle: q=0} e^{-K} \right] \\
 & \times \left[\prod_{\langle ij \rangle: p=1} (1 - e^{-K}) \text{X} \right] \left[\prod_{\langle ij \rangle: p=0} e^{-K} \right] \\
 & \times \left[\prod_{\langle ij \rangle: r=1} (1 - e^{-K}) \text{~} \right] \left[\prod_{\langle ij \rangle: r=0} e^{-K} \right] \\
 & \times \left[\prod_{\langle ij \rangle: s=1} (1 - e^{-K}) \text{~} \right] \left[\prod_{\langle ij \rangle: s=0} e^{-K} \right] \\
 & \times \left[\prod_{i: t=1} (1 - e^{-A}) \text{~} \right] \left[\prod_{i: t=0} e^{-A} \right] \tag{23}
 \end{aligned}$$

This is the joint partition function we referred to in the beginning of the section. In order to find an equivalent percolation model, we must sum over spin states.

The first step in this direction is to identify clusters as sets of spins joined by a continuous path of bonds. In the Ising model all spins connected to a cluster must have the same sign, but here the situation is far more complicated, since due to the r and s bonds all three types of spin (in the S representation) may appear in the same cluster. Furthermore, the number of states compatible to a given configuration depends on the number of different spins belonging to the cluster, and the type of bond present. A cluster may be constituted by spins on both lattices, μ and ν , or by spins in just one of them (when constituted only by q bonds, or no bond at all). Each possible case has to be analysed separately.

The simplest case is when only one type of spin appears in the cluster. There are two different possibilities, depending on whether there is a p bond present or not. If it is present, we may have only $+$ or $-$ spins in the cluster, whilst if it is not 0 spins are also possible.

The second possibility is a cluster made only of two different types of spin. Here we can match two spin configurations to a given bond configuration, for example,

$$\begin{array}{ccc}
 \mu & + \text{---} + & - \\
 & & \text{~} \\
 \nu & + \text{---} + & +
 \end{array} \tag{24}$$

or

$$\begin{array}{ccc}
 \mu & - & - & - \\
 & \text{---} & & \\
 & & \diagdown & / \\
 \nu & + & + & - \\
 & \text{---} & & \\
 & & \diagup & \backslash
 \end{array} \tag{25}$$

Finally, a cluster may have all three different types of spin, with zero spins providing a bridge between spins with different signs. For this type of cluster there is a unique correspondence between spin and bond configurations. An example is given in (26).

$$\begin{array}{ccccccc}
 \mu & + & + & - & - & - & - \\
 & \text{---} & & & & & \\
 & & \diagdown & / & \diagdown & / & \\
 \nu & + & + & + & - & - & - \\
 & \text{---} & & & & & \\
 & & \diagup & \backslash & \diagup & \backslash &
 \end{array} \tag{26}$$

Spins of different signs may get connected to the same cluster only through zero spins, and these connections are always through r or s bonds. Spins zero work like a glue that binds the cluster together.

It is useful now to introduce the concept of a spin island. We may see the cluster as a collection of different blocks, each block containing spins with the same value. Each one of these blocks we call a spin island. In (26) we had three different islands: one of $+$ spins, one of $-$ spins, and a 0 spin island connecting these two islands of opposite signs.

Note that permuting all r and s in a given configuration results in a change of sign of spins belonging to all islands constituted by non-zero spins. For example, permuting r and s bonds in the last figure would result in the following configuration:

$$\begin{array}{ccccccc}
 \mu & - & - & - & + & + & + \\
 & \text{---} & & & & & \\
 & & \diagdown & / & \diagdown & / & \\
 \nu & - & - & + & + & + & + \\
 & \text{---} & & & & & \\
 & & \diagup & \backslash & \diagup & \backslash &
 \end{array} \tag{27}$$

Let us now sum up over spin configurations taking into account the more general Hamiltonian where there is also a magnetic field present in the form $B \sum_i S_i = (B/2) \sum_i (\mu_i + \nu_i)$. Its contribution to the partition function is $\prod_i \exp[(B/2)(\mu_i + \nu_i)]$. Our strategy consists in writing every possible cluster contribution in a hyperbolic cosine form, in the same way as in the Ising model.⁽¹⁸⁾

When the cluster is made of just one type of spin we have two different contributions, as explained before. When p bonds are present each of these clusters give a contribution

$$(e^{Bn} + e^{-Bn}) \tag{28}$$

whilst when they are absent the contribution is

$$(1 + e^{Bn} + e^{-Bn}) \tag{29}$$

since now the cluster may also be constituted by zero spins. In the above expressions n is the number of spins in an island, which in this case corresponds to the whole cluster.

When there are two different types of spin in the cluster, two different spin configurations are compatible with a given set of bonds, but they do not contribute with a cosh. This is the reason why we introduced the island concept: if we permute all r and s bonds present, the sign of all spins in the cluster will flip, resulting in the desired cosh contribution. After permutations, the contribution of this type of cluster is of the form

$$\left[\prod_{i1} (e^{Bn} + e^{-Bn}) \right] + \left[\prod_{i0} (e^{Bn_0} + e^{-Bn_0}) \right] \tag{30}$$

where the first product is over all islands of spins $+$ or $-$, while the second one is over 0 spin islands. By n_0 we mean the number of zero spins in this type of island. If a t -like bond is present, the second product will not appear, since zero spins would be frozen.

When the cluster has three spin types there is only one possible spin configuration. However, as before, if we add $r \leftrightarrow s$ permutations we again get a hyperbolic cosine. The resulting contribution is

$$\left[\prod (e^{Bn} + e^{-Bn}) \right] \tag{31}$$

where the product is over all spin islands.

Let us introduce now the following notation for the part of the partition function that generates bonds:

$$\begin{aligned} \Pi(G', p) = & \left[\prod_{\langle ij \rangle: q=1} (1 - e^{-K}) \text{---} \right] \left[\prod_{\langle ij \rangle: q=0} e^{-K} \right] \\ & \times \left[\prod_{\langle ij \rangle: p=1} (1 - e^{-K}) \text{X} \right] \left[\prod_{\langle ij \rangle: p=0} e^{-K} \right] \\ & \times \left[\prod_{\langle ij \rangle: r=1} (1 - e^{-K}) \text{~} \right] \left[\prod_{\langle ij \rangle: r=0} e^{-K} \right] \\ & \times \left[\prod_{\langle ij \rangle: s=1} (1 - e^{-K}) \text{~} \right] \left[\prod_{\langle ij \rangle: s=0} e^{-K} \right] \\ & \times \left[\prod_{i: t=1} (1 - e^{-A}) \text{~} \right] \left[\prod_{i: t=0} e^{-A} \right] \end{aligned} \tag{32}$$

Joining all different contributions and summing over spin states the partition function becomes

$$\begin{aligned} \mathcal{Z} = & \sum_{G' \subseteq G}^{per} \Pi(G', p) \left[\prod_A (e^{Bn} + e^{-Bn}) \right] \left[\prod_C (1 + e^{Bn} + e^{-Bn}) \right] \\ & \times \left[\prod_D (e^{Bn} + e^{-Bn}) + \prod_E (e^{Bn_0} + e^{-Bn_0}) \right] \end{aligned} \quad (33)$$

where the *per* symbol over the sum means that configurations that differ by a permutation of all *r* and *s* bonds should be summed only once. As before, the constraint between variables must be obeyed. The *A* product is over spin islands that belong to any cluster where there are *p* or *t* bonds present, or to clusters with three different types of spin. The *C* product runs over clusters made only of *q* type bonds. *D* and *E* products belong to clusters with two different spin types, the first one being over islands of spins +1 or -1 and the second one over zero-spin islands.

This is the partition function of the corresponding percolation model. In order to reobtain the original partition function it is necessary to reintroduce spin variables obeying the original constraint between variables and are compatible with a given bond configuration. Summing then over bond configurations would bring us back to the Blume-Capel model. Note, however, that some bond configurations are not compatible with a spin configuration, due to the constraint between variables. This is the case of configurations that have a consecutive set of *ppp* or *psp*. These kind of configurations have zero statistical weight.

It is possible now to compute physical quantities in terms of properties of the clusters generated by the percolation model. The magnetization is given by

$$M(G, p) = \lim_{B \rightarrow 0^+} \lim_{N \rightarrow \infty} \frac{1}{N} \frac{\partial}{\partial B} \ln \mathcal{Z} \quad (34)$$

or

$$\begin{aligned} M = & \lim_{B \rightarrow 0^+} \lim_{N \rightarrow \infty} \frac{1}{N \mathcal{Z}} \sum_{G' \subseteq G}^{per} \Pi(G', p) \frac{\partial}{\partial B} \\ & \times \left\{ \left[\prod_A (e^{Bn} + e^{-Bn}) \right] \left[\prod_C (1 + e^{Bn} + e^{-Bn}) \right] \right. \\ & \left. \times \left[\prod_D (e^{Bn} + e^{-Bn}) + \prod_E (e^{Bn_0} + e^{-Bn_0}) \right] \right\} \end{aligned} \quad (35)$$

The hyperbolic sine that results from the derivative will be zero in the limit $B \rightarrow 0$, unless the cluster percolates. When crossing the second-order line from the paramagnetic region, the first cluster to percolate comes from the A product, since they may contain a p bond, while others may not. As a consequence, others clusters have a substantially smaller bond probability. Using this hypothesis, we find that the magnetization through the second-order line is given by

$$M = \lim_{N \rightarrow \infty} \frac{1}{N \mathcal{Z}} \sum_{G' \subseteq G}^{per} \Pi(G', p) N_A 2^{n_a-1} 3^{n_c} [2^{n_d} + 2^{n_e}] \quad (36)$$

where N_A is the number of spins in a type A cluster that percolates, n_a is the number of A clusters that do not percolate and so on. To derive this expression it is assumed that only type- A clusters are percolating. Redefining

$$\Pi^*(G', p) = \Pi(G', p) 2^{n_a} 3^{n_c} [2^{n_d} + 2^{n_e}] \quad (37)$$

and considering that in the limit $\lim_{N \rightarrow \infty} n_a - 1 \sim n_a$, we rewrite the magnetization as

$$M = \lim_{N \rightarrow \infty} \frac{1}{\mathcal{Z}} \sum_{G' \subseteq G}^{per} \Pi^*(G', p) \frac{N_A}{N} \quad (38)$$

which shows that the magnetization is the probability that a spin belongs to a cluster that percolates.

The magnetic susceptibility is given by

$$\chi(G, p) = \lim_{B \rightarrow 0^+} \lim_{N \rightarrow \infty} \frac{\partial^2}{\partial B^2} \ln \mathcal{Z} \quad (39)$$

We find a similar expression to the one found for the Ising Model⁽¹⁸⁾:

$$\begin{aligned} \chi(G, p) = & \lim_{N \rightarrow \infty} \mathcal{Z}^{-1} \sum_{G' \subseteq G} \Pi(G', p) \sum_c^f \frac{n_c^2(G')}{N} \\ & + \lim_{N \rightarrow \infty} \mathcal{Z}^{-2} \sum_{G' \subseteq G} \sum_{G'' \subseteq G} \Pi(G', p) \Pi(G'', p) \frac{[N_A(G') - N_A(G'')]^2}{2N} \end{aligned} \quad (40)$$

where \sum_c^f is over all islands that do not percolate, including islands that do not result from the A product. The first term in this expression is the mean cluster size, and similarly to the Ising model we may argue that the second

term contribution remains finite. The conclusion is that the susceptibility is the mean size of islands built by the algorithm.

The expression for the correlation function between spins is

$$\langle \sigma_a \sigma_b \rangle = \mathcal{Z}^{-1} \sum_{\{\mu\nu\}} \Pi(G', p) \left[\prod_A (e^{Bn} + e^{-Bn}) \right] \cdot \left[\prod_C (1 + e^{Bn} + e^{-Bn}) \right] \cdot \left[\prod_D (e^{Bn} + e^{-Bn}) + \prod_E (e^{Bn_0} + e^{-Bn_0}) \right] \sigma_a \sigma_b \quad (41)$$

If σ_a and σ_b belong to the same island $\sigma_a \sigma_b = 1$, otherwise $\sigma_a \sigma_b = \pm 1$ with the same probability and the contribution of this graph G' to the sum of configurations is zero. The correlation function is given then by

$$\langle \sigma_a \sigma_b \rangle = \lim_{N \rightarrow \infty} \mathcal{Z}^{-1} \left[\sum_{G' \subseteq G}^{per} \Pi^*(G', p) \delta(a, b) \right] \quad (42)$$

where

$$\delta(a, b) = \begin{cases} 1 & \text{if } a \text{ and } b \text{ are in the same island} \\ 0 & \text{if } a \text{ and } b \text{ are in different islands} \end{cases} \quad (43)$$

The last expression is the pair-connectedness function of the percolation model, when spin islands generated by different clusters are taken into account. This result shows that the exponent ν_p relative to the correlation length when $p \rightarrow p_c$ is the same as the ν exponent of the Blume–Capel model. Similar results may be deduced for the energy and specific-heat exponents.

Clusters that control the second-order transition line are type A clusters or, more specifically, clusters constituted by p or q bonds. These clusters are built with probability $p = 1 - e^{-2K}$, as in the Ising Model. A consequence is that second-order transitions in both models belong to the same universality class. Although the lattice may be substantially different due to the presence of zero spins, the above results show that the mechanism controlling the transition is basically the same.

An interesting point is that the relevant percolation properties are connected to the size of islands belonging to a cluster, rather than to the cluster itself.

We have described so far the second-order transition line in terms of percolation properties. Where does the first-order transition line and the tricritical point fit in the scheme? The key point is that although second-order transitions are always controlled by Ising like clusters, transitions between

spin's modulo will have the extra help of t bonds. These bonds fix zero spins into the lattice, and so it is to be expected that as the cristaline field increases the number of zero spins also increase, despite the fact that bond probabilities for these clusters are smaller than for Ising clusters. At a certain point, Ising and zero-spin clusters will compete on the same footing, and this is the tricritical point. If we keep increasing the t probabilities (by increasing the cristaline field) clusters that result from a change of the spin's modulo will predominate, resulting in the first-order line.

5. CONCLUSIONS

The mapping just presented is not unique. Reference 11 presents a different approach, where as in this paper the mapping is done through a decomposition into Ising variables, but the constraint between variables is dealt with a correction term in the partition function. However, the authors were interested only in the case $D = \ln 2$, a special point where the scheme proved efficient. It is also possible to simulate the model using an embedding algorithm that alternates Ising cluster steps with Metropolis ones, to ensue ergodicity.

Our own simulations show that both procedures are sufficiently good only on the second-order line. On the first-order transition line both procedures are not sufficient to eliminate hysteresis effects, and the simulation close to the tricritical point becomes problematic. The reason why an embedding algorithm does not perform well is clear: it fails to incorporate the necessary transitions that change spin modulus. The reasons for the failure of the other algorithm are more subtle. Although it does generate transitions between spin's modulo, it fails to generate the appropriate clusters, in the sense that clusters generated will be too large and more likely will not be updated, due to a transition probability associated with the internal energy.

The algorithm dynamics is very sensitive to this kind of balance, it is very ease to formulate a cluster algorithm whose clusters do not move at all. The good performance of our algorithm (tested in ref. 9) is due to the fact that it does generate the appropriate clusters, in the sense that it is possible to relate thermodynamics properties like magnetization in terms of cluster properties. This is an important criterion when analysing more complicated models where frustration plays a significant role.

Unfortunately, we still have not been able to measure thermodynamical properties directly from the percolation model, for example, magnetization as a probability of a given spin to belonging to a percolating cluster. The percolation model obtained is too complicated to be used as a Swendsen–Wang type algorithm, since it would require our keeping track of individual

properties of each cluster, like the number of zero spins. The use of a Wolff-type algorithm that generates just one cluster per step is more efficient.

The possibility of representing a model that has a tricritical point in a percolation formulation is also very promising from a theoretical point of view, since up to now only models where Ising-like symmetries play a predominant role have been mapped. The existence of two competing clusters at the tricritical point is a very interesting feature of the model. In this paper we have not explored further these possibilities from a more quantitative point of view, keeping it at a more qualitative level. A quantitative analysis may be quite fruitful, providing a different approach to analyse effects like finite-size scale, which has been quite controversial in asymmetric first-order transitions. Also appealing is the possibility of generating FKG inequalities, and being able to characterize the tricritical point in terms of percolation concepts.

ACKNOWLEDGMENTS

The authors thank FAPESP for financial support.

REFERENCES

1. P. W. Kastelyn and C. M. Fortuin, *J. Phys. Soc. Japan Suppl.* **26**:11 (1969).
2. C. M. Fortuin and P. W. Kastelyn, On the random-cluster model, *Physica* **57**:536 (1972).
3. C. M. Fortuin, P. W. Kastelyn, and J. Ginibre, Correlation inequalities on some partially ordered sets, *Commun. Math. Phys.* **22**:89 (1971).
4. D. Ioffe, Exact large deviation bounds up to T_c for the Ising model in two dimensions, *Probab. Theory Relat. Fields* **102**:313 (1995).
5. A. Pisztora, Surface order deviations for the Ising, Potts and percolation models, *Probab. Theory Relat. Fields* **104**:427 (1996).
6. M. Aizenman, J. T. Chayes, L. Chayes, and C. M. Newman, Discontinuity of the magnetization in one-dimensional $1/|x-y|^2$ Ising and Potts model, *J. Stat. Phys.* **50**:1 (1988).
7. C.-E. Pfister and Y. Velenik, Random-cluster representation of the Ashkin–Teller model, *J. Stat. Phys.* **88**:1295 (1997).
8. R. H. Swendsen and J. S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, *Phys. Rev. Lett.* **58**:86 (1987).
9. M. B. Bouabci and C. E. I. Carneiro, Eliminating metastability in first-order phase transitions, *Phys. Rev. B* **54**:359 (1996).
10. M. Blume, V. J. Emery, and R. B. Griffiths, Ising model for the λ transition and phase separation in $\text{He}^3\text{-He}^4$ mixtures, *Phys. Rev. A* **4**:1071 (1971).
11. H. W. J. Blöte, E. Luijten, and J. R. Heringa, Ising universality in three dimensions: A Monte Carlo study, *J. Phys. A* **28**:6289 (1995).
12. M. Blume, Theory of the first-order magnetic phase change in UO_2 , *Phys. Rev.* **141**:517 (1966).
13. H. W. Capel, On the possibility of first-order phase-transitions in Ising systems of triplet ions with zero-field, *Physica* **32**:966 (1966).

14. I. D. Lawrie and S. Sarbach, Theory of tricritical points, in *Phase Transitions and Critical Phenomena*, J. L. Lebowitz and C. Domb, eds. (Academic, London, 1984), Vol. 8.
15. U. Wolff, Collective Monte Carlo updating for spin systems, *Phys. Rev. Lett.* **62**:361 (1989).
16. G. T. Barkema and M. E. J. Newman, New Monte Carlo algorithms for classical spin systems, in *Monte Carlo Methods in Chemical Physics* (Wiley, New York, 1997).
17. D. Kandel and E. Domany, General cluster Monte Carlo dynamics, *Phys. Rev. B* **43**:8539 (1991).
18. C.-K. Hu, Percolations, clusters, and phase transitions in spin models, *Phys. Rev. B* **29**:5103 (1984).