

# Generalization of the Fortuin–Kasteleyn Transformation and Its Application to Quantum Spin Simulations

N. Kawashima<sup>1</sup> and J. E. Gubernatis<sup>1</sup>

*Received October 21, 1994*

---

We generalize the Fortuin–Kasteleyn (FK) cluster representation of the partition function of the Ising model to represent the partition function of quantum spin models with an arbitrary spin magnitude in arbitrary dimensions. This generalized representation enables us to develop a new cluster algorithm for the simulation of quantum spin systems by the worldline Monte Carlo method. Because the Swendsen–Wang algorithm is based on the FK representation, the new cluster algorithm naturally includes it as a special case. As well as the general description of the new representation, we present an illustration of our new algorithm for some special interesting cases: the Ising model, the antiferromagnetic Heisenberg model with  $S=1$ , and a general Heisenberg model. The new algorithm is applicable to models with any range of the exchange interaction, any lattice geometry, and any dimensions.

---

**KEY WORDS:** Quantum Monte Carlo; cluster algorithm;  $XXZ$  model; Heisenberg model;  $XY$  model.

## 1. INTRODUCTION

In 1987, Swendsen and Wang<sup>(1)</sup> used the Fortuin and Kasteleyn (FK) representation<sup>(2)</sup> of the partition function of Ising models in a Monte Carlo simulation of these models. With this representation, they were able to produce an algorithm whose key feature was the global updating of Ising spin configurations in contrast to the local updating in the standard Metropolis algorithm. With global updating, their cluster algorithm greatly reduced the autocorrelation times in the simulation near a critical point. Since then, several attempts have been made (see ref. 3 for a brief review) to reduce autocorrelation times of various systems by various forms of

---

<sup>1</sup> Center for Nonlinear Studies and Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545.

cluster algorithms, and recently the construction of a cluster algorithm has been formulated on more general grounds.<sup>(4-7)</sup> Still, most applications of cluster algorithms have been restricted to classical models.

In this paper, we discuss cluster algorithms for worldline Monte Carlo (WLMC) simulations of general classes of quantum spin systems. For most quantum spin systems, exact knowledge of the properties of the systems is very restricted. To obtain information of a wider range, one often resorts to numerical methods such as exact diagonalization, expansions with respect to small parameters, and quantum Monte Carlo methods. Among these, only quantum Monte Carlo methods are available for relatively large systems. Several variants of the quantum Monte Carlo simulation exist: the Green's function Monte Carlo (GFMC) method,<sup>(8)</sup> the projector Monte Carlo method,<sup>(9)</sup> the auxiliary-field method based on Hubbard-Stratonovich transformation,<sup>(10)</sup> Handscomb's method,<sup>(11)</sup> and the worldline Monte Carlo method based on the Suzuki-Trotter (ST) approximation.<sup>(12)</sup> For the study of ground-state properties, the GFMC is particularly useful. The auxiliary-field method is powerful for Hubbard models and related problems. Handscomb's method lacks the systematic error caused by the ST approximation. The WLMC has enjoyed a wide range of applicability to fermion, boson, and quantum spin models. For various models, it is the simplest and perhaps the most widely known.

For fermion problems and frustrated spin systems, the main difficulty shared by all the above-mentioned methods is the well-known negative sign problem. In some special cases, one of the methods can be particularly useful compared to others, as is the case for the auxiliary field method applied to the Hubbard model with particle-hole symmetry. However, for many other models, such as frustrated spin models, the WLMC is commonly used because no other method is particularly efficient in reducing the difficulty. In this paper, we do not address the sign problem. We will rather focus on another difficulty which has been underappreciated. The WLMC suffers from long autocorrelation times even when the system is not near a finite-temperature critical point.<sup>(13)</sup> Although a similar problem may exist for the other quantum Monte Carlo techniques, this problem has not yet been studied systematically. One way to overcome this difficulty is to develop a cluster updating as in the SW method. In this paper we present the details of such a method.

In general, it is nontrivial to find a cluster algorithm. The generalized approaches,<sup>(4-7)</sup> however, provide a starting point. They first require the specification of a proper set of local graphs by which the whole system is decomposed into clusters and also require a nonnegative solution to a system of linear equations (weight equations) that is often underdetermined. Little *a priori* guidance is given on the construction of these graphs,

and even the existence of a nonnegative solution is not guaranteed. Nevertheless, solutions exist in some simple cases. The Swendsen–Wang (SW) algorithm, for instance, is one such case. Here, the number of weight equations is only two, as is the number of independent variables. A slightly more complicated case is the loop algorithm<sup>(14,15)</sup> for the six-vertex model. In the massless case, for example, both the number of equations and the number of variables are three. This algorithm was successfully applied in a WLMC simulation of the spin-1/2 antiferromagnetic Heisenberg model<sup>(16)</sup> because the  $S = 1/2$  quantum spin systems can be mapped to the six-vertex model by using ST approximation.<sup>(12)</sup> This particular algorithm can also be viewed as the simplest example of the general method that we present in this paper.

For a general quantum spin system, the number of the weight equations and independent variables can be very large. As we will see below, even in the next simplest case, the case of the  $XY$ -like  $XXZ$  model, is already somewhat difficult to handle, as the number of the equations is seven, whereas the number of the variables is 11. Here, it does not seem guaranteed that a meaningful solution exists. In fact, however, at least one meaningful solution exists for any system regardless of the magnitude of spins or the coupling constants. This rather surprising result is what we will present in this paper as well as a practical method for obtaining the solution. Instead of working on the complicated weight equation itself, we will take another approach which naturally leads to a proper choice of graphs and a solution of the equations; that is, we will generalize the FK cluster representation of the Boltzmann weight of Ising models to the quantum models. We thereby propose a new algorithm which potentially reduces the autocorrelation times greatly. We have already applied the new algorithm to the  $S = 1$  antiferromagnetic Heisenberg chain,<sup>(17)</sup> reducing some autocorrelation times by as much as four orders of magnitude. These algorithms differ from previous attempts<sup>(18)</sup> to generalize the FK transformation in that they produce clusters whose states can naturally be specified by a single variable and can be changed independently. In this paper, we present the details of the construction of the algorithm.

It is useful to outline the algorithm before the detailed explanations and the mathematical proofs. The lattice on which we will work is not the original lattice on which the quantum problem is defined. Instead, we will consider many lattices, each of which is geometrically equivalent to the original lattice, and take this set of lattices as a hyperlattice which has a dimension one greater than the dimension of the original model. This hyperlattice can also be viewed as a collection of local units which we call plaquettes and links. One Monte Carlo step of the new algorithm consists of a labeling process applied to the plaquettes and a flipping process for

clusters formed by these labels. In the labeling process, we assign a label, i.e., a graph, stochastically to each local unit. More specifically, in each local unit, we connect vertices by edges which are chosen probabilistically. After this graph assignment is done for all plaquettes, the union of all local graphs forms a global graph defined for the entire lattice: the connected vertices form clusters and these clusters constitute the global graph. In the flipping process, each cluster is flipped randomly with probability  $1/2$ . The question to be answered is how do we obtain the local graphs and the labeling probabilities.

In this paper, we will develop various notations and definitions needed to describe our algorithm without ambiguity. To this end, in Section 2 we first summarize the general framework of cluster representation and cluster algorithm, and then we illustrate how a problem fits into this general framework by presenting two examples, the SW algorithm for the Ising model and the  $S=1$  antiferromagnetic Heisenberg chain. These examples will not only illustrate the notation, but also give a peek at the generalization of the FK representation and the Monte Carlo algorithms that we will derive from it. After these examples, formal definitions for the words introduced in the examples will be given in Section 3. In this section, we also present, without derivation, the new cluster representation for the general  $XXZ$  models. Section 4 is the essential part of this paper. In this section, we will show how to obtain a solution of the weight equation for the general  $XXZ$  spin model. We also propose a practical method for computing the solution. In Section 5 we present the compact formula for the solution of the weight equation for the Heisenberg models. In Section 6 the ergodicity of the new algorithm is discussed. Section 7 is a brief summary. We also discuss a further generalization of our results to the  $XYZ$  models in the Appendix.

## 2. FRAMEWORK AND EXAMPLES

In this section, we first show how the FK-type representation of the partition function is used in a Monte Carlo simulation and then present two pedagogical examples to introduce symbols and notation which we will use in the subsequent sections to generalize the FK representation and to develop the new algorithms.

### 2.1. The Cluster Representation

In general, the partition function of a lattice spin system is the sum of some weight function over all possible "spin configurations"

$$Z = \sum_{\mathbf{n}} W(\mathbf{n}) \quad (2.1)$$

where

$$\mathbf{n} \equiv (n_1, n_2, \dots, n_{N_v}) \quad (2.2)$$

denotes a set of one-bit variables each of which takes a value 0 or 1. This type of configurational summation is the case even with quantum spin systems when they are mapped by the ST decomposition to problems with classical degrees of freedom. Here, we are considering  $N_v$  vertices, which are lattice points in the case of the Ising model, but will not necessarily be lattice points in the more general cases discussed below. In the subsequent sections, we will show that this weight function  $\mathcal{W}(\mathbf{n})$  itself can be expressed as a sum of another weight function over a variable different from the “spin configurations”  $\mathbf{n}$ ,

$$\mathcal{W}(\mathbf{n}) = \sum_G \tilde{\mathcal{W}}(\mathbf{n}, G) \quad (2.3)$$

This new variable  $G$  is a graph defined on the lattice. Equation (2.1) with (2.3) can be viewed as a partition function of a system that consists of vertex variables  $\mathbf{n}$  and graphs  $G$  interacting with each other. A graph consists of *edges* each of which connects two vertices. It imposes a strong restriction on the values that the variable  $\mathbf{n}$  can take because the weight function  $\tilde{\mathcal{W}}(\mathbf{n}, G)$  will be zero for many “spin configurations.” Therefore, we can express the weight function  $\tilde{\mathcal{W}}(\mathbf{n}, G)$  as

$$\tilde{\mathcal{W}}(\mathbf{n}, G) = V(G) \Delta(\mathbf{n}, G) \quad (2.4)$$

where the function  $\Delta(\mathbf{n}, G)$  represents the restriction imposed by  $G$ , i.e., it takes the value 1 if  $\mathbf{n}$  is *compatible* with  $G$  and 0 otherwise.

All functions that have appeared so far also factorize into products where each factor is defined on a local *unit* denoted by  $u$ , i.e.,

$$\mathcal{W}(\mathbf{n}) = \prod_u w(\mathbf{n}(u)) \quad (2.5)$$

$$V(G) = \prod_u v(G(u)) \quad (2.6)$$

$$\Delta(\mathbf{n}, G) = \prod_u \Delta(\mathbf{n}(u), G(u)) \quad (2.7)$$

A local unit is a bond, i.e., a pair of vertices, in the case of the SW algorithm, whereas it is more than two vertices in general. The symbol  $\mathbf{n}(u)$  is the part of  $\mathbf{n}$  that concerns the unit  $u$ . The symbol  $G(u)$  has the similar meaning.

Equation (2.7) means that the restriction imposed by the graph  $G$  is collection of local restrictions imposed by its subgraphs  $G(u)$ . Reexpressing (2.3) in the factorized form, we obtain

$$w(\mathbf{n}(u)) = \sum_{G(u) \in \Gamma(u)} v(G(u)) \Delta(\mathbf{n}(u), G(u)) \quad (2.8)$$

where  $\Gamma(u)$  is the set of local graphs for the problem at hand.

We must assume another property of the restriction imposed by  $G$ : when we define a one-bit cluster variable  $\sigma_c = 0$  or 1 on each cluster in  $G$ , a way must exist for specifying an arbitrary state compatible with  $G$  so that the  $n_i$  depends on  $i$  only through the cluster  $c$  to which  $i$  belongs. Here, a cluster is a maximal group of vertices connected by edges to which another vertex cannot be added without loss of connectivity. In general, a graph  $G$  has many clusters. To be more specific, defining  $N_c(G)$  as the number of clusters in  $G$  and  $\Sigma(G)$  as a set of configurations compatible with  $G$ , we assume a bijection exists which maps  $\{0, 1\}^{N_c(G)}$  onto  $\Sigma(G)$  in such a way that  $(\sigma_1, \sigma_2, \dots, \sigma_{N_c})$  maps to  $(n_1, n_2, \dots, n_{N_c})$  with  $n_i$  depending only on  $\sigma_c$  and  $i \in c$ .

Thus, our problem is clearly defined. Of course, it is not obvious *a priori* that a given partition function can be expressed in the form described here. Most of this paper will be dedicated to the derivation of this type of representation for quantum spin systems. In the next subsection, we will discuss how we can construct a proper Markov process assuming that we are already given the above representation.

## 2.2. Cluster Monte Carlo Method

We consider a Markov process  $(\mathbf{n}^{(t)}, G^{(t)})$  ( $t = 1, 2, 3, \dots$ ) on the extended phase space  $\Sigma \times \Gamma$ ,<sup>(7)</sup> where  $\Sigma$  is the configuration space, i.e., the set of possible values that  $\mathbf{n}$  can take, and  $\Gamma$  is the space of graphs. The Markov process is characterized by two transition matrices: the first one  $T_L(G' | \mathbf{n}, G)$  gives the probability of having a graph  $G'$  given a state  $(\mathbf{n}, G)$ , and the second one  $T_F(\mathbf{n}' | \mathbf{n}, G)$  gives the probability of having a new set of vertex variables  $\mathbf{n}'$ . More specifically,  $T_L$  and  $T_F$ , called the *labeling* probability and the *flipping* probability, respectively, are defined by

$$T_L(G' | \mathbf{n}, G) \equiv \Pr(G^{(t+1)} = G' | \mathbf{n}^{(t)} = \mathbf{n}, G^{(t)} = G)$$

$$T_F(\mathbf{n}' | \mathbf{n}, G) \equiv \Pr(\mathbf{n}^{(t+1)} = \mathbf{n}' | \mathbf{n}^{(t)} = \mathbf{n}, G^{(t+1)} = G)$$

where  $\Pr(X|Y)$  is the conditional probability of having an event  $X$  given an event  $Y$ .

In ref. 7 a special class of cluster algorithms is discussed in which each cluster can be flipped independently as if it were a single noninteracting degree of freedom. This type of cluster algorithm is called *free*. To obtain a free cluster algorithm,<sup>(7)</sup> we need to choose the flipping probability as

$$T_F(\mathbf{n}' | \mathbf{n}, G) = \Delta(\mathbf{n}', G) / N(G) \tag{2.9}$$

where  $N(G) \equiv |\Sigma(G)| = \sum_{\mathbf{n}} \Delta(\mathbf{n}, G)$ . As mentioned in the last subsection, the symbol  $\Sigma(G)$  stands for a set of configurations defined by

$$\Sigma(G) \equiv \{ \mathbf{n} \in \Sigma | \Delta(\mathbf{n}, G) = 1 \} \tag{2.10}$$

Because of the property of  $\Delta(\mathbf{n}, G)$  discussed in the last section, all variables in a cluster are “locked” into a single one-bit degree of freedom and  $N(G)$  equals  $2^{N_c(G)}$ , where  $N_c(G)$  is the total number of clusters.

Therefore, the implication of the specific form (2.9) for the flipping probability is that we flip each cluster in the given graph  $G$  at random with probability  $1/2$  as if each were a single one-bit degree of freedom not interacting with the others. Although we could choose the flipping probability different from (2.9), the choice we made is obviously advantageous in some respect. Computational simplicity is such an advantage. In addition, we can use improved estimators<sup>(19)</sup> for various quantities.

Given the flipping probability (2.9), we need to choose the labeling probability as

$$T_L(G' | \mathbf{n}, G) = V(G') \Delta(\mathbf{n}, G') / W(\mathbf{n}) \tag{2.11}$$

in order that the limiting distribution is the one desired. It is easy to see that the two transition probabilities (2.9) and (2.11) have the distribution  $\tilde{W}(\mathbf{n}, G)$  as their stationary distribution. In other words, the distribution  $\tilde{W}(\mathbf{n}, G)$  is an eigenstate with the eigenvalue 1 of both transition matrices, i.e.,

$$\tilde{W}(\mathbf{n}', G) = \sum_{\mathbf{n} \in \Sigma} T_F(\mathbf{n}' | \mathbf{n}, G) \tilde{W}(\mathbf{n}, G) \tag{2.12}$$

$$\tilde{W}(\mathbf{n}, G') = \sum_{G \in \Gamma} T_L(G' | \mathbf{n}, G) \tilde{W}(\mathbf{n}, G) \tag{2.13}$$

Here we used the fact that  $T_F(\mathbf{n}' | \mathbf{n}, G)$  and  $T_L(G' | \mathbf{n}, G)$  are independent of  $\mathbf{n}$  and  $G$ , respectively, and also used the abbreviated notation  $T_F(\mathbf{n} | G)$  and  $T_L(G | \mathbf{n})$  for the transition probabilities. Therefore, when the ergodicity holds for the algorithm, the only possible limiting probability distribution is proportional to  $\tilde{W}(\mathbf{n}, G)$ . Indeed, we can prove that the algorithm is ergodic in the case of ferromagnetic and  $XY$ -like models. In the case of the

antiferromagnetic model, although we have not proven ergodicity, we can show that if the conventional algorithm is ergodic, the new algorithm is also ergodic. This issue is discussed in Section 6.

We also note that because of (2.3)

$$\sum_{G \in \Gamma} \tilde{W}(\mathbf{n}, G) \quad (2.14)$$

is the distribution that we want to generate. This fact implies that we can obtain the distribution  $W(\mathbf{n})$  simply by generating a Markov sequence described above, picking the vertex-variable part  $\mathbf{n}^{(t)}$  from the state  $(\mathbf{n}^{(t)}, G^{(t)})$  and ignoring the graph part  $G^{(t)}$ .

In an actual Monte Carlo simulation, the labeling process is done locally because of the decomposability of  $V(G) \Delta(\mathbf{n}, G)$ , i.e., (2.6) and (2.7). Therefore, we can generate a graph  $G$  with the probability (2.11) simply by picking a graph  $G(u)$  for each local unit with probability

$$P_L(G(u) | \mathbf{n}(u)) \equiv v(G(u) \Delta(\mathbf{n}(u), G(u)) / w(\mathbf{n}(u))) \quad (2.15)$$

and then taking the union of these  $G(u)$ 's as  $G$ . We can easily see that the probability (2.15) is properly normalized because of (2.8).

### 2.3. The Fortuin–Kasteleyn Representation and the Swendsen–Wang Algorithm

In this subsection, we briefly review the simplest and best-known cluster algorithm, the SW algorithm for the Ising model, and its connection with the FK representation. This algorithm and connection have already been discussed<sup>(3)</sup>; however, we need a more formal discussion to achieve our goal of establishing a mathematically rigorous background for general cluster algorithms. We also found that Fortuin and Kasteleyn's definitions and notation are not completely convenient to describe more general and complicated algorithms. Therefore, in this subsection, we will review the FK representation and the SW algorithm in a language which we will eventually develop to accommodate more general ideas as we proceed.

The Hamiltonian of the Ising model can be written as

$$\mathcal{H} = -J \sum_{(i,j)} (2n_i - 1)(2n_j - 1) \quad (2.16)$$

where  $n_i = 0, 1$ . Correspondingly, the partition function of the Ising model is written as

$$Z = \sum_{\mathbf{n}} W(\mathbf{n}) \quad (2.17)$$



Here  $\mathbf{n} \equiv (n_1, n_2, \dots, n_{|L|})$  is a set of  $|L|$  one-bit variables, where  $|L|$  is the number of lattice points of the lattice  $L$ . The function  $W(\mathbf{n})$  is the Boltzmann weight, which can be decomposed into a product of local factors defined on local units, i.e., bonds in this case,

$$W(\mathbf{n}) = \prod_b w(\mathbf{n}(b)) \tag{2.18}$$

where  $\mathbf{n}(b) \equiv (n_i, n_j)$  is a part of  $\mathbf{n}$  concerning the two endpoints  $i$  and  $j$  of a bond  $b$ . The local Boltzmann factor  $w(\mathbf{n}(b))$  is then defined by

$$w(\mathbf{n}(b)) \equiv \exp[K(2n_i - 1)(2n_j - 1)] \tag{2.19}$$

where  $K \equiv \beta J$ . Equation (2.19) can be expressed as a sum of two terms each of which corresponds to one of two graphs often called “deleted” and “frozen”

$$w(\mathbf{n}(b)) = \sum_{g=d,f} v(g) \Delta(\mathbf{n}(b), g) \tag{2.20}$$

where

$$v(d) = e^{-K} \quad \text{and} \quad \Delta(\mathbf{n}(b), d) = 1 \tag{2.21}$$

for a “deleted” bond and

$$v(f) = e^K - e^{-K} \quad \text{and} \quad \Delta(\mathbf{n}(b), f) = \delta(n_i, n_j) \tag{2.22}$$

for a “frozen” bond. Substituting (2.20) into (2.18), we have

$$\begin{aligned} W(\mathbf{n}) &= \prod_b \sum_{g_b} v(g_b) \Delta(\mathbf{n}(b), g_b) \\ &= \sum_{\{g_b\}} \left[ \prod_b v(g_b) \right] \left[ \prod_b \Delta(\mathbf{n}(b), g_b) \right] \end{aligned} \tag{2.23}$$

The product  $\prod_b \Delta(\mathbf{n}(b), g_b)$  implies that a configuration  $\mathbf{n}$  is allowed only if the two variables  $n_i$  and  $n_j$  at the endpoints of any frozen bond are equal. Otherwise, this product vanishes and the corresponding term does not contribute to the sum. We can visualize this situation by placing edges on frozen bonds and nothing on deleted bonds. These edges form a graph which we denote as  $G$ . Obviously, this graph has the same information as the set of variables  $\{g_b\}$ . Therefore, we will identify these two and simply write  $G = \{g_b\}$ . A cluster is a maximal set of vertices connected to each other by edges. A graph consists of many clusters, in general. It is clear that for a given graph  $G$  an allowed state is one where all variables  $n_i$  in the

same cluster in  $G$  have the same value 0 or 1. In other words, all variables in a cluster are locked into a single degree of freedom represented by  $\sigma_c = 0$  or 1, where  $c$  specifies a cluster in  $G$ .

If we adopt the notation used in (2.7), we can write  $g_b = G(b)$  and then rewrite (2.23) as

$$W(\mathbf{n}) = \sum_G V(G) \Delta(\mathbf{n}, G) \quad (2.24)$$

where

$$V(G) \equiv \prod_b v(G(b)) \quad (2.25)$$

$$\Delta(\mathbf{n}, G) \equiv \prod_b \Delta(\mathbf{n}(b), G(b)) \quad (2.26)$$

$$(2.27)$$

Substituting (2.24) into (2.17), we have

$$Z = \sum_{\mathbf{n}} \sum_G V(G) \Delta(\mathbf{n}, G) \quad (2.28)$$

This equation is the Fortuin–Kasteleyn representation for the Ising model. Although it is not explicitly indicated,  $Z$  and  $V(G)$  depend on the model and the coupling constant  $K$ . On the other hand,  $\Delta(\mathbf{n}, G)$  will be used for other models by generalizing its definition. In what follows, we will see exactly the same form as (2.28) with different  $Z$  and  $V(G)$  for other models.

Because we have rewritten the partition function in the form discussed in Section 2.1, we can construct the Markov process following the general prescription given in Section 2.2. The resulting algorithm is the SW algorithm. To be specific, given a configuration  $\mathbf{n}$ , we first choose a graph  $G$  with the weight  $V(G) \Delta(\mathbf{n}, G)$  and then pick a new configuration  $\mathbf{n}'$  with the weight  $\Delta(\mathbf{n}', G)$ . The first step is equivalent to choosing a local graph  $G(b)$  with the weight  $v(G(b)) \Delta(\mathbf{n}(b), G(b))$ . Because of (2.20), the probability of assigning  $G(b)$  to a given bond in state  $\mathbf{n}(b)$  is

$$P_L(G(b) | \mathbf{n}(b)) \equiv v(G(b)) \Delta(\mathbf{n}(b), G(b)) / w(\mathbf{n}(b)) \quad (2.29)$$

This result agrees with the ordinary SW labeling probability

$$P_L(\text{"deleted"} | (0, 0)) = P_L(\text{"deleted"} | (1, 1)) = e^{-2K} \quad (2.30)$$

$$P_L(\text{"deleted"} | (0, 1)) = P_L(\text{"deleted"} | (1, 0)) = 1 \quad (2.31)$$

$$P_L(\text{"frozen"} | \mathbf{n}(b)) = 1 - P_L(\text{"deleted"} | \mathbf{n}(b)) \quad (2.32)$$

On the other hand, the second step is, as discussed in the last subsection, equivalent to setting each cluster variable  $\sigma_c$  to 0 or 1 with equal probability.

### 2.4. Generalization to Quantum Spin Systems

In this subsection, we will give an example of the generalization of the FK representation to quantum spin systems. We will discuss the  $S = 1$  antiferromagnetic Heisenberg model in one dimension. The Hamiltonian is written as

$$\mathcal{H} = J \sum_{i=1}^N \mathbf{S}_i \cdot \mathbf{S}_{i+1} \tag{2.33}$$

where  $\mathbf{S}_i^2 = 2$  and

$$[S_i^\alpha, S_j^\beta] = \sqrt{-1} \delta_{i,j} S_j^\gamma, \quad (\alpha, \beta, \gamma) = (x, y, z), (y, z, x), \text{ or } (z, x, y) \tag{2.34}$$

The periodic boundary condition, i.e.,  $\mathbf{S}_{N+1} = \mathbf{S}_1$ , is assumed. We also assume that  $N$  is even. Then, by applying the unitary transformation

$$S_i^x \rightarrow -S_i^x \quad \text{and} \quad S_i^y \rightarrow -S_i^y \tag{2.35}$$

to all sites with even  $i$ , we have

$$\mathcal{H} = -J \sum_{i=1}^N \hat{A}_{i,i+1} \tag{2.36}$$

where

$$\hat{A}_{i,j} \equiv S_i^x S_j^x + S_i^y S_j^y - S_i^z S_j^z \tag{2.37}$$

We remark that these assumptions are not essential to the general algorithm described in the following sections.

Usually, we take the basis in which  $S_i^z$  is diagonalized for Monte Carlo simulations. Then, a basis vector is specified by a set of  $N$  classical variables, each of which takes a value  $-1, 0, \text{ or } 1$ . The Hilbert space is, of course,  $3^N$  dimensional. However, this basis is inconvenient for our purpose of mapping the problem into a one-bit problem of the form discussed in Section 2.1. Therefore, we will first map the original quantum problem into another quantum problem with one-bit degrees of freedom. To do this, we first decompose each spin operator into a sum of two Pauli operators:

$$S_i^\alpha = \frac{1}{2}(\sigma_{(i,1)}^\alpha + \sigma_{(i,2)}^\alpha), \quad \alpha = x, y, \text{ or } z \tag{2.38}$$

where

$$[\sigma_{(i,\mu)}^\alpha, \sigma_{(j,\nu)}^\beta] = 2\sqrt{-1} \delta_{i,j} \delta_{\mu,\nu} \sigma_{(i,\mu)}^\gamma \quad (2.39)$$

Then, we take a basis in which the  $z$ -components of the Pauli operators are diagonal. Therefore, a basis vector can be expressed as  $|\mathbf{n}\rangle$ , where  $\mathbf{n}$  is a set of  $2N$  one-bit variables

$$\mathbf{n} = (n_{(1,1)}, n_{(1,2)}, n_{(2,1)}, n_{(2,2)}, n_{(3,1)}, n_{(3,2)}, \dots, n_{(N,1)}, n_{(N,2)}) \quad (2.40)$$

and has the property

$$\sigma_{(i,\mu)}^z |\mathbf{n}\rangle = (2n_{(i,\mu)} - 1) |\mathbf{n}\rangle, \quad n_{(i,\mu)} = 0 \text{ or } 1 \quad (2.41)$$

Clearly, the Hilbert space is  $2^{2N}$  dimensional and larger than the original Hilbert space. The new Hilbert space is larger because it includes some singlet states which are unphysical. Since we should not count such states in the partition function, the partition function in this new basis becomes

$$Z = \sum_{\mathbf{n}} \langle \mathbf{n} | P e^{-\beta \mathcal{H}} P | \mathbf{n} \rangle \quad (2.42)$$

The projection operator  $P$  is the product of local projection operators  $P_i$  which projects out states with  $S_i^z = 0$ ,

$$P = \prod_i P_i \quad (2.43)$$

It is easy to see that  $P_i$  is the symmetrization operator

$$P_i = \frac{1}{2}(\hat{I} + \hat{X}_i) \quad (2.44)$$

where  $\hat{I}$  is the identity operator and  $\hat{X}_i$  is the operator that exchanges  $n_{(i,1)}$  and  $n_{(i,2)}$ .

Thus, the original problem is mapped to a problem with one-bit degrees of freedom (2.42). This problem still is a quantum problem which needs the evaluation of matrix elements of the Boltzmann density operator. Therefore, the next thing we must do is to map this problem to a classical one-bit problem by the ST decomposition. We can write  $P e^{-\beta \mathcal{H}} P$  in (2.42) as

$$P e^{-\beta \mathcal{H}} P = (P e^{-\Delta\tau \mathcal{H}} P)^{M_T} \approx [(P e^{-\Delta\tau \mathcal{H}_E} P)(P e^{-\Delta\tau \mathcal{H}_O} P)]^{M_T} \quad (2.45)$$

where  $M_T \Delta\tau \equiv \beta$  and

$$\mathcal{H}_E \equiv -J \sum_{i: \text{even}} \hat{A}_{i,i+1}, \quad \mathcal{H}_O \equiv -J \sum_{i: \text{odd}} \hat{A}_{i,i+1} \quad (2.46)$$

The integer  $M_T$  is called the *Trotter number*. By inserting the identity operator

$$I = \sum_{\mathbf{n}'} |\mathbf{n}'\rangle \langle \mathbf{n}'| \tag{2.47}$$

between the factors in (2.45), we can reexpress the partition function (2.42) as

$$\begin{aligned} Z = & \sum_{\mathbf{n}_M} \sum_{\mathbf{n}_{M-1}} \dots \sum_{\mathbf{n}_1} \langle \mathbf{n}_1 | P e^{-\Delta\tau \mathcal{H}_E} P | \mathbf{n}_M \rangle \langle \mathbf{n}_M | P e^{-\Delta\tau \mathcal{H}_O} P | \mathbf{n}_{M-1} \rangle \\ & \times \dots \langle \mathbf{n}_2 | P e^{-\Delta\tau \mathcal{H}_O} P | \mathbf{n}_1 \rangle \end{aligned} \tag{2.48}$$

where  $M \equiv 2M_T$ .

Because of the form of (2.48), it is natural to introduce a hyperlattice which consists of  $M$  lattices, each of which is equivalent to the original lattice. Because the original lattice is a ring in the present case and we are automatically imposing the periodic boundary condition in the new direction, the hyperlattice is a torus. We will refer to this hyperlattice by a symbol  $\tilde{L}$  and each ring in this hyperlattice by a symbol  $\tilde{L}_k$  ( $k = 1, 2, \dots, M$ ). We call  $MN$  lattice points in the hyperlattice *sites*. As we have seen, two one-bit variables are defined on each site. They can be viewed as variables defined on two *vertices* inside the site. Thus, the hyperlattice is a set of  $2MN$  vertices. In what follows, we refer to the set of vectors  $\{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_M\}$  simply as  $\mathbf{n}$ . We also refer to the variable  $n_{(i,\mu)}$  on  $\tilde{L}_k$  as  $n_{(k,i,\mu)}$ . Accordingly, the vertex on which  $n_{(k,i,\mu)}$  is defined will be referred to as  $(k, i, \mu)$ . The site that contains the vertices  $(k, i, 1)$  and  $(k, i, 2)$  will be referred to as  $(k, i)$ . In Fig. 1, the hyperlattice is shown for the case of  $N = 6$  and  $M_T = 2$ . Next, we introduce the symbol  $\mathbf{n}(V)$ , which stands for the part of  $\mathbf{n}$  concerning a set  $V$ , where  $V$  is any set of vertices. For example,  $\mathbf{n}((k, i, \mu))$  is simply  $n_{(k,i,\mu)}$ ,  $\mathbf{n}((k, i))$  stands for  $(n_{(k,i,1)}, n_{(k,i,2)})$ ,  $\mathbf{n}(\tilde{L}_k)$  is  $\mathbf{n}_k$ , and  $\mathbf{n}(\tilde{L})$  is  $\mathbf{n}$  itself. Actually, a special case of this notation has already been used in (2.7).

Having defined this notation, we note that the state  $|\mathbf{n}(\tilde{L}_k)\rangle$  can be expressed by a direct product of local wave functions as

$$|\mathbf{n}(\tilde{L}_k)\rangle = \bigotimes_{i: \text{even}} |\mathbf{n}(l_{(k,i)})\rangle_2 \tag{2.49}$$

where  $l_{(k,i)}$  is a pair of sites  $\{(k, i), (k, i + 1)\}$ , or equivalently a set of four vertices  $\{(k, i, 1), (k, i, 2), (k, i + 1, 1), (k, i + 1, 2)\}$ . To be more specific,  $|\mathbf{n}(l_{(k,i)})\rangle_2$  is an eigenvector of four Pauli operators  $\sigma_{(i,\mu)}^z$  and  $\sigma_{(i+1,\mu)}^z$

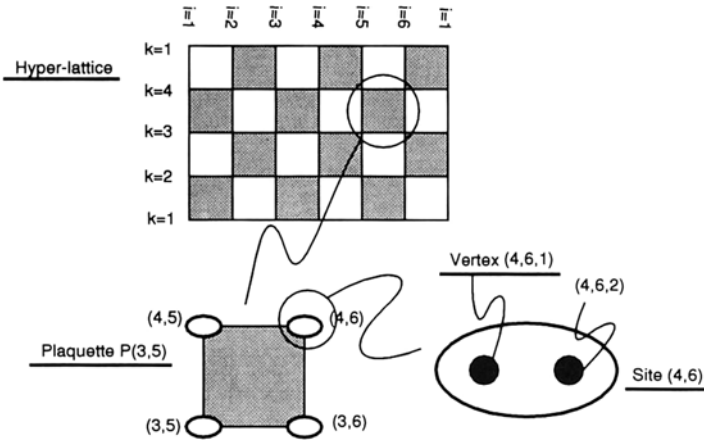


Fig. 1. The “checkerboard” hyperlattice for the  $S = 1$   $XY$  model on a chain of the length 6 with Trotter number 2. Only shaded plaquettes are called “plaquettes” here. On each plaquette a four-body Boltzmann factor is defined.

( $\mu = 1, 2$ ) in a  $2^4$ -dimensional local Hilbert space. Therefore, because  $[\hat{A}_{i,i+1}, \hat{A}_{j,j+1}] = 0$  if  $|i - j| \geq 2$ , we obtain

$$\langle \mathbf{n}(\tilde{L}_{k+1}) | Pe^{-\mathcal{H}P} | \mathbf{n}(\tilde{L}_k) \rangle = \prod_{i: \text{even}} w(\mathbf{n}(p_{(k,i)})) \quad (2.50)$$

where  $p_{(k,i)}$  is a *plaquette* defined by  $p_{(k,i)} \equiv l_{(k,i)} \cup l_{(k+1,i)}$  and

$$w(\mathbf{n}(p_{(k,i)})) \equiv \langle \mathbf{n}(l_{(k+1,i)}) | P_i P_{i+1} [\exp(K\hat{A}_{i,i+1})] P_i P_{i+1} | \mathbf{n}(l_{(k,i)}) \rangle_2 \quad (2.51)$$

with  $K \equiv \mathcal{A}J$ . An expression similar to (2.50) is available for  $\mathcal{H}_O$ . Therefore, the partition function (2.48) can be rewritten as

$$Z = \sum_{\mathbf{n}} W(\mathbf{n}) \quad (2.52)$$

where

$$W(\mathbf{n}) = \prod_p w(\mathbf{n}(p)) \quad (2.53)$$

The product in (2.53) is taken over  $p_{(k,i)}$  with even  $k + i$ .

Here we note that  $w(\mathbf{n}(p))$  depends on  $\mathbf{n}(p)$  only through  $m_{(k,i)} \equiv \sum_{\mu} n_{(k,i,\mu)}$ , where  $(k, i)$  is one of four sites in the plaquette  $p$ . Therefore, we can reexpress  $w(\mathbf{n}(p))$  in the form

$$w(\mathbf{n}(p)) = \tilde{w} \begin{pmatrix} m_{(k+1,i)} & m_{(k+1,i+1)} \\ m_{(k,i)} & m_{(k,i+1)} \end{pmatrix} \quad (2.54)$$

We further note that this  $\tilde{w}$  has several symmetry properties:

$$\begin{aligned} \tilde{w} \begin{pmatrix} m_{tl} & m_{tr} \\ m_{bl} & m_{br} \end{pmatrix} &= \tilde{w} \begin{pmatrix} m_{bl} & m_{br} \\ m_{tl} & m_{tr} \end{pmatrix} = \tilde{w} \begin{pmatrix} m_{tr} & m_{tl} \\ m_{br} & m_{bl} \end{pmatrix} \\ &= \tilde{w} \begin{pmatrix} 2S - m_{bl} & 2S - m_{br} \\ 2S - m_{tl} & 2S - m_{tr} \end{pmatrix} \end{aligned} \quad (2.55)$$

where  $S = 1$  in the present case and the suffix “tl” means top-left corner of the plaquette, “tr” means top-right, etc. These symmetry properties define classes of states which correspond to the same local weight  $\tilde{w}$ . In what follows, we will specify one of these classes by a symbol  $\mathcal{S}(p)$ . The “particle number conservation” imposes a restriction on  $\tilde{w}$ , i.e.,  $\tilde{w}$  is nonzero only if

$$m_{bl} + m_{br} = m_{tl} + m_{tr} \quad (2.56)$$

We can eliminate from consideration all classes that violate this condition. As a result, only seven distinct values among  $2^8 = 256$  values of  $w(\mathbf{n}(p))$  exist, i.e., there are seven relevant classes of states,

$$\begin{aligned} \tilde{w}(1) &= \tilde{w} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} = 12r \\ \tilde{w}(2) &= \tilde{w} \begin{pmatrix} 0 & 2 \\ 0 & 2 \end{pmatrix} = 2r + 6r^{-1} + 4r^{-2} \\ \tilde{w}(3) &= \tilde{w} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = 2r - 6r^{-1} + 4r^{-2} \\ \tilde{w}(4) &= \tilde{w} \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} = -4r + 4r^{-2} \\ \tilde{w}(5) &= \tilde{w} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = -6r + 6r^{-1} \\ \tilde{w}(6) &= \tilde{w} \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} = 6r + 6r^{-1} \\ \tilde{w}(7) &= \tilde{w} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = 8r + 4r^{-2} \end{aligned} \quad (2.57)$$

where  $r \equiv \exp(-K)$ .

Equations (2.52) and (2.53) have the same form as (2.1) and (2.5). Therefore, the next thing we have to find is a set of local graphs  $\Gamma(p)$  and coefficients  $v(G(p))$  that satisfy (2.8) with  $u$  replaced by  $p$ ,

$$w(\mathbf{n}(p)) = \sum_{G(p) \in \Gamma(p)} v(G(p)) \Delta(\mathbf{n}(p), G(p)) \tag{2.58}$$

To this end, we consider the graphs which correspond to pairing of the eight vertices in  $p$ , i.e., graphs which consist of four edges sharing no vertices. From the 105 such graphs, we take only those graphs that consist of vertical or horizontal edges. Here, a vertical edge connects two vertices whose spatial indices are the same and the temporal indices are different, whereas a horizontal edge connects two vertices whose spatial indices are different and temporal indices are the same. There are 24 such graphs. When we neglect distinctions between vertices in the same site, these 24 graphs are classified into three classes which are represented by the diagrams in the leftmost column in Fig. 2. We use a symbol  $\mathcal{G}(p)$  to specify a class of graphs.

We next define  $\Delta(\mathbf{n}(p), G(p))$  in (2.58) as the function that for any edge connecting two vertices  $v$  and  $v'$  takes the value of 1 if  $\mathbf{n}(v) = \mathbf{n}(v')$  and the edge is vertical and if  $\mathbf{n}(v) = 1 - \mathbf{n}(v')$  and the edge is horizontal.

$\mathcal{G}(p) \backslash \mathcal{S}(p)$		1	2	3	4	5	6	7
		$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 \\ 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$
1		4	4	0	0	0	2	1
2		0	16	0	4	4	4	2
3		0	4	4	2	0	0	1

Fig. 2. The coefficient  $N(\mathcal{S}(p), \mathcal{G}(p))$  for the  $S=1$  antiferromagnetic Heisenberg model.  $\mathcal{S}(p)$  specifies a class of states and  $\mathcal{G}(p)$  specifies a class of graphs. In the diagrams of the leftmost column, a solid line stands for a green edge and a dashed line a red edge.



Otherwise, this function takes the value of zero. For later convenience, we call edges for which two vertex variables have the same value *green* and all other edges *red*. In the present case, all vertical edges are green, whereas all horizontal edges are red. All edges in what follows have this additional attribute, i.e., *color*. With this definition, we can formally specify the set of graphs  $\Gamma(p)$  in (2.58) as

$$\Gamma(p) \equiv \{G(p) \mid \text{“Every vertex is an endpoint of one and only one edge” and} \\ \text{“An edge is either green vertical or red horizontal”}\} \quad (2.59)$$

Given (2.57) and (2.59), we next find a solution  $v(G(p))$  of (2.58). It is natural to seek a symmetric solution in which  $v(G(p))$  depends on  $G(p)$  only through the class to which  $G(p)$  belongs, i.e.,  $v(G(p)) = \tilde{v}(\mathcal{G}(p))$  where  $G(p) \in \mathcal{G}(p)$ . Taking this into account, we can rewrite (2.58) as

$$\tilde{w}(\mathcal{S}(p)) = \sum_{\mathcal{G}(p)} N(\mathcal{S}(p), \mathcal{G}(p)) \tilde{v}(\mathcal{G}(p)) \quad (2.60)$$

Here, the function  $N$  is defined by

$$N(\mathcal{S}(p), \mathcal{G}(p)) \equiv \sum_{G(p) \in \mathcal{G}(p)} A(\mathbf{n}(p), G(p)) \quad (2.61)$$

where  $\mathbf{n}(p) \in \mathcal{S}(p)$ . The function  $N$  is shown in Fig. 2. We can easily see that the solution

$$\begin{aligned} \tilde{v}(1) &= \frac{1}{4} \tilde{w}(1) \\ \tilde{v}(2) &= \frac{1}{4} \tilde{w}(5) \\ \tilde{v}(3) &= \frac{1}{4} \tilde{w}(3) \end{aligned} \quad (2.62)$$

satisfies (2.60).

Now, since we obtained the representation in the form discussed in Section 2.1, the Monte Carlo method in Section 2.2 applies. The resulting algorithm is the loop algorithm used in ref. 17.

In this subsection, we found the solution  $v(G(p))$  given a set of graphs  $\Gamma(p)$ , but did not show how we choose  $\Gamma(p)$  or how we obtain the solution  $v(G(p))$  in general. These questions will be answered in the following sections.

### 3. CLUSTER REPRESENTATION OF GENERAL XXZ MODEL

In Section 2.4 we saw how the  $S=1$  antiferromagnetic Heisenberg chain is mapped to a one-bit classical problem. In this section, we will see

how the mapping is done in the case of general  $XXZ$  spin systems. Our task can be divided into two parts. In Section 3.1 we formulate the problem as a one-bit classical problem with a weight function that factorizes into a product of local factors. Each one of the local factors is defined on a local unit called a plaquette, as we saw already. At this stage, graphs are not yet introduced. In short, in Section 3.1 we will formulate the problem in the form of (2.1) with (2.5). Then, in Section 3.3 we show without derivation how  $w(\mathbf{n}(u))$  can be expressed as a sum of terms each of which corresponds to a local graph  $G(u)$ ; that is, we will present an explicit form of (2.8) for the  $XXY$  model. This expression leads to (2.3). The derivation will be presented in Section 4. To make the discussions clearer, in Section 3.2 we summarize and generalize the notations and definitions introduced in Section 2.4 concerning graphs.

### 3.1. Mapping to a Classical Problem

Our Hamiltonian is

$$\begin{aligned} \mathcal{H} &= - \sum_{(i,j)} (J_{i,j}^x S_i^x S_j^x + J_{i,j}^y S_i^y S_j^y + J_{i,j}^z S_i^z S_j^z) \\ &= - \sum_{(i,j)} J_{i,j} (S_i^x S_j^x + S_i^y S_j^y + \lambda_{i,j} S_i^z S_j^z) \end{aligned} \quad (3.1)$$

with  $J_{i,j}^x = J_{i,j}^y = J_{i,j}$  and  $J_{i,j}^z = J_{i,j} \lambda_{i,j}$ . The operator  $S_i^\alpha$  is the spin operator which satisfies

$$\mathbf{S}_i^2 = (S_i^x)^2 + (S_i^y)^2 + (S_i^z)^2 = S(S+1) \quad (3.2)$$

and

$$[S_i^\alpha, S_j^\beta] = \sqrt{-1} \delta_{i,j} S_i^\gamma \quad (3.3)$$

for  $(\alpha, \beta, \gamma) = (x, y, z), (y, z, x),$  or  $(z, x, y)$ . The symbol  $(i, j)$  in (3.1) is an arbitrary undirected pair of elements of the set of lattice points  $L$ . The constants  $J_{i,j}$  and  $\lambda_{i,j}$  can be any real numbers. We are not assuming any particular geometric feature of the lattice. It can have any dimension and does not even have to be translationally symmetric.

The first important step is to express a spin operator in terms of sum of  $2S$  Pauli operators as we did in Section 2.4,

$$S_i^\alpha = \frac{1}{2} \sum_{\mu=1}^{2S} \sigma_{i,\mu}^\alpha \quad (3.4)$$

To each Pauli matrix, we will assign a vertex. Therefore, it is convenient to define  $\tilde{L}$  as a set of vertices defined on the lattice  $L$ . We work with a complete set of states that are simultaneous eigenfunctions of the operators  $\sigma_{i,\mu}^z$  ( $i = 1, 2, \dots, |L|$ ;  $\mu = 1, 2, \dots, 2S$ ). Here,  $|L|$  is the number of the lattice points in the lattice  $L$ . To be more specific,

$$\sigma_{i,\mu}^z |\mathbf{n}(\tilde{L})\rangle \equiv (2n_{i,\mu} - 1) |\mathbf{n}(\tilde{L})\rangle \quad (3.5)$$

where  $\mathbf{n}(\tilde{L})$  represents  $2S|L|$  one-bit variables,

$$\mathbf{n}(\tilde{L}) \equiv (n_{(1,1)}, n_{(1,2)}, \dots, n_{(|L|,2S)}) \quad (3.6)$$

The partition function in this basis becomes

$$Z = \sum_{\mathbf{n}(\tilde{L})} \langle \mathbf{n}(\tilde{L}) | P e^{-\beta \mathcal{H}} P | \mathbf{n}(\tilde{L}) \rangle \quad (3.7)$$

where  $P$  is the projection operator defined by

$$P = \prod_{i \in L} P_i \quad (3.8)$$

and  $P_i$  is the projection operator to the space in which  $S_i^z$  takes the value  $S(S+1)$ . Since this representation with Pauli matrices provides a single representation for which  $S_i^z = S(S+1)$ , (3.7) is the correct representation of the original model.

As we saw in Section 2.4, using the ST approximation<sup>(12)</sup> and (3.4), we can map the original quantum problem into a problem with classical one-bit degrees of freedom. However, since there are so many variants of the ST approximation, it is impractical to describe all of them. Therefore, in what follows, we will only describe properties which are shared by all known variants.

Once mapped, the problem has  $2SM|L|$  one-bit variables, where  $M$  is some integer proportional to the Trotter number  $M_T$ . The proportionality constant depends on the variant of the ST approximation. The  $2SM|L|$  variables naturally fit into a  $(d+1)$ -dimensional hyperlattice, where  $d$  is the dimension of the original lattice  $L$ . We call the first dimension *temporal* or *vertical* and the other  $d$  dimensions, which correspond to the dimensions of the original lattice, *spatial* or *horizontal*. This lattice consists of  $M$  layers of  $d$ -dimensional lattices each of which is equivalent to the original lattice  $L$ . We number these layers with an index  $k = 1, 2, \dots, M$  and use  $L_k$  to denote the  $k$ th layer. We call a lattice point in this hyperlattice a *site*. A total of  $2S$  vertices are associated with each site, and a one-bit variable is defined on each vertex. We label a site with two indices  $(k, i)$ , with the first index

specifying the temporal location of the site and the second index specifying the spatial location. As for vertices, we use symbols  $(k, i, \mu)$  by adding an additional index  $\mu = 1, 2, \dots, 2S$  specifying the  $\mu$ th vertex in the site  $(k, i)$ . The boundary condition in the temporal direction is periodic, i.e.,  $k = M + 1$  is identified with  $k = 1$ . On the other hand, we are not assuming any particular spatial geometry; therefore, the present scheme can be applied to models in any dimension, with any boundary condition, and with any range of the interactions.

A *horizontal link* is a pair of sites whose temporal indices are the same. A *vertical link* is a pair of sites whose spatial indices are the same and temporal indices differ by one. Some of the squares which consist of two vertical links and two horizontal links play a special role, because a local weight function is defined on them. We call such a square a *plaquette* and often use a symbol  $p$  for it. The set of  $8S$  vertices associated with four corners (sites) of a plaquette will also be referred to as a plaquette and represented by the same symbol. Which square among all possible squares is to be a plaquette depends on the variant of the ST approximation. In any variant of the ST approximation, no vertical link is shared by more than one plaquette. The symbol  $\mathcal{U}_p$  denotes the set of plaquettes, whereas the symbol  $\mathcal{U}_l$  denotes the set of vertical links which do not belong to any plaquette. In the example presented in Section 2.4,  $\mathcal{U}_l$  is an empty set. In other cases, such as the systems defined on a triangular lattice,  $\mathcal{U}_l$  is non-empty. We also define  $\mathcal{U} \equiv \mathcal{U}_p \cup \mathcal{U}_l$ . This is the set of local units that we will consider in what follows and the products in (2.5)–(2.7) should be taken over this set in the present case.

It is convenient to have a notation by which we can refer to a specific site or a link in a given plaquette  $p$ . To this end, we consider a plaquette  $p$  consisting of four sites  $(k, i)$ ,  $(k, j)$ ,  $(k + 1, i)$ , and  $(k + 1, j)$ . We first take either one of spatial indices  $i$  and  $j$  and call it “left” and the other “right.” We use the symbols  $i_l(p)$  for “left” index and  $i_r(p)$  for the other. (Which one we call “left” does not matter in what follows.) We also define symbols  $l_t(p)$  and  $l_b(p)$  as the top and bottom horizontal links of  $p$ , respectively. The symbol  $s_{tl}(p)$  stands for a top-left site. Other symbols  $s_{tr}(p)$ ,  $s_{bl}(p)$ , and  $s_{br}(p)$  are defined in a similar fashion. The relative locations of these symbols are illustrated in Fig. 3.

With these definitions and the ST approximation, we can rewrite the partition function (3.7) as

$$Z \equiv \sum_{\mathbf{n} \in \mathcal{Z}} \text{sgn}(\mathbf{n}) W(\mathbf{n}) \quad (3.9)$$

with the sign factor  $\text{sgn}(\mathbf{n})$  and the Boltzmann weight  $W(\mathbf{n})$  defined below. The sign factor is defined so that  $W(\mathbf{n})$  is always nonnegative. The

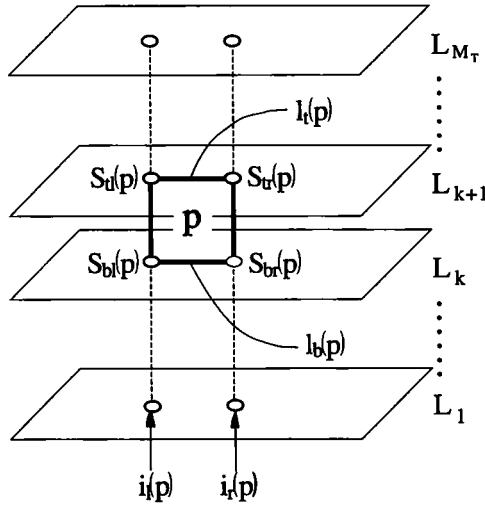


Fig. 3. The hyperlattice on which the transformed problem is defined and a plaquette on which a four-body Boltzmann factor is defined. The upper link of the plaquette  $p$  belongs to the layer  $L_{k+1}$  and its lower link belongs to the layer  $L_k$ . The open ovals represent sites.

configurations with negative sign appear only for a frustrated system such as the antiferromagnetic Heisenberg model on a triangular lattice. Although the sign factor is physically important, from a computational point of view, it is needed only when adding up measured values at each Monte Carlo step and does not affect the Markov process. In other words, we usually neglect the sign factor in defining the Markov process and the limiting distribution of the resulting process is  $\mathcal{W}(\mathbf{n})$ . Since the goal of the present paper is to accelerate the Markov process, we neglect the sign factor in (3.9). The weight function  $\mathcal{W}(\mathbf{n})$  in (3.9) is defined by

$$\mathcal{W}(\mathbf{n}) \equiv \prod_{p \in \mathcal{Q}_p} w(\mathbf{n}(p)) \prod_{l \in \mathcal{Q}_l} \delta_l(\mathbf{n}(l)) \tag{3.10}$$

Since there is no interaction which corresponds to a vertical link in  $\mathcal{Q}_l$ ,  $\delta_l(\mathbf{n}(l))$  should be a function which takes the nonzero value 1 if and only if  $n_{(k,i,\mu)} = n_{(k+1,i,\mu)}$  for all  $\mu$ , where  $l = \{(k, i), (k + 1, i)\}$ .

The function  $w(\mathbf{n}(p))$  in (3.10) is the absolute value of matrix element of the local Boltzmann operator, that is,

$$w(\mathbf{n}(p)) \equiv |\langle \mathbf{n}(l_t(p)) | \tilde{\rho}(p) | \mathbf{n}(l_b(p)) \rangle| \tag{3.11}$$

Here,  $\langle \dots \rangle_2$  is the matrix element in  $2^{4S}$ -dimensional Hilbert space on which the operators  $\sigma_{i,\mu}^\alpha$  and  $\sigma_{j,\mu}^\alpha$  operate, where  $\mu = 1, 2, \dots, 2S$  and  $\alpha = x, y, z$ . The operator  $\tilde{\rho}(p)$  in (3.11) is defined by

$$\tilde{\rho}(p) \equiv P_{i(p)} P_{i(p)} \rho(p) P_{i(p)} P_{i(p)} \quad (3.12)$$

with

$$\rho(p) \equiv \exp \left[ \sum_{\mu, \nu} K_p \hat{A}(\mu, \nu) \right] \quad (3.13)$$

Here  $\hat{A}(\mu, \nu)$  is defined by

$$\hat{A}(\mu, \nu) \equiv \sigma_{i(p), \mu}^x \sigma_{i(p), \nu}^x + \sigma_{i(p), \mu}^y \sigma_{i(p), \nu}^y + \lambda_p \sigma_{i(p), \mu}^z \sigma_{i(p), \nu}^z \quad (3.14)$$

The constant  $K_p$  in (3.13) depends on the variant of the ST decomposition. However, in any variant of the ST decomposition, the following equation holds:

$$\sum_{\substack{p \\ i(p)=i \\ i(p)=j}} K_p = \beta J_{i,j}/4 \quad \text{for all } (i, j) \quad (3.15)$$

where the summation is over the plaquettes whose spatial location is specified by two spatial indices  $i$  and  $j$ .

As a function of  $K_p$  and  $\lambda_p$ , the local weight function  $w(\mathbf{n}(p))$  has the following property:

$$[w(\mathbf{n}(p))]_{K_p, \lambda_p} = [w(\mathbf{n}(p))]_{-K_p, -\lambda_p} \quad (3.16)$$

Because of this identity, we can assume positive  $K_p$  without loss of generality. With this assumption, we can rewrite (3.11) simply as

$$w(\mathbf{n}(p)) \equiv \langle \mathbf{n}(l_i(p)) | \tilde{\rho}(p) | \mathbf{n}(l_b(p)) \rangle_2 \geq 0 \quad (3.17)$$

In order to prove the identity (3.16), we notice that

$$\begin{aligned} & \langle \mathbf{n}(l_i(p)) | \tilde{\rho}(p) | \mathbf{n}(l_b(p)) \rangle_2 \\ &= \langle \langle m_{tl}, m_{tr} | \rho(p) | m_{bl}, m_{br} \rangle \rangle / \left[ \prod_{X=tl, tr, bl, br} \binom{2S}{m_X} \right]^{1/2} \end{aligned} \quad (3.18)$$

where

$$m_X \equiv \sum_{v \in s_X(p)} n_v, \quad X = tl, tr, bl, \text{ or } br \quad (3.19)$$

and  $|m_1, m_2\rangle\rangle$  is the eigenstate of  $S_i^2$ ,  $S_j^2$ ,  $S_i^z$ , and  $S_j^z$  which satisfies

$$S_i^2 |m_1, m_2\rangle\rangle = S_j^2 |m_1, m_2\rangle\rangle = S(S+1) |m_1, m_2\rangle\rangle \quad (3.20)$$

$$S_i^z |m_1, m_2\rangle\rangle = (-S + m_1) |m_1, m_2\rangle\rangle \quad (3.21)$$

$$S_j^z |m_1, m_2\rangle\rangle = (-S + m_2) |m_1, m_2\rangle\rangle \quad (3.22)$$

and  $m_i = 0, 1, \dots, 2S$ . Since the term

$$S_{i(p)}^x S_{i(p)}^x + S_{i(p)}^y S_{i(p)}^y \quad (3.23)$$

in (3.13) corresponds to transferring a “particle” (i.e., a vertex with a value of 1) from one side (left or right) to the opposite, a matrix element

$$\langle\langle m_{ul}, m_{tr} | \rho(p) | m_{bl}, m_{br} \rangle\rangle \quad (3.24)$$

with an odd value of  $m_{ul} - m_{bl} = -m_{tr} + m_{br}$  must be an odd function of  $J_x = J_y = J$ . For the same reason, if  $m_{ul} - m_{bl}$  is even, the matrix element must be an even function of  $J$ . Therefore, changing the sign of  $K$  and  $\lambda$  at the same time, which is equivalent to changing the sign of  $J_x = J_y$  with  $J_z$  fixed, may change the sign of some of matrix elements, not the absolute values. Therefore, because of (3.11) and (3.18), (3.16) follows.

In (3.10), we employed a method of decomposition in which the two-body interaction in the original Hamiltonian corresponds to a four-body interaction on the plaquettes. It is possible to decompose the partition function in such a way that the local units are cubes, for example, instead of plaquettes. This possibility will be briefly discussed in Section 6.

### 3.2. Notation and Definitions

In Section 2 we introduced notation and definitions to present two pedagogical examples. Although most essential concepts have been presented there already, we need to extend them to describe the algorithm for more general cases in an unambiguous fashion. In this subsection, we will summarize and generalize those notations and definitions.

A graph  $G = (V_G, E_G)$  is defined by two sets  $V_G$  and  $E_G$ . We call elements of  $V_G$  *vertices* and elements of  $E_G$  *edges*. In what follows, two symbols  $v$  and  $e$ , often with subscripts and superscripts, denote a vertex and an edge. Every edge has two attributes: *endpoints* (two vertices that the edge connects) and *color* (green or red).

A graph  $G' = (V_{G'}, E_{G'})$  is called *subgraph* of  $G = (V_G, E_G)$  if  $V_{G'} \subset V_G$ ,  $E_{G'} \subset E_G$ . A *path*  $P$  is a special graph whose vertices and edges can be numbered in such a way that the  $i$ th edge’s endpoints are the  $i$ th

and the  $(i + 1)$ th vertices. Two vertices  $v$  and  $v'$  are called *connected* in  $G$  when a path in  $G$  exists whose first and last vertices are  $v$  and  $v'$ . A path is called *green* when it contains an even number of red edges. Otherwise, it is called *red*. If the first and the last vertices of a path are the same, i.e., if the path is closed, it is called a *loop*. A subgraph  $C$  of  $G$  is called a *cluster* if any two vertices in  $C$  are connected in  $G$ , any edge in  $G$  connecting two vertices in  $C$  belongs to  $C$ , and no vertices or edges can be added to  $C$  without violating the two previous conditions. A *union* of two graphs  $G$  and  $G'$  is defined by

$$G \cup G' \equiv (V_G \cup V_{G'}, E_G \cup E_{G'}) \quad (3.25)$$

In Section 2.4 we saw that a special type of edge, namely, green vertical or red horizontal, played a particularly important role in representing the local weight  $w(\mathbf{n}(p))$  in the form (2.58). We will see in the next subsection that different types of edges are important for different types of anisotropy of the models. Therefore, it is useful to classify open paths and edges into several categories. They are classified according to the relative locations of their endpoints. A *vertical* path connects vertices whose spatial indices are the same but have different temporal indices, a *diagonal* path connects vertices whose spatial and temporal indices are different, a *horizontal* path connects vertices whose spatial indices are different but have the same temporal indices, and a *recurrent* path connects vertices whose spatial and temporal indices are the same. The vertical paths and the diagonal paths are both called *temporal*, whereas the horizontal paths and the recurrent paths are called *spatial*. Some of these paths are special for ferromagnetic models and some others for antiferromagnetic ones. We will call a green vertical or a green diagonal path *ferromagnetic*, and a green vertical or red horizontal path *antiferromagnetic*. A green vertical path is both ferromagnetic and antiferromagnetic, while a recurrent path is neither. Since a graph with only one edge and two vertices that the edge connects can be viewed as a special open path, we can apply the above classification to edges as well. These definitions are summarized in Fig. 4.

As we saw in the last section, we define not only graphs, but also a set of vertex variables  $\mathbf{n}$  on the same vertex set as graphs. With these one-bit variables  $\mathbf{n}$  and graphs  $G$ , we can express the partition function in the form of (2.1) with (2.3). These equations can be viewed as representing a system that consists of vertex variables and graphs interacting with each other. From (2.4), it is clear that the function  $\Delta(\mathbf{n}, G)$  is the "interaction" between vertex variables and graphs. In the two examples presented in the last section, we saw two of its special forms. Here, we restate its definition given in Section 2.4 for the  $S = 1$  Heisenberg model in a clearer fashion and



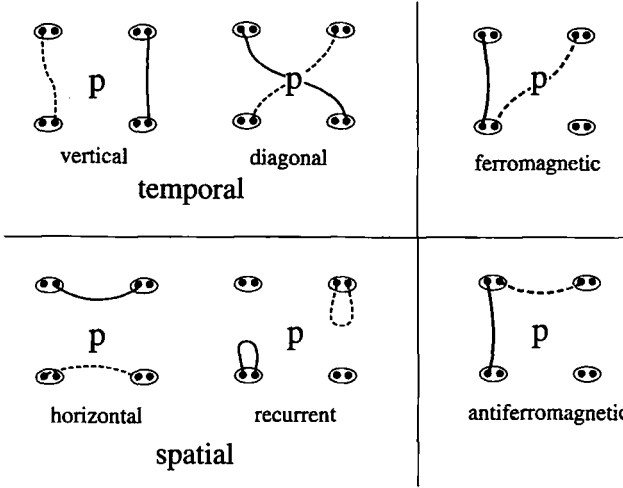


Fig. 4. The types of edges. Red edges are represented by dashed curves and green edges are represented by solid curves.

show a few of its important properties. The function  $\Delta(\mathbf{n}(V_G), G)$  is defined by

$$\Delta(\mathbf{n}(V_G), G) \equiv \prod_{e \in E_G} \varepsilon_e(\mathbf{n}(e)) \tag{3.26}$$

with a one-bit function  $\varepsilon_e(\mathbf{n}(e))$  which takes value 1 if  $e$  is green and the vertex variables take the same value at the two endpoints, or if  $e$  is red and the variables take different values. Otherwise, the function takes the value 0.

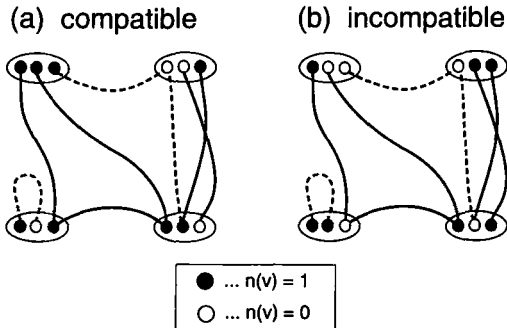


Fig. 5. Two local configurations (a) compatible and (b) incompatible with a given graph. The vertices for which the vertex variables are 1 are represented by solid circles, the other by open circles.

If  $\Delta(\mathbf{n}(V_G), G) = 1$ , the graph  $G$  and the state  $\mathbf{n}(V_G)$  are called *compatible* with each other. Examples for compatible and incompatible cases are shown in Fig. 5. It is obvious from the definition that  $\Delta(\mathbf{n}(V_G), G)$  is non-zero if  $n_v = n_{v'}$  for any pair of vertices  $v$  and  $v'$  connected by a green path and if  $n_v = \bar{n}_{v'} \equiv 1 - n_{v'}$  for any pair of vertices connected by a red path. For two graphs  $G$  and  $G'$ , the following identities hold:

$$\begin{aligned} \Delta(\mathbf{n}(V_G), G) \Delta(\mathbf{n}(V_{G'}), G') &= \prod_{e \in E_G} \varepsilon_e(\mathbf{n}(e)) \prod_{e \in E_{G'}} \varepsilon_e(\mathbf{n}(e)) \\ &= \prod_{e \in E_G \setminus E_{G'}} \varepsilon_e(\mathbf{n}(e)) \prod_{e \in E_{G'} \setminus E_G} \varepsilon_e(\mathbf{n}(e)) \prod_{e \in E_G \cap E_{G'}} [\varepsilon_e(\mathbf{n}(e))]^2 \\ &= \prod_{e \in E_{G \cup G'}} \varepsilon_e(\mathbf{n}(e)) = \Delta(\mathbf{n}(V_{G \cup G'}), G \cup G') \end{aligned} \tag{3.27}$$

In deriving this identity, we have used the fact that  $\varepsilon^2 = \varepsilon$ . This identity is useful and will be used in what follows. As a corollary, we obtain

$$\Delta(\mathbf{n}(V_G), G) = \prod_{C \in \mathcal{C}(G)} \Delta(\mathbf{n}(V_C), C) \tag{3.28}$$

where  $\mathcal{C}(G)$  is the set of clusters in  $G$  and  $V_C$  is the vertex set of a graph  $C$ . The proof is straightforward using (3.27) if we note that

$$\bigcup_{C \in \mathcal{C}(G)} C = G \tag{3.29}$$

### 3.3. Local Weight Equation

Given the  $\Delta$  functions defined in the last subsection, our task is to find a set of graphs  $\Gamma(p)$  and the weight  $v(G(p))$  of graphs that satisfy the local weight equation

$$w(\mathbf{n}(p)) = \sum_{G(p) \in \Gamma(p)} v(G(p)) \Delta(\mathbf{n}(p), G(p)) \tag{3.30}$$

for various given weights  $w(\mathbf{n}(p))$ . Once we obtain the form (3.30), by substituting it into (3.10), we can express the Boltzmann weight as

$$W(\mathbf{n}) = \prod_{u \in \mathcal{U}} \sum_{G(u) \in \Gamma(u)} v(G(u)) \Delta(\mathbf{n}(u), G(u)) \tag{3.31}$$

where  $v(G(l))$  for  $l \in \mathcal{U}_l$  is 1 and  $\Delta(\mathbf{n}(l), G(l))$  is  $\delta_l(\mathbf{n}(l))$ , i.e.,  $\Gamma(l)$  contains only one graph  $G(l)$  that consists of  $2S$  green vertical edges each of which connects two vertices with the same spatial indices  $i$  and vertex indices  $\mu$ . With the use of (3.27), the above equation can be written as

$$\begin{aligned} W(\mathbf{n}) &= \sum_{G \in \Gamma} \left[ \prod_{u \in \mathcal{U}} v(G(u)) \right] \left[ \prod_{u \in \mathcal{U}} \Delta(\mathbf{n}(u), G(u)) \right] \\ &= \sum_{G \in \Gamma} V(G) \Delta(\mathbf{n}, G) \end{aligned} \tag{3.32}$$

where  $V(G) \equiv \prod_u v(G(u))$  and  $G \equiv \bigcup_u G(u)$ . Since this representation has the form discussed in Section 2.1, we can construct a cluster algorithm following the prescription given in Section 2.2.

In Section 2.4 we defined graphs  $G(p)$  whose vertex set is a plaquette with eight vertices. We also defined a special set of graphs in (2.59) over which the summation in (3.30) should be taken. Obviously, for the general  $XXZ$  models, we have to define  $G(p)$  on a plaquette of  $8S$  vertices, not on one of only eight vertices. Furthermore, as we will see in the next section, we have to take different types of sets as  $\Gamma(p)$  in (3.30). In this subsection, we present the essential result of the next section, i.e., we present explicitly the set of graphs  $\Gamma(p)$  with which the local weight equation (3.30) can have meaningful solution  $v(G(p))$ .

In the next section, we will find that the proper set of graphs  $\Gamma(p)$  depends on the anisotropy of the problem. Therefore, we will separately treat five distinct cases: (1)  $XY$ -like ( $|\lambda_{i,j}| < 1$ ), (2) isotropic ferromagnetic ( $\lambda_{i,j} = 1$ ), (3) isotropic antiferromagnetic ( $\lambda_{i,j} = -1$ ), (4) ferromagnetic Ising-like ( $\lambda_{i,j} > 1$ ), and (5) antiferromagnetic Ising-like ( $\lambda_{i,j} < -1$ ) anisotropies. Corresponding to these five cases, we will define five sets of graphs. All graphs belonging to any one of these five sets share a property in common. To be more specific, all five sets are included in a larger set of graphs

$$\begin{aligned} \Gamma^{XXZ}(p) &\equiv \{ G \mid V_G = p, \\ &\quad \text{“Any cluster in } G \text{ has an even number of vertices” and} \\ &\quad \text{“Any edge is green temporal or red spatial”} \} \end{aligned} \tag{3.33}$$

The first condition reflects the rule that only local configurations  $\mathbf{n}(p)$  for which  $\sum_{v \in p} n_v$  is even have a nonzero weight  $w(\mathbf{n}(p))$ . The second restriction imposed upon the edges for the  $XXZ$  model reflects the fact that  $z$  component of total magnetization commutes with the Hamiltonian and is

conserved. Flipping those edges, green temporal or red horizontal, automatically results in a state for which this conservation law holds. As we will see in the Appendix, we do not have this restriction for the  $XYZ$  model, in which  $z$  component is not conserved while the first condition is still valid.

The five subsets are defined as

$$\Gamma^{XY}(p) \equiv \{G \in \Gamma^{XXZ}(p) \mid \text{“No two edges in } G \text{ share a vertex”}\} \quad (3.34)$$

$$\Gamma^{FH}(p) \equiv \{G \in \Gamma^{XY}(p) \mid \text{“Every edge in } G \text{ is ferromagnetic”}\} \quad (3.35)$$

$$\Gamma^{AFH}(p) \equiv \{G \in \Gamma^{XY}(p) \mid \text{“Every edge in } G \text{ is antiferromagnetic”}\} \quad (3.36)$$

$$\Gamma^F(p) \equiv \{G \in \Gamma^{XXZ}(p) \mid \text{“Every edge in } G \text{ is ferromagnetic”}\} \quad (3.37)$$

$$\Gamma^{AF}(p) \equiv \{G \in \Gamma^{XXZ}(p) \mid \text{“Every edge in } G \text{ is antiferromagnetic”}\} \quad (3.38)$$

We note that

$$\Gamma^{FH} = \Gamma^{XY} \cap \Gamma^F \quad (3.39)$$

$$\Gamma^{AFH} = \Gamma^{XY} \cap \Gamma^{AF} \quad (3.40)$$

A typical graph for each case is illustrated for the  $S = 3/2$  case in Fig. 6.

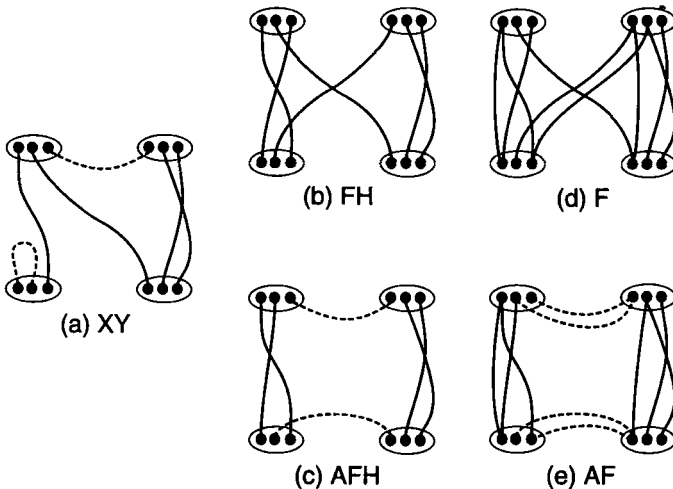


Fig. 6. Typical examples for the five classes of graphs for (a)  $XY$ -like, (b) isotropic ferromagnetic, (c) isotropic antiferromagnetic, (d) ferromagnetic Ising-like, and (e) antiferromagnetic Ising-like anisotropies.

In the next section, we will show that with these sets of graphs, at least one nonnegative set of coefficients  $v(G(p))$  exists that satisfies (3.30) for each type of anisotropy. In this paper, we do not give the explicit form of those solutions except for the special case of Heisenberg models. Instead, we will present the method to compute the coefficients.

It is interesting to consider what kind of clusters will be formed when we take the union of local graphs, assuming that all plaquettes in the system have the same type of anisotropy. One striking difference is seen between the  $XY$ -like and the Ising-like anisotropic cases. For  $XY$ -like anisotropy, in a local graph  $G(p)$ , a vertex is an endpoint of one and only one edge. This means that every vertex in  $G$  is shared by two and only two edges, when we take into account the fact that every site is shared by two local units, i.e., plaquettes or vertical links. Therefore, when we take the union graph  $G \equiv \bigcup_u G(u)$ , it consists of a number of loops without branching. Therefore, the present scheme naturally leads to a loop algorithm in the case of  $XY$ -like anisotropy. On the other hand, we generally have branching in the case of Ising-like anisotropy, because a vertex can be shared by any number of edges. Another important difference is that, in the case of  $XY$ -like and ferromagnetic Ising-like anisotropies, for any worldline in the current configuration, a loop can form that coincides with the worldline because the graphs include both vertical and diagonal edges and a worldline also consists of vertical and diagonal segments. Here a worldline is a line connecting vertices with vertex variable 1. This means that, in those cases, any worldline can vanish in a Monte Carlo step. On the other hand, in the case of antiferromagnetic Ising-like anisotropy, a worldline with diagonal segments cannot vanish because a loop which overlaps with this worldline cannot form. This may cause a problem concerning ergodicity, as we will discuss in Section 6.

#### 4. CLUSTER REPRESENTATION OF THE LOCAL BOLTZMANN FACTOR

In this section, we show that the local Boltzmann factor  $w(\mathbf{n}(p))$ , defined on a plaquette  $p$ , can be expressed by a sum of terms, each of which corresponds to a graph. In other words, the goal of this section is to prove the weight equation (3.30) has at least one meaningful solution with a set of graphs (3.34), (3.35), (3.36), (3.37), or (3.38), depending on the anisotropy of the problem. We also propose a method to compute the coefficients  $v(G(p))$ . For the general case, we will show how to compute them numerically, and for the Heisenberg models, we will provide compact formulas.

Because we will focus on only one pair of interacting points in the original lattice  $L$ , which we refer to as  $a$  and  $b$ , the corresponding lattice we consider in this section has just two lattice points when projected onto a horizontal plane. What we will discuss is the multiplication of two operators defined in the local Hilbert space on these two lattice points (in the original lattice). Because an operator in this space can be represented by a plaquette, it is sufficient to take two plaquettes, one stacked on the other, in order to discuss the multiplication. Therefore, the size of "hyperlattice" we consider here in the temporal dimension is three.

To avoid confusion, we emphasize that this hyperlattice, whose size is two in the spatial direction and three in the temporal direction, is not necessarily the subset of the hyperlattice discussed in the previous sections. The spatial indices  $a$  and  $b$ , of course, correspond to the spatial indices in the last section, because they correspond to the lattice points in the original lattice  $L$ . The index  $\mu$  also corresponds to the same index in the last section and specifies a vertex in a given site. However, the temporal indices in this section do not correspond to those in the last section. Here, they are merely labels introduced to distinguish one state vector from another. For example,  $\mathbf{n}_k$  stands for  $4S$  one-bit variables  $n_{(k,i,\mu)}$  ( $i = a, b; \mu = 1, 2, \dots, 2S$ ). The state  $|\mathbf{n}_k\rangle_2$  is an eigenvector of the  $z$ -components of the Pauli operators, i.e.,

$$\sigma_{i,\mu}^z |\mathbf{n}_k\rangle_2 = (2n_{(k,i,\mu)} - 1) |\mathbf{n}_k\rangle_2 \quad (4.1)$$

Defining  $l_k$  as a set of two sites with the temporal index  $k$ , i.e.,  $\{(k, a), (k, b)\}$ , we can rewrite  $\mathbf{n}_k$  as  $\mathbf{n}(l_k)$ . We will use this notation in what follows.

Other useful tools for the discussion are the operators whose matrix elements are given by the  $\Delta$  functions defined in Section 3.2. We define an operator  $\hat{\Delta}(G(p))$  by

$$\langle \mathbf{n}(l_t(p)) | \hat{\Delta}(G(p)) | \mathbf{n}(l_b(p)) \rangle_2 \equiv \Delta(\mathbf{n}(p), G(p)) \quad (4.2)$$

Here,  $l_t(p)$  and  $l_b(p)$  are the top and bottom links of the plaquette  $p$  as defined in Section 3.2. The operator  $\hat{\Delta}(G(p))$  depends on  $p$  only through  $l_t(p)$  and  $l_b(p)$  and does not depend on the temporal index  $k$ .

To outline the proof before going into its detail is useful. We will introduce an operation called the *contraction* of two graphs in such a way that the product of two operators  $\hat{\Delta}_{G_p}$  and  $\hat{\Delta}_{G'_p}$  is  $\hat{\Delta}_{G''}$  multiplied by a scalar factor, where  $G''$  is the graph resulting from the contraction of  $G_p$  and  $G'_p$ . In other words, the contraction of two graphs corresponds to multiplication of two operators. This relationship enables us to discuss the nature of the operators in a graph-theoretic language. The most important

feature that we will prove is that a set of operators  $O^*$  is closed with respect to multiplication. Here, the asterisk stands for  $XY$ ,  $FH$ ,  $FAH$ ,  $F$ , or  $AF$ , corresponding, respectively, to  $XY$ -like, isotropic-ferromagnetic, isotropic-antiferromagnetic, ferromagnetic-Ising-like, or antiferromagnetic-Ising-like anisotropy, and  $O^*$  is the set of operators  $\hat{\delta}$ , which can be expressed as a sum of elements of  $\Gamma^*$  multiplied by some nonnegative coefficients  $v(G(p))$ . Formally,

$$O^* \equiv \left\{ \hat{\delta} \mid \exists v(G(p)) \geq 0 \left[ \hat{\delta} = \sum_{G(p) \in \Gamma^*(p)} v(G(p)) \hat{A}(G(p)) \right] \right\} \quad (4.3)$$

The sets of graphs  $\Gamma^*$  are the ones defined in Section 3.3. Then, we can show that  $\tilde{\rho}$  in (3.12) belongs to one of these sets of operators because (1) the product of projection operators in (3.12) belongs to all the above sets of operators, (2) all the operators  $\hat{A}(\mu, \nu)$  in (3.13) become elements of one of the above sets of operators when a scalar operator  $x\hat{I}$  is added to them ( $\hat{I}$  is the identity operator and  $x$  is some real number), (3) all the above sets of operators are closed with respect to the multiplication of two elements, the multiplication of an element by a nonnegative real number, and the addition of two elements, (4) the number of graphs which correspond to distinct  $\hat{A}$  operators is finite, and (5) the Taylor expansion series of  $\tilde{\rho}$  with respect to  $K_p$  converges. In short, we will show that a set of nonnegative coefficients  $v(G(p))$  exists that satisfies

$$\tilde{\rho} = \sum_{G(p) \in \Gamma^*} v(G(p)) \hat{A}(G(p)) \quad (4.4)$$

When expressed as an equation between matrix elements, (4.4) reduces to (3.30).

### 4.1. The $XY$ -Like Anisotropy ( $|\lambda_p| \leq 1$ )

The first step in proving the statement (4.4) in the case where  $|\lambda_p| \leq 1$  is to note the following identity:

$$\begin{aligned} \hat{A}(\mu, \nu) &= \sigma_{a,\mu}^x \sigma_{b,\nu}^x + \sigma_{a,\mu}^y \sigma_{b,\nu}^y + \lambda \sigma_{a,\mu}^z \sigma_{b,\nu}^z \\ &= -1 + (1 + \lambda) \hat{A}(\mu, \nu) + (1 - \lambda) \hat{B}(\mu, \nu) \end{aligned} \quad (4.5)$$

where  $\hat{A}$  and  $\hat{B}$  are operators defined by

$$\begin{aligned} &\langle \mathbf{n}(l_2) | \hat{A}(\mu, \nu) | \mathbf{n}(l_1) \rangle_2 \\ &= \delta(n_{(2,a,\mu)}, n_{(1,b,\nu)}) \delta(n_{(2,b,\nu)}, n_{(1,a,\mu)}) \\ &\quad \times \prod_{\substack{\alpha \neq \mu \\ \beta \neq \nu}} \delta(n_{(2,a,\alpha)}, n_{(1,a,\alpha)}) \delta(n_{(2,b,\beta)}, n_{(1,b,\beta)}) \end{aligned} \quad (4.6)$$

$$\begin{aligned}
 &\langle \mathbf{n}(l_2) | \hat{B}(\mu, \nu) | \mathbf{n}(l_1) \rangle_2 \\
 &= \delta(\bar{n}_{(2,a,\mu)}, n_{(2,b,\nu)}) \delta(n_{(1,a,\mu)}, \bar{n}_{(1,b,\nu)}) \\
 &\quad \times \prod_{\substack{\alpha \neq \mu \\ \beta \neq \nu}} \delta(n_{(2,a,\alpha)}, n_{(1,a,\alpha)}) \delta(n_{(2,b,\beta)}, n_{(1,b,\beta)}) \tag{4.7}
 \end{aligned}$$

Here, the symbol  $\bar{n}$  denotes  $1 - n$ . It is easy to see that the operator  $\hat{A}(\mu, \nu)$  can be expressed as

$$\hat{A}(\mu, \nu) \equiv \hat{A}(A(\mu, \nu)) \tag{4.8}$$

where  $A(\mu, \nu)$  is a graph shown in Fig. 7a. Similarly, for the operator  $\hat{B}_{(\mu,\nu)}$  we have

$$\hat{B}(\mu, \nu) \equiv \hat{A}(B(\mu, \nu)) \tag{4.9}$$

with the graph  $B(\mu, \nu)$  shown in Fig. 7b.

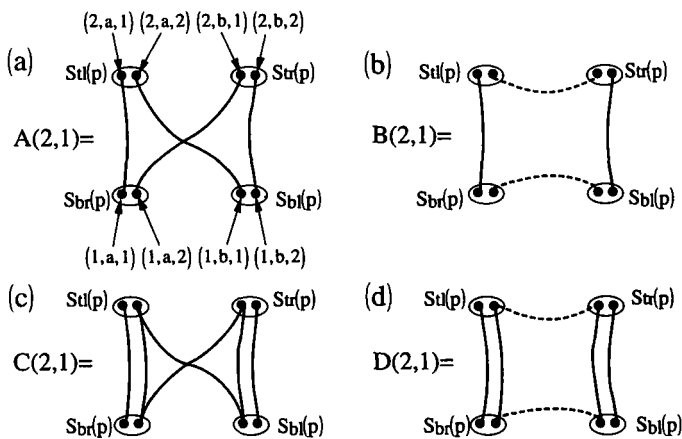


Fig. 7. Examples for the graphs which define (a)  $\hat{A}$ , (b)  $\hat{B}$ , (c)  $\hat{C}$ , and (d)  $\hat{D}$  in the case of  $S = 1$ . The Hamiltonian of the  $XXZ$  model can be expressed as a sum of these four types of operators. For models with  $XY$ -like anisotropy, graphs of type (a) and (b) are sufficient to express the Hamiltonian, whereas (b) is replaced by (c) in the case of ferromagnetic Ising-like anisotropy and by (d) in the case of antiferromagnetic Ising-like anisotropy. The ovals are sites, the solid circles are vertices, and the curves are edges. In the graphs for the  $XXZ$  models, all spatial edges are “red” and all temporal edges are “green.”



Using the identity (4.5), we can rewrite the exponential operator in (3.13) as

$$\begin{aligned} \rho &= \exp\left(\sum_{\mu,\nu} K_p A^{(\mu,\nu)}\right) \\ &= \exp\left\{\frac{K_p}{4} \sum_{\mu,\nu} [-1 + (1 + \lambda_p) \hat{A}(\mu, \nu) + (1 - \lambda_p) \hat{B}(\mu, \nu)]\right\} \\ &= \exp(-K_p S^2) \exp\left\{\frac{K_p}{4} \sum_{\mu,\nu} [(1 + \lambda_p) \hat{A}(\mu, \nu) + (1 - \lambda_p) \hat{B}(\mu, \nu)]\right\} \end{aligned} \quad (4.10)$$

The exponent of the last factor

$$\frac{K_p}{4} \sum_{\mu,\nu} [(1 + \lambda_p) \hat{A}(\mu, \nu) + (1 - \lambda_p) \hat{B}(\mu, \nu)] \quad (4.11)$$

is an element of  $O^{XY}$  because  $K_p$ ,  $1 + \lambda_p$ , and  $1 - \lambda_p$  are nonnegative and  $O^{XY}$  is obviously closed with respect to the addition of two elements and the multiplication of an element by a nonnegative number. Additionally, both  $A(\mu, \nu)$  and  $B(\mu, \nu)$  belong to  $\Gamma^{XY}(p)$ , or equivalently,  $\hat{A}(\mu, \nu)$  and  $\hat{B}(\mu, \nu)$  belong to  $O^{XY}$ .

On the other hand, matrix elements of the projection operators in (3.12) can simply be written in the following form:

$$\begin{aligned} &\langle \mathbf{n}(l_2) | P_a P_b | \mathbf{n}(l_1) \rangle_2 \\ &= \frac{1}{[(2S)!]^2} \sum_{\pi, \pi'} \prod_{\mu=1}^{2S} \delta(n_{(2,a,\pi(\mu))}, n_{(1,a,\mu)}) \delta(n_{(2,b,\pi'(\mu))}, n_{(1,b,\mu)}) \end{aligned} \quad (4.12)$$

where  $\pi$  and  $\pi'$  stand for permutations of the set  $\{1, 2, \dots, 2S\}$  and the summation is taken over all possible permutations. Therefore, this product of projection operators can be viewed as a sum of  $\mathcal{A}$  operators for which the graph consists of  $4S$  green vertical edges which connect  $(1, a, \mu)$  to  $(2, a, \pi(\mu))$  and  $(1, b, \mu)$  to  $(2, b, \pi'(\mu))$ . Therefore, the operator  $P_a P_b$  belongs to all five sets of operators defined in (4.3). To summarize, we have shown that  $\tilde{\rho}$  in (3.17) can be expressed in the form

$$\tilde{\rho} = \hat{Y} e^{\hat{X}} \hat{Y} \quad (4.13)$$

with two elements  $\hat{X}$  and  $\hat{Y}$  of  $O^{XY}$ .

We now consider an arbitrary set  $\Gamma(p)$  of graphs defined on a plaquette  $p$  and its two arbitrary elements,  $G_1$  and  $G_2$ . If there exists an element  $G$  in  $\Gamma(p)$  for which

$$\hat{\Delta}(G_1) \hat{\Delta}(G_2) = 2^m \hat{\Delta}(G) \quad (4.14)$$

holds with some nonnegative integer  $m$ , we call  $\Gamma(p)$  *closed* with respect to multiplication. With this definition, we will prove the following statement:

**Lemma 1.**  $\Gamma^{XY}(p)$  is closed with respect to multiplication.

If we assume Lemma 1, it is obvious that  $O^{XY}$  defined in (4.3) is closed with respect to multiplication.

We first assume that the above lemma is true and show that  $\tilde{\rho}$  expressed in the form (4.13) belongs to  $O^{XY}$ . To this end, we consider the Taylor expansion of  $\exp(\hat{X})$  with respect to  $\hat{X}$ . We then neglect the terms of the  $(n+1)$ th order and higher. The  $k$ th-order term in the Taylor expansion is  $(1/k!) \hat{X}^k$  and obviously belongs to  $O^{XY}$ . So does the  $k$ th-order term of  $\tilde{\rho}$ , i.e.,  $(1/k!) \hat{Y} \hat{X}^k \hat{Y}$ . Thus the  $n$ th-order approximant of  $\tilde{\rho}$  also belongs to  $O^{XY}$ . In other words, the operator  $\tilde{\rho}$  in (3.12) can be approximated up to the  $n$ th order by  $\tilde{\rho}^{(n)} \in O^{XY}$ . Namely,

$$\tilde{\rho} \approx \tilde{\rho}^{(n)} \equiv \sum_{G(p) \in \Gamma^{XY}(p)} v^{(n)}(G(p)) \hat{\Delta}(G(p)) \quad (4.15)$$

with nonnegative coefficients  $v^{(n)}(G(p))$ . Here we note that the number of distinct elements of  $\Gamma^{XY}(p)$  is finite. We also note that the series  $v^{(n)}(G(p))$  ( $n = 1, 2, 3, \dots$ ) is monotonically convergent to a nonnegative value because the contribution from each order term to  $v^{(n)}(G)$  is nonnegative. These facts and (4.15) make it obvious that in the limit of  $n \rightarrow \infty$  the operator  $\tilde{\rho}^{(n)}$  can still be expressed in the form (4.15), i.e., it belongs to  $O^{XY}$ . Thus we get the following theorem:

**Theorem 1.** If  $|\lambda_p|$  is not greater than unity, the operator  $\tilde{\rho}$  can be decomposed into a sum of  $\Delta$  operators of the form

$$\tilde{\rho} = \sum_{G(p) \in \Gamma^{XY}(p)} v(G(p)) \hat{\Delta}(G(p)) \quad (4.16)$$

with nonnegative coefficients  $v(G(p))$ .

Taking the matrix elements of the both sides of the above equation, we have

$$w(\mathbf{n}(p)) = \sum_{G(p) \in \Gamma^{XY}(p)} v(G(p)) \Delta(\mathbf{n}(p), G(p)) \quad (4.17)$$

This is the statement that we wanted to show for  $XY$ -like anisotropic models.

### 4.2. Contraction Operation and Proof of Lemma 1

Now we prove Lemma 1. We consider two arbitrary elements  $G_1$  and  $G_2$  of  $\Gamma_p^{XY}$ . Since  $\hat{A}(G_1)$  and  $\hat{A}(G_2)$  do not depend on a particular plaquette, as we discussed above, we assume that  $G_1$  is defined on  $p_1 \equiv l_1 \cup l_2$  and  $G_2$  on  $p_2 \equiv l_2 \cup l_3$  (see Fig. 8).

Using (3.27), we can express the matrix element of  $\hat{A}(G_2) \hat{A}(G_1)$  as

$$\begin{aligned}
 & \langle \mathbf{n}(l_3) | \hat{A}(G_2) \hat{A}(G_1) | \mathbf{n}(l_1) \rangle_2 \\
 &= \sum_{\mathbf{n}(l_2)} \langle \mathbf{n}(l_3) | \hat{A}(G_2) | \mathbf{n}(l_2) \rangle_2 \langle \mathbf{n}(l_2) | \hat{A}(G_1) | \mathbf{n}(l_1) \rangle_2 \\
 &= \sum_{\mathbf{n}(l_2)} \mathcal{A}(\mathbf{n}(p_2), G_2) \mathcal{A}(\mathbf{n}(p_1), G_1) \\
 &= \sum_{\mathbf{n}(l_2)} \mathcal{A}(\mathbf{n}(l_3 \cup l_2 \cup l_1), G_1 \cup G_2) \tag{4.18}
 \end{aligned}$$

Any cluster in the union  $G_1 \cup G_2$  is a single path because a vertex in  $l_1$  can only be an endpoint of one and only one of the edges in  $G_1$ , a vertex in  $l_3$  can only be in  $G_2$ , and a vertex in  $l_2$  is shared by an edge in  $G_1$  and another in  $G_2$ . Therefore, the clusters (in this case, paths) have the following properties:

**Property 1.** The vertices on the top link ( $l_3$ ) and on the bottom link ( $l_1$ ), which correspond to  $(3, i, \mu)$  and  $(1, i, \mu)$ , are endpoints of paths. They cannot be intermediate points.

**Property 2.** The vertices on the middle link ( $l_2$ ), which correspond to  $(2, i, \mu)$ , are intermediate points of paths. They cannot be endpoints.

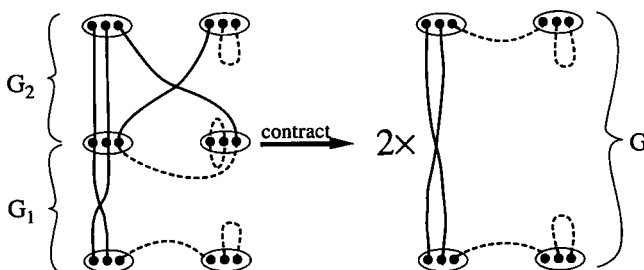


Fig. 8. Two graphs  $G_1 \in \Gamma^{XY}(p_1)$  and  $G_2 \in \Gamma^{XY}(p_2)$  and their contraction resulting in a graph  $G$  with a factor two. The factor comes from the small loop in the right middle part of the graph on the left-hand side.

**Property 3.** If an edge belongs to the lower graph ( $G_1$ ), the adjacent edge(s) in the same path belongs to the upper graph ( $G_2$ ), and vice versa.

These elemental properties lead to the following properties:

**Property 1'.** A temporal edge can appear only as the first or the last edge in a path.

**Property 2'.** A spatial edge can appear only as the intermediate edge in a path.

From Property 3, the following properties are derived:

**Property 4.** The length of any loop is even.

Here, the length of a path is the number of its edges. Similar to Property 4, for an open spatial path we have the following property:

**Property 5.** The length of any open spatial path is odd.

This is because the first and the last edges in such an open path belong to the same graph ( $G_1$  or  $G_2$ ). For the same reason, we have the following:

**Property 6.** The length of any temporal path is even.

Using (3.28), we can express the matrix elements (4.18) in terms of contributions from these paths as

$$\langle \mathbf{n}(l_3) | \hat{A}(G_2) \hat{A}(G_1) | \mathbf{n}(l_1) \rangle_2 = \sum_{\mathbf{n}(l_2)} \prod_P \Delta(\mathbf{n}(V_P), P) = \prod_P f(P) \quad (4.19)$$

where  $P$  stands for a path,  $V_P$  is the vertex set of  $P$ , and  $f(P)$  is the contribution from the path  $P$  defined by

$$f(P) \equiv \sum_{\mathbf{n}(V_P \cap l_2)} \Delta(\mathbf{n}(V_P), P) \quad (4.20)$$

Note that the paths  $P$  in (4.19) can be classified into three categories: (1) loops, (2) open spatial paths, and (3) open temporal paths. In what follows, we discuss contributions from these three types of paths separately.

We first consider the contribution in (4.19) from loops, i.e.,  $f(P)$  for a loop  $P$ . Because there are no endpoints, all vertices are on the middle link  $l_2$  (Property 1). Therefore, all edges are red because of the definition of  $\Gamma^{XZX}(p)$ . Hence, we can write the contribution from a loop  $P$  whose length is  $2m$  (Property 4) in the following way:

$$f(P) = \sum_{n_1^P} \sum_{n_2^P} \cdots \sum_{n_{2m}^P} \delta(n_1^P, \bar{n}_2^P) \delta(\bar{n}_2^P, n_3^P) \cdots \delta(n_{2m-1}^P, \bar{n}_{2m}^P) \delta(\bar{n}_{2m}^P, n_1^P) \quad (4.21)$$

where  $n_i^P$  is the value of the one-bit function  $\mathbf{n}$  on the  $i$ th vertex in the path  $P$ . The product of the  $\delta$  functions is nonzero if and only if

$$n_1^P = \bar{n}_2^P = \dots = \bar{n}_{2m}^P \tag{4.22}$$

Therefore, a contribution from a closed path is a mere numerical factor of 2, i.e.,  $f(P) = 2$ .

Next we consider the contribution from a spatial path. If the length of the path is one, the edge must be spatial and there is no intermediate vertex. Therefore, the contribution is  $\delta(n_1^P, \bar{n}_2^P)$  and we will assume the length is larger than one. For such a path, the first and the last edges are temporal and all others are spatial, because of Properties 1 and 2. The contribution can then be written as

$$\begin{aligned} f(P) &= \sum_{n_2^P} \sum_{n_3^P} \dots \sum_{n_{2m-1}^P} \delta(n_1^P, n_2^P) \delta(n_2^P, \bar{n}_3^P) \delta(\bar{n}_3^P, n_4^P) \\ &\quad \dots \delta(n_{2m-2}^P, \bar{n}_{2m-1}^P) \delta(\bar{n}_{2m-1}^P, \bar{n}_{2m}^P) \\ &= \delta(n_1^P, \bar{n}_{2m}^P) \end{aligned} \tag{4.23}$$

Therefore, by tracing out the intermediate variables, this path is transformed into a red spatial edge in the resulting graph, whether the length is one or larger.

Finally we consider the contribution from a temporal path. In this case, the first and the last edges are temporal as the previous case. All other intermediate edges are spatial. Since the length is even (Property 6) in this case, the contribution is

$$\begin{aligned} f(P) &= \sum_{n_2^P} \sum_{n_3^P} \dots \sum_{n_{2m}^P} \delta(n_1^P, n_2^P) \delta(n_2^P, \bar{n}_3^P) \delta(\bar{n}_3^P, n_4^P) \\ &\quad \dots \delta(\bar{n}_{2m-1}^P, n_{2m}^P) \delta(n_{2m}^P, n_{2m+1}^P) \\ &= \delta(n_1^P, n_{2m+1}^P) \end{aligned} \tag{4.24}$$

Hence the contraction results in a green vertical or green diagonal edge, i.e., a green temporal edge. In Fig. 8, we can see examples of the three types of contribution we have discussed.

Thus we have shown that the right-hand side of (4.19) is a product of three kinds of factors: (1) the numerical factor  $2^m$ , where  $m$  is the number of loops, (2) the factor which corresponds to antiferromagnetic edges, and (3) the factor which corresponds to ferromagnetic edges. Hence Lemma 1 is proven.

In this subsection, we traced out the intermediate variables to obtain a new  $\mathcal{A}$  function with a multiplicative factor. As illustrated in Fig. 8, it can be done graphically by (1) erasing intermediate vertices, (2) replacing open paths by edges with the same color, and (3) replacing each loop by a factor 2. In what follows, we call this graphical operation a *contraction*.

### 4.3. A Numerical Method to Compute the Coefficients $v(G_p)$

The discussion based on the Taylor expansion that led to (4.15) provides us not only a proof of Theorem 1, but also a numerical method for computing  $v(G) \equiv \lim_{n \rightarrow \infty} v^{(n)}(G)$ . What we have to do is:

1. Make a table of the multiplication rules among the elements of  $\Gamma^{XY}(p)$ , i.e., a table which tells us which graph and what scalar multiplicative factor results from contraction of two arbitrary graphs in  $\Gamma^{XY}(p)$ .
2. Compute the Taylor expansion of the exponential operator in (4.10) up to the  $n$ th order using the table.
3. Compute the result of multiplication of the projection operators. Or, equivalently, symmetrize the expression obtained in the last procedure with respect to the vertex indices.

Considering the fact that expansion series of an exponential function of bounded matrices such as (4.10) generally converges quickly, the  $v^{(n)}(G(p))$  in (4.15) should be a good approximant for  $v(G(p))$  with not too large  $n$ . We remark that the above procedure does not become combinatorially difficult as  $n$  increases. The amount of computation is only proportional to  $n$ . Therefore, this procedure to compute the coefficient is reasonably practical, although it may not be optimal. We also remark that the above procedure applies to other types of anisotropy to be now discussed.

### 4.4. The Isotropic Cases ( $|\lambda_p| = 1$ )

We now consider the special cases where  $\lambda_p = 1$  or  $\lambda_p = -1$ . These cases have already been considered in the last subsection. However, we take up these cases separately because the set of graphs we must consider for the expression (4.16) is truly smaller than  $\Gamma^{XY}(p)$ . This reduction means that the set of graphs needed in the labeling process of the simulation can be simpler than in the general case. It will also be clear in the next section that in these special cases the coefficients can even be computed analytically without resorting to the numerical method discussed above. For the isotropic cases, the sets of graphs  $\Gamma^{FH}(p)$  and  $\Gamma^{AFH}(p)$  play the

same role as  $\Gamma^{XY}(p)$  did for the  $XY$ -like anisotropy. In what follows, we call a graph which belongs to  $\Gamma^{\text{FH}}(p)$  a *special ferromagnetic graph* and one which belongs to  $\Gamma^{\text{AFH}}(p)$  a *special antiferromagnetic graph*.

We can show the following for the special ferromagnetic graphs:

**Lemma 2.**  $\Gamma^{\text{FH}}(p)$  is closed with respect to multiplication.

It is sufficient to prove that the following property exists in the union graph defined on  $p_1 \cup p_2$ :

**Property 7.** Given a graph defined on two plaquettes sharing a link, if all edges in a path are ferromagnetic, the path is also ferromagnetic.

As we did in the last subsection, we consider two plaquettes stacked on top of each other. In the present case, we consider a special ferromagnetic graph defined on each plaquette. To show Property 7, it is sufficient to prove that we can obtain only temporal paths from ferromagnetic edges and that those paths are green. Because of Property 1', any possible path in the stacked special ferromagnetic graphs consists of two temporal edges: one in the lower graph and the other in the upper graph. Therefore, the resulting path must be temporal. Because these two edges are both green, the resulting path is also green. Thus, Lemma 2 is proven. As a corollary, we can easily show that  $O^{\text{FH}}$  is closed with respect to multiplication.

A statement similar to Lemma 2 holds for the antiferromagnetic graphs:

**Lemma 3.**  $\Gamma^{\text{AFH}}(p)$  is closed with respect to multiplication.

This is equivalent to the following statement.

**Property 8.** Given a graph defined on two plaquettes sharing a link, if all edges in a path are antiferromagnetic, the path is also antiferromagnetic.

It is sufficient to prove that there are no diagonal or recurrent paths in the stacked special antiferromagnetic graphs and that all horizontal paths are red while all vertical paths are green. First, we assume that a path  $P$  is diagonal. Because all edges are either horizontal or vertical, we must then have an odd number of horizontal edges and two vertical edges in  $P$ . Therefore, the length of  $P$  must be odd. However, this contradicts Property 6, so there are no diagonal paths. Next, we assume that a path  $P$  is recurrent. With the same reasoning as above, the length of  $P$  must be even. However, this contradicts Property 5, so no recurrent edge can appear. As for the colors of the resulting path, horizontal ones must be red while vertical ones must be green because of Lemma 1 and the fact that  $\Gamma_p^{\text{AFH}} \subset \Gamma_p^{XY}$ . Thus, Lemma 3 is proven. It follows that  $O^{\text{AFH}}$  is closed with respect to multiplication.

Because the graphs for the projection operators (4.12) belong to  $\Gamma^{\text{FH}}(p)$ , by using Lemma 2 and following the same line of argument that proved Theorem 1, we can expand the operator  $\tilde{\rho}$  in terms of ferromagnetic  $\mathcal{A}$  operators:

**Theorem 2.** The ferromagnetic operator  $\tilde{\rho}$  can be decomposed into a sum of  $\mathcal{A}$  operators of the form

$$\tilde{\rho} = \sum_{G(p) \in \Gamma^{\text{FH}}(p)} v(G(p)) \hat{\mathcal{A}}(G(p)) \tag{4.25}$$

with nonnegative coefficients  $v(G(p))$ .

By the same argument, using Lemma 3, we have the following theorem for the antiferromagnetic operator  $\tilde{\rho}$ :

**Theorem 3.** The antiferromagnetic operator  $\tilde{\rho}$  can be decomposed into a sum of  $\mathcal{A}$  operators of the form

$$\tilde{\rho} = \sum_{G(p) \in \Gamma^{\text{AFH}}(p)} v(G(p)) \hat{\mathcal{A}}(G(p)) \tag{4.26}$$

with nonnegative coefficients  $v(G(p))$ .

### 4.5. The Ferromagnetic Ising-Like Anisotropy ( $\lambda \geq 1$ )

In the case of Ising-like anisotropy, we cannot use expression (4.5) as a starting point, because the coefficient  $(1 - \lambda)$  may be negative. Therefore, instead of (4.5), we consider the following identity with positive coefficients as the new starting point for the ferromagnetic case ( $\lambda \geq 1$ ):

$$\sigma_{a,\mu}^x \sigma_{b,\nu}^x + \sigma_{a,\mu}^y \sigma_{b,\nu}^y + \lambda \sigma_{a,\mu}^z \sigma_{b,\nu}^z = -\lambda + 2\hat{\mathcal{A}}(\mu, \nu) + 2(\lambda - 1) \hat{\mathcal{C}}(\mu, \nu) \tag{4.27}$$

where the new operator  $\hat{\mathcal{C}}(\mu, \nu)$  is defined by

$$\begin{aligned} &\langle \mathbf{n}(l_2) | \hat{\mathcal{C}}(\mu, \nu) | \mathbf{n}(l_1) \rangle_2 \\ &= \delta(n_{(2,a,\mu)}, n_{(1,a,\mu)}) \delta(n_{(2,b,\nu)}, n_{(1,b,\nu)}) \delta(n_{(2,a,\mu)}, n_{(1,b,\nu)}) \delta(n_{(2,b,\nu)}, n_{(1,a,\mu)}) \\ &\times \prod_{\alpha \neq \mu} \delta(n_{(2,a,\alpha)}, n_{(1,a,\alpha)}) \prod_{\beta \neq \nu} \delta(n_{(2,b,\beta)}, n_{(1,b,\beta)}) \end{aligned} \tag{4.28}$$

The graph  $C(\mu, \nu)$  which expresses this operator is shown in Fig. 7c. With this graph, we can simply write  $\hat{\mathcal{C}}(\mu, \nu) \equiv \hat{\mathcal{A}}(C(\mu, \nu))$ . Obviously, the new graph does not correspond to any graph in  $\Gamma^{XY}(p)$ , because some of the vertices are shared by multiple edges. However, it still belongs to  $\Gamma^{XXZ}(p)$ . In fact,  $A(\mu, \nu)$  and  $C(\mu, \nu)$  both belong to  $\Gamma^F(p)$  defined in (3.37). In what



follows, we call such a graph *ferromagnetic*. We also call the corresponding  $\Delta$  operator ferromagnetic. The difference from  $\Gamma_p^{\text{FH}}$  is that for  $\Gamma_p^{\text{F}}$  there is no restriction on the number of edges sharing a vertex.

For the ferromagnetic graphs, we can show the following statement:

**Lemma 4.**  $\Gamma^{\text{F}}(p)$  is closed with respect to multiplication.

We will consider the contraction of these operators. We first take two ferromagnetic graphs stacked on top of one another. This time a cluster is not necessarily a path. In addition, not all the paths satisfy Property 3. We will call paths that satisfy Properties 1–3 *alternating* paths. For alternating paths all the properties (Properties 1–8) hold since Properties 4–8 are derived from Properties 1–3. In particular, Property 7 holds also for these alternating paths. It is important to notice that any ferromagnetic graph is equal to the union of all the alternating paths in it. Since the alternating paths overlap each other in the present case, additional considerations are needed to find the contribution from the paths when they are contracted. The result is, however, the same: a ferromagnetic alternating path contracts to a ferromagnetic edge.

To see this, we resort to an example rather than rigorous argument. The example is shown in Fig. 9. In Fig. 9a, a typical graph is shown in the case of  $S = 1$ . There are three clusters in the graph. The result of the contraction is merely the product of corresponding three factors. Since the contributions of the left and the right clusters are trivial, we focus on the middle cluster, whose vertex set is  $\{v_0, v_1, v_2, v_3, v_4\}$ . There are four alternating paths in this cluster. Their edge sets are  $\{e_1, e_3\}$ ,  $\{e_1, e_4\}$ ,  $\{e_2, e_3\}$ , and  $\{e_2, e_4\}$ . Correspondingly, the contraction of this cluster results in

$$\begin{aligned} & \sum_{n_0} \delta(n_0, n_1) \delta(n_0, n_2) \delta(n_0, n_3) \delta(n_0, n_4) \\ &= \sum_{n_0} [\delta(n_1, n_0) \delta(n_0, n_3)] [\delta(n_1, n_0) \delta(n_0, n_4)] \\ & \quad \times [\delta(n_2, n_0) \delta(n_0, n_3)] [\delta(n_2, n_0) \delta(n_0, n_4)] \\ &= \delta(n_1, n_3) \delta(n_1, n_4) \delta(n_2, n_3) \delta(n_2, n_4) \end{aligned} \tag{4.29}$$

where  $n_i \equiv n_{v_i}$ . The last expression in the above equation is represented by Fig. 9b. It is easy to extract essential points from this example and construct a rigorous proof. Thus we obtain Lemma 4.

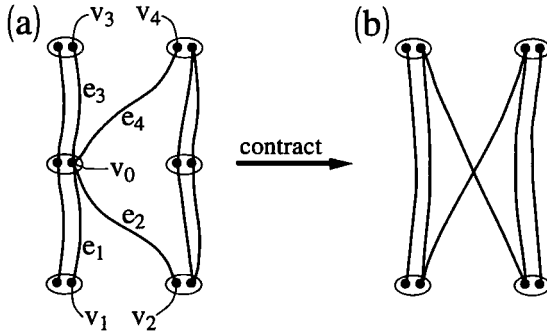


Fig. 9. An example of contraction of two ferromagnetic graphs.

Combining Lemma 4 and the expression (4.27), we obtain the following theorem:

**Theorem 4.** For  $\lambda$  greater than or equal to 1, we can expand the operator  $\tilde{\rho}$  in the form

$$\tilde{\rho} = \sum_{G(p) \in \mathcal{F}^{\mathcal{F}(p)}} v(G(p)) \hat{\Delta}(G(p)) \tag{4.30}$$

with nonnegative coefficients  $v(G(p))$ .

We can compute the coefficients  $v(G(p))$  numerically in this case, too, by the procedure described in Section 4.3. The first step may seem impossible to do in the present case, where we have infinitely many distinct graphs. However, even if the number of graphs is infinite, it is still possible that the number of graphs which correspond to distinct  $\Delta$  operators is finite. In such a case, by identifying all distinct graphs which give the same operator, we can keep the first step of the procedure manageable. In fact, the number of distinct  $\Delta$  operators is finite in the present case. Therefore, the procedure discussed in Section 4.3 can apply.

#### 4.6. The Antiferromagnetic Ising-Like Anisotropy ( $\lambda \leq -1$ )

In this case, we start with the following identity:

$$\sigma_{a,\mu}^x \sigma_{b,\nu}^x + \sigma_{a,\mu}^y \sigma_{b,\nu}^y + \lambda \sigma_{a,\mu}^z \sigma_{b,\nu}^z = \lambda + 2\hat{B}(\mu, \nu) + 2(-\lambda - 1) \hat{D}(\mu, \nu) \tag{4.31}$$

where the new operator  $\hat{D}(\mu, \nu)$  is defined by

$$\begin{aligned} \langle \mathbf{n}(l_2) | \hat{D}(\mu, \nu) | \mathbf{n}(l_1) \rangle_2 &= \delta(n_{(2,a,\mu)}, n_{(1,a,\mu)}) \delta(n_{(2,b,\nu)}, n_{(1,b,\nu)}) \\ &\times \delta(n_{(1,a,\mu)}, \bar{n}_{(1,b,\nu)}) \delta(n_{(2,b,\nu)}, \bar{n}_{(2,a,\mu)}) \\ &\times \prod_{\alpha \neq \mu} \delta(n_{(2,a,\alpha)}, n_{(1,a,\alpha)}) \prod_{\beta \neq \nu} \delta(n_{(2,b,\beta)}, n_{(1,b,\beta)}) \end{aligned} \quad (4.32)$$

The graph which expresses this operator is shown in Fig. 7d.

In the last subsection, we defined  $\Gamma^F(p)$  by removing the restriction on the multiple occupation of vertices from  $\Gamma^{FH}(p)$ . Similarly, in the present case, we consider a graph which consists of an arbitrary number of antiferromagnetic edges. Every vertex must be an endpoint of one or more edges. We call this type of graph *antiferromagnetic* and represent the set of the antiferromagnetic graphs by  $\Gamma^{AF}(p)$ . For the antiferromagnetic operators, we have the following lemma:

**Lemma 5.**  $\Gamma^{AF}(p)$  is closed with respect to multiplication.

The proof can be done in the same way as the one in the last subsection. In other words, using Property 8, we can show that all alternating paths in the graph  $G_1 \cup G_2$  are antiferromagnetic. The contribution from each alternating path can be represented by an antiferromagnetic edge, unless the path is a loop. If the path is a loop which is connected to an open path, it contributes a factor 1. If the path is a loop which is not connected to any open path, it contributes a factor 2 together with all other loops connected to it. Therefore, contraction results in antiferromagnetic edges and numerical factors. Thus follows the lemma.

Combining Lemma 5 and the expression (4.31), we obtain the following theorem:

**Theorem 5.** For  $\lambda$  smaller than or equal to  $-1$ , we can expand the operator  $\tilde{\rho}$  in the form

$$\tilde{\rho} = \sum_{G(p) \in \Gamma^{AF}(p)} v(G(p)) \hat{D}(G(p)) \quad (4.33)$$

with nonnegative coefficients  $v(G(p))$ .

## 5. HEISENBERG MODELS

In the isotropic cases, i.e.,  $|\lambda_p| = 1$ , we can obtain compact formulas for the coefficients  $v(G(p))$ . To obtain this formula, we have to examine the nature of the coefficients in more detail. As we did in Section 2.4, we can

classify local configurations into several categories by using symmetry properties. The function  $w(\mathbf{n}(p))$  has various symmetries: invariance under (1) the vertical and horizontal mirror-image transformation, (2) the simultaneous flipping of all spins, and (3) the permutation of vertices within each site. We define classes of configurations  $\mathbf{n}(p)$  in such a way that two configurations  $\mathbf{n}(p)$  and  $\mathbf{n}'(p)$  belong to the same class if and only if they can be transformed into each other by these symmetry transformations. We use symbol  $\mathcal{S}(p)$  to specify such a class, which is a special set of states. For the same reason, we expect that the solution  $v(G(p))$  possesses the same symmetry, i.e.,  $v(G'(p)) = v(G(p))$  if a graph  $G'(p)$  can be obtained from  $G(p)$  through these symmetry transformations. Therefore, it is convenient to define classes of graphs. We say that  $G'(p)$  belongs to the same class as  $G(p)$  if and only if  $G(p)$  can be transformed into  $G'(p)$  through the symmetry transformations. A class is a subset of  $\Gamma^{XXZ}(p)$  and denoted by the symbol  $\mathcal{G}(p)$ . Note that

$$\bigcup_{\mathcal{G}(p) \in \Gamma^*(p)} \mathcal{G}(p) = \Gamma^*(p) \quad (5.1)$$

where the asterisk stands for XY, FH, AFH, F, or AF. Taking these definitions into account, we can reduce (3.30) to

$$\tilde{w}(\mathcal{S}(p)) = \sum_{\mathcal{G} \in \Gamma^*(p)} N(\mathcal{S}(p), \mathcal{G}(p)) \tilde{v}(\mathcal{G}(p)) \quad (5.2)$$

where

$$\tilde{w}(\mathcal{S}(p)) \equiv w(\mathbf{n}(p)), \quad \mathbf{n}(p) \in \mathcal{S}(p) \quad (5.3)$$

$$\tilde{v}(\mathcal{G}(p)) \equiv v(G(p)), \quad G(p) \in \mathcal{G}(p) \quad (5.4)$$

and

$$N(\mathcal{S}(p), \mathcal{G}(p)) \equiv \sum_{G(p) \in \mathcal{G}(p)} \Delta(\mathbf{n}(p), G(p)), \quad \mathbf{n}(p) \in \mathcal{S}(p) \quad (5.5)$$

We first note that the class of graphs  $\mathcal{G}(p)$  is characterized by 16 integers,  $m_{X_1-X_2}$  ( $X_1, X_2 = \text{bl}, \text{br}, \text{tl}, \text{tr}$ ). The integer  $m_{X_1-X_2}$  is the number of edges which connect vertices in the site  $s_{X_1}(p)$  to vertices in the site  $s_{X_2}(p)$ . For example,  $m_{\text{bl-tr}}$  is the number of vertical edges which connects a vertex in the bottom-left site and a vertex in the top-right site. In the case of ferromagnetic Heisenberg model, these integers vanish except for  $m_{\text{bl-tl}}$ ,  $m_{\text{bl-tr}}$ ,  $m_{\text{br-tl}}$ , and  $m_{\text{br-tr}}$ . Since there are constraints

$$\begin{aligned} m_{\text{bl-tl}} + m_{\text{bl-tr}} &= m_{\text{br-tl}} + m_{\text{br-tr}} = m_{\text{bl-tl}} + m_{\text{br-tl}} \\ &= m_{\text{bl-tr}} + m_{\text{br-tr}} = 2S \end{aligned} \quad (5.6)$$

only one of the above four numbers is independent. Therefore, we take  $m_{\text{bl-tl}}$  as the representative. Thus, we can simply express the coefficient  $\tilde{v}(\mathcal{G}(p))$  as

$$\tilde{v}(\mathcal{G}(p)) = \tilde{v}(m_{\text{bl-tl}}) \tag{5.7}$$

Next, we notice that a class of configurations is characterized by four integers  $m_{\text{bl}}$ ,  $m_{\text{br}}$ ,  $m_{\text{tl}}$ , and  $m_{\text{tr}}$ , where  $m_{\text{bl}}$  is the sum of vertex variables at the bottom-left site and the other three are similarly defined. This time, three of them are independent. We express the local weight function as in Section 2.4:

$$\tilde{w}(\mathcal{S}(p)) = \tilde{w} \begin{pmatrix} m_{\text{tl}} & m_{\text{tr}} \\ m_{\text{bl}} & m_{\text{br}} \end{pmatrix} \tag{5.8}$$

Using this notation, we can rewrite (4.25) as

$$\tilde{w} \begin{pmatrix} m_{\text{tl}} & m_{\text{tr}} \\ m_{\text{bl}} & m_{\text{br}} \end{pmatrix} = \sum_{m_{\text{bl-tl}}} N_{\text{FH}} \begin{pmatrix} m_{\text{tl}} & m_{\text{tr}} \\ m_{\text{bl}} & m_{\text{br}} \end{pmatrix} \Big|_{m_{\text{bl-tl}}} \tilde{v}(m_{\text{bl-tl}}) \tag{5.9}$$

We now consider a configuration which belongs to a class characterized by  $(m_{\text{tl}}, m_{\text{tr}}, m_{\text{bl}}, m_{\text{br}}) = (m, 2S - m, 2S, 0)$ . If a graph  $G$  is compatible with this configuration, all  $m$  occupied vertices in the top-left site must be connected to vertices in the bottom-right site because all edges in the graph are temporal and green. For a similar reason, all  $2S - m$  occupied vertices in the top-right site must be connected to those in the bottom-right site. Therefore, we can easily see that the function  $N$  defined above must have the following property:

$$N_{\text{FH}} \begin{pmatrix} m & 2S - m \\ 2S & 0 \end{pmatrix} \Big|_{m_{\text{bl-tl}}} = \begin{cases} [(2S)!]^2 & \text{if } m_{\text{bl-tl}} = m \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

Having this equation, we can derive from (5.9) the following equation, which determines  $v_{\text{FH}}$  completely:

$$\tilde{w} \begin{pmatrix} m & 2S - m \\ 2S & 0 \end{pmatrix} = [(2S)!]^2 \tilde{v}(m) \tag{5.11}$$

Using the identity (3.18), we finally get

$$\tilde{v}(m) = \frac{1}{[(2S)!]^2} \binom{2S}{m}^{-1} \langle \langle m, 2S - m | \exp(-\Delta\tau \mathcal{H}_{i,j}) | 2S, 0 \rangle \rangle \tag{5.12}$$

Therefore, the problem has been reduced to the diagonalization of  $\exp(-\Delta\tau \mathcal{H}_{i,j})$ , a matrix in  $(2S+1)$  dimensions, which is numerically simple and also can be done exactly by various symbolic computational languages.

Now, we will discuss the isotropic antiferromagnetic case. In this case, too, the class of graphs  $\mathcal{G}(p)$  is characterized by four integers,  $m_{\text{bl-tl}}$ ,  $m_{\text{br-tr}}$ ,  $m_{\text{bl-br}}$ , and  $m_{\text{tl-tr}}$ . As in the case of the ferromagnetic Heisenberg model, only one of the above four numbers is independent. We express the coefficient  $\tilde{v}(\mathcal{G}(p))$  as

$$\tilde{v}(\mathcal{G}(p)) = \tilde{v}(m_{\text{bl-tl}}) \quad (5.13)$$

Equation (4.26) is rewritten as

$$\tilde{w} \begin{pmatrix} m_{\text{tl}} & m_{\text{tr}} \\ m_{\text{bl}} & m_{\text{br}} \end{pmatrix} = \sum_{m_{\text{bl-tl}}} N_{\text{AFH}} \begin{pmatrix} m_{\text{tl}} & m_{\text{tr}} \\ m_{\text{bl}} & m_{\text{br}} \end{pmatrix} \Big|_{m_{\text{bl-tl}}} \tilde{v}(m_{\text{bl-tl}}) \quad (5.14)$$

We next consider a configuration which belong to a class characterized by  $(m_{\text{tl}}, m_{\text{tr}}, m_{\text{bl}}, m_{\text{br}}) = (0, m, m, 0)$ . In this case, in any graph compatible to this configuration, all the  $m$  occupied vertices in the bottom-left site must be connected to vertices in the bottom-right site because we cannot connect them by green edges to the top-left site, where there are no occupied vertices. Hence, we obtain the following property of  $N_{\text{AFH}}$ :

$$N_{\text{AFH}} \begin{pmatrix} 0 & m \\ m & 0 \end{pmatrix} \Big|_{m_{\text{bl-tl}}} = \begin{cases} [(2S)!]^2 & \text{if } m_{\text{bl-tl}} = 2S - m \\ 0 & \text{otherwise} \end{cases} \quad (5.15)$$

Accordingly, we obtain

$$\tilde{v}(2S - m) = \frac{1}{[(2S)!]^2} \binom{2S}{m}^{-1} \langle \langle 0, m | \exp(-\Delta\tau \mathcal{H}_{i,j}) | m, 0 \rangle \rangle \quad (5.16)$$

This is the generalization of (2.62).

## 6. ERGODICITY

Here we call an algorithm ergodic if an arbitrary state can be reached with a nonzero probability within a finite number of Monte Carlo steps regardless of the initial state. The ergodicity and the detailed balance condition are two important features that any Monte Carlo algorithm must possess. In the main text of this paper, we focused on the detailed balance condition and left out the discussion of ergodicity. It is sometimes non-trivial to show that an algorithm possesses this property. For example, the ergodicity of conventional algorithms has not yet been proven, as far as we know. However, for the new algorithm presented in this paper, it is almost straightforward to prove that ergodicity holds in the case of the

ferromagnetic and  $XY$ -like models, as we will see below. In the antiferromagnetic models, we can show that the new algorithm is ergodic if the conventional algorithm is.

The main part of conventional worldline algorithms is local loop flips where the size and shape of the loops are fixed. In order to achieve ergodicity, the shape and the location must be chosen carefully. Even if we choose them properly, however, using only local movements does not constitute an ergodic algorithm, because some quantities are conserved by these movements.<sup>(20)</sup> Global winding numbers are such conserved quantities. Therefore, we have to include several different kinds of global updates that can change the winding numbers. What local updates and global updates we should include depend on the model. We always have to face this annoying question as long as we use the conventional algorithms. In fact, this difficulty also makes the actual computer programs complicated in order to accommodate different kinds of procedures.

Now we prove the ergodicity of the new algorithm in the ferromagnetic and  $XY$ -like cases. To this end, we consider an arbitrary worldline configuration. If we note that any worldline consists of vertical line segments and diagonal ones, it is easy to realize that in a labeling process a loop configuration can be generated with a finite probability in such a way that any worldline in the initial state coincides with one of the loops in the loop configuration. (Of course this coincidence is possible only in the case of ferromagnetic or  $XY$ -like systems where graphs with diagonal edges can be chosen.) Once such loops are formed, in the subsequent flipping process, it can happen that all existing worldlines vanish (flip) and no new worldlines are created. The outcome is the vacuum state. Thus, within one Monte Carlo step, the vacuum state can be reached from an arbitrary state. The inverse process can also take place with a finite probability, i.e., the transition from the vacuum state to an arbitrary state. Therefore, we can conclude that every state can be reached from any state within two Monte Carlo steps via the vacuum state. Thus, ergodicity is proven.

This proof was possible because vertical and diagonal segments appear in the graphs corresponding to the operators  $\hat{f}$  and  $\hat{A}$  in the decomposition (4.5) and (4.27). Since the decomposition (4.31) does not contain the operator  $\hat{A}$ , the ergodicity of the antiferromagnetic model is not obvious. However, we can argue that if conventional algorithms are ergodic, the new algorithm is also ergodic. We can see this simply by noticing that most global flips introduced in the conventional algorithms can happen in the new algorithm with finite probability. To be more specific, the  $n$ -direction global flips in ref. 20 are simply flipping of loops with the temporal winding number 1, and the  $x$ -direction global flips are those of loops with spatial

winding number 1. Both types of loops can form in the new algorithm. The  $t$ -direction global flips are equivalent to flipping of loops whose temporal and spatial winding numbers are 1, as far as the effect on the global winding numbers is concerned. Again, this type of loop can form in the new algorithm.

Here, we emphasize that these global movements are intrinsically included in the new algorithm and they do not introduce any additional complication into the algorithm. In other words, we can achieve ergodicity in a simple and systematic way in the new algorithm, in contrast to the conventional algorithms.

## 7. SUMMARY

In this paper, we described how the Boltzmann factor of the lattice quantum spin model can be decomposed into a sum of terms each of which corresponds to a graph. Based on this decomposition, we have shown that a nontrivial cluster algorithm exists for any quantum spin model which can be described by this Hamiltonian, regardless of geometrical properties of the original lattice such as the number of dimensions, the boundary condition, and the range of the interactions. This decomposition also determines how to choose the proper set of local graphs and how to assign one of them to each plaquette. The new algorithm is advantageous for several reasons: (1) it may reduce the autocorrelation time drastically, (2) with the improved estimators,<sup>(19)</sup> it can reduce variances of distribution functions of important physical quantities and therefore reduce the statistical error, (3) it can achieve ergodicity without introducing any *ad hoc* global updates, and (4) the resulting computer programs can be simpler than those for the conventional algorithms. For some cases, such as the Ising model and the  $S=1$  antiferromagnetic, we explicitly gave the labeling probabilities that determine the Monte Carlo algorithm. For the general case, a method for computing the labeling probabilities numerically was presented.

Our representation can be viewed as the extension of the FK cluster representation of the Ising model. Therefore, the resulting Monte Carlo algorithms are generalizations of the SW algorithm to the quantum spin problems. As we showed in ref. 7, the present scheme reduces to the SW algorithm in the limit of strong Ising-like anisotropy. The present algorithm also includes the loop algorithm for the  $S=1/2$  model.<sup>(15)</sup>

It is remarkable that in contrast to the conventional algorithm, the cluster algorithm must be quite different, depending on the anisotropy of the model. For the  $XY$ -like anisotropy, we can have a loop algorithm. A loop algorithm is advantageous because we can identify loops in a computational time proportional to the number of vertices  $N_v = 2SM|L|$ . For



the Ising-like anisotropy, we inevitably create clusters with branching. In this case, the computational time is proportional to  $N_v \log(N_v)$ .

Although the efficiency of the algorithms has not yet been demonstrated systematically, some encouraging facts are already known. The efficiency of the SW algorithm near a critical point is well known. Other examples are the efficiency of the algorithms for the  $S = 1/2$  antiferromagnetic Heisenberg model<sup>(16)</sup> in two dimensions and  $S = 1$  antiferromagnetic Heisenberg model in one dimension.<sup>(7)</sup> In the latter, the autocorrelation time was at least three orders of magnitude smaller than that of the conventional method. The difference tends to be greater for large Trotter numbers. We also found that the present algorithm dramatically reduces the autocorrelation time of the  $S = 1/2$   $XY$  model in two dimensions, which is equivalent to the hard-core boson model. This result will be published elsewhere.<sup>(21)</sup> Based on these findings, we believe that for the homogeneous  $XXZ$  spin model without a symmetry-breaking field such as a magnetic field, the present algorithm provides a more efficient alternative to the standard local Metropolis algorithm. On the other hand, we know much less about the efficiency of the algorithm for disordered system and systems with symmetry-breaking fields. As for disorder, it is known<sup>(3)</sup> that the SW algorithm is not advantageous for the random-bond Ising model, i.e., the Edwards–Anderson spin-glass model. As for the effect of the symmetry-breaking field, we empirically found in the case of  $S = 1/2$  models in two dimensions that the algorithm works well for the Ising model regardless of the strength of a magnetic field, whereas it does not for the  $XY$  model when the field in the  $z$  direction is strong.<sup>(22)</sup> More work is needed for these cases.

## APPENDIX. THE COMPLETELY ANISOTROPIC CASE— THE $XYZ$ MODEL

Since the case where  $J_x = J_y$  is much more frequently studied than completely asymmetric case where  $J_x \neq J_y \neq J_z \neq J_x$ , we focused on the  $XXZ$  model in the main text to avoid too much complication. However, the  $XYZ$  model can be treated in a fashion similar to that presented for the  $XXZ$  model. In this appendix, we give a brief outline of the algorithm for the  $XYZ$  model. A more detailed discussion will be given elsewhere.<sup>(22)</sup>

We first note that

$$\begin{aligned}
 & K_0 + K_x \sigma_\mu^x \sigma_\nu^x + K_y \sigma_\mu^y \sigma_\nu^y + K_z \sigma_\mu^z \sigma_\nu^z \\
 &= \begin{pmatrix} K_1 & 0 & 0 & K_4 \\ 0 & K_2 & K_3 & 0 \\ 0 & K_3 & K_2 & 0 \\ K_4 & 0 & 0 & K_1 \end{pmatrix} \equiv \hat{\Lambda}_{K_1, K_2, K_3, K_4}(\mu, \nu) \quad (\text{A1})
 \end{aligned}$$

where

$$K_1 \equiv K_0 + K_z, \quad K_2 \equiv K_0 - K_z, \quad K_3 \equiv K_x + K_y, \quad K_4 \equiv K_x - K_y \quad (A2)$$

Here the constant  $K_0$  is irrelevant, i.e., its value does not affect the result. We introduce it for appearance. We also define  $w_{K_1, K_2, K_3, K_4}$  as the absolute value of the matrix element of  $\tilde{\rho}$  which is defined by (3.12) and (3.13) with  $\hat{A}(\mu, \nu)$  replaced by  $\hat{A}_{K_1, K_2, K_3, K_4}(\mu, \nu)$ . We then can show, by almost the same argument as we presented to show (3.16), that

$$w_{K_1, K_2, K_3, K_4} = w_{K_1, K_2, -K_3, K_4} = w_{K_1, K_2, K_3, -K_4} = w_{K_1, K_2, -K_3, -K_4} \quad (A3)$$

This generalizes (3.16). Because of (A3) we can assume nonnegative values for  $K_3$  and  $K_4$  without loss of generality. We can also assume nonnegative  $K_1$  and  $K_2$  because we can take as large a value as we want for the irrelevant constant  $K_0$  without changing final result. Therefore, we assume that all the constants are nonnegative.

For the XYZ model, we consider ten types of graphs (Fig. 10) instead of the four types in Fig. 7 for the XXZ model. We denote these graphs as  $G^X(\mu, \nu)$  where  $X = (1), (2), (3), (4), (12), (13), (14), (23), (24),$  and  $(34)$  as indicated in Fig. 10. Here, we omit the index  $p$  specifying a plaquette.

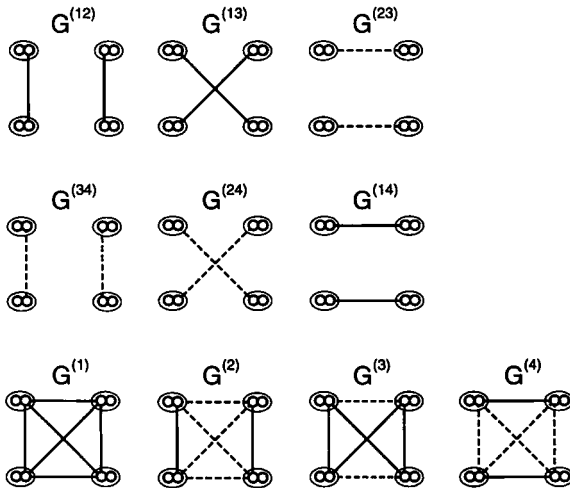


Fig. 10. Ten types of graphs for the decomposition of the local Boltzmann factor of the XYZ model. For clarity, two green vertical edges which connect leftmost vertices and rightmost vertices in each graph are not drawn.

We note that the graphs for the four operators discussed for the  $XXZ$  model can be rewritten in this notation as

$$A(\mu, \nu) = G^{(1,3)}(\mu, \nu)$$

$$B(\mu, \nu) = G^{(2,3)}(\mu, \nu)$$

$$C(\mu, \nu) = G^{(1)}(\mu, \nu)$$

$$D(\mu, \nu) = G^{(2)}(\mu, \nu)$$

The decomposition of the Hamiltonian which is analogous to (4.5), (4.27), and (4.31) is

$$\hat{A}_{K_1, K_2, K_3, K_4}(\mu, \nu) = \sum_X a_X \hat{A}(G^X(\mu, \nu)) \tag{A4}$$

where the undetermined variables  $a_X$  must be nonnegative. There is at least one nontrivial solution to this equation. In fact there are many in general. Which solution gives the most effective algorithm has not been studied extensively. A more detailed discussion about the solutions of (A4) will be presented elsewhere.

We should point out that the matrix element of (A1) can be viewed as the vertex weight of an eight-vertex model with symmetry with respect to simultaneous inversion of all arrows. Therefore, the solution of Eq. (A4) gives us a cluster algorithm of an eight-vertex model as well as the foundation for the cluster algorithm of the  $XYZ$  model. A similar remark applies to the  $XXZ$  model, where the  $S = 1/2$  problem was a special case of the six-vertex model.

Here we will only give a relatively simple but useful example of the solution of (A4) instead of listing all possible solutions. We consider the homogeneous system where the coupling constant does not depend on the location. We also assume, without loss of generality, that

$$|K_z| \geq |K_x| \geq |K_y| \geq 0 \tag{A5}$$

(If this is not the case, we can “rotate” the space of spins so that the above inequality holds.) We consider only the case where  $K_z > 0$ . We can do this when the original lattice is bipartite. Furthermore, because of (A3) and (A5), we can assume that  $K_x \geq 0$  without loss of generality. With these assumptions, we obtain a solution

$$a_1 = K_1 - K_2 - K_3 - K_4 = 2(K_z - K_x)$$

$$a_{(12)} = K_2 = 0$$

$$a_{(13)} = K_3 = K_x + K_y$$

$$a_{(14)} = K_4 = K_x - K_y$$

Correspondingly, (A4) becomes

$$A_{K_1, K_2, K_3, K_4}(\mu, \nu) = 2(K_z - K_x) \hat{A}(G^{(1)}(\mu, \nu)) \\ + (K_x + K_y) \hat{A}(G^{(13)}(\mu, \nu)) + (K_x - K_y) \hat{A}(G^{(14)}(\mu, \nu)) \quad (\text{A6})$$

Here we have chosen  $K_0 = K_z$ .

In Section 4 we gave an graph-theoretic argument to show the closure of various sets of graphs with respect to multiplication. By a similar argument, we can show that  $\Gamma_p^{\text{FXYZ}}$  is closed with respect to multiplication. Here  $\Gamma_p^{\text{FXYZ}}$  is defined by

$$\Gamma^{\text{FXYZ}}(p) \equiv \{G \mid V_G = p, \text{ and "All edges in } G \text{ are green"}\} \quad (\text{A7})$$

We note that

$$\Gamma^{\text{FXYZ}}(p) \cap \Gamma^{\text{XXZ}}(p) = \Gamma^{\text{F}}(p) \quad (\text{A8})$$

Following the same line of argument as the one in the main text, we can conclude that a set of nonnegative variables exists that satisfies

$$\tilde{\rho}_{K_1, K_2, K_3, K_4} = \sum_{G(p) \in \Gamma^{\text{FXYZ}}(p)} \nu(G(p)) \hat{A}(G(p)) \quad (\text{A9})$$

This, of course, leads to a cluster algorithm for the XYZ model with  $K_z \geq K_x \geq |K_y| \geq 0$ . It is straightforward to calculate  $\nu(G(p))$  analytically for small values of  $S$ . We note that one useful application of this type of algorithm is to the XY model with the  $x$ -representation basis, i.e., the representation basis in which the  $x$  components of spins are diagonalized. This is useful because by this representation we can calculate the two-point correlation between in-plane spin components. This correlation may be more interesting to study than the correlation between out-of-plane components which can be calculated with the  $z$ -representation basis.

## ACKNOWLEDGMENTS

The authors are grateful to G. A. Baker, Jr. for his critical reading of the manuscript and many suggestions. One of the authors (N.K.) thanks G. McNamara and B. Luce for useful comments on the proofs of the theorems. The work of J.E.G. was supported by the High Performance Computing and Communication program of the Department of Energy.

## REFERENCES

1. R. H. Swendsen and J.-S. Wang, *Phys. Rev. Lett.* **58**:86 (1987).
2. P. W. Kasteleyn and F. M. Fortuin, *J. Phys. Soc. Jpn.* **26**(Suppl.):11 (1969); C. M. Fortuin and P. W. Kasteleyn, *Physica* **57**:536 (1972).
3. R. H. Swendsen, J.-S. Wang, and A. M. Ferrenberg, in *The Monte Carlo Method in Condensed Matter Physics*, K. Binder, ed. (Springer, Berlin, 1991).
4. D. Kandel and E. Domany, *Phys. Rev. B* **43**:8539 (1991).
5. R. Brower and P. Tamayo, *Physica A* **193**:314 (1992).
6. P. D. Coddington and L. Han, unpublished.
7. N. Kawashima and J. E. Gubernatis, *Phys. Rev. E* **51**:1547 (1995).
8. M. H. Kalos, *Phys. Rev.* **128**:1791 (1962).
9. R. Blankenbecler and R. L. Sugar, *Phys. Rev. D* **27**:1304 (1983).
10. J. E. Hirsch, *Phys. Rev. Lett.* **51**:1900 (1983); *Phys. Rev. B* **31**:4403 (1985).
11. D. C. Handscomb, *Proc. Camb. Phil. Soc.* **58**:594 (1962); A. W. Sandvik and J. Kurkijärvi, *Phys. Rev. B* **43**:5950 (1991); A. W. Sandvik, *J. Phys. A* **25**:3667 (1992).
12. M. Suzuki, *Prog. Theor. Phys.* **56**:1454 (1976).
13. N. Kawashima, J. E. Gubernatis, and H. G. Evertz, *Phys. Rev. B* **50**:136 (1994).
14. H. G. Evertz, M. Marcu, and G. Lana, *Phys. Rev. Lett.* **70**:875 (1993).
15. H. G. Evertz and M. Marcu, in *Quantum Monte Carlo Method in Condensed Matter Physics*, M. Suzuki, ed. (World Scientific, Singapore, 1992), p. 65.
16. U.-J. Wiese and H.-P. Ying, *Phys. Lett. A* **168**:141 (1992); *Z. Phys. B* **93**:147 (1994).
17. N. Kawashima and J. E. Gubernatis, *Phys. Rev. Lett.* **73**:1295 (1994).
18. R. G. Edwards and A. D. Sokal, *Phys. Rev. D* **38**:2009 (1988).
19. U. Wolff, *Phys. Rev. Lett.* **60**:1461 (1988); *Nucl. Phys.* **300**[FS22]:501 (1988).
20. S. Miyashita, in *Quantum Simulations of Condensed Matter Phenomena*, J. D. Doll and J. E. Gubernatis, eds. (World Scientific, Singapore, 1989), p. 238, and references cited therein.
21. S.-W. Zhang, N. Kawashima, J. Carlson, and J. E. Gubernatis, *Phys. Rev. Lett.* **74**:1500 (1995).
22. N. Kawashima, unpublished.