

## Order- $N$ cluster Monte Carlo method for spin systems with long-range interactions

Kouki Fukui<sup>a,1</sup>, Synge Todo<sup>a,b,\*</sup>

<sup>a</sup> Department of Applied Physics, University of Tokyo, 7-3-1 Hongo, Tokyo 113-8656, Japan

<sup>b</sup> CREST, Japan Science and Technology Agency, Kawaguchi 332-0012, Japan

### ARTICLE INFO

#### Article history:

Received 2 February 2008

Received in revised form 11 November 2008

Accepted 18 December 2008

Available online 27 December 2008

#### PACS:

02.50.Ga

02.70.Ss

05.10.Ln

64.60.De

#### Keywords:

Long-range interaction

Cluster algorithm

$O(N)$  method

Ising model

Quantum Monte Carlo

Kosterlitz–Thouless transition

### ABSTRACT

An efficient  $O(N)$  cluster Monte Carlo method for Ising models with long-range interactions is presented. Our novel algorithm does not introduce any cutoff for interaction range and thus it strictly fulfills the detailed balance. The realized stochastic dynamics is equivalent to that of the conventional Swendsen–Wang algorithm, which requires  $O(N^2)$  operations per Monte Carlo sweep if applied to long-range interacting models. In addition, it is shown that the total energy and the specific heat can also be measured in  $O(N)$  time. We demonstrate the efficiency of our algorithm over the conventional method and the  $O(N \log N)$  algorithm by Luijten and Blöte. We also apply our algorithm to the classical and quantum Ising chains with inverse-square ferromagnetic interactions, and confirm in a high accuracy that a Kosterlitz–Thouless phase transition, associated with a universal jump in the magnetization, occurs in both cases.

© 2008 Elsevier Inc. All rights reserved.

## 1. Introduction

Systems with long-range interactions exhibit more involved phase diagrams and richer critical phenomena than those with only nearest-neighbor interactions. One of the most prominent examples is the Ising model with long-range interactions, whose Hamiltonian is defined as

$$-\beta\mathcal{H} = \sum_{i<j} \beta J_{ij} \sigma_i^z \sigma_j^z, \quad (1)$$

where  $\sigma_i^z = \pm 1$ ,  $J_{ij}$  is the coupling constant between the  $i$ th and  $j$ th sites ( $i, j = 1, 2, \dots, N$ ), and  $N$  is the total number of spins. Among the models described by Eq. (1), the one-dimensional chain with algebraically decaying interactions has been studied most intensely so far. The interaction  $J_{ij}$  for the model is written as

\* Corresponding author. Address: Department of Applied Physics, University of Tokyo, 7-3-1 Hongo, Tokyo 113-8656, Japan.

E-mail address: [wistaria@ap.t.u-tokyo.ac.jp](mailto:wistaria@ap.t.u-tokyo.ac.jp) (S. Todo).

<sup>1</sup> Present address: Morgan Stanley, 4-20-3 Ebisu, Tokyo 150-6008, Japan.

$$J_{ij} = \frac{1}{r_{ij}^{1+\alpha}}, \quad (2)$$

where  $\alpha$  is the parameter characterizing the range of interaction. (Note that  $\alpha$  should be positive to assure the energy convergence.) In spite of the extremely simple form of its Hamiltonian, the model is known to exhibit various critical behavior with respect to the parameter  $\alpha$ : When  $\alpha$  is sufficiently large, the system belongs to the same universality class as the nearest-neighbor model, i.e., no finite-temperature phase transitions [1,2]. At  $\alpha = 1$ , however, the system exhibits a Kosterlitz–Thouless phase transition at a finite temperature [3–5]. In the regime  $1/2 < \alpha < 1$ , the critical exponents of the system change continuously as  $\alpha$  is decreased [6]. Finally, when  $\alpha$  is equal to or smaller than  $1/2$ , the system shows the critical exponents of the mean-field universality [6]. Such rich and nontrivial phenomena associated with the long-range interactions have attracted much interest and many researches have been done both theoretically and numerically.

In general, Monte Carlo simulations of long-range interacting spin models are quite difficult, since the CPU time per Monte Carlo sweep grows quite rapidly as the system size increases. This is simply because there are  $N(N-1)/2$  different pairs of spins to be considered in an  $N$ -spin system. For unfrustrated spin models, the cluster methods, such as the Swendsen–Wang [7] or Wolff [8] algorithms, are the methods of choice, since they almost completely eliminate correlations between successive spin configurations on the Markov chain. Unfortunately, the cluster algorithms share the same difficulty with the single spin flip update. In 1995, however, Luijten and Blöte introduced a very efficient cluster algorithm [9]. What they noticed was that, on average, only  $O(N)$  among  $O(N^2)$  bonds contribute to cluster construction. By employing a rejection-free method based on binary search on a cumulative probabilities, they succeeded in reducing the number of operations per Monte Carlo sweep drastically to  $O(N \log N)$ . Recently, the same strategy has been applied to the quantum Monte Carlo method for long-range ferromagnetic Ising models in a transverse external field [10].

In the present paper, we propose a still faster cluster Monte Carlo algorithm for long-range interacting ferromagnets. Our method is based on the extended Fortuin–Kasteleyn representation of partition function [11,12] and an extremely effective technique for integral random number generation, so-called *Walker's method of alias* [13,14]. As the method of Luijten and Blöte, the proposed algorithm does not introduce any cutoff for interaction range and realizes the identical stochastic dynamics with the original  $O(N^2)$  Swendsen–Wang method. The CPU time per Monte Carlo sweep is, on the other hand, merely proportional to  $N$  instead of  $N \log N$  or  $N^2$ , and is an order of magnitude shorter than that of Luijten and Blöte for sufficiently large systems. In addition to its speed, our algorithm has several advantages: First, it is quite robust, that is, it works efficiently both for short-range and long-range interacting models as it stands. Second, the calculation of the total energy and the specific heat are also possible in  $O(N)$  time without any extra cost. Third, our Monte Carlo algorithm is straightforwardly extended for quantum models, such as the transverse-field Ising model, the Heisenberg model, etc.

The organization of the present paper is as follows: In Section 2, we briefly review the Swendsen–Wang cluster algorithm and its  $O(N \log N)$  variant by Luijten and Blöte. In Section 3, we present our new algorithm in detail. We also show how the total energy and the specific heat are calculated in  $O(N)$  time, and the extension of the  $O(N)$  algorithm to the transverse-field Ising model. In Section 4, a benchmark test of our new algorithm is presented. As an application of the  $O(N)$  algorithm, the Kosterlitz–Thouless transition of the Ising chain with inverse-square interaction is investigated in Section 5. Especially, we confirm the universality between the classical and quantum Ising models in a high accuracy. Section 6 includes a summary and discussion, followed by appendices on some technical details about Walker's method of alias.

## 2. Conventional cluster algorithms for Ising model with long-range interactions

### 2.1. Swendsen–Wang method

In this section, first we briefly review the cluster Monte Carlo method by Swendsen and Wang [7]. Each Monte Carlo sweep of the Swendsen–Wang algorithm consists of two procedures, graph assignment and cluster flip. In the former procedure, one inspects all the bonds sequentially, and each bond is activated or deactivated with probability

$$P_{ij} = \delta_{\sigma_i^z, \sigma_j^z} [1 - \exp(-2\beta J_{ij})] \quad (3)$$

and  $(1 - P_{ij})$ , respectively. Then, after the trials for all the bonds, each cluster of spins connected by active bonds is flipped at once with probability  $1/2$ , and a terminate configuration is generated.

The stochastic process achieved by the Swendsen–Wang algorithm is ergodic. It is also proved, with the help of the Fortuin–Kasteleyn representation of the partition function [11,12], that the algorithm satisfies the detailed balance. The partition function of the Hamiltonian (1) is written as

$$Z = \sum_{\mathbf{c}} \prod_{i < j} e^{\beta J_{ij} \sigma_i^z \sigma_j^z} = \sum_{\mathbf{c}} \prod_{\ell=1}^{N_b} e^{\beta J_{\ell} \sigma_{\ell}^z}, \quad (4)$$

where  $N_b (= N(N-1)/2)$  denotes the total number of bonds,  $\ell$  is the bond index, and  $J_{\ell} = J_{ij}$  and  $\sigma_{\ell}^z = \sigma_i^z \sigma_j^z$  are the coupling constant and the product of the spin states of the both ends of the bond  $\ell$ , respectively. We first extend the original phase space of Ising spins  $\{\mathbf{c}\} (= \{(\sigma_1^z, \sigma_2^z, \dots, \sigma_N^z)\})$  to the direct product of phase spaces of spins  $\{\mathbf{c}\}$  and *graphs*  $\{\mathbf{g}\}$ . A graph  $\mathbf{g}$  is defined by a set of variables  $\mathbf{g}_{\ell}$  ( $\ell = 1, 2, \dots, N_b$ ), each of which is defined on each bond (or link). The graph variable  $\mathbf{g}_{\ell}$

describes whether the  $\ell$ th bond is activated ( $g_\ell = 1$ ) or not ( $g_\ell = 0$ ). By using the extended phase space, the partition function (4) is expressed as

$$Z = C \sum_c \sum_g \omega(c, g) \tag{5}$$

with

$$\omega(c, g) = \prod_{\ell=1}^{N_b} \Delta(\sigma_\ell, g_\ell) V_\ell(g_\ell), \tag{6}$$

where  $C$  is a constant and the summation  $\sum_g$  runs over  $2^{N_b}$  possible graph configurations. The weight functions  $\Delta$  and  $V_\ell$  are defined as

$$\Delta(\sigma_\ell, g_\ell) = \begin{cases} 0 & \text{if } g_\ell = 1 \text{ and } \sigma_\ell = -1, \\ 1 & \text{otherwise,} \end{cases} \tag{7}$$

$$V_\ell(g_\ell) = (e^{2\beta J_\ell} - 1)^{g_\ell}, \tag{8}$$

respectively. The equality between Eqs. (4) and (5) is verified by figuring out the summation with respect to  $g$  in the latter

$$\begin{aligned} \sum_g \omega(c, g) &= \sum_g \prod_{\ell=1}^{N_b} \Delta(\sigma_\ell, g_\ell) V_\ell(g_\ell) = \prod_{\ell=1}^{N_b} (\Delta(\sigma_\ell, 0) V_\ell(0) + \Delta(\sigma_\ell, 1) V_\ell(1)) = \prod_{\ell=1}^{N_b} (1 + \delta_{\sigma_\ell, 1} (e^{2\beta J_\ell} - 1)) \\ &= e^{\beta \sum_{\ell=1}^{N_b} J_\ell} \prod_{\ell=1}^{N_b} (e^{-\beta J_\ell} + \delta_{\sigma_\ell, 1} (e^{\beta J_\ell} - e^{-\beta J_\ell})) = e^{\beta \sum_{\ell=1}^{N_b} J_\ell} \prod_{\ell=1}^{N_b} e^{\beta J_\ell \sigma_\ell} \end{aligned} \tag{9}$$

and thus  $C = \exp(-\beta \sum J_\ell)$ . From Eq. (5), we can consider  $\omega(c, g)$  as a weight of the configuration  $(c, g)$  in the extended phase space.

The Swendsen–Wang method is a procedure to update the spin configuration dynamically by going through an intermediate graph configuration. Consider that the initial spin configuration is  $c$ . We assign a graph  $g$  for the spin configuration  $c$  with the following probability:

$$P(g|c) = \frac{\omega(c, g)}{\sum_{g'} \omega(c, g')} = \prod_{\ell=1}^{N_b} \frac{V_\ell(g_\ell) \Delta(\sigma_\ell, g_\ell)}{\sum_{g'_\ell} V_\ell(g'_\ell) \Delta(\sigma_\ell, g'_\ell)}. \tag{10}$$

That is, for each bond we assign  $g_\ell = 1$  with probability

$$P(g_\ell = 1 | \sigma_\ell) = \frac{V_\ell(1) \Delta(\sigma_\ell, 1)}{V_\ell(1) \Delta(\sigma_\ell, 1) + V_\ell(0) \Delta(\sigma_\ell, 0)} = \delta_{\sigma_\ell, 1} (1 - e^{-2\beta J_\ell}). \tag{11}$$

This probability turns out to be the same as the one in Eq. (3). The cluster construction in the Swendsen–Wang algorithm is thus equivalent to assigning graph variables in the Fortuin–Kasteleyn language. The second procedure (cluster flip) in the Swendsen–Wang algorithm is also represented clearly in the Fortuin–Kasteleyn representation: Under a given graph configuration  $g$ , a new spin configuration  $c'$  is selected according to probability

$$P(c'|g) = \frac{\omega(c', g)}{\sum_c \omega(c, g)} = \frac{\prod_{\ell=1}^{N_b} \Delta(\sigma'_\ell, g_\ell)}{\sum_c \prod_{\ell=1}^{N_b} \Delta(\sigma_\ell, g_\ell)}. \tag{12}$$

Note that  $\prod_{\ell=1}^{N_b} \Delta(\sigma_\ell, g_\ell)$  takes either 1 or 0, depending on whether all the active bonds have  $\sigma_\ell = 1$ , or not. The last equation means that among all the allowed spin configurations, which have nonzero  $\omega(c, g)$ , for a given  $g$ , a configuration is chosen with equal probability. This is equivalent to flipping each cluster of spins connected by active bonds independently with probability 1/2.

The detailed balance condition of the Swendsen–Wang method is thus represented in a concrete form:

$$P(c'|c) \omega(c) = \sum_g P(c'|g) P(g|c) \omega(c) = \sum_g \frac{\omega(c, g)}{\omega(g)} \omega(c', g). \tag{13}$$

Since the most right-hand side of Eq. (13) is symmetric under the exchange of initial and terminal spin states  $c$  and  $c'$ , the detailed balance is satisfied automatically.

### 2.2. $O(N \log N)$ Method by Luijten and Blöte

It has turned out that the Swendsen–Wang cluster algorithm works quite well for wide variety of systems without frustration. Especially, it removes almost completely the so-called critical slowing down near the continuous phase transition point. Since there is no constraint about the range of interactions in its construction, the Swendsen–Wang algorithm is also applicable to long-range interacting systems without any modification. However, since the number of bonds  $N_b$  is  $O(N^2)$  in

such systems, the number of operations required for one Monte Carlo sweep is proportional to  $N^2$ , which is significantly more expensive than those for the nearest-neighbor models.

A nifty solution for reducing drastically the number of operations from  $O(N^2)$  to  $O(N \log N)$  was devised by Luijten and Blöte [9]. What they noticed is separating the activation probability  $P_\ell$  into the two parts:

$$P_\ell = p_\ell \delta_{\sigma_\ell, 1}, \quad (14)$$

$$p_\ell = 1 - \exp(-2\beta J_\ell). \quad (15)$$

If one chooses candidate bonds with probability  $p_\ell$  and then activate them with probability  $\delta_{\sigma_\ell, 1}$  afterward, the probability  $P_\ell$  is realized eventually. For choosing the candidates bonds, one could use a more efficient method than the exhaustive search, since  $p_\ell$  is independent of the spin state  $\sigma_\ell$  and predetermined statically at the beginning of the Monte Carlo simulation. Indeed, it is seen that the number of candidate bonds are typically much smaller than  $N_b$ . The average number of candidate bonds is estimated as

$$\sum_{\ell=1}^{N_b} p_\ell = \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} (1 - e^{-2\beta J_{ij}}) \sim \frac{1}{2} \sum_{i=1}^N \int_1^L dr r^{d-1} (1 - e^{-2\beta J(r)}) \sim \beta N \int_1^L dr r^{d-1} J(r). \quad (16)$$

Here,  $L = N^{1/d}$  is the linear extent of the system, and we assume the translational invariance and that  $J_{ij}$  depends only on the distance, i.e.,  $J_{ij} = J(r_{ij})$ . If  $J(r)$  decays faster than  $r^{-d}$ , which is equivalent to the condition of energy convergence for the ferromagnetic models, the integral in the last expression converges to a finite value in the thermodynamic limit. Thus, at a fixed temperature, the number of candidate bonds increases as  $N$  instead of  $N^2$ .

For choosing candidate bonds, Luijten and Blöte adopted a kind of rejection-free method, which is based on the binary search of cumulative probability tables. Let us define

$$q_m^{(0)} = p_m \prod_{\ell=1}^{m-1} (1 - p_\ell) \quad (q_1^{(0)} = p_1), \quad (17)$$

$$C_m^{(0)} = \sum_{\ell=1}^m q_\ell^{(0)} \quad (C_0^{(0)} = 0 \quad \text{and} \quad C_{N_b+1}^{(0)} = 1), \quad (18)$$

where  $q_m^{(0)}$  is the probability that the  $m$ th bond is eventually chosen as a candidate after the failure for the first, second, ..., and  $(m-1)$ th bonds, and  $C_m^{(0)}$  is the cumulative probability of  $q_m^{(0)}$ . When an uniform real random variable  $U$  ( $\in [0, 1)$ ) is generated,  $U$  satisfies  $C_{m-1}^{(0)} \leq U < C_m^{(0)}$  with probability  $q_m^{(0)}$ . The first candidate bond  $m$  can then be chosen by searching the first element larger than  $U$ . After the  $m$ th bond is activated or deactivated depending on its spin state  $\sigma_m$ , one can continue the same procedure using the tables

$$q_n^{(m)} = p_n \prod_{\ell=m+1}^{n-1} (1 - p_\ell) \quad (q_{m+1}^{(m)} = p_{m+1}), \quad (19)$$

$$C_n^{(m)} = \sum_{\ell=m+1}^n q_\ell^{(m)} \quad (C_m^{(m)} = 0 \quad \text{and} \quad C_{N_b+1}^{(m)} = 1). \quad (20)$$

In practice, one does not have to prepare  $C_n^{(m)}$  for all  $m$ 's, since  $C_n^{(m)}$  is readily expressed in terms of  $C_m^{(0)}$  and  $q_m^{(0)}$  as

$$C_n^{(m)} = \frac{p_{m+1}}{q_{m+1}^{(0)}} (C_n^{(0)} - C_m^{(0)}). \quad (21)$$

In other words, comparing  $C_n^{(m)}$  to a random number  $U$  ( $U \in [0, 1)$ ) is equivalent to comparing  $C_n^{(0)}$  to  $C_m^{(0)} + (q_{m+1}^{(0)}/p_{m+1})U$ .

Besides an initial table setup, which requires  $O(N_b)$  operations, searching an element in the table can be performed very quickly by using the binary search algorithm. The number of operations required for each search is  $O(\log N_b) = O(\log N)$ , which is significantly smaller than  $O(N_b)$  for the naive sequential search. Since the average number of candidate bonds is  $O(N)$ , a whole Monte Carlo sweep is accomplished by  $O(N \log N)$  operations on average.

### 3. New $O(N)$ cluster algorithm

#### 3.1. Formulation of $O(N)$ method

The factor  $\log N$  in the method of Luijten and Blöte is due to the fact that they use the binary search algorithm in looking for a next candidate. This factor might be removed if one can use some  $O(1)$  method instead of the binary search. Walker's method of alias [13,14] has been known as such an  $O(1)$  method to generate integral random numbers according to arbitrary probability distribution for a long time (Appendix A), and is a potential candidate for the replacement. Unfortunately, for the Walker method one can not use the smart trick presented in Eq. (21) for reducing the number of tables. It means that one has to prepare a table of length  $O(N_b)$  for each  $m$  before starting the simulation. The total amount of memory storage for storing all the tables is thus  $O(N_b^2) = O(N^4)$ , which is not acceptable in practice. In the following, we present a different approach

based on the extended Fortuin–Kasteleyn representation, which solves the storage problem and enables us to use the efficient  $O(1)$  method by Walker with reasonable storage requirement,  $O(N_b)$  (or  $O(N)$  for systems with translational invariance).

Our central idea is assigning a nonnegative integer to each bond instead of a binary (active or inactive). The integer to be assigned is generated according to the Poisson distribution. The probability that a Poisson variable takes an integer  $k$  is given by

$$f(k; \lambda) = \frac{e^{-\lambda} \lambda^k}{k!}, \tag{22}$$

where  $\lambda$  is the mean of the distribution. Note that  $f(0; \lambda) = e^{-\lambda}$  and therefore

$$\sum_{k=1}^{\infty} f(k; \lambda) = 1 - e^{-\lambda}, \tag{23}$$

which is equal to  $p_\ell$  in Eq. (15), if one puts  $\lambda$  to be  $2\beta J_\ell$ . That is, if one generates an integer according to the Poisson distribution with mean

$$\lambda_\ell = 2\beta J_\ell, \tag{24}$$

then it takes a nonzero value with probability  $p_\ell$ . Thus the bond-activation procedure in the conventional Swendsen–Wang algorithm can be modified as follows: Generate a Poisson variable for each bond with mean  $2\beta J_\ell$ , then activate the bond only when the variable is nonzero and the spins are parallel. At first glance, it seems that the situation is getting worse, since a Poisson random number, instead of a binary, is needed for each bond. At this point, however, we leverage an important property of the Poisson distribution: the Poisson process is that for random events and there is no statistical correlation between each two events. It allows us to realize the whole distribution by calculating just one Poisson random variable with mean  $\lambda_{\text{tot}} = \sum_\ell \lambda_\ell$ . The following identity clearly represents the essence:

$$\prod_{\ell=1}^{N_b} f(k_\ell; \lambda_\ell) = f(k_{\text{tot}}; \lambda_{\text{tot}}) \frac{(k_{\text{tot}})!}{k_1! k_2! \dots k_{N_b}!} \prod_{\ell=1}^{N_b} \left( \frac{\lambda_\ell}{\lambda_{\text{tot}}} \right)^{k_\ell}, \tag{25}$$

where  $k_{\text{tot}} = \sum_\ell k_\ell$ . This identity is verified in a straightforward way by substituting Eq. (22) in both hands. The left-hand side of Eq. (25) is the probability that  $k_\ell$  is assigned to each bond. The right-hand side, on the other hand, stands for the probability of generating a single Poisson number  $k_{\text{tot}}$  and then distributing  $k_\ell$  events to each bond with the weight proportional to  $\lambda_\ell$ . Distributing each event can be carried out in a constant time using Walker’s method of alias. Since generating a Poisson number with mean  $\lambda_{\text{tot}}$  takes only  $O(\lambda_{\text{tot}})$  time on average, the number of operations of the whole procedure is also proportional to  $\lambda_{\text{tot}} = 2\beta \sum_\ell J_\ell$ , which is  $O(N)$  for energy converging models.

Before closing this section, let us describe our  $O(N)$  algorithm in terms of an extended Fortuin–Kasteleyn representation. Introducing a configuration  $k = (k_1, k_2, \dots, k_{N_b})$  instead of  $g = (g_1, g_2, \dots, g_{N_b})$  in the original representation, the partition function is expressed as

$$Z = \sum_c \sum_k \prod_{\ell=1}^{N_b} \Delta(\sigma_\ell, k_\ell) V_\ell(k_\ell) = \sum_c \prod_{\ell=1}^{N_b} \sum_{k_\ell=0}^{\infty} \Delta(\sigma_\ell, k_\ell) V_\ell(k_\ell) \tag{26}$$

with

$$\Delta(\sigma, k) = \begin{cases} 0 & \text{if } k \geq 1 \text{ and } \sigma = -1, \\ 1 & \text{otherwise,} \end{cases} \tag{27}$$

$$V_\ell(k) = \frac{e^{-\beta J_\ell} (2\beta J_\ell)^k}{k!}. \tag{28}$$

The original partition function is easily recovered by performing the summation over  $k_\ell$ ’s first.

### 3.2. Pseudo-algorithm

Based on the extended Fortuin–Kasteleyn representation formulated above, we now describe one Monte Carlo sweep of the present  $O(N)$  algorithm:

- (1) Generate a nonnegative integer  $k$  according to the Poisson distribution with mean  $\lambda_{\text{tot}}$ .
- (2) Repeat the following procedure  $k$  times:
  - (2-a) Choose a bond  $\ell$  with probability

$$\frac{\lambda_\ell}{\lambda_{\text{tot}}} = \frac{J_\ell}{\sum_{\ell'=1}^{N_b} J_{\ell'}} \tag{29}$$

by using Walker’s method of alias.

(2-b) If  $\sigma_\ell = 1$  then activate bond  $\ell$ . If the bond is already activated, just do nothing.

(3) Flip each cluster with probability 1/2.

The probability that a particular bond  $\ell$  is activated during the iteration in step (2) is evaluated as

$$\delta_{\sigma_\ell, 1} \sum_{k=0}^{\infty} f(k; \lambda_{\text{tot}}) \sum_{k_\ell=1}^k \frac{k!}{(k-k_\ell)!k_\ell!} \left(1 - \frac{\lambda_\ell}{\lambda_{\text{tot}}}\right)^{k-k_\ell} \left(\frac{\lambda_\ell}{\lambda_{\text{tot}}}\right)^{k_\ell} = \delta_{\sigma_\ell, 1} \sum_{k=0}^{\infty} f(k; \lambda_{\text{tot}}) \left[1 - \left(1 - \frac{\lambda_\ell}{\lambda_{\text{tot}}}\right)^k\right] = \delta_{\sigma_\ell, 1} [1 - \exp(-\lambda_\ell)], \quad (30)$$

which is equal to the probability given in Eq. (3) or Eqs. (14) and (15). Thus, one can confirm that the present  $O(N)$  algorithm realizes the identical stochastic dynamics as the Swendsen–Wang and Luijten–Blöte methods.

Furthermore, for a system with translational invariance, step (2-a) in the above procedure can be replaced by

(2-a') Choose a site  $i$  with probability  $1/N$ , then choose another site  $j$  with probability

$$\frac{J_{ij}}{\sum_{j' \neq i} J_{ij'}}. \quad (31)$$

In this way, the size of tables for the modified probabilities and alias numbers used in the Walker method (see Appendices A and B for details) can also be reduced from  $O(N^2)$  down to  $O(N)$ .

### 3.3. Total energy and specific heat measurement

Measuring the total energy is also costly for long-range interacting models. In Ref. [15], Krech and Luijten proposed a method based on the fast Fourier transform. In this section, however, we show that the total energy and the specific heat are also calculated in  $O(N)$  time in the present algorithm. Indeed, the both quantities are obtained free of charge during Monte Carlo sweeps.

Let us consider the expression for the energy in the extended Fortuin–Kasteleyn representation. Differentiating the partition function (26) with respect to the inverse temperature, we obtain

$$E = -\frac{\partial}{\partial \beta} \ln \left[ \sum_c \sum_k W(c, k) \right] = \frac{\sum_c \sum_k \sum_\ell (J_\ell - k_\ell/\beta) W(c, k)}{\sum_c \sum_k W(c, k)} = J_{\text{tot}} - \frac{1}{\beta} \left\langle \sum_\ell k_\ell \right\rangle_{\text{MC}}, \quad (32)$$

where  $J_{\text{tot}} = \sum_\ell J_\ell$ , and  $\langle \cdots \rangle_{\text{MC}}$  denotes the Monte Carlo average of an observable in the present  $O(N)$  algorithm. Thus, in order to calculate the total energy, nothing more than the information one uses during Monte Carlo sweeps is needed. It also applies to the specific heat. Differentiating the right-hand side of Eq. (32) once again, one obtains the following expression

$$\begin{aligned} C &= -\frac{\beta^2}{N} \frac{dE}{d\beta} = -\frac{\beta^2}{N} \left[ \frac{1}{\beta^2} \left\langle \sum_\ell k_\ell \right\rangle_{\text{MC}} - \left\langle \left( J_{\text{tot}} - \frac{1}{\beta} \sum_\ell k_\ell \right)^2 \right\rangle_{\text{MC}} + \left\langle J_{\text{tot}} - \frac{1}{\beta} \sum_\ell k_\ell \right\rangle_{\text{MC}}^2 \right] \\ &= \frac{1}{N} \left[ \left\langle \left( \sum_\ell k_\ell \right)^2 \right\rangle_{\text{MC}} - \left\langle \sum_\ell k_\ell \right\rangle_{\text{MC}}^2 - \left\langle \sum_\ell k_\ell \right\rangle_{\text{MC}} \right] \end{aligned} \quad (33)$$

for the specific heat, which is not simply a variance of energy (32) but has an extra term  $\langle \sum_\ell k_\ell \rangle_{\text{MC}}$ . We note that the expressions for the total energy (32) and the specific heat (33) have a close relation with those for the quantum Monte Carlo method in the continuous imaginary-time path integral or the high-temperature series representations [16].

### 3.4. Quantum cluster algorithm for transverse-field Ising model

The Monte Carlo method has also been applied to various quantum spin systems and achieved great success [17], though for frustrated quantum models, in which the quantum Monte Carlo method suffers from the notorious negative sign problem, alternative numerical approaches, such as the transfer-matrix renormalization group [18,19], might be preferable. The present  $O(N)$  Monte Carlo algorithm can be extended quite naturally to quantum spin systems with long-range interactions. In this section, as a simplest example, we present a quantum cluster algorithm for the long-range Ising model in a transverse external field. Application to other quantum spin models, such as the Heisenberg or the XY models, is also straightforward.

The Hamiltonian of the transverse-field Ising model with long-range interactions is defined as

$$\mathcal{H} = -\sum_{i < j} J_{ij} \sigma_i^z \sigma_j^z - \sum_{i=1}^N \Gamma \sigma_i^x, \quad (34)$$

where  $\Gamma$  denotes the strength of transverse external field, and  $\sigma_i^x$  and  $\sigma_i^z$  are the Pauli spin operators at site  $i$ , which form an orthogonal basis together with the identity operator and  $\sigma_i^y$ . According to the standard prescription [20], we start with divid-

ing the Hamiltonian (34) into two parts, bond terms  $\mathcal{H}_b = -\sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z$  and site terms  $\mathcal{H}_s = -\sum_{i=1}^N \Gamma \sigma_i^x$ . The partition function is then expanded as

$$Z = \text{Tr} e^{-\beta \mathcal{H}} = \lim_{M \rightarrow \infty} \sum_{\phi_1} \langle \phi_1 | \left( e^{-\frac{\beta}{M} \mathcal{H}_b} e^{-\frac{\beta}{M} \mathcal{H}_s} \right)^M | \phi_1 \rangle = \lim_{M \rightarrow \infty} \sum_{\phi_1, \dots, \phi_M} \prod_{m=1}^M e^{-\frac{\beta}{M} E_m} \langle \phi_m | e^{-\frac{\beta}{M} \mathcal{H}_s} | \phi_{m+1} \rangle, \tag{35}$$

where  $M$  is called the Trotter number, which represents the number of slices ( $m = 1, 2, \dots, M$ ) along the imaginary-time axis. In Eq. (35), we inserted the identities  $\sum_{\phi_m} |\phi_m\rangle \langle \phi_m|$  between the operators. The basis set  $\{\phi_m\}$  is chosen so that  $\{\sigma_i^z\}$  are diagonalized (and so is  $\mathcal{H}_b$ ), and  $E_m \equiv \langle \phi_m | \mathcal{H}_b | \phi_m \rangle$ . We impose the periodic boundary conditions in the imaginary-time direction:  $\phi_{M+1} = \phi_1$ . Expanding the exponential operators of the site Hamiltonian to the first order, we obtain the following discrete imaginary-time path integral:

$$\begin{aligned} Z &= \lim_{M \rightarrow \infty} \sum_{\phi_1, \dots, \phi_M} \prod_{m=1}^M e^{-\frac{\beta}{M} E_m} \langle \phi_m | \prod_{i=1}^N \left[ 1 + \frac{\beta \Gamma}{M} (\sigma_i^+ + \sigma_i^-) \right] | \phi_{m+1} \rangle \\ &= C \lim_{M \rightarrow \infty} \sum_{\phi_1, \dots, \phi_M} \prod_{m=1}^M \left[ \prod_{i<j} e^{\frac{\beta J_{ij}}{M} \sigma_i^{(m)} \sigma_j^{(m)}} \right] \left[ \prod_{i=1}^N e^{-\frac{1}{2} \ln \left( \frac{\beta \Gamma}{M} \right) \sigma_i^{(m)} \sigma_i^{(m+1)}} \right] \\ &= C \lim_{M \rightarrow \infty} \sum_{\phi_1, \dots, \phi_M} \exp \left[ \sum_{m=1}^M \sum_{i<j} \frac{\beta J_{ij}}{M} \sigma_i^{(m)} \sigma_j^{(m)} - \sum_{m=1}^M \sum_{i=1}^N \frac{1}{2} \ln \left( \frac{\beta \Gamma}{M} \right) \sigma_i^{(m)} \sigma_i^{(m+1)} \right], \end{aligned} \tag{36}$$

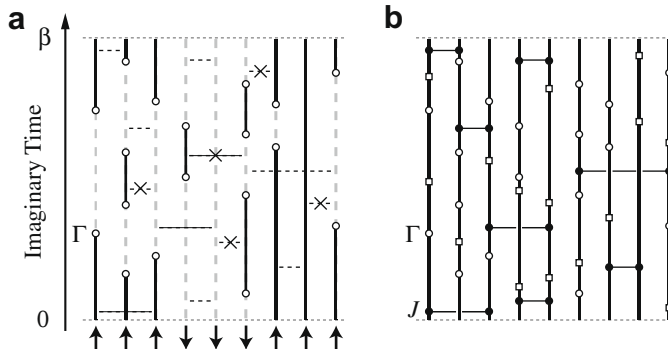
where  $\sigma_i^\pm \equiv (\sigma_i^x \pm i \sigma_i^y)/2$  are the spin ladder operators,  $\sigma_i^{(m)} \equiv \langle \phi_m | \sigma_i^z | \phi_m \rangle$ , and  $C$  is a constant. Thus, the partition function of the transverse-field Ising chain of  $N$  sites is represented by that of a two-dimensional classical Ising model of  $M \times N$  sites, where the interactions are long-ranged along one axis (real space direction) and short-ranged along the other axis (imaginary-time direction). The coupling constants in both directions are  $\beta_{\text{cl}} J_{ij}^{\text{(space)}} = \beta J_{ij}/M$  and  $\beta_{\text{cl}} J^{\text{(time)}} = -\frac{1}{2} \ln(\beta \Gamma/M)$ , respectively, where  $\beta_{\text{cl}}$  is a fictitious inverse temperature of the mapped system. The  $O(N)$  cluster algorithm presented in the previous subsection is then applied to this classical Ising model straightforwardly.

Furthermore, it has been shown that one can take the Trotter limit ( $M \rightarrow \infty$ ) in Eq. (36), and perform Monte Carlo simulations directly in the imaginary-time continuum [21,22]. It is possible because the coupling constant along the imaginary-time axis  $J^{\text{(time)}}$  increases as  $M$  does. The average number of antiparallel pairs (or kinks) remains finite even in the continuous-time limit, and therefore one does not have to take configurations with infinite number of kinks into account. Specifying the number of kinks by  $n$  and its space-time position by  $(\tau_p, s_p)$  ( $p = 1, 2, \dots, n$ ), we obtain the continuous-time path integral representation of the partition function

$$Z = \sum_{\phi_0} \left[ e^{-\beta E_0} + \sum_{n=1}^{\infty} \sum_{\{s_p\}} \int_0^\beta d\tau_1 \int_{\tau_1}^\beta d\tau_2 \cdots \int_{\tau_{n-1}}^\beta d\tau_n \Gamma^n \prod_{p=1}^{n+1} e^{-(\tau_p - \tau_{p-1}) E_{p-1}} \right], \tag{37}$$

where  $\tau_0 = 0$ ,  $\tau_{n+1} = \beta$ , and  $E_p$  is the diagonal energy  $\langle \phi | \mathcal{H}_b | \phi \rangle$  of the spin configuration between the  $p$ th and  $(p + 1)$ th kinks ( $E_n = E_0$ ). In Fig. 1(a) an example of path integral configuration is shown.

The cluster algorithm is also defined directly in the Trotter limit. Since the activation probability of temporal bond with parallel spins,  $1 - \exp(2\beta_{\text{cl}} J^{\text{(time)}}) = 1 - \beta \Gamma/M$ , becomes almost unity for  $M \gg 1$ , the probability of finding  $n$  inactive links (open squares in Fig. 1) in a uniform temporal segment of unit imaginary time, which contains  $M/\beta$  Trotter slices, is given by a Poisson distribution,



**Fig. 1.** (a) Example of the space-time configuration in the continuous-time path integral representation. The arrows at the bottom denote  $\phi_0$ . Solid and broken lines denote the continuously aligned up and down spins, respectively. The open circles represent the space-time position where a ladder operator is inserted. (b) Possible graph configuration assigned to spin configuration (a). The open squares represent the positions where the temporal bond is deactivated, while the filled circle represent those where a spatial bond is activated. The spatial long-range bonds are activated only when the spins are parallel at both ends as depicted in (a), where the candidates connecting antiparallel spins are rejected (x-marks).



$$M/\beta C_n (1 - \beta\Gamma/M)^{(M/\beta-n)} (\beta\Gamma/M)^n \approx f(n, \Gamma). \quad (38)$$

Similarly, the probability of finding  $n$  spatial candidate links (horizontal dashed lines in Fig. 1) between parallel spins at site  $i$  and  $j$  in unit imaginary time is  $f(n, 2J_{ij})$ . After all, the overall probability of finding  $n$  events in total at some site or bond is given by  $f(n, A)$  with

$$A = N\Gamma + 2J_{\text{tot}}. \quad (39)$$

Since these events are statistically independent with each other, a series of events is generated successively by using the exponential distribution for the temporal interval  $t$  between two events:

$$p(t)dt = Ae^{-At}dt. \quad (40)$$

At each imaginary time, then a site or bond is chosen according to the probabilities  $\Gamma/A$  or  $2J_{ij}/A$ , respectively. This is again done in a constant time by using the Walker method. If a site is chosen, the temporal bond is deactivated, i.e., clusters are disconnected at this space-time position. If a bond is selected (and if the spins on its ends are parallel), on the other hand, a spatial link is inserted, i.e., two sites are connected at this imaginary time (horizontal solid lines in Fig. 1). At the space-time points where the spin changes its direction (open circles in Fig. 1), we always deactivate the temporal bond. By repeating this procedure until the imaginary time  $\beta$  is reached, the whole lattice is divided into several clusters [Fig. 1(b)]. Finally each cluster is flipped with probability  $1/2$  to generate a terminal configuration.

The number of operations per Monte Carlo sweep is proportional to the number of generated events. Its average is given by  $\beta A$ , which is proportional to the system size  $N$  as the  $O(N)$  algorithm for classical models. We note that the  $O(N)$  quantum cluster algorithm presented in this section is also formulated in the same way in the high-temperature series representation [10].

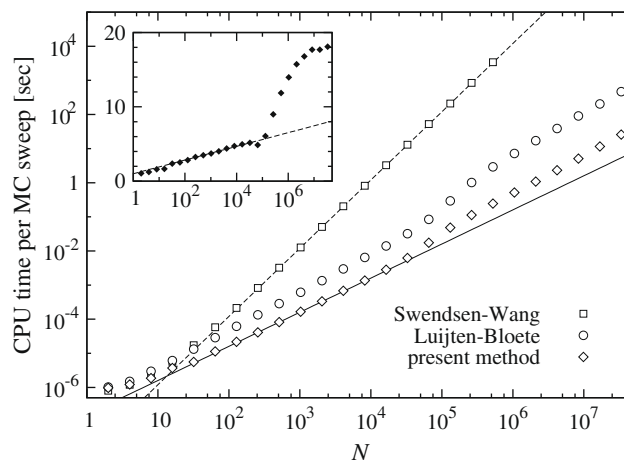
#### 4. Performance test

In order to demonstrate the efficiency of the present method, we carried out Monte Carlo simulations for the classical mean-field (or infinite-range) model of various system sizes ( $N = 2, 4, \dots, 2^{25}$ ). We use the naive Swendsen–Wang and Luijten–Blöte methods as benchmarks. The coupling constants of the mean-field model is given by

$$J_{ij} = \frac{1}{N} \quad (41)$$

for all  $i \neq j$ . The denominator  $N$  is introduced to prevent the energy density of the system from diverging in the thermodynamic limit. We choose the mean-field model as a severest test case for these algorithms, though simpler and faster algorithms, even exact analytic results, exist for this specific model. The benchmark test was performed on a PC workstation (CentOS Linux 5.1, Intel Xeon 3.2GHz, 1MB cache, GNU C++ 4.1.1).

We confirm that all these algorithms produce the same results, total energy, specific heat, magnetization density squared, Binder cumulant, etc, within the error bar in the whole temperature range we simulated. The CPU time spent for one Monte Carlo sweep at the critical temperature ( $T = 1$ ) is shown in Fig. 2. For the naive Swendsen–Wang algorithm, as one expects,



**Fig. 2.** System size dependence of CPU time per Monte Carlo sweep of the Swendsen–Wang (squares), Luijten–Blöte (circles), and the present (diamonds) methods for the mean-field model at  $T = 1$ . The solid and dashed lines indicate the theoretical asymptotic scaling for the Swendsen–Wang ( $N^2$ ) and the present ( $N$ ) methods, respectively. The Luijten–Blöte and the present methods both show an anomaly at  $N \approx 10^5$ , which is attributed to the occurrence of cache miss. In the inset, the relative speed of the present algorithm to that of the Luijten–Blöte method is also shown, where the dashed line indicates a  $\log N$  scaling.



the CPU time grows rapidly as  $N^2$ . On the other hand, it is clearly seen in Fig. 2 that the present algorithm has a different scaling, linear to the system size, and is indeed much faster than the Swendsen–Wang method except for very small system sizes ( $N \leq 4$ ). The present and Luijten–Blöte methods exhibit a similar scaling behavior, but the former is faster for all the system sizes we simulated. To see the difference in scaling behavior in detail, we plot the relative speed of the present algorithm to the latter in the inset of Fig. 2. For  $N \leq 10^5$ , it scales as  $\log N$ , which is consistent with the performance difference between the Walker and the binary search algorithms. Around the system size  $N \approx 10^5$ , however, the results for the present algorithm start to deviate from the  $N$ -linear scaling. Those for the Luijten–Blöte method shows a similar anomalous behavior, but the situation is much worse in this case as seen in the inset of Fig. 2. We attribute these anomalies to the occurrence of cache miss, for simulating system of  $N = 10^5$  spins requires about 1.6 MB of memory ( $4N$  byte for spin configuration and  $(8 + 4)N$  byte for modified probabilities and aliases in the Walker method), which is comparable with the cache memory size (1 MB) of the system used for the present benchmark test. The naive Swendsen–Wang method should also suffer from the same problem, but in the present benchmark test its effect seems to be hidden under the quadratic growth in the number of operations.

In summary, among the existing three algorithms the present  $O(N)$  method is the fastest except for very small system sizes. Especially, it outperforms the Swendsen–Wang method by four orders of magnitude at  $N = 2^{19}$  and the Luijten–Blöte method by about factor twenty at  $N = 2^{25}$ . This efficiency of the present method enables us to simulate much larger systems or further improve statistics as compared with the previous Monte Carlo studies, as demonstrated in the next section.

### 5. Kosterlitz–Thouless transition in Ising chain with inverse-square interactions

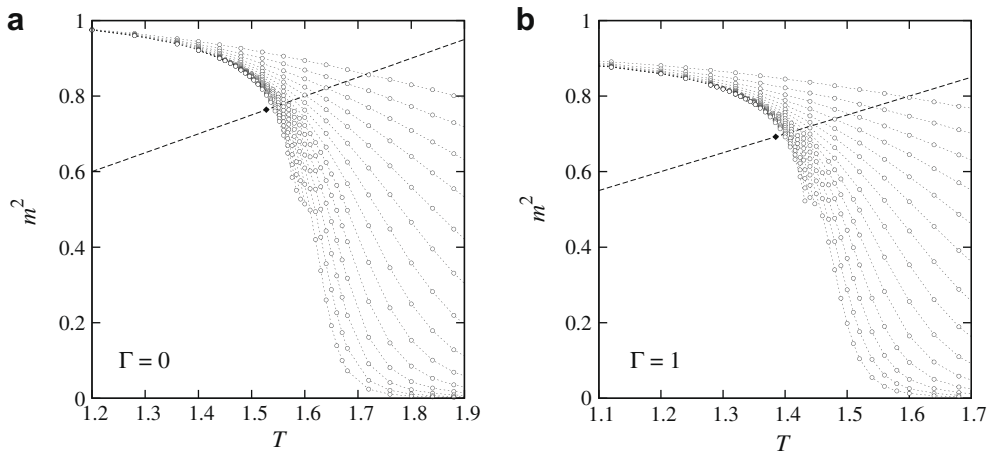
In this section, as a nontrivial example, we apply our  $O(N)$  cluster algorithm to the phase transition of the one-dimensional Ising model with inverse-square interactions [Eq. (2) with  $\alpha = 1$ ]. As we mentioned in the introduction, among the models with algebraically decaying interactions, this model is special as a boundary case, i.e., it has the weakest (or shortest) interactions to trigger a finite-temperature phase transition. What is more, this phase transition belongs to the same universality class as the Kosterlitz–Thouless transition [3–5], where logarithmic excitations brought by formation of domain walls compete with the entropy generation. The Kosterlitz–Thouless transition is characterized by an exponential divergence of the correlation length toward the critical temperature  $T_{KT}$  and a finite jump in the magnetization. Especially, the amount of the magnetization gap at the critical point is conjectured to satisfy the following universal relation:

$$2m^2 = T_{KT}, \tag{42}$$

where  $m^2 = \langle (\sum_i \sigma_i^z)^2 \rangle / N^2$ , being the square of magnetization density. For the classical Ising chain with inverse-square interactions, it is confirmed that a phase transition of Kosterlitz–Thouless universality occurs by an extensive Monte Carlo study [23].

We perform Monte Carlo simulations by using the  $O(N)$  cluster algorithm for the chain length  $L = 2^3, 2^4, \dots, 2^{20}$  ( $= 1048576$ ). We impose periodic boundary conditions. In order to minimize the effect of boundary conditions, we use the following renormalized coupling constant

$$\tilde{J}_{ij} = \sum_{n=-\infty}^{\infty} \frac{1}{(i-j-nL)^2} = \frac{\pi^2}{L^2 \sin^2 \frac{\pi(i-j)}{L}}, \tag{43}$$



**Fig. 3.** Temperature dependence of magnetization density squared for (a) classical ( $\Gamma = 0$ ) and (b) quantum ( $\Gamma = 1$ ) Ising chains with inverse-square interactions. System sizes are  $L = 2^3, 2^4, \dots, 2^{20}$  from the top to the bottom. The error bar of each data point is much smaller than the symbol size. The dashed lines denote the universal jump relation Eq. (42). The filled diamond indicates the critical temperature obtained from the finite-size scaling analysis (see Fig. 4 below) and the magnetization density squared just below the critical point.

in which contribution from all periodic images is taken into account. It reduces to the bare coupling constant (2) in the thermodynamic limit  $L \rightarrow \infty$ . Measurement of physical quantities is performed for 524288 Monte Carlo sweeps after discarding 8192 sweeps for thermalization.

In Fig. 3, we show the temperature dependence of the magnetization density squared for (a)  $\Gamma = 0$  and (b)  $\Gamma = 1$ . For the classical system ( $\Gamma = 0$ ), our results coincide quite well with the previous Monte Carlo study [23]. In both cases,  $m^2$  decreases monotonically as the temperature increases. At high temperatures,  $m^2$  vanishes quite rapidly as the system size increases, while it seems converging to a finite value in the low temperature regime though the convergence is rather slow. This suggests an emergence of long-range order at some finite critical temperature. At low temperatures,  $m^2$  of the quantum system is smaller than the classical one. Indeed,  $m^2 < 1$  even at  $T = 0$  for  $\Gamma = 1$ , which is in contrast to the classical case,  $m^2 = 1$ . This is due to quantum fluctuations introduced by the transverse external field. In a previous quantum Monte Carlo study [10], intersections of magnetization curves for different system sizes at intermediate temperatures have been reported. In the present study, however, we do not observe such a nonmonotonic behavior regardless of the system size. We would attribute this discrepancy to a relaxation problem in the Monte Carlo calculation in Ref. [10], where only a local flip scheme is used for updating spin configurations.

In order to discuss the critical behavior in detail, next we perform a finite-size scaling analysis. As is well known, the standard finite-size scaling technique does not work in the case of the Kosterlitz–Thouless transition, for the correlation length exhibits an exponential divergence. Instead of the ordinary finite-size scaling, which depends on an algebraic divergence of the correlation length, an alternative scaling form for the magnetization has been suggested from the renormalization group equations [5,24,25]:

$$\frac{2m^2}{T} - 1 = \ell^{-1} F(t\ell^2), \quad (44)$$

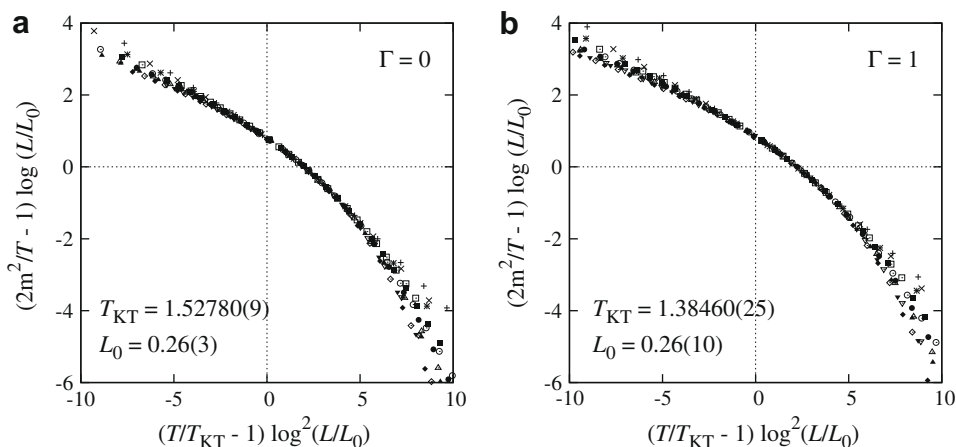
where  $F(x)$  is a scaling function,  $t = T/T_{\text{KT}} - 1$ ,  $\ell = \log(L/L_0)$ , and  $L_0$  a constant. It is confirmed that this finite-size scaling assumption works well for the two-dimensional XY model [25], in which the helicity modulus, instead of the magnetization, is the quantity exhibiting a universal jump.

In Fig. 4, we show the scaling plots for  $\Gamma = 0$  and 1. Both data are scaled excellently by using the same scaling form (44), where we have only two fitting parameters,  $T_{\text{KT}}$  and  $L_0$ . This strongly supports that the magnetization shows the universal jump (42) at the critical point. From these scaling plots we conclude

$$T_{\text{KT}} = \begin{cases} 1.52780(9) & \text{for } \Gamma = 0 \\ 1.38460(25) & \text{for } \Gamma = 1 \end{cases} \quad (45)$$

for the Kosterlitz–Thouless critical temperature. The result for the classical case is compared with that in the previous Monte Carlo study,  $T_{\text{KT}} = 1.5263(4)$  [23], which differs slightly beyond the error bar. This tiny discrepancy might be due to the difference in the way of scaling analysis. In Ref. [23] the magnetization data at low temperatures are first extrapolated to the thermodynamic limit, then further extrapolated towards the critical point, whereas the Monte Carlo data are directly used to estimate the critical temperature in the present finite-size scaling analysis. Thus, we expect that the present estimate for  $T_{\text{KT}}$  is more reliable.

As for the quantum system ( $\Gamma = 1$ ), the critical temperature is lower than the classical one due to the quantum fluctuations. However, the finite-scaling analysis confirms that the phase transition belongs to the Kosterlitz–Thouless universality class as in the classical case. The finite-size scaling plots shown in Fig. 4 suggest that the scaling function itself is universal as



**Fig. 4.** Scaling plot of magnetization density squared for (a) classical ( $\Gamma = 0$ ) and (b) quantum ( $\Gamma = 1$ ) Ising chains with inverse-square interactions. System sizes are  $L = 2^8$  (crosses),  $2^{10}$  (x-marks),  $\dots$ ,  $2^{20}$  (filled diamonds). The error bar of each data point is much smaller than the symbol size.

well. As  $\Gamma$  is increased further,  $T_{KT}$  decreases monotonically, and it finally vanishes at a critical transverse field  $\Gamma_c$  ( $\approx 2.52$  [26]), where a quantum phase transition occurs. At this point some exotic quantum critical behavior with a nontrivial dynamical exponent  $z$  is expected, since the  $(1+1)$ -dimensional system is extremely anisotropic, i.e., in real space direction the system has long-range interactions, whereas the interaction in the imaginary-time direction is still short ranged. More detailed analyses on the quantum criticality of the transverse-field Ising model will be presented elsewhere [26].

## 6. Summary and discussion

We presented an  $O(N)$  cluster algorithm for Ising models with long-range interactions. The algorithm proposed in the present paper is ‘exact’ in the sense that it does not introduce any cutoff for interaction range and thus it strictly fulfills the detailed balance. Our algorithm is formulated based on the extended Fortuin–Kasteleyn representation, where bond variables have a nonnegative integral value instead of a binary number. For each bond, an integer is generated according to the Poisson distribution. However, it does not necessarily mean that each Poisson variable has to be generated one by one. We show that generating an overall Poisson distribution and ex-post assignment of events, using Walker’s method of alias, are statistically equivalent to the naive Swendsen–Wang method. In Section 4, we demonstrated the  $N$ -linear scaling behavior in the CPU time for the mean-field model.

The present method has several advantages over the existing methods, such as the Metropolis method, the Swendsen–Wang algorithm [7], the improvement by Luijten and Blöte [9], or the recently proposed  $O(N)$  method [27], in several aspects: (a) The CPU time per Monte Carlo sweep is  $O(N)$ . (b) It works effectively both for short-range and long-range interacting models. (c) It is a cluster algorithm and free from the critical slowing down near the critical point. (d) It is possible to formulate a single-cluster variant [8]. (e) It is very easy to implement the algorithm, based on an existing Swendsen–Wang code. (f) It can also be used for systems without translational invariance, though it once costs  $O(N^2)$  to initialize lookup tables. (g) Calculation of the total energy and the specific heat can be done all together in  $O(N)$  time. (h) It can be applied to Potts, XY, and Heisenberg models with the help of Wolff’s embedding technique [8]. (i) Extension to quantum models, such as the transverse-field Ising model or the Heisenberg model, is also possible straightforwardly.

In Section 5, we have applied our new algorithm to the phase transition of Ising model with inverse-square interactions, where we see that the  $O(N)$  method works ideally for both of the classical and quantum systems. It is confirmed in a high accuracy that the phase transition belongs to the same universality as the Kosterlitz–Thouless transition.

Finally, let us discuss the efficiency of the present algorithm at very low temperatures. The Swendsen–Wang algorithm itself works quite effectively even away from the critical point. At high temperatures, since the lattice will be divided into a set of tiny clusters, the stochastic dynamics becomes essentially the same as the conventional single spin flip update. On the other hand, at low temperatures, a huge cluster tends to span the whole lattice, and thus the ordered states with all up and all down spin configurations can be connected with each other by a single Monte Carlo step. However, at low temperatures, the calculation cost of the present method for a fixed system size  $N$  grows linearly as  $\beta$  increases since  $\lambda_{\text{tot}}$  is proportional to  $\beta$ , whereas that of the naive Swendsen–Wang method is constant regardless of the temperature for classical Ising models. It indicates that at lower temperatures than some threshold  $1/\beta_{\text{thresh}}$ , the naive method outperforms the present method. At extremely low temperatures, almost all the bonds are activated. The present method then activates such bonds many times, which is the cause of the slowing down. Although the  $\beta$ -linear increase of CPU time is inevitable for quantum systems, where the standard quantum Monte Carlo algorithms for short-range models also suffer from the same slowing down, however, one can adopt a ‘‘hybrid’’ scheme to optimize the calculation cost at intermediate temperatures,  $\max(J_\ell) \lesssim \beta < \beta_{\text{thresh}}$  for classical models. Suppose  $J_\ell$ ’s are sorted in descending order, and we use the naive method for the first  $n$  bonds and the  $O(N)$  method for the others. The CPU time  $C(n)$  per Monte Carlo sweep is estimated as

$$C(n) \simeq An + B \sum_{\ell=n+1}^{N_b} 2\beta J_\ell, \quad (46)$$

where  $A$  and  $B$  are some constants. The optimal value of  $n$  is then given by  $\Delta C = C(n+1) - C(n) = A - 2B\beta J_n = 0$ . For the one-dimensional model with algebraically decaying interactions (2), for example, we have

$$\frac{n_{\text{opt}}}{N_b} \approx N^{-1} \left( \frac{2B\beta}{A} \right)^{\frac{1}{1+z}}. \quad (47)$$

The threshold  $\beta_{\text{thresh}}$  is defined as the inverse temperature where  $n_{\text{opt}} = N_b \simeq N^2$ , that is,  $\beta_{\text{thresh}} \approx (A/2B)N^{1+z}$ , which grows as the system size  $N$  increases.

## Acknowledgments

The authors would like to thank M. Sasaki and K. Hukushima for critical reading of the manuscript. The part of the simulations in the present paper has been done by using the facility of the Supercomputer Center, Institute for Solid State Physics, University of Tokyo. The simulation code has been developed based on the ALPS/looper library [28–30]. The authors

acknowledge support by Grant-in-Aid for Scientific Research Program (Nos. 18540369 and 20540364) from JSPS, and also by the Grand Challenge to Next-Generation Integrated Nanoscience, Development and Application of Advanced High-Performance Supercomputer Project from MEXT, Japan.

**Appendix A. Walker’s method of alias**

Consider a random variable  $X$  which takes an integral value  $i$  according to a probability  $p_i$  ( $1 \leq i \leq N$  and  $\sum p_i = 1$ ). In this appendix, we discuss how to generate such random numbers effectively. One of the simplest and the most well-known methods is the one based on rejection:

Rejection Method

- (1) Generate a uniform integral random variable  $M$  ( $1 \leq M \leq N$ ).
- (2) Generate a uniform real random variable  $U$  ( $0 \leq U < 1$ ).
- (3) If  $U$  is smaller than  $p_M/p_{\max}$  then  $X = M$ , otherwise repeat from (1).

Here,  $p_{\max} = \max(p_i)$ . Since the acceptance rate in step (3) is  $1/(Np_{\max})(\equiv q)$ , the probability of obtaining  $X = i$  eventually is  $\sum_{r=1}^{\infty} p_i(1 - q)^{r-1}q = p_i$ . Notice that the number of iterations is  $\sum_{r=1}^{\infty} r(1 - q)^{r-1}q = 1/q = Np_{\max}$  on average, and therefore it would take  $O(N)$  time for each generation. Especially, the efficiency decreases quite rapidly as the variance of  $p_i$  increases. One may reduce the number of operations down to  $O(\log N)$  by employing the binary search on the table of cumulative probabilities (see Section 2.2). However, there exists a further effective method, called “Walker’s method of alias” [13,14], which is rejection free and generates a random integer in a constant time.

The Walker algorithm requires two tables of size  $N$ , which need to be calculated in advance. One is the table of integral alias numbers  $\{A_i\}(1 \leq A_i \leq N)$  and the other is that of modified probabilities  $\{P_i\}(0 \leq P_i \leq 1)$ . Using these tables a random integer is generated by the following procedure:

Walker’s Method of Alias

- (1) Generate a uniform integral random variable  $M$  ( $1 \leq M \leq N$ ).
- (2) Generate a uniform real random variable  $U$  ( $0 \leq U < 1$ ).
- (3) If  $U$  is smaller than  $P_M$  then  $X = M$ , otherwise  $X = A_M$ .

This procedure has no iterations, and thus completes in a constant time. The meaning of the tables  $\{A_i\}$  and  $\{P_i\}$  and the correctness of the algorithm is readily understood with the following example:

$i$	1	2	3	4	5	6	7	8	9	10	11	12
$p_i$	0	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$
$P_i$	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0	$\frac{2}{3}$	$\frac{1}{3}$
$A_i$	10	9	8	*	*	5	6	6	7	7	8	8

The modified probabilities  $P_i$  are determined from  $p_i$  (see Appendix B), which gives the probabilities whether one should accept the firstly chosen number or choose the alias number  $A_i$ . Let us consider, for example, the probability of  $X = 9$ . There are two possibilities: One is  $M = 9$  and  $U < P_9$ , and the other is  $M = 2$  and  $U \geq P_2$  since  $A_2 = 9$ . The sum of these two probabilities is

$$\frac{1}{12}[P_9 + (1 - P_2)] = \frac{1}{9}, \tag{A.1}$$

which is equal to  $p_9$  as expected. One can confirm that  $\{P_i\}$  and  $\{A_i\}$  are given in the example so that

$$p_i = \frac{1}{N} \left[ P_i + \sum_{j=1}^N (1 - P_j) \delta_{iA_j} \right] \tag{A.2}$$

holds for  $i = 1, 2, \dots, N$ . Together with the ordinary requirement for probabilities,  $0 \leq P_i \leq 1 (i = 1, 2, \dots, N)$ , Eq. (A.2) is the necessary condition for  $\{P_i\}$  and  $\{A_i\}$  to satisfy.

In practice, when  $N$  is not a power of two, we expand the size of tables from  $N$  to  $N_{\text{opt}}$ , where  $N_{\text{opt}}$  is the smallest integer satisfying  $N_{\text{opt}} \geq N$ . For  $N + 1 \leq i \leq N_{\text{opt}}$ , we assume  $p_i = 0$ . In this way, generating  $M$  in step (1) is optimized as a bit shift operation on a 32- or 64-bit integral random number [14]. Furthermore, steps (2) and (3) can be replaced by a comparison between two integral variables by preparing a table of integers  $\{2^{32}P_i\}$  (or  $\{2^{64}P_i\}$ ) instead of floating point numbers  $\{P_i\}$ , by which a costly conversion from an integer to a floating point variable can also be avoided.

In summary, by using the Walker method, integral random numbers according to arbitrary probabilities can be generated in a constant time. This extreme efficiency is essential for the present  $O(N)$  cluster Monte Carlo method. In the next appendix, we describe how to prepare the tables  $\{P_i\}$  and  $\{A_i\}$ .

**Appendix B. Preparation of modified probabilities and aliases**

In the original paper by Walker [13] and also in the standard literature [14], only a naive  $O(N^2)$  method is presented for initializing  $\{P_i\}$  and  $\{A_i\}$ . Here we propose for the first time an efficient alternative procedure, which takes only  $O(N)$  time. Consider the following table of probabilities for  $\{p_i\}$ :

$i$	1	2	3	4	5	6	7	8	9	10	11	12
$p_i$	0	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$
$P_i$	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{3}{3}$	$\frac{4}{3}$	$\frac{5}{3}$	$\frac{6}{3}$	$\frac{5}{3}$	$\frac{4}{3}$	$\frac{3}{3}$	$\frac{2}{3}$	$\frac{1}{3}$

Here  $P_i$  is initially set to a tentative value  $Np_i$  for  $i = 1, \dots, N$ . First we rearrange the table so that all the elements with  $P_i \geq 1$  precede those with  $P_i < 1$

							$\nabla$											$\blacktriangledown$
$i$	4	5	6	7	8	9	10	12	11	3	2	1						
$P_i$	$\frac{3}{3}$	$\frac{4}{3}$	$\frac{5}{3}$	$\frac{6}{3}$	$\frac{5}{3}$	$\frac{4}{3}$	$\frac{3}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	0						

The rearrangement can be done by  $N$  steps in contrast to the perfect sorting, which is an  $O(N \log N)$  procedure. The white triangle points to the rightmost element with  $P_i \geq 1$  and the black triangle points to the rightmost element in the rearranged table.

Next, we determine the alias numbers  $A_i$  sequentially from the right. We fill the “shortfall”  $(1 - P_i)$  of the element pointed by the solid triangle by the one pointed by the white triangle. The latter is always large enough, since  $P_i \geq 1$  by definition. In the present example, first the shortfall of the rightmost element  $(1 - P_1) = 1$  is filled by the element with  $i = 10$ . The alias number for  $i = 1$  is then set to 10 and  $P_{10}$  is replaced by  $P_{10} - (1 - P_1) = 0$ . Since  $P_{10}$  is no more larger than nor equal to unity, we shift the white triangle to the left by one. We repeat the same for the next “unfilled” element. After four iterations, the table is transformed as follows:

							$\nabla$												$\blacktriangledown$
$i$	4	5	6	7	8	9	10	12	11	3	2	1							
$P_i$	$\frac{3}{3}$	$\frac{4}{3}$	$\frac{5}{3}$	$\frac{6}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	0							
$A_i$									8	8	9	10							

Here the solid and white triangles are shifted four and three times from their original positions, respectively. The above procedure is repeated until the black triangle points to the same element as the white one, i.e., all the elements get filled. One should note that the black triangle always moves by one after each iteration, though the white one may stay on the same element depending whether  $P_i \geq 1$  or not after the step. The whole procedure is thus completed at most after  $(N - 1)$  iterations. In the present example, after 10 iterations we end up with

																			$\blacktriangledown$
$i$	4	5	6	7	8	9	10	12	11	3	2	1							$\nabla$
$P_i$	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	0							
$A_i$	*	*	5	6	6	7	7	8	8	8	9	10							

which is equivalent to the table presented in Appendix A.

**References**

- [1] D. Ruelle, Statistical mechanics of a one-dimensional lattice gas, Commun. Math. Phys. 9 (1968) 267.
- [2] F.J. Dyson, Existence of a phase-transition in a one-dimensional Ising ferromagnet, Commun. Math. Phys. 12 (1969) 91.
- [3] P.W. Anderson, G. Yuval, Some numerical results on the Kondo problem and the inverse square one-dimensional Ising model, J. Phys. C 4 (1971) 607.
- [4] J.M. Kosterlitz, D.J. Thouless, Ordering, metastability and phase transitions in two-dimensional systems, J. Phys. C 6 (1973) 1181.
- [5] J.M. Kosterlitz, The critical properties of the two-dimensional XY model, J. Phys. C 7 (1974) 1046.
- [6] M.E. Fisher, S.K. Ma, B.G. Nickel, Critical exponents for long-range interactions, Phys. Rev. Lett. 29 (1972) 917.
- [7] R.H. Swendsen, J.S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, Phys. Rev. Lett. 58 (1987) 86.
- [8] U. Wolff, Collective Monte Carlo updating for spin systems, Phys. Rev. Lett. 62 (1989) 361.
- [9] E. Luijten, H.W.J. Blöte, Monte Carlo method for spin models with long-range interactions, Int. J. Mod. Phys. C 6 (1995) 359.

- [10] A.W. Sandvik, Stochastic series expansion method for quantum Ising models with arbitrary interactions, *Phys. Rev. E* 68 (2003) 056701.
- [11] C.M. Fortuin, P.W. Kasteleyn, On the random-cluster model I. introduction and relation to other models, *Physica* 57 (1972) 536.
- [12] N. Kawashima, J.E. Gubernatis, Generalization of the Fortuin–Kasteleyn transformation and its application to quantum spin simulations, *J. Stat. Phys.* 80 (1995) 169.
- [13] A.J. Walker, An efficient method for generating discrete random variables with general distributions, *ACM Trans. Math. Softw.* 3 (1977) 253.
- [14] D.E. Knuth, *The Art of Computer Programming, Seminumerical Algorithms*, third ed., vol. 2, Addison Wesley, Reading, 1997. 119.
- [15] M. Krech, E. Luijten, Optimized energy calculation in lattice systems with long-range interactions, *Phys. Rev. E* 61 (2000) 2058.
- [16] H.G. Evertz, The loop algorithm, *Adv. Phys.* 52 (2003) 1.
- [17] M. Suzuki (Ed.), *Quantum Monte Carlo Methods in Condensed Matter Physics*, World Scientific, 1993.
- [18] A. Klümper, R. Raupach, F. Schönfeld, Finite temperature density-matrix-renormalization-group investigation of the spin-peierls transition in  $\text{CuGeO}_3$ , *Phys. Rev. B* 59 (1999) 3612.
- [19] J. Sirker, A. Klümper, K. Hamacher, Ground-state properties of two-dimensional dimerized heisenberg models, *Phys. Rev. B* 65 (2002) 134409.
- [20] M. Suzuki, Relationship between  $d$ -dimensional quantum spin systems and  $(d + 1)$ -dimensional Ising systems, *Prog. Theor. Phys.* 56 (1976) 1454.
- [21] B.B. Beard, U.J. Wiese, Simulations of discrete quantum systems in continuous Euclidean time, *Phys. Rev. Lett.* 77 (1996) 5130.
- [22] H. Rieger, N. Kawashima, Application of a continuous time cluster algorithm to the two-dimensional random quantum Ising ferromagnet, *Eur. Phys. J. B* 9 (1999) 233.
- [23] E. Luijten, H. Meßingfeld, Criticality in one dimension with inverse square-law potential, *Phys. Rev. Lett.* 86 (2001) 5305.
- [24] H. Weber, P. Minnhagen, Monte Carlo determination of the critical temperature for the two-dimensional XY model, *Phys. Rev. B* 37 (1987) 5986.
- [25] K. Harada, N. Kawashima, Universal jump in the helicity modulus of the two-dimensional quantum XY model, *Phys. Rev. B* 55 (1997) R11949.
- [26] S. Todo, K. Fukui, in preparation.
- [27] M. Sasaki, F. Matsubara, Stochastic cutoff method for long-range interacting systems, *J. Phys. Soc. Jpn.* 77 (2008) 024004.
- [28] <http://wistaria.comp-phys.org/alps-looper/>.
- [29] S. Todo, K. Kato, Cluster algorithms for general-S quantum spin systems, *Phys. Rev. Lett.* 87 (2001) 047203.
- [30] A. Albuquerque, F. Alet, P. Corboz, P. Dayal, A. Feiguin, S. Fuchs, L. Gamper, E. Gull, S. Gürtler, A. Honecker, R. Igarashi, M. Körner, A. Kozhevnikov, A. Läuchli, S. Manmana, M. Matsumoto, I. McCulloch, F. Michel, R. Noack, G. Pawłowski, L. Pollet, T. Pruschke, U. Schollwöck, S. Todo, S. Trebst, M. Troyer, P. Werner, S. Wessel, The ALPS project release 1.3: Open-source software for strongly correlated systems, *J. Mag. Mag. Mat.* 310 (2007) 1187.