TITLE: CLUSTER ALGORITHMS WITH EMPHASIS ON QUANTUM SPIN SYSTEMS

Authors: James E. Gubernatis, T-11
Naoki Kawashima, T-11

SUBMITTED TO: Proceedings of the Euroconference on Computer Simulation on Physics and Chemistry; Editors: J. Binder and G. Cicotti; Publisher: Italian Physical Society

By acceptance of this article, the publisher recognized that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution or to allow others to do so for U.S. Government National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.
Cluster Algorithms with Emphasis on Quantum Spin Systems

J.E. Gubernatis
Center for Nonlinear Studies and Theoretical Division
Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A.

N. Kawashima*
Department of Physics, University of Tokyo
Hongo, Bunkyo, Tokyo 113, Japan
(October 6, 1995)

Abstract

The purpose of this lecture is to discuss in detail the generalized approach of Kawashima and Gubernatis for the construction of cluster algorithms. We first present a brief refresher on the Monte Carlo method, describe the Swendsen-Wang algorithm, show how this algorithm follows from the Fortuin-Kasteleyn transformation, and re-interpret this transformation in a form which is the basis of the generalized approach. We then derive the essential equations of the generalized approach. This derivation is remarkably simple if done from the viewpoint of probability theory, and the essential assumptions will be clearly stated. These assumptions are implicit in all useful cluster algorithms of which we are aware. They lead to a quite different perspective on cluster algorithms than found in the seminal works and in Ising model applications. Next, we illustrate how the generalized approach leads to a cluster algorithm for world-line quantum Monte Carlo simulations of Heisenberg models with $S = 1/2$. More succinctly, we also discuss the generalization of the Fortuin-Kasteleyn transformation to higher spin models and illustrate the essential steps for a $S = 1$ Heisenberg model. Finally, we summarize how to go beyond $S = 1$ to a general spin, XYZ model.

Presented at the Euroconference on Computer Simulation in Condensed Matter Physics and Chemistry, July 18, 1995, Como, Italy
I. INTRODUCTION

In 1987, Swendsen and Wang [1] used the Fortuin and Kasteleyn (FK) representation [2] of the partition function of Ising models in a Monte Carlo simulation of these models. With the use of this representation, they produced 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 [3] to reduce autocorrelation times of various systems by various forms of cluster algorithms, and recently the construction of a cluster algorithm has been formulated on more general grounds [4–6,19]. Still, most applications of cluster algorithms have been restricted to classical models.

In general, it is non-trivial to find a cluster algorithm. The recently formulated generalized approaches [4–7], however, provide a starting point. They first require the specification of a proper set of local graphs by which a configuration for the whole system is decomposed into clusters and also require a non-negative solution to a system of linear equations (weight equations) that is often under-determined. Little a priori guidance is given on the construction of these graphs, and even the existence of a non-negative solution of these equations 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 [8,9] 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 world-line Monte Carlo simulation of the spin 1/2 anti-ferromagnetic Heisenberg model [10] because the $S = 1/2$ quantum spin systems can be mapped to the six-vertex model by using the Suzuki-Trotter approximation [11].

The purpose of this lecture is to discuss in detail the generalized approach of Kawashima and Gubernatis [7]. In Section II, we will present a brief refresher on the Monte Carlo method, describe the Swendsen-Wang algorithm, show how this algorithm follows form the Fortuin-Kasteleyn transformation, and re-interpret this transformation in a form which is basis of the generalized approach. In the Section III, we will derive the essential equations of the generalized approach. This derivation is remarkably simple if done from the viewpoint of probability theory, and the essential assumptions will be clearly stated. These assumptions are implicit in all useful cluster algorithms of which we are aware. They lead to a quite different perspective on cluster algorithms than found in the seminal works and in Ising model applications. In Section IV, we illustrate how the generalized approach leads to a cluster algorithm for world-line quantum Monte Carlo simulations of Heisenberg models with $S = 1/2$. Then, we briefly discuss the generalization of the Fortuin-Kastelyn transformation to higher spin models and illustrate the essential steps for a $S = 1$ Heisenberg model. Finally, we describe how to go beyond $S = 1$ to a general spin, XYZ model.

As one proceeds through the paper, he will find that the level of detail decreases while the level of difficulty increases. Our intent is to provide the basis for the reader to understand the essential features of a cluster algorithm. What we provide should enable the reader to understand more easily the published literature where more details are published [12–14]. At the same time, we will try to orient the reader to several remarkable results about the
existence of cluster algorithms for quantum spin systems. These results underscore the triviality of the cluster algorithms now routinely being applied to Ising and Potts models. Our hope is that the now informed reader will be equipped and motivated to develop and apply cluster algorithms to the rich number of physical system for which such algorithms presently do not exist. In the more general cases, detail is unnecessary unless an implementation is attempted.

II. BACKGROUND

A. Monte Carlo Refresher

In statistical mechanics, we want to calculate averages of the type

$$\langle f \rangle = \sum_A f(A)W(A)$$

where $W(A)$ is the Boltzmann weight for states $A$. The Monte Carlo method replaces this average by

$$\langle f \rangle = M^{-1} \sum_{i=1}^{M} f(A_i)$$

where $A_i$ is a member of the set $\{A\}$ of allowed states that has a weight $W(A_i)$. The $A_i$ are generated with weight $W(A_i)$ by a random walk

$$\ldots \rightarrow A \rightarrow A' \rightarrow A'' \rightarrow \ldots$$

in phase space along a Markov chain defined by a transition probability $T(A'|A)$ to go from $A$ to $A'$. At the $n$-th step,

$$T^{(n)}(A'|A) = \sum_{A''} T(A'|A'')T^{(n-1)}(A''|A)$$

$$T^{(1)}(A'|A) = T(A'|A)$$

For the Monte Carlo procedure to produce states with the proper weight $W(A)$, we need

$$\lim_{n \to \infty} T^{(n)}(A'|A) = W(A) > 0$$

The proper limit exits if [15]

1. the random walk is ergodic,

2. $\sum_A W(A) = 1$ and $W(A') = \sum_{A} T(A'|A)W(A)$

The last equation will be satisfied if the condition of detailed balance holds

$$T(A'|A)W(A) = T(A|A')W(A')$$

Metropolis et al. [16] showed that detailed balance condition will be satisfied if the transition from $A$ to $A'$ is accepted with probability

$$\min[1, W(A')/W(A)]$$
This simple prescription for specifying the transition probability is often called the *Metropolis algorithm*. The Metropolis algorithm, however, is a sufficient not a necessary to construct a Markov chain with the proper limiting distribution. Cluster algorithms need only be based on the more general condition. Satisfying this condition does not preclude detailed balancing from also being satisfied.

A key feature of any Monte Carlo simulation is not just the computation of averages but also estimates of the errors associated with these averages. The estimate of the error on Monte Carlo averages is usually taken to be the square root of the variance of the average. Because most measurements are not statistically independent, the measured variance

\[ \sigma_{\text{meas}}^2 = \frac{1}{M-1} \sum_{i=1}^{M} [f(A_i) - \langle f \rangle]^2 \]

and the true variance are not equal but are related by

\[ \sigma_{\text{true}}^2 = 2\tau \sigma_{\text{meas}}^2 \]

where \( \tau \) is the *auto-correlation time* between measurements. The correlation time represents a loss of useful measurements. What one wants is \( 2\tau \approx 1 \). Achieving \( 2\tau \approx 1 \) is attempted by improving the algorithm, by methods of data analysis, or by both. Cluster algorithms can tremendously reduce \( \tau \).

**B. Swendsen-Wang Algorithm**

In 1987, Swendsen and Wang [1] introduced an algorithm for Ising-like models that changed the orientation of clusters of identically aligned spins. Near the critical temperature \( T_c \), the computational auto-correlation time, i.e. \( \tau \), was reduced by several orders of magnitude. A practical description of the algorithm is now found in textbooks [17].

The essential features of the algorithm are illustrated in Fig. 1. These features corresponds to the following steps [17]:

1. Take a spin configuration and form a network of bonds joining together all aligned neighbors.
2. Keep these bonds with probability \( 1 - e^{2\beta J} \) or delete them with probability \( e^{2\beta J} \).
3. Treat the spin orientation of the new, smaller clusters as basic units and orient them randomly, i.e., up or down with equal probability.
4. Finally reconstruct the original lattice of spins from the re-oriented clusters.

From where do these steps come? Their analytic basis is given by the Fortuin-Kasteleyn transformation.
C. The Fortuin-Kasteleyn Transformation

In this subsection, we first describe the FK representation using the original notation adopted by Fortuin and Kasteleyn, then we re-describe it later using the notation which makes it easier to generalize the representation.

For a finite, spin-$\frac{1}{2}$ Ising system, Fortuin and Kasteleyn [2] define a “graph” $\Gamma$ to specify the location of the Ising spins and the connectivity of the interactions between the spins, i.e.,

1. at each vertex $v$ of the graph is an Ising spin $\sigma_v$,

2. along each edge $e$ of the graph is the interaction between a pair of spins at vertices $v$ and $v'$.

They expressed the ferromagnetic Ising model

$$H = - \sum_{v,v'} J_{vv'}(\sigma_v \sigma_{v'} - 1), \quad J_{vv'} > 0$$

as

$$H = - \sum_{e \in E(\Gamma)} J_e (\sigma_e - 1), \quad J_e > 0$$

where $E(\Gamma)$ is the set of the edges of $\Gamma$ and $\sigma_e = \sigma_v \sigma_{v'}$, and the partition function for the system is

$$Z(\Gamma) = \sum_\sigma \exp \left\{ \sum_{e' \in E(\Gamma)} \beta J_{e'} (\sigma_{e'} - 1) \right\}$$

Considering arbitrary edge $e$, they re-wrote the partition function as

$$Z(\Gamma) = \sum_{\sigma \sigma_e = +1} \exp \left\{ \sum_{e' \in E(\Gamma) - e} \beta J_{e'} (\sigma_{e'} - 1) \right\}$$

$$+ e^{-2\beta J_e} \sum_{\sigma \sigma_e = -1} \exp \left\{ \sum_{e' \in E(\Gamma) - e} \beta J_{e'} (\sigma_{e'} - 1) \right\}$$

Next they defined a contraction $C_e$ and deletion $D_e$ operation relative to edge $e$ by

$$Z(C_e \Gamma) = \lim_{\beta J_e \to \infty} Z(\Gamma) = \sum_{\sigma \sigma_e = +1} \exp \left\{ \sum_{e' \in E(\Gamma) - e} \beta J_{e'} (\sigma_{e'} - 1) \right\}$$

and

$$Z(D_e \Gamma) = \lim_{\beta J_e \to 0} Z(\Gamma) = Z(C_e \Gamma) + \sum_{\sigma \sigma_e = -1} \exp \left\{ \sum_{e' \in E(\Gamma) - e} \beta J_{e'} (\sigma_{e'} - 1) \right\}$$
Therefore,
\[ Z(\Gamma) = p_e Z(C_e \Gamma) + q_e Z(D_e \Gamma) \]
where
\[ q_e = e^{-2\beta J_e}, \quad p_e + q_e = 1 \]

For a finite system, a closed expression for \( Z(\Gamma) \) was found recursively. The result is
\[
Z(\Gamma) = \sum_{C \in \mathcal{E}(\Gamma)} p^C q^D Z(C^C D^D \Gamma) \\
= \sum_{C \in \mathcal{E}(\Gamma)} p^C q^D 2^{N_e}
\]
where \( N_e = N_e(C^C D^D \Gamma) \) is the number of clusters. \( C \) stands for both the set of “contracted” (frozen) edges and the number of its elements and \( D \) is both the set of “deleted” edges and the number of its elements. For example, \( C^C \) stands for \( \Pi_{c \in C} C_c \) and \( p^C \) stands for \( p \) raised to the \(|C|\)-th power.

Clusters are composed of identically aligned contracted spins with \( J_e \to \infty \) and associated with each cluster \( c \) is a cluster variable \( \Phi_e = \pm 1 \) designating the orientation of the spins in the cluster. The clusters do not interact. \( Z(C^C D^D \Gamma) \) is the partition function of a system of \( N_e \) two-state variables \( \{\Phi_e\} \) with no interaction between them. Thus, it equals \( 2^{N_e} \).

Now we restate the FK representation using different notations which turns out to yield much simpler description of the general algorithm. The partition function
\[
Z(\Gamma) = \sum_{\sigma} \exp \left\{ \sum_{e' \in E(\Gamma)} \beta J_{e'} (\sigma_{e'} - 1) \right\}
\]
can be rewritten as
\[
Z(\Gamma) = \sum_{\sigma} W(\sigma)
\]
where
\[
W(\sigma) = \prod_{e} w(\sigma_e)
\]
and \( w(\sigma_e) = \exp(\beta J_e (\sigma_e - 1)) \). The definition for \( w(\sigma_e) \) can be represented as linear system of equations
\[
w(\sigma_e) = \sum_{b_e = c, d} \delta(\sigma_e, b_e) v(b_e)
\]
with
\[
w(\sigma_e = \pm 1) \to \left( \begin{array}{c} 1 \\ e^{-2\beta J_e} \end{array} \right), \quad \delta(\sigma_e, b_e) \to \left( \begin{array}{cc} 1 & 1 \\ 0 & 1 \end{array} \right), \quad v(b_e = c, d) \to \left( \begin{array}{cc} 1 & 0 \\ 1 & e^{-2\beta J_e} \end{array} \right)
\]
The elements of the vector \( v(b_e) \) are just the contraction and deletion probabilities defined by the Fortuin-Kasteleyn transformation. Thus, we have
The last product implies that a state of spins contributes to the partition function only if the values of $\sigma_v$ and $\sigma_{v'}$ of any contracted edge are equal. This product defines the clusters. In Sec. III, we will illustrate that similar linear systems of equations are characteristic of all known cluster algorithms in which the cluster can be "re-oriented" freely.

D. Cluster Algorithm Characteristics

Before moving onto the more formal discussion, we will now state several general characteristics of all successful cluster algorithms to-date. Most of these characteristics are illustrated by the Swendsen-Wang algorithm.

Suppose for a given state $A$, e.g. $\{\sigma_e\}$, we can associate a graph $B$, e.g., $\{b_e\}$ and write

\[
W(A) = \sum_B V(B)\Delta(A, B)
\]

We will show that the above form leads to a cluster algorithm. Several features of the FK transformation also seem characteristic of a successful cluster algorithm:

1. a graph is assigned to each possible state,
2. the graph is specified by the stochastic assignment of local "labels," e.g., $b_e = \{c, d\}$, that depend on the local state of the system,
3. parts of the graph with the same label form a cluster,
4. to each cluster a variable specifying the state of the cluster can be assigned,
5. "flipping" the cluster involves changing the cluster variable to produce a new, allowed state.

Experience shows that successful cluster algorithms also produce large numbers of small clusters and produce non-interacting clusters. An important point here is that we do not need any information or prior-knowledge about critical phenomena in constructing cluster algorithms.

III. CLUSTER ALGORITHM CONSTRUCTION

A. Dual Monte Carlo

In a standard Monte Carlo algorithm, the states of the system progress through the sequence

$$\ldots \rightarrow A \rightarrow A' \rightarrow A'' \rightarrow \ldots$$
In a standard cluster algorithm, the states of the system A and the graphs B progress through the sequence

\[ \ldots \rightarrow A \rightarrow B \rightarrow A' \rightarrow B' \rightarrow A'' \rightarrow B'' \rightarrow \ldots \]

More generally, a cluster Monte Carlo algorithm can be viewed as progressing through the dual sequence

\[ \ldots \rightarrow (A, B) \rightarrow (A', B') \rightarrow (A'', B'') \rightarrow \ldots \]

From this point of view, we want to construct a Markov process that produces \( \mathcal{W}(A, B) \) as its limiting distribution.

Several ways exist to produce the proper limiting sequence. A simple way is to choose conditional probabilities that satisfy

\[
\begin{align*}
\mathcal{W}(A', B) &= \sum_A T_F(A'|A, B) \mathcal{W}(A, B) \\
\mathcal{W}(A, B') &= \sum_A T_L(B'|A, B) \mathcal{W}(A, B)
\end{align*}
\]

Most cluster algorithms however assume *heat-bath* type of conditions on the transition probabilities

\[
\begin{align*}
\mathcal{W}(A', B) &= \sum_A T_F(A'|B) \mathcal{W}(A, B) \\
\mathcal{W}(A, B') &= \sum_B T_L(B'|A) \mathcal{W}(A, B)
\end{align*}
\]

That is, a transition to a new state or label is independent of the current value of that state of label. We note that

\[
T_F(A|B) = \frac{\mathcal{W}(A, B)}{\sum_A \mathcal{W}(A, B)} = \frac{\mathcal{W}(A, B)}{\bar{W}(B)}
\]

\[
T_L(B|A) = \frac{\mathcal{W}(A, B)}{\sum_B \mathcal{W}(A, B)} = \frac{\mathcal{W}(A, B)}{\bar{W}(A)}
\]

where \( \bar{W}(A) = \sum_B \mathcal{W}(A, B) \) and \( \bar{W}(B) = \sum_A \mathcal{W}(A, B) \), standard results from probability theory. We will construct heat-bath type algorithms.

A defining property of a cluster algorithm seems to be the limited range of states to which the final state A is restricted by the label B. We can express this situation by writing

\[
T_F(A|B) = W'(A) \Delta(A, B) / N(B)
\]

where \( W'(A) \) is a weight function consistent with \( W(A) = W_0(A) W'(A) \), \( \Phi(B) \) is the set of all allowed states consistent with the graph B, \( N(B) \) is the number of members of \( \Phi(B) \), and \( \Delta(A, B) \) is defined by

\[
\Delta(A, B) = \begin{cases} 
1, & \text{if } A \in \Phi(B) \\
0, & \text{otherwise}
\end{cases}
\]
B. Free Cluster Algorithms

We can obtain an important subclass of cluster algorithms by setting $W'(A) = 1$. These algorithms are the analogs of non-interacting cluster algorithms. That is, from a given graph $B$, the new states are chosen with equal probability from any of the allowed states compatible with $B$. We will call such algorithms free cluster algorithms. Here, we will only consider the construction of free cluster algorithms.

C. The Linear System of Equations

To specify $W(A, B)$, we start by recalling a standard relation from probability theory

$$W(A) = \sum_B T_F(A|B)\tilde{W}(B)$$

For free cluster algorithms and with our choice for $T_F(A|B)$, this relation becomes

$$W(A) = \sum_B \Delta(A, B)\tilde{W}(B)/N(B)$$

With the definition

$$V(B) = \tilde{W}(B)/N(B)$$

we will rewrite the equation for $W(A)$ as the linear system of equations

$$\sum_B \Delta(A, B)V(B) = W(A)$$

If a solution to this equation can be found such that for all admissible $A$, $V(B) > 0$ then a (free) cluster algorithm exists.

D. Labeling and Joint Probabilities

Usually, we are given $W(A)$ and specify $\Delta(A, B)$. Specifying $\Delta(A, B)$ fixes $T_F(A|B)$ through the relation

$$T_F(A|B) = \Delta(A, B)/\sum_A \Delta(A, B)$$

If we obtain an admissible solution for $V(B)$, we can calculate $T_L(B|A)$ by starting with Bayes' theorem

$$T_L(B|A) = \frac{T_F(A|B)\tilde{W}(B)}{W(A)}$$

and substituting our various definitions to find
\[ T_L(B|A) = \frac{[\Delta(A, B)/N(B)][V(B)N(B)]}{W(A)} = \frac{\Delta(A, B)V(B)}{W(A)}. \]

Because \( T_L(A|B) = \mathcal{W}(A, B)/W(A) \), the last equation implies

\[ \mathcal{W}(A, B) = \Delta(A, B)V(B) \]

Important point: We emphasize that neither the uniqueness of the solution for \( V(B) \) or its existence is guaranteed.

E. Cluster Algorithms with Local Labeling Rules

To construct a local algorithm, we assume

\[ W(A) = \prod_i w(a_i), \]
\[ \Delta(A, B) = \prod_i \delta(a_i, b_i), \]
\[ V(B) = \prod_i v(b_i). \]

Now, we can arrive at a set of equations with the number of degrees of freedom of order \( \mathcal{O}(1) \)

\[ \sum_b \delta(a, b)v(b) = w(a) \]

where \( w(a) \equiv w(a)/w'(a) \). Once we get a solution \( v(b) > 0 \) of this equation, we can obtain the transition probability \( T_L(B|A) \) as follows

\[ T_L(B|A) = \prod_i t_L(b_i|a_i) \]

where

\[ t_L(b|a) = \delta(a, b)v(b)/w(a). \]

We also have

\[ T_F(A|B) = \prod_i t_F(a_i|b_i) \]

where

\[ t_F(a|b) = \delta(a, b)/\sum_a \delta(a, b) \]

F. Interim Summary

In summary, if one can re-express the partition function

\[ Z = \sum_A W(A) \]
in the form
\[ Z = \sum_{A,B} \mathcal{W}(A, B) \]
where
\[ \mathcal{W}(A) = \sum_{B} \mathcal{W}(A, B) \]
then we can simulate the system via a cluster algorithm by
1. generating a graph \( B \) from a given state \( A \) with a probability
\[ T_L(B|A) = \mathcal{W}(A, B) / \sum_{B} \mathcal{W}(A, B) = \mathcal{W}(A, B) / \mathcal{W}(A) \]
2. generating a state \( A \) from a given graph \( B \) with a probability
\[ T_R(A|B) = \mathcal{W}(A, B) / \sum_{A} \mathcal{W}(A, B) = \mathcal{W}(A, B) / \mathcal{W}(B) \]
The choice for \( \mathcal{W}(A, B) \) is
\[ \mathcal{W}(A, B) = V(B)\Delta(A, B) \]
provided a solution \( V(B) > 0 \) can be found from
\[ \mathcal{W}(A) = \sum_{B} \Delta(A, B)V(B) \]
We note that \( \mathcal{W}(B) = V(B)\sum_{A} \Delta(A, B) \).
The procedure is in principle quite simple. It only provides a way to think about constructing cluster algorithms. It does not provide the algorithm. Remarkably, in a number of complex cases, thinking about cluster algorithms in this way has provided such algorithms. For quantum Heisenberg-type models, it is theorem (due to Kawashima) that such algorithms exist [12–14].

IV. QUANTUM SPIN SYSTEMS

A. Cluster Algorithm: \( S = \frac{i}{2} \) Ferromagnetic Heisenberg Model

In one dimension, the Hamiltonian for the ferromagnetic, spin-\( \frac{1}{2} \) Heisenberg model is
\[ H = -J \sum_{i=1}^{N} \vec{S}_i \cdot \vec{S}_{i+1}, \quad J > 0 \]
where
\[ \vec{S}_i = \frac{1}{2}(\sigma_i^x, \sigma_i^y, \sigma_i^z) \]
and \( \vec{S}_i^2 = \frac{3}{2} \). The \( \sigma_i \) are Pauli spin operators and \( \vec{S}_N+1 = \vec{S}_1 \). The world-line method is developed by first expressing the partition function as a path-integral. Before doing this, we first choose a basis in which to represent the path. We take
\[ \sigma^z|n\rangle = (2n - 1)|n\rangle \]
where eigenvalue $n = 0$ or $1$. The path integral representation for the partition function becomes

$$Z = \sum_n \langle n | e^{-\beta H} | n \rangle$$

$$Z = \sum_n \langle n_1 | e^{-\Delta \tau H} | n_M \rangle \cdots \langle n_3 | e^{-\Delta \tau H} | n_2 \rangle \langle n_2 | e^{-\Delta \tau H} | n_1 \rangle$$

where $\beta = M \Delta \tau$, $n_k = \{n_{(k,1)}, n_{(k,2)}, \ldots, n_{(k,N)}\}$, and $| n \rangle = | n_1 \rangle | n_2 \rangle \cdots | n_M \rangle$

In order to simplify the evaluation of the matrix elements in the path-integral, we break the Hamiltonian into easily diagonalizable odd and even parts

$$H = H_O + H_E$$

$$H_O = -J \sum_{i \text{odd}} \vec{S}_i \cdot \vec{S}_{i+1}$$

$$H_E = -J \sum_{i \text{even}} \vec{S}_i \cdot \vec{S}_{i+1}$$

and then approximate

$$e^{-\Delta \tau H} \approx e^{-\Delta \tau H_E} e^{-\Delta \tau H_O}$$

Finally, we write

$$Z = \sum_n \langle n_1 | e^{-\Delta \tau H_E} | n_2 \rangle \cdots \langle n_4 | e^{-\Delta \tau H_O} | n_5 \rangle$$

$$\times \langle n_3 | e^{-\Delta \tau H_E} | n_2 \rangle \langle n_2 | e^{-\Delta \tau H_O} | n_1 \rangle$$

For a typical factor involving $H_O$, for example, we have

$$\langle n_{k+1} | e^{-\Delta \tau H_O} | n_k \rangle = \prod_{i \text{odd}} \langle n_{k+1,i} | n_{k+1,i+1} | e^{\Delta \tau J} \vec{S}_i \cdot \vec{S}_{i+1} | n_{k,i} \rangle$$

The time-space co-ordinates $(k, i)$, $(k, i + 1)$, $(k + 1, i)$, and $(k + 1, i + 1)$ define the corners of a plaquette and we can write

$$Z = \sum_n W(n) = \sum_n \prod_p w(n_p)$$

where the weight for a plaquette is given by

$$w(n_p) = w \left( \begin{array}{cc} n_{tl} & n_{tr} \\ n_{bl} & n_{br} \end{array} \right) = \langle n_{tl} n_{tr} | e^{\Delta \tau J} \vec{S}_i \cdot \vec{S}_{i+1} | n_{bl} n_{br} \rangle$$

The break-up of the Hamiltonian leads to a pictorial representation of a given configuration, i.e., a term in the partition function is represented as "world-lines" of the spin state. This is illustrated in Fig. 2.

To evaluate the path-integral, i.e. to move the world-lines, Monte Carlo importance sampling is used. Moves are accepted with probability

$$\min \left[ 1, \prod_p \frac{w(n'_p)}{w(n_p)} \right]$$
where the product is over the plaquettes which are affected by the local move.

Because of conservation of angular momentum

\[ n_{tl} + n_{tr} = n_{bl} + n_{br} \]

only a limited number of local moves is possible. (See Fig. 3.) Out of the 16 possible local plaquette configurations only 6 are allowed by the conservation rule. We designate these as

\[
1 \equiv \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \bar{1} \equiv \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \\
2 \equiv \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \quad 2 \equiv \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \\
3 \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad 3 \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.
\]

Besides the above, another restriction, namely, \( n_{(k,i)} = n_{(k+1,i)} \) occurs when the vertical edge \((k, i) - (k+1, i)\) does not belong to any shaded plaquette. We depict the allowed plaquettes, the local labels, and \( \delta(a, b) \) in Fig. 4. Then we have

\[
w(1) = w(\bar{1}) = \exp(\pm \Delta \tau J), \\
w(2) = w(2) = \exp(\mp \Delta \tau J) \cosh(2 \Delta \tau J), \\
w(3) = w(3) = \exp(\mp \Delta \tau J) \sinh(2 \Delta \tau J),
\]

The resulting weight equation is

\[
v(0) + v(2) + v(3) = \exp(\Delta \tau J), \\
v(1) + v(3) = \exp(-\Delta \tau J) \cosh(2 \Delta \tau J), \\
v(1) + v(2) = \exp(-\Delta \tau J) \sinh(2 \Delta \tau J).
\]

The solution is

\[
v(0) = e^{\Delta \tau J} - p, \\
v(1) = (e^{\Delta \tau J} - p)/2, \\
v(2) = (p - e^{-3\Delta \tau J})/2, \\
v(3) = (p + e^{-3\Delta \tau J})/2,
\]

where \( p \) is an adjustable parameter.

The guideline we use to determine \( p \) is to make the clusters as small as possible. This means making \( v(0) \) as small as possible. The range of \( p \) is given by

\[ 0 \leq p \leq \min(1, e^{\Delta \tau J}). \]

Therefore, \( p = e^{\Delta \tau J} \).

Thus, the local graphs \( b = 0 \) and \( b = 1 \) are not allowed. With the remaining two local graphs, \textit{loops} are formed. The flipping probability for the loops is \( \frac{1}{2} \). Along a loop, both \( \pm \frac{1}{2} \) spin states appear. The flipping probability is \( \frac{1}{2} \) because we have a two-state \textit{cluster variable}. The local labeling probabilities are shown in Table 1.
In Fig. 5, we show a sequence for the cluster algorithm of the $S = 1/2$ Heisenberg ferromagnetic chain that is the analog of the steps shown for the Swendsen-Wang algorithm in Fig. 1. We note the starting configuration had no world-lines while the resulting one had 3. In general, the algorithm does not conserve world-lines, which is to say the magnetization. We also note that loops, not clusters, are the long-range coherent structures formed. In the intermediate step, three loops are formed. (The system has periodic boundary conditions in both the spatial and temporal directions.)

B. Cluster Algorithm: $S = 1$ Ferromagnetic Heisenberg Model

In one-dimension the $S = 1$ ferromagnetic Heisenberg Hamiltonian is

$$H = -J \sum_{i=1}^{N} \vec{S}_i \cdot \vec{S}_{i+1}, \quad J > 0$$

where $\vec{S}_i^2 = 2$. As a basis, we could choose a set of $(2S + 1) \times (2S + 1) = 3 \times 3$ matrices in which $S^z$ is diagonal. Instead, we first decompose each spin operator into a sum of two Pauli operators.

$$S^\alpha_i = \frac{1}{2}[\sigma^\alpha_{i,1} + \sigma^\alpha_{i,2}], \quad \alpha = x, y, \text{or } z.$$ (Notice the extra label in the subscript.) We again take a basis in which the $z$-components of the Pauli operators are again diagonalized.

With this choice of a representation, we already have most of the necessary analytic results available from the $S = 1/2$ problem. Also as we will see, this choice will permit us to assign a two-state cluster variable to whatever clusters we form. This assignment means we “flip” clusters with probability 1/2.

In this over-complete basis, the partition function becomes

$$Z = \sum_n \langle n | e^{-\beta H} P | n \rangle.$$ 

The projection operator $P$ is the product of local projection operators $P_i$ which projects out states with $\vec{S}_i^2 \neq 2$

$$P = \prod_i P_i.$$ 

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

$$P_i = \frac{1}{2} (I + X_i)$$

where $I$ is the identity operator and $X_i$ is the operator that exchanges $\sigma^z_{i,1}$ and $\sigma^z_{i,2}$.

We now re-express the partition function as

$$Z = \sum_n \langle n_1 | e^{-\Delta \tau H_B} P | n_{2M} \rangle \cdots \langle n_4 | e^{-\Delta \tau H_O} P | n_3 \rangle \times \langle n_3 | e^{-\Delta \tau H_B} P | n_2 \rangle \langle n_2 | e^{-\Delta \tau H_O} P | n_1 \rangle$$

where $\beta = M \Delta \tau$ and $n_k = \{n_{(k,1,1)}, n_{(k,1,2)}, n_{(k,2,1)}, \ldots, n_{(k,N,2)}\}$. 
A typical factor in the expression for the partition function reduces to a product of "plaquettes"

\[ \langle n_{k+1} | P e^{-\Delta \tau H_0} P | n_k \rangle = \prod_{i: \text{even}} w(n_{p(k,i)}) \]

where \( p(k,i) \) is a plaquette consisting of four sites \((k,i), (k,i+1), (k+1,i)\) and \((k+1,i+1)\), and \( n_p \) is, as above, the set of one bit variables defined on the plaquette \( p \), i.e., \( n(k,i,1), n(k,i,2), n(k,i+1,1), \) etc., in this case. The local weight is defined as

\[ w(n_{p(k,i)}) \equiv \langle n(k+1,i,i), n(k+1,i,1), n(k+1,i+1,1), n(k+1,i+1,2) | P_{i} P_{i+1} e^{K_S}, S_i+1 P_{i} P_{i+1} | n(k,i,1), n(k,i,2), n(k,i+1,1), n(k,i+1,2) \rangle \]

with \( K \equiv \Delta \tau J \).

A similar expression is available for \( H_0 \). Therefore, the partition function can be rewritten as

\[ Z = \sum_n W(n), \]

where

\[ W(n) = \prod_p w(n_p). \]

The product is taken over \( p(k,i) \) with even \( k + i \). In Fig. 6, we illustrate the properties and definitions for a plaquette appropriate for the \( S = 1 \) case.

Here we note that \( w(n_p) \) depends on \( n_p \) 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 re-express \( w(n_p) \) in the form

\[ w(n_p) = w \left( \begin{array}{cc} m(k+1,i) & m(k+1,i+1) \\ m(k,i) & m(k,i+1) \end{array} \right). \]

We further note that this \( w \) has several symmetry properties that define classes of states which correspond to the same local weight regardless of the value of \( K \). In what follows, we will specify one of these classes by a symbol \( S \).

The "particle number conservation,"

\[ m_{bl} + m_{br} = m_{dl} + m_{tr}, \]

also imposes a restriction on \( w \), i.e., we can eliminate from consideration the classes that violate this condition. As a result, only 7 distinct classes of states among \( 2^8 = 256 \) values of \( w(n_p) \) exist.

The next thing we have to find is a set of local graphs. 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 diagonal edges. These 24 graphs fall into 3 classes \( G \). These classes of states and graphs satisfy

\[ w(S) = \sum_{G} v(G) \delta(S, G). \]

for which an analytic solution \( v(G) > 0 \) exists.

In Fig. 7, we present the local labels and \( \delta(S, G) \) for the \( S = 1 \) case. This figure is the analog of Fig. 4 which was for the \( S = 1/2 \) case.
C. Cluster Algorithm: General XYZ Model

The XYZ Hamiltonian we consider is

$$H = -\sum_{\langle ij \rangle} \left( J_{i,j}^x S_i^x S_j^x + J_{i,j}^y S_i^y S_j^y + J_{i,j}^z S_i^z S_j^z \right)$$

where $S_i^z = S(S + 1)$.

To develop a cluster algorithm we proceed in analogy with the $S = 1$ case: we first decompose each spin operator into a sum of Pauli operators.

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

take a basis in which the $z$-components of the Pauli operators are diagonalized, break the Hamiltonian into "odd" and "even" parts, make the Trotter approximation, and finally re-express the partition function as

$$Z = \sum_n \langle n_1 | P e^{-\Delta \tau H_E} P | n_{2M} \rangle \cdots \langle n_4 | P e^{-\Delta \tau H_O} P | n_1 \rangle$$

where $\beta = M \Delta \tau$ and $n_k = \{ n(k,1,1), n(k,1,2), \ldots, n(k,1,2S), n(k,2,1), \ldots, n(k,N,2S) \}$. A typical factor in the expression for the partition function again reduces to a product of "plaquettes." Symmetry operations can be used to reduce the states and local graphs to a small number of classes.

The remarkable thing is that Kawashima has proven [12-14] for any magnitude of spin $S$, dimension of space, arrangement of spins, and range of interactions, these classes of states and graphs will satisfy

$$w(S) = \sum_\mathcal{G} v(\mathcal{G}) \delta(S, \mathcal{G}).$$

and have a positive solution for $v(\mathcal{G})$. For the XYZ model [14], in contrast to the $S = \frac{1}{2}$ or $S = 1$ Heisenberg models, a larger variety of graphs can exist, depending on the sign and anisotropy of the interaction. The graphs for the $S = 1/2$ are shown in Fig. 8.

V. SUMMARY

A general procedure to construct cluster algorithms exists. The procedure requires a set of local graphs that permit a positive solution of a specific linear system of equations. This system is generally under-determined; thus, a cluster algorithms need not exist or be unique. Effective clusters algorithms exist for quantum spin models [10,12-14], fermion models [19], hard-core boson models [18], 6 and 8 vertex models [8,9,14], Ising glass models [7], and classical XY [20] and Potts [1] models. It is amazing that for a very general class of quantum spin models, a cluster algorithm is guaranteed to exist. Effective cluster algorithms have yet been developed for soft-core boson models and models with continuous degrees of freedom. An important point: current cluster algorithms do not specifically address disorder,
frustration, and fermion sign problems. In contrast to cluster algorithms for Potts models, which are inefficient away from a critical point because of the computational overhead of the algorithms, for quantum spin models, these algorithms are orders of magnitude more efficient even far away from a phase transition.

The efficiency of the cluster algorithms for quantum spin systems is striking. The standard world-line algorithm is often plagued with exceptionally long auto-correlation times. In Fig. 9, we compare such times for the standard algorithm for a Heisenberg anti-ferromagnetic chain and a loop algorithm for the same problem. Plotted are the auto-correlation time as a function of the square of the inverse of the Trotter step $\Delta \tau$. As $\Delta \tau$ is decreased, the accuracy of the Suzuki-Trotter approximation increases. Hence, for a fixed value of $\beta$, decreasing $\Delta \tau$ to reduce the size of the systematic error caused by the approximation seems reasonable. Fig. 9 indicates, however, a huge increase in variance can occur which necessitates a huge increase in computer time to recoup accuracy. In contrast, the variance of the cluster algorithm is insensitive to the change in Trotter step size. Our experience also indicates that the variance is also insensitive to changing the size of the system, the model parameters, etc.

In many instances, ergodicity is present in the formulation of the algorithm even if it might not exist in the standard algorithm. How ergodic is the algorithm when implemented, at this time seems to be a matter than can only be settled by testing. An added bonus with a cluster algorithm is often improved estimators based on the clusters can give an additional performance gain. The existence of an algorithm does not guarantee the efficiency of the algorithm. The efficiency too must be determined by implementation and testing. In general, clusters algorithms, despite their overhead, are orders of magnitude more efficient than the standard algorithms.

ACKNOWLEDGMENTS

This work was supported by the High Performance Computing and Communication program of the U.S. Department of Energy.
REFERENCES

* Present Address: Department of Physics, Toho University, Miyama 2-2-1, Funabashi, Chiba 274, Japan.

FIGURES

FIG. 1. The sequence of steps involved in the Swendsen-Wang algorithm for the Ising model: (a) formation of the bonds between neighboring sites, (b) random deletion of bonds, and (c) random reorientation of resulting clusters of spins.

FIG. 2. The checkerboard and a world-line configuration for a $S = 1/2$ Heisenberg ferromagnet for a 8 site chain. The world-lines connect states with $S_z = 1/2$.

FIG. 3. The allowed local moves. The movement can go from the solid to dashed line and vice versa.

FIG. 4. The allowed plaquettes, local labels, and the matrix $\delta(a,b)$ for a $S = 1/2$ Heisenberg ferromagnet.

FIG. 5. (a) The initial world-line configuration. (b) A possible loop structure compatible with the initial configuration. Because the labels are assigned stochastically, other configurations are possible. (c) A possible spin configuration compatible with the loop configuration in (b). Because the loops are flipped with equal probability to one of two possible state, other configurations are possible.

FIG. 6. A schematic of the hyper-lattice for the $S = 1$ Heisenberg ferromagnetic chain.

FIG. 7. The local labels and and $\delta(S, \mathcal{G})$ for the $S = 1$ case.

FIG. 8. Ten possible graphs of a plaquette for $S = 1/2$. Note the vertices are connect by dotted and solid lines, corresponding to red and green edges.

FIG. 9. A comparison of the auto-correlation times of the standard algorithm and a loop algorithm for a 32 site $S = 1$ Heisenberg anti-ferromagnetic chain as a function of $\Delta \tau$. Shown are the magnetization and static susceptibility.
### TABLE 1. The probabilities $t_L(a|b)$.

<table>
<thead>
<tr>
<th></th>
<th>b=0</th>
<th>b=1</th>
<th>b=2</th>
<th>b=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>a=1,1</td>
<td>0</td>
<td>0</td>
<td>$\frac{1 - e^{-\Delta r_J}}{2}$</td>
<td>$\frac{1 + e^{-\Delta r_J}}{2}$</td>
</tr>
<tr>
<td>a=2,2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>a=3,3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
\( z = 3A \)
\( z = A T \)
\( z = 0 \)

\[
\begin{array}{ccccccc}
|n_0\rangle & |n_1\rangle & |n_2\rangle & |n_3\rangle & |n_4\rangle & |n_5\rangle & |n_6\rangle \\
\hline
\tau = 0 & & & & & & \\
\tau = 1\Delta\tau & & & & & & \\
\tau = 2\Delta\tau & & & & & & \\
\tau = 3\Delta\tau & & & & & & \\
\tau = \beta = 0 & & & & & & \\
\end{array}
\]
\[ \tau = \beta = 0 \]
\[ \tau = 3\Delta \tau \]
\[ \tau = 2\Delta \tau \]
\[ \tau = 1\Delta \tau \]
\[ \tau = 0 \]
Hyper-lattice

Site (4,6)

Plaquette $P(3,5)$

Vertex $(4,6,1)$

$(4,5)$

$(3,5)$

$(3,6)$

$(4,6)$

$(4,6,2)$