

Geometric allocation approach to accelerating directed worm algorithm

Hidemaro Suwa 

*Department of Physics and Astronomy, University of Tennessee, Knoxville, Tennessee 37996, USA
and Department of Physics, University of Tokyo, Tokyo 113-0033, Japan*

 (Received 6 November 2019; revised 16 November 2020; accepted 28 December 2020; published 13 January 2021)

The worm algorithm is a versatile technique in the Markov chain Monte Carlo method for both classical and quantum systems. The algorithm substantially alleviates critical slowing down and reduces the dynamic critical exponents of various classical systems. It is crucial to improve the algorithm and push the boundary of the Monte Carlo method for physical systems. We here propose a directed worm algorithm that significantly improves computational efficiency. We use the geometric allocation approach to optimize the worm scattering process: worm backscattering is averted, and forward scattering is favored. Our approach successfully enhances the diffusivity of the worm head (kink), which is evident in the probability distribution of the relative position of the two kinks. Performance improvement is demonstrated for the Ising model at the critical temperature by measurement of exponential autocorrelation times and asymptotic variances. The present worm update is approximately 25 times as efficient as the conventional worm update for the simple cubic lattice model. Surprisingly, our algorithm is even more efficient than the Wolff cluster algorithm, which is one of the best update algorithms. We estimate the dynamic critical exponent of the simple cubic lattice Ising model to be $z \approx 0.27$ in the worm update. The worm and the Wolff algorithms produce different exponents of the integrated autocorrelation time of the magnetic susceptibility estimator but the same exponent of the asymptotic variance. We also discuss how to quantify the computational efficiency of the Markov chain Monte Carlo method. Our approach can be applied to a wide range of physical systems, such as the $|\phi|^4$ model, the Potts model, the $O(n)$ loop model, and lattice QCD.

DOI: [10.1103/PhysRevE.103.013308](https://doi.org/10.1103/PhysRevE.103.013308)

I. INTRODUCTION

The Markov chain Monte Carlo (MCMC) method is a powerful numerical tool for studying a wide variety of statistical mechanical problems [1,2]. Many kinds of nontrivial phases and phase transitions in both classical and quantum systems have been uncovered by the MCMC method. The essence of the method is to construct a global transition kernel as a series of *local* kernels acting on local state variables. One can sample states from an arbitrary target distribution even in a vast number of dimensions (or degrees of freedom) of state space.

In the MCMC method, one has to care about autocorrelation between samples. The autocorrelation function [1,2] of an estimator \hat{O} is defined by

$$A_{\hat{O}}(t) = \frac{\langle O_{i+t} O_i \rangle - \langle \hat{O} \rangle^2}{\langle \hat{O}^2 \rangle - \langle \hat{O} \rangle^2}, \quad (1)$$

where O_s is the sample of a physical quantity O , such as the total energy, at the s th Monte Carlo step. The Monte Carlo average is denoted by the bracket $\langle \cdot \rangle$. The autocorrelation function eventually becomes (almost) independent of i in Eq. (1) after the distribution convergence, namely, the thermalization (the burn-in). In many cases, the function decays exponentially for large t : $A_{\hat{O}}(t) \sim e^{-t/\tau_{\text{exp},\hat{O}}}$,

where

$$\tau_{\text{exp},\hat{O}} = \limsup_{t \rightarrow \infty} \frac{t}{-\ln |A_{\hat{O}}(t)|} \quad (2)$$

is the exponential autocorrelation time of \hat{O} . Autocorrelation reduces the effective number of independent Monte Carlo samples to $M_{\text{eff}} \approx M/2\tau_{\text{int},\hat{O}}$, where M is the number of samples obtained in a simulation and

$$\tau_{\text{int},\hat{O}} = \frac{1}{2} + \sum_{t=1}^{\infty} A_{\hat{O}}(t) \quad (3)$$

is the integrated autocorrelation time of \hat{O} . The constant $\frac{1}{2}$ comes from the discrete nature of the Monte Carlo time evolution. The needed computation time for a certain precision is proportional to these autocorrelation times, $\tau_{\text{exp},\hat{O}}$ and $\tau_{\text{int},\hat{O}}$. They may depend on estimators and update methods.

The MCMC method can be applied to many kinds of phase transitions in principle, but the convergence (relaxation) rate and the sampling efficiency can become very poor in some cases, such as critical slowing down [3,4]. As the system approaches a critical point, the exponential autocorrelation time diverges: $\tau_{\text{exp}} \propto \xi^z \propto |t|^{-\nu z}$, where ξ is the correlation length, t is the temperature difference from a critical point, and ν is the critical exponent of the correlation length. The exponent z is called the dynamic critical exponent, which is

given by

$$z = \lim_{L \rightarrow \infty} \frac{\ln(\max_{\hat{\rho}} \tau_{\text{exp},\hat{\rho}})}{\ln L} \quad (4)$$

at the critical point, where L is the system length. Note that most estimators share the maximum exponential autocorrelation time. Thus, $\tau_{\text{exp},\hat{\rho}} \propto L^z$ asymptotically at the critical point. The exponent of the integrated autocorrelation time may differ from z , but they are identical in many cases. For example, in the case of the square lattice Ising model, the Metropolis algorithm for the single spin update suffers from the rapid growth of the autocorrelation times: $\tau_{\text{exp}} \sim \tau_{\text{int}} \propto L^z$ with $z \approx 2.17$ [5–8]. The dynamic critical exponent is expected to be universal among many MCMC updates [6]. Such a large dynamic critical exponent hampers efficient sampling near a phase transition: the spectral gap of a Markov chain $\Delta \approx \tau_{\text{exp}}^{-1} \propto \xi^{-z}$ is reduced to zero at the critical point. It is thus crucial to devise a smart update method that alleviates or avoids slowing down.

In the case of unfrustrated models, the cluster algorithms, such as the Swendsen-Wang [9] and the Wolff [10] algorithms, reduce the dynamic critical exponent significantly [8,11,12]: for example, $z \approx 0.3$ for the Ising model in two dimensions. The Wolff algorithm is known to be more efficient than the Swendsen-Wang algorithm in $d \geq 3$ dimensions. The size of a cluster corresponds to the correlation length, and the flip of clusters, which can be performed with probability 1, achieves an efficient nonlocal spin update. Forming such an efficient cluster, however, is nontrivial or impractical in general cases. The application of the cluster updates is thus limited to specific models.

In the meantime, the worm algorithm has been one of the most versatile techniques in the world line quantum Monte Carlo method [13,14]. In quantum cases, a naive local update of world lines is often not allowed: for example, a local spin flip breaking up the world line is not allowed in the XXZ quantum spin model because the total magnetization is conserved by the Hamiltonian. The worm algorithm works especially well for cases in which the allowed configurations are restricted by such constraints.

The main idea of the worm algorithm is to achieve an eventual nonlocal update resulting from sequential local updates in extended state space. In practice, the extended space is composed of configurations that contain kinks, which break the constraint. We insert a pair of kinks, which is called the worm, and move one of them, which is called the worm head.

The whole procedure of the worm algorithm is described by the repetition of the following processes.

- (i) A pair of kinks is inserted at a randomly chosen position of the system.
- (ii) One of the kinks moves in a stochastic way, updating the configuration.
- (iii) When meeting each other, the kinks are removed.

The worm algorithm for classical systems [15] was proposed as well, which we call the classical algorithm hereafter. During the process (ii) mentioned above, the position of the worm (the kinks) randomly shifts from site to site of a lattice. The algorithm aims at a random walk of the kink at sites (vertices). The next site is chosen at random among the nearest sites. The worm shifting process is then accepted or rejected

using the Metropolis algorithm. The detailed balance holds in every shifting process. Although each worm move is local in the extended space, a nonlocal update in the original space is eventually achieved after the whole worm update (from insertion to removal). Despite its local nature, the worm algorithm significantly reduces the dynamic critical exponents for several classical models [16,17]. We review the details of the classical algorithm for the Ising model in Sec. II.

It is critical to optimize the stochastic worm update for efficient computation. How can we improve the worm algorithm? The stochastic worm move can be viewed as a diffusion process of the kink in the real space. Thus, higher diffusivity of the kink is expected to yield higher sampling efficiency. In particular, the worm backscattering process, which cancels the previous update, should be averted for efficient sampling.

The directed loop (or the directed worm) algorithm was proposed to improve the efficiency of the world line quantum Monte Carlo method [14]. The directed worm has an additional feature, the direction to move in. The update does not hold the detailed balance for each local worm process but does for the whole worm update from insertion to removal. Thanks to the directed path, backscattering is successfully suppressed.

In the meantime, the geometric allocation approach was proposed to optimize the transition probability in a flexible manner [18]. It is a versatile technique for the MCMC method. The basic concept of this approach is that *the flows* between the states are purposely allocated in a geometric fashion. One can easily find a set of probabilities that holds the global (total) balance even without detailed balance and minimize the rejection probability. The efficiency of the directed worm update in the world line quantum Monte Carlo method is significantly improved by the geometric allocation approach [18,19].

The purpose of the present paper is to enhance the diffusivity of the kink in the worm algorithm. We propose a directed worm algorithm accelerated by the geometric allocation approach. The key ideas of our approach are the following.

- (1) The kinks are located on *bonds* (edges) instead of sites of a lattice.
- (2) The worm move is directed.
- (3) The worm backscattering probability is minimized, and the forward scattering probability is maximized using the geometric allocation.

We confirm enhanced diffusivity by calculating the probability distribution of the relative position of the two kinks. The present algorithm is detailed in Sec. III.

We also discuss how to compare MCMC samplers in Sec. IV. We stress that the sampling efficiency of the MCMC method should be quantified by the asymptotic variance, the prefactor of the asymptotic scaling of the statistical error squared.

We demonstrate, in Sec. V, that the present worm algorithm for the Ising model significantly improves computational efficiency. We show that the efficiency of the present worm update is approximately 25 times as high as that of the classical worm update for the simple cubic lattice Ising model at the critical temperature. There is no extra computational cost in the present algorithm, as compared to the classical algorithm. Our algorithm is even more efficient than the Wolff cluster algorithm, which is one of the best update methods for the Ising model. We estimate the dynamic critical exponent

of the simple cubic lattice Ising model to be $z \approx 0.27$ in the worm update.

Our approach is applicable to many physical models, such as the $|\phi|^4$ model [15], the Potts model [20], the $O(n)$ loop model [17,21,22], and lattice QCD [23], and expected to improve the efficiency of the MCMC update for these models as well as for the Ising model.

The present paper is summarized with discussions in Sec. VI.

II. CLASSICAL ALGORITHM

We review the conventional (classical) worm algorithm [15] for the Ising model in this section. Let the model be represented by $-H/T = K \sum_{\langle ij \rangle} \sigma_i \sigma_j$, where H is the Hamiltonian, T is the temperature, and $\sigma_i = \pm 1$ is the Ising spin variable at each site (vertex) i of a lattice (graph). The partition function of the canonical ensemble can be represented by

$$\begin{aligned} Z &= \sum_{\sigma_i = \pm 1} e^{K \sum_{\langle ij \rangle} \sigma_i \sigma_j} = \sum_{\sigma_i = \pm 1} \prod_{b = \langle ij \rangle} e^{K \sigma_i \sigma_j} \\ &= \sum_{\sigma_i = \pm 1} \prod_{b = \langle ij \rangle} \cosh(K) \sum_{n_b = 0, 1} [\sigma_i \sigma_j \tanh K]^{n_b} \\ &= 2^N [\cosh K]^{N_b^{\text{tot}}} \sum_{\{n_b\}}^{\text{loops}} [\tanh K]^\ell, \end{aligned} \quad (5)$$

where the bond variable on bond b is denoted by n_b , the identity $e^{K \sigma_i \sigma_j} = \cosh(K) \sum_{n_b = 0, 1} [\sigma_i \sigma_j \tanh K]^{n_b}$ is used in the second line, and N and N_b^{tot} are the total number of sites and bonds of a lattice, respectively. In the last line, the sum runs over all the bond configurations that only have closed loops formed by the activated bonds ($n_b = 1$). The configurations that have open strings of activated bonds do not contribute to the partition function. The total length of the closed loops is denoted by $\ell \equiv \sum_b n_b$. The bond variables are sampled by means of the MCMC method under the constraint of the loop structure: the number of activated bonds meeting at each site is even. Any set of bond variables can be used as the initial state in the simulation as long as the loop constraint is satisfied. As the initial state, we chose the vacuum state, in which the bond variables are all deactivated ($n_b = 0 \forall b$).

The worm algorithm is an efficient update method for sampling under such a constraint or a conservation law. The fundamental idea is to extend the state space and allow configurations containing kinks, which break the constraint. Let us consider inserting two kinks and move one of them in a stochastic way. The moving kink is called the worm head, and the other is the worm tail. The classical worm algorithm [15] is described as follows.

(1) Choose a site i_0 at random as the starting point and set $i \leftarrow i_0$. Insert the worm head and tail at i_0 . Go to step 2.

(2) Choose a site j at random among the nearest neighbor sites of site i and shift the worm head from i to j with probability $p = [\tanh K]^{1-n_b}$, where $n_b (= 0 \text{ or } 1)$ is the bond variable on $b = \langle ij \rangle$ before the shift. If the shift is accepted, update n_b ($0 \leftrightarrow 1$) and set $i \leftarrow j$. If $j = i_0$, go to step 3. Otherwise, repeat step 2.

(3) Measure observables. Go to step 1 after removing the worm with probability p_{move} , or go to step 2 with probability $1 - p_{\text{move}}$.

The probability p_{move} can be set to an arbitrary value in $(0, 1]$: $p_{\text{move}} = 1/2$ in Ref. [15].

In the measurement, the total energy can be measured by the total number of activated bonds:

$$\begin{aligned} E &= -\frac{\partial \ln Z}{\partial \beta} \\ &= -N_b^{\text{total}} \tanh K - \left(\frac{1}{\tanh K} - \tanh K \right) \langle \ell \rangle, \end{aligned} \quad (6)$$

where $\beta = 1/T$ is the inverse temperature. The spin correlation function, $G_{ij} \equiv \text{tr}[\sigma_i \sigma_j e^{-\beta H}] / Z$, can be estimated by $\langle N_{ij} \rangle / \langle N_j \rangle$, where N_{ij} is how many times the head is at site i and the tail is simultaneously at site j in step 2, and N_j is how many times the head and tail are both at site j in step 3. The magnetic susceptibility, $\chi \equiv \frac{\beta}{N} \sum_{ij} G_{ij}$, can be estimated by

$$\chi = \beta \langle \ell_{\text{worm}} \rangle, \quad (7)$$

where ℓ_{worm} is the worm length, the total number of worm shifting processes in step 2 including the rejection process. It is straightforward to calculate the Fourier transformed correlation function: one only needs to take into account a phase factor depending on the kink position. The correlation length can be calculated using the Fourier transformed correlation functions and the moment method [24].

The worm algorithm significantly reduces the dynamic critical exponents of several models [16]. It has been applied to many fundamental physical systems, such as the Potts model [20], the $|\phi|^4$ model [15], the $O(n)$ loop model [17,21,22], and lattice QCD [23].

One can use the worm algorithm for dual variables on a dual lattice [15,25–27]. The dual worm algorithm samples domain walls of the original spin variables; in other words, it samples “unsatisfied” bonds that increase the total energy. While the classical worm algorithm is formulated in the high temperature expansion, the dual worm algorithm is in the low temperature expansion for the dual inverse temperature: $\beta' = -\frac{1}{2} \ln \tanh \beta$ [28,29]. One of the advantages of the dual worm algorithm is that it is applicable also to frustrated cases, in which the original worm algorithm suffers from the negative sign problem [26,27].

In two dimensions, the dual variables are under an additional constraint: the winding number of the loops formed by the unsatisfied bonds is even (odd) for periodic (antiperiodic) boundaries. If this constraint is ignored in simulation, nevertheless, the domain wall free energy, the difference between the free energy of the system with periodic boundaries and the free energy of the system with antiperiodic boundaries, can be estimated from the winding number histogram [25]: $e^{-\beta(F_{\text{AP}} - F_{\text{P}})} = Z_{\text{AP}} / Z_{\text{P}} = \langle N_{\text{AP}} \rangle / \langle N_{\text{P}} \rangle$, where F_{AP} is the free energy of the system with periodic (antiperiodic) boundaries in one (the other) direction, F_{P} is the free energy of the system with periodic boundaries in both directions, Z_{AP} and Z_{P} are the associated partition functions, N_{AP} is how many times the winding number is even (odd) in one (the other) direction, and N_{P} is how many times the winding numbers in the two

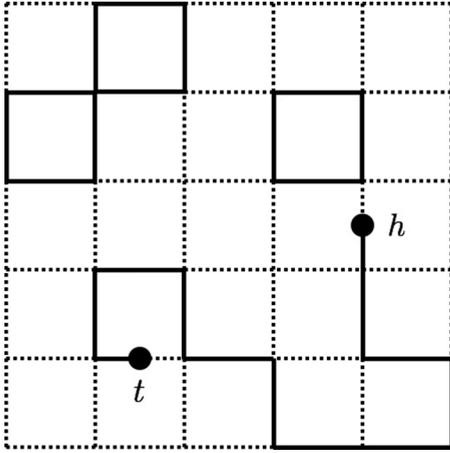


FIG. 1. Example of a configuration containing the present worm in the square lattice Ising model. The solid lines show the activated bonds, and the broken lines show the deactivated bonds. The solid circles indicate the worm head (h) and the worm tail (t), both of which break the loop constraint of the activated bonds.

directions are both even, respectively. Because the square lattice is self-dual, the dual worm update at the critical temperature is identical to the original worm update, except for the winding number constraint. The domain wall free energy is, therefore, accessible in both formalisms.

III. PRESENT APPROACH

We present a modified worm update in this section. The worm backscattering (rejection) probability is minimized, and the forward scattering probability is maximized using the geometric allocation approach. Our algorithm is indeed free from rejection at the critical temperatures of the square lattice and the cubic lattice Ising models. As a result, the diffusivity of the worm head is enhanced, which improves computational efficiency. We show the ergodicity of the Markov chain created by the present method and describe how to measure relevant physical quantities, such as the magnetic susceptibility. We also discuss a possible bias and how to avoid it in the worm algorithm.

A. Worm on bonds

We adopt the same representation of the partition function [Eq. (5)] with the classical worm algorithm. Our goal is to sample bond variables $\{n_b\}$ efficiently under the loop constraint. We here consider inserting the worm, namely, a pair of kinks, on a bond, or an edge, of a lattice. Our worm is distinct in this respect from the classical worm, which is always located at sites. We then move the worm head, that is, one of the pair, from one bond to another in a stochastic way: when coming to a site, the worm head scatters to another (or possibly the same) bond with a certain probability. This scattering process continues until the head comes back to the tail, that is, the other of the pair.

A typical configuration containing the present worm in the $L = 6$ square lattice Ising model with open boundaries is illustrated in Fig. 1. In our algorithm, each kink is located at

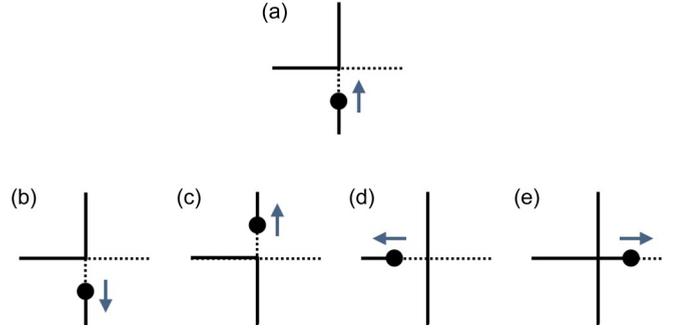


FIG. 2. Example of the worm scattering process for the square lattice case. The solid circle in each graph shows the worm head, and the arrow shows the moving direction of the head. When coming to a vertex (state a), the worm head scatters to another (or possibly the same) bond (states b , c , d , and e) with a certain probability.

the center of a bond; the bond variables can take $n_b = 0, \frac{1}{2},$ or 1 . The head has a moving direction in a fashion similar to the directed loop algorithm [14].

Suppose the moving direction of the head is upward in Fig. 1. Then the head scatters at the next site (vertex) and moves to a bond connecting to the site stochastically, which we call the worm scattering process. The four possible states after the scattering are shown as b , c , d , and e in Fig. 2. The next state is chosen between the four states with a certain probability. We discuss probability optimization in Sec. III B. After the worm scattering, bond variables are updated, as shown in Fig. 2; the halves of bonds are updated ($n_b = 0, \frac{1}{2}, 1$) as the kink is assumed to be at the center of a bond. We repeat this worm scattering process until the head comes back to the tail position.

The whole procedure of the present algorithm is described as follows.

(1) Choose a bond b_0 at random as the starting point and $b \leftarrow b_0$. Insert the worm head and tail at the center of b_0 . Choose the moving direction at random. Go to step 2.

(2) Choose the next bond c with the probability optimized using the geometric allocation. If $b \neq c$, update the bond variables n_b and n_c , and set $b \leftarrow c$. If $b = b_0$, go to step 3. Otherwise, repeat step 2.

(3) Measure observables and go to step 1 after removing the kinks (worm).

As compared with the classical worm algorithm, the probability p_{move} at step 3 is fixed to 1 in our algorithm. Note that the present worm carries extra weight (a factor of $1/2$) such that the insertion and the removal are accepted with probability 1, as discussed in Sec. III E.

One of the advantages of our approach is that it is straightforward to optimize the worm scattering probability. In general, the transition probability is set under global balance in the MCMC method. If one did not resort to the Metropolis algorithm in the classical worm update, the worm shifting probability at a site would depend on other shifting processes at the nearest neighbor sites. The shifting probabilities at the nearest neighbor sites would further depend on the processes at the next nearest neighbor sites. Thus, it is nontrivial to write down the global balance condition in a closed form.

The Metropolis scheme reduces the condition to a local form, but no room for optimization is left, except for increasing the number of possible states. In contrast, the balance condition of the worm scattering process in our approach is expressed in a closed form without using the Metropolis algorithm, as shown in the next subsection. This simple structure of the balance condition leaves much room for optimization.

B. Geometric allocation approach

We here detail the optimization of the transition probability in the present worm algorithm. In the MCMC method, it is crucial to optimize the transition probability for practical and efficient sampling. The problem we tackle here is how to prepare a set of appropriate transition probabilities between given states. The geometric allocation [18,19] is a versatile approach to optimizing the transition probability. The fundamental concept of this approach is that *the flows* between the states are purposely allocated using a geometric graph. This geometric manner is very distinct from the conventional approaches, such as the Metropolis and the heat bath algorithms. They provide algebraic solutions that satisfy the detailed balance condition, which is a sufficient condition for the global balance. In contrast, the allocation approach converts the optimization problem into a geometric puzzle and provides a graphic solution. Although the geometric allocation was originally introduced to break the detailed balance in Ref. [18], one of the main advantages of this approach is that we can easily arrange the transition probability in a flexible manner. We stress that the geometric allocation is not merely a representation of solutions but a versatile and efficient way to find optimal solutions.

Let us describe the rule of the puzzle game. Let $v_{ij} := \pi_i p_{i \rightarrow j}$ be the raw flow from state i to j , where π_i is the weight, or the measure, of state i apart from the normalization factor of a target distribution, and $p_{i \rightarrow j}$ is the transition probability from i to j . Given possible states and their weights $\{\pi_i\}$, we allocate v_{ij} under the two conditions: the law of probability conservation and the global balance condition, which are expressed by

$$\pi_i = \sum_{j=1}^n v_{ij} \quad \forall i \quad (8)$$

and

$$\pi_j = \sum_{i=1}^n v_{ij} \quad \forall j, \quad (9)$$

respectively, where n is the number of possible states. In the worm scattering process for the square lattice Ising model, there are four possible states because a square lattice has a coordination number of 4. For example, in the case of Fig. 2, the possible states are b , c , d , and e , with $n = 4$.

Let us reinterpret the conventional algorithms in this picture. It is easy to understand that the flows allocated by the Metropolis and the heat bath algorithms are represented by

$$v_{ij} = \frac{1}{n-1} \min(\pi_i, \pi_j) \quad i \neq j \quad (10)$$

and

$$v_{ij} = \frac{\pi_i \pi_j}{\sum_{k=1}^n \pi_k} \quad \forall i, j, \quad (11)$$

respectively. Both algorithms satisfy the detailed balance condition, which is expressed by the symmetry of the flow: $v_{ij} = v_{ji}$.

Let us set a cost function in this optimization problem. The cost function we first consider is the average rejection (worm backscattering) probability, which is given by $\sum_i v_{ii} / \sum_j \pi_j$. We optimize the flow for the average rejection probability to be minimized. This choice should be desirable because, in general, the rejection reduces the sampling efficiency of the MCMC method. Rejection minimization has also been discussed in the previous applications [18,19].

To increase further the diffusivity of the worm head, we maximize the forward scattering probability under the condition of backscattering minimization. Our choice of the local transition probability is expected to reduce the variance of the worm length, namely, the variance of the first return time for the head to come back to the tail position.

We found the optimal solution through the geometric allocation shown in Fig. 3. Any local configuration in the worm scattering can be mapped into the case of Figs. 3(a) or 3(b) through a possible rotation and flip. It is easy to confirm that Eqs. (8) and (9) are both satisfied: the area of each weight (color) is conserved, which is nothing but the probability conservation; the entire *box* shape is intact after the allocation, which guarantees the global balance. We also obtained the analytical form of the flow v_{ij} corresponding to the optimal allocation, which is shown in the caption.

The rejection-free condition [18] is, in general, given by

$$\pi_1 \leq \sum_{i=2}^n \pi_i. \quad (12)$$

This condition is equivalent to $\tanh K \geq 1/3 \iff T \leq 2/\ln 2$ in the case of Fig. 3(a) and always satisfied in the case of Fig. 3(b). Here, the ratio $\pi_4/\pi_1 (= \tanh K)$ depends on the temperature in the simulation. Our update is rejection free for $T \leq 2/\ln 2 \simeq 2.885$, including the critical temperature $T_c = 2/\ln(1 + \sqrt{2}) \simeq 2.269$ [28]. In addition, the forward scattering probability $(v_{12} + v_{21} + v_{34} + v_{43}) / \sum_j \pi_j$ is maximized under rejection minimization in both cases.

We chose the unique solution satisfying the detailed balance condition under backscattering minimization and forward scattering maximization. Technically, directed worm scattering always breaks the detailed balance in the extended state space. Nevertheless, if local worm scattering satisfies the detailed balance condition without taking the direction into account, the whole worm update from insertion to removal ensures the detailed balance in the original state space [14]. It is easy to find many (actually infinite) solutions to satisfy the required conditions [Eqs. (8) and (9)] thanks to the geometric picture. Even solutions breaking detailed balance can be readily found [18]. For example, starting from the solution shown in Fig. 3(a), we can increase a certain amount of v_{13} , v_{34} , and v_{41} , while decreasing the same amount of v_{31} , v_{43} , and v_{14} . This modified solution again satisfies Eqs. (8) and (9) without the detailed balance because $v_{ij} \neq v_{ji}$

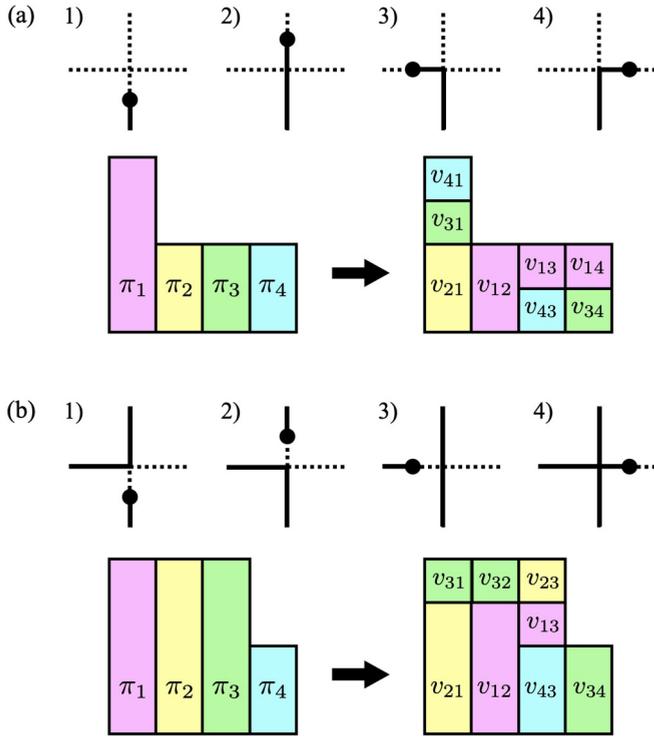


FIG. 3. Geometric allocation for the square lattice Ising model. There are two cases: (a) and (b). The solid (broken) lines show the activated (deactivated) bonds, and the solid circles show the worm head. The weight, or the measure, of each state is denoted by π_i ($i = 1, 2, 3, 4$) apart from the normalization factor of the target distribution, and the allocated raw flow from i to j is denoted by v_{ij} . The detailed balance condition is satisfied in both cases: $v_{ij} = v_{ji}$. In the case of (a), we set $v_{12} = \pi_4$, $v_{13} = v_{14} = \frac{1}{2}(\pi_1 - \pi_4)$, and $v_{34} = \frac{1}{2}(3\pi_4 - \pi_1)$. This rejection-free allocation can be performed if $3\pi_4 > \pi_1 \Leftrightarrow T < 2/\ln 2$. In the case of (b), we set $v_{12} = \frac{1}{2}(\pi_1 + \pi_4)$, $v_{13} = v_{23} = \frac{1}{2}(\pi_1 - \pi_4)$, and $v_{34} = \pi_4$, which is possible at any temperature. The scale of the area π_i is arbitrary; only the ratio π_4/π_1 matters.

for $(i, j) = (1, 3), (1, 4), (3, 4)$. The modified solution, as well as the original solution [Fig. 3(a)], has the minimized (zero) backscattering rate $(= v_{11} + v_{22} + v_{33} + v_{44})/\sum_j \pi_j$ and the maximized forward scattering rate $(= v_{12} + v_{21} + v_{34} + v_{43})/\sum_j \pi_j$. If local flows break the detailed balance condition ($v_{ij} \neq v_{ji}$), the whole worm update breaks the detailed balance in the original state space as well, which is called irreversible. Although it is possible to improve the efficiency by breaking detailed balance, we have not yet found any irreversible solution that works significantly better than the present choice in the case of the Ising model. We selected the present reversible solution because it is unique and easy to prove the ergodicity (discussed below in Sec. III C). Irreversible Markov chains, nevertheless, have the potential to play an essential role in Monte Carlo dynamics.

We can calculate all the transition probabilities, $p_{i \rightarrow j} = v_{ij}/\pi_i \forall i, j$, before simulation and prepare a look-up table storing the probabilities. In the actual simulations, we choose the next state in each worm scattering by using Walker's method of alias [30,31]. The advantage of Walker's method is that the computation time, which is $O(1)$, does not increase

with the number of possible states n in contrast to the computation time of a simple binary search, which is $O(\log n)$. The present algorithm needs no extra computational cost, compared to the classical worm algorithm.

In the simple cubic lattice case, we chose a set of flows, as illustrated in Fig. 4. The six possible states are indexed such that (1, 2), (3, 4), and (5, 6) are the pairs of the states from and to which the worm head forward scatters like the square lattice case. The allocation patterns depending on the local configuration are all shown in Fig. 4. We express the analytical form of v_{ij} as well in the caption. The rejection-free condition [Eq. (12)] is satisfied for $T \leq 2/\ln(3/2) \simeq 4.933$, including the critical temperature $T_c \approx 4.511$ [32]. In addition to the conditions of backscattering minimization and forward scattering maximization, we here impose an additional condition to find the unique solution; the variance of the forward scattering flow, $\sum_{k=1,3,5} (v_{kk+1} - \bar{v})^2$, where $\bar{v} = \frac{1}{3} \sum_{k=1,3,5} v_{kk+1}$, is minimized. Other solutions, nevertheless, are expected to work as well as our choice does as long as the backscattering probability is minimized, and the forward scattering probability is maximized.

Our geometric allocation approach to optimizing the worm scattering probability can be generalized to many physical models, such as the $|\phi|^4$ model, the Potts model, the $O(n)$ loop model, and lattice QCD. It is expected to improve the computational efficiency of the worm update for these models as well as for the Ising model.

C. Ergodicity

We here show that the Markov chain created by the present worm algorithm is (uniformly) ergodic in the extended state space; equivalently, it is irreducible and aperiodic [33]. Any configuration under the loop constraint is represented by a combination of loops formed by activated bonds. The worm can create any loop with a finite probability. (Note that the forward scattering probability is always positive in the present flow allocation.) Hence any state with and without kinks can be visited from the vacuum state, in which $n_b = 0 \forall b$. Because our solution of the transition probability holds the detailed balance, any two states in the extended state space are connected by the transition kernel; the Markov chain is irreducible.

Let us next consider the aperiodicity. Even if it never backscatters, the worm can come back to the same physical state (with no kink). There are many paths for the kink to start from and end at the vacuum state. For example, the paths formed by 9 and 11 worm scattering steps (going around a plaquette twice) exist for both the square and cubic lattice models. Let p and q be 9 and 11, respectively. Because p and q are coprime, Bézout's identity states $\exists a, b \in \mathbb{Z}$ such that $ap + bq = \text{greatest common divisor}(p, q) = 1$. We can choose a and b such that $-q < a < 0$ and $0 < b < p$.

We prove $\forall n \geq pq - 1, \exists c, d \in \mathbb{N}$ such that $n = cp + dq$. First, we can express $pq - 1 = -ap + (p - b)q$, where $-a > 0$ and $p - b > 0$. We here use mathematical induction: if $\exists m \in \mathbb{N}$ such that $m \geq pq - 1$ and $\exists \alpha, \beta \in \mathbb{N}$ such that $m = \alpha p + \beta q$, then $\exists \alpha', \beta' \in \mathbb{N}$ such that $m + 1 = \alpha' p + \beta' q$. We can express $m + 1 = (\alpha + a)p + (\beta + b)q = (\alpha + a + q)p + (\beta + b - p)q$. If $\alpha + a \geq 0$, simply $\alpha' = \alpha + q$ and $\beta' = \beta + b$. If $\alpha + a < 0$, we can take $\alpha' = \alpha + a + q \geq 0$

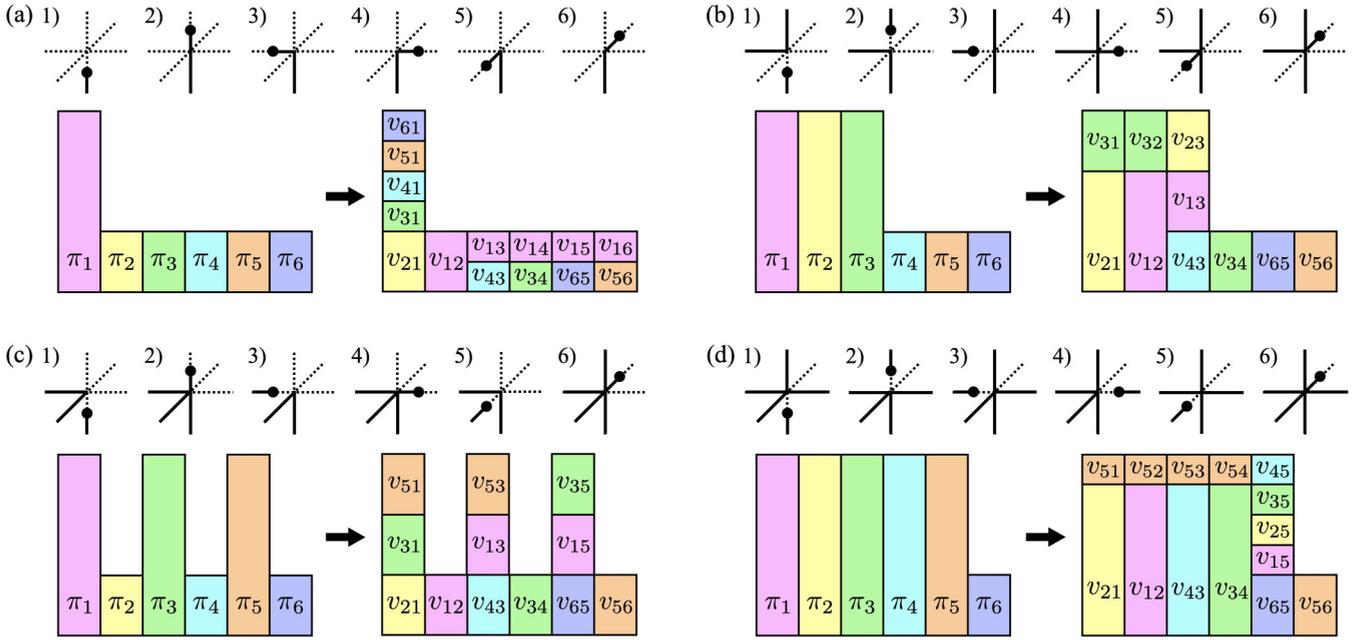


FIG. 4. Geometric allocation for the simple cubic lattice Ising model. There are four cases: (a), (b), (c), and (d). The six possible states are indexed such that (1, 2), (3, 4), and (5, 6) are the pairs of the states from and to which the worm head forward scatters like the square lattice case. The detailed balance condition is satisfied in all the cases: $v_{ij} = v_{ji}$. In the case of (a), we set $v_{12} = \pi_6$, $v_{13} = v_{14} = v_{15} = v_{16} = \frac{1}{4}(\pi_1 - \pi_6)$, and $v_{34} = v_{56} = \frac{1}{4}(5\pi_6 - \pi_1)$; in (b), $v_{12} = \frac{1}{2}(\pi_1 + \pi_6)$, $v_{13} = v_{23} = \frac{1}{2}(\pi_1 - \pi_6)$, and $v_{34} = v_{56} = \pi_6$; in (c), $v_{12} = v_{34} = v_{56} = \pi_6$ and $v_{13} = v_{15} = v_{35} = \frac{1}{2}(\pi_1 - \pi_6)$; in (d), $v_{12} = v_{34} = \frac{1}{4}(3\pi_1 + \pi_6)$, $v_{15} = v_{25} = v_{35} = v_{45} = \frac{1}{4}(\pi_1 - \pi_6)$, and $v_{56} = \pi_6$. This rejection-free allocation in (a) can be performed if $5\pi_6 > \pi_1 \Leftrightarrow T < 2/\ln(3/2)$. The allocations in (b), (c), and (d) are possible at any temperature.

and $\beta' = \beta + b - p \geq 0$. The last inequality follows from $\alpha + a < 0 \Rightarrow \alpha p + ap < 0 \Rightarrow m + 1 - (\beta + b)q < 0 \Rightarrow m + 1 - (\beta + b - p)q < pq \Rightarrow (\beta + b - p)q > m + 1 - pq \geq 0 \Rightarrow \beta + b - p \geq 0$.

Therefore, the vacuum state can be revisited from itself with a finite probability after $n \geq pq - 1$ worm scattering steps: the vacuum state is aperiodic in the extended space. Hence, the Markov chain is aperiodic.

We can choose an irreversible solution instead of the present reversible one, as mentioned in Sec. III B. It is not trivial to prove the ergodicity of irreversible Markov chains. Nevertheless, we have tested a couple of irreversible solutions and confirmed that the results are consistent. We thus expect many solutions to generate ergodic Markov chains even without detailed balance.

D. Enhanced diffusivity

We demonstrate here that the present worm algorithm indeed enhances the diffusivity of the worm head. Figure 5 shows the probability distribution of the difference between the coordinates of the two kinks (the worm head and tail) in the classical and the present worm updates for the $L = 128$ square lattice Ising model at the critical temperature. The coordinates were measured 256 worm shifting or scattering steps after the insertion. The distribution in the present worm update is much broader than in the classical worm update. The removed worms at $(x, y) = (0, 0)$ before 256 worm steps are not shown in Fig. 5 but counted in the normalization.

The distribution tail of the kink distance is well approximated by a Gaussian distribution, as shown in Fig. 6. We

estimated the variances of the Gaussian distributions 64, 128, and 256 local worm steps after the insertion. In Fig. 6, although some faster decay is observed in the distribution after 64 scattering steps of the present worm update, the tails of the distributions after 128 and 256 steps are well fitted to Gaussian distributions up to longer distances. We found a linear growth of the variance as a function of the number of local worm steps, as shown in the inset. The variance in the present worm update is six times as large as in the classical worm update. These observations indicate that the present method successfully enhances the diffusivity of the worm head, which is expected to improve sampling efficiency.

E. Estimators

Many physical quantities can be measured in the present worm simulation as well as in the classical worm simulation. For example, the total energy can be estimated by the same observable [Eq. (6)]. Nevertheless, estimators associated with the extended state space need to be slightly modified. As relevant quantities, we here explain how to measure the spin correlation function and the magnetic susceptibility.

In the classical worm algorithm, the state space is extended to include the configurations that contain (up to) two kinks at sites of a lattice. How many times the two kinks are at sites i and j directly contributes to the estimator of the spin correlation between sites i and j , as mentioned in Sec. II. On the other hand, the present worm is never located at the sites during the update processes.

Let us here consider a *virtual* process of shifting the two kinks from bonds to adjacent sites of the bonds. There are four

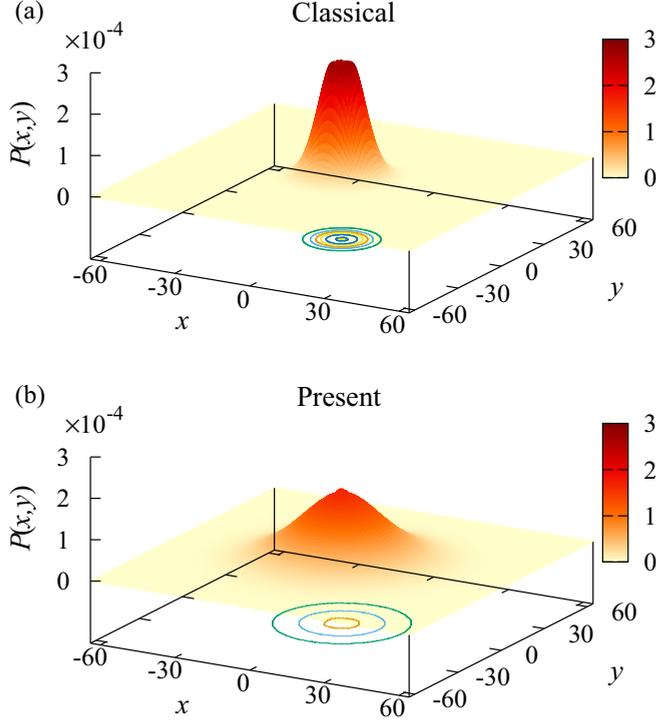


FIG. 5. Probability distributions of the difference between the coordinates of the two kinks (head and tail) for the $L = 128$ square lattice model at the critical temperature in (a) the classical and (b) the present worm updates. The coordinates were measured 256 worm shifting or scattering steps after the insertion. The contours show the coordinates at which $P(x, y) = 0.5, 1.0, 1.5, 2.0,$ and 2.5×10^{-4} . Because the two kinks are removed when meeting each other, the distribution is lowered near the center, which is more significant in the classical worm update. The removed worms are not shown here but counted in the normalization.

choices of sites because each bond connects two sites. We choose a pair of sites at random, one from the two adjacent sites of the head and the other from the two adjacent sites of the tail. We then consider using the Metropolis algorithm to accept or reject the virtual shift. The acceptance probability depends on the change of $\ell = \sum_b n_b$. If this virtual shift were accepted, we would count one for measuring the associated spin correlation in a manner similar to the classical worm algorithm. We next consider a reverse process of shifting the kinks from the sites to the original bonds, using the Metropolis algorithm again. If this reverse shift were rejected, we would count one for measuring the spin correlation again. We would repeat the reverse shifting process and continue counting one while the kinks would be on sites. The average count through these virtual processes is given by the ratio of the weight of a site-kink configuration to the weight of a bond-kink configuration. Thus, we can use the weight ratio as the reweighting factor from a bond-kink configuration to a site-kink configuration. Since we assume each kink to be at the center of a bond in the present algorithm, the reweighting factor is given by a simple form. To calculate the magnetic susceptibility, we take the average over the four possible choices and sum up the reweighting factors during the worm

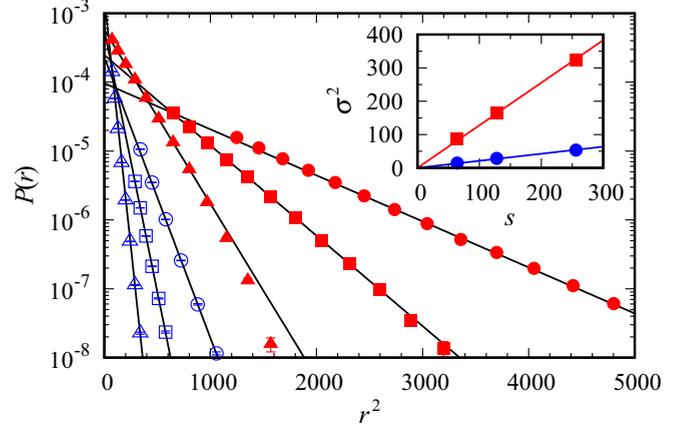


FIG. 6. Tails of the probability distributions of the distance between the two kinks in the classical (open) and the present (solid) worm updates 64 (triangles), 128 (squares), and 256 (circles) local worm steps after the insertion, measured in the $L = 128$ square lattice Ising model at the critical temperature. The distribution was measured at $r = |\mathbf{r}|$, where $\mathbf{r} = (x, y)$ and $|x| = |y|$. The tails are fitted to Gaussian distributions: $P(r) \propto e^{-r^2/2\sigma^2}$, where σ^2 is a parameter (variance). The inset shows the linear scaling of the estimated variance in the classical (circles) and the present (squares) worm updates as a function of the number of worm steps (s). The variance in the present worm update is approximately six times as large as in the classical worm update.

scattering process from insertion to removal without going through the virtual processes.

From the above argument, a magnetic susceptibility estimator in the present algorithm is given by

$$\hat{\chi} = \frac{\beta}{4zw} \sum_{\text{path}} f_{\text{rew}}, \quad (13)$$

where β is the inverse temperature, z is the coordination number (four for a square lattice and six for a cubic lattice), w is the extra weight the worm carries,

$$f_{\text{rew}} = \left(s + \frac{1}{s} \right) f_h \quad (14)$$

is the reweighting factor after a worm scattering process, and $s \equiv \sqrt{\tanh K}$. In Eq. (14), f_h is $2/s$ if the head is on an activated bond, $2s$ on a deactivated bond, and $s + \frac{1}{s}$ on a half-activated and half-deactivated bond. In other words, f_h takes $2/s$ or $2s$ if the head comes back to the tail. It takes $s + \frac{1}{s}$ otherwise. The summation in Eq. (13) means that f_{rew} is calculated after each worm scattering process and summed over the scattering processes in step 2. The sum of the reweighting factors is divided by four because we take the average of the four reweighting factors depending on the choice of adjacent sites. Furthermore, it is divided by the coordination number because of the multiple counts in the virtual shift from bonds to sites.

Let us consider the extra weight w in Eqs. (13) and (15). In the worm algorithm, we can arbitrarily set the weight of each state in the extended space: we assume that the present worm has extra weight, a factor of $1/2$, to insert and remove the worm with probability 1 in the Ising model. The value of the

extra weight comes from the fact that there are two possible directions for the worm head to go in. We need to take this extra weight into account for estimators related to the extended space. In the susceptibility estimator (13), the reweighting factor needs to be divided by the extra weight ($w = 1/2$ for the Ising model). It is straightforward to calculate other quantities, such as the Fourier transformed correlation function and the correlation length. Note that while being a factor of 1/2 for the Ising model, the extra weight that the worm carries may depend on models and worm variants used in the simulation.

We easily find

$$\chi \sim \frac{\beta}{4zw} \left(s + \frac{1}{s} \right)^2 \langle \ell_{\text{worm}} \rangle, \quad (15)$$

where ℓ_{worm} is the worm length. It is because f_h takes $s + \frac{1}{s}$ unless the head and tail are located at the same position. This estimation is comparable to Eq. (7) in the case of the classical worm algorithm.

F. Avoiding bias

Before closing this section on the methodology, we discuss a possible bias introduced by the fixed time simulation of the worm algorithm and the Wolff cluster algorithm [10]. In these methods, the computation time for a Monte Carlo step depends on the worm length or the cluster size (in the Wolff algorithm). The mean worm length, which is approximately proportional to the magnetic structure factor ($= \chi/\beta$) as shown in Eqs. (7) and (15), is usually a decreasing function of temperature and indeed so in the present Ising models. In contrast, the total energy is an increasing function of temperature. When the configuration in the simulation is a higher energy configuration, the computation time for the subsequent Monte Carlo step (from worm insertion to removal) will be shorter on average. In other words, the needed computational time to sample a high energy configuration is shorter on average than to sample a low energy configuration. As a result, given a simulation time, say, 1 hour, high energy configurations tend to be sampled more often than low energy configurations. Therefore, such a fixed time simulation creates a bias. For example, if a parallel simulation is run for a certain period using independent Markov chains, an estimator that naively averages over chains has a bias. To avoid this bias, we need to fix the total Monte Carlo steps for each chain instead of the total run time and take the average over chains that run the same Monte Carlo steps. The bias we discuss here can be caused in the worm algorithm for quantum systems as well. Although the bias might be tiny, we carefully run simulations avoiding it.

IV. HOW TO COMPARE MCMC SAMPLERS

We discuss how to quantify the computational efficiency of the MCMC sampler. There are mainly two points to consider [19]: the relaxation rate and the sampling efficiency. In the former, as Monte Carlo samples are taken after the thermalization, faster relaxation to a target distribution allows for sampling from an earlier Monte Carlo step; in the latter, more efficient sampling yields a smaller statistical error. The mean squared error of an estimator is proportional to the inverse

of the number of samples (Monte Carlo steps) according to the central limit theorem [33]. The sampling efficiency of the MCMC update should be quantified by the prefactor of the scaling, that is, the asymptotic variance [19]. We explain here how to measure relevant quantities in the present method.

The relaxation rate is quantified by the exponential autocorrelation time. The autocorrelation function exponentially decays in large Monte Carlo steps, which is the case for the finite size systems we study in the present paper. We calculate the function by running independent simulations and estimate the exponential autocorrelation time as a fitting parameter. In the present paper, we use a single exponential as the fitting function and estimate the error bar of the fitting parameter by using bootstrapping [34,35].

In the worm algorithm, we consider each Monte Carlo step to be a one-time worm update from insertion to removal. In other words, the number of Monte Carlo steps is equal to how many times the head comes back to the tail. Here, the number of Monte Carlo steps should be measured in units of the number of sites for a fair comparison. An autocorrelation time τ'_{exp} estimated by fitting to an exponential function is rescaled:

$$\tau_{\text{exp}} = \tau'_{\text{exp}} \frac{\langle \ell_{\text{worm}} \rangle}{N}, \quad (16)$$

where $\langle \ell_{\text{worm}} \rangle$ is the mean worm length, and N is the number of sites. The mean worm length differs for the classical and the present worm updates as the state space is extended in different manners.

The sampling efficiency of the MCMC method is related to the integrated autocorrelation time. It can be estimated by the relation

$$\tau'_{\text{int}} = \frac{\sigma^2}{2\bar{\sigma}^2}, \quad (17)$$

where σ^2 is the mean squared error, namely, the square of the statistical error, calculated by binning analysis using a much larger bin size than the exponential autocorrelation time, and $\bar{\sigma}^2$ is calculated without binning. The above estimator (17) gives the exact integrated autocorrelation time (3) in the limit of a large number of Monte Carlo steps [1]. In a manner similar to Eq. (16), it is rescaled:

$$\tau_{\text{int}} = \tau'_{\text{int}} \frac{\langle \ell_{\text{worm}} \rangle}{N}. \quad (18)$$

Although the integrated autocorrelation time is useful for studying Monte Carlo dynamics, we stress that the sampling efficiency of the Monte Carlo method should be quantified by the asymptotic variance, which is the prefactor of the asymptotic scaling:

$$\sigma_{\hat{O}}^2 \approx \frac{v_{\text{asyp},\hat{O}}}{M}, \quad (19)$$

where $\sigma_{\hat{O}}^2$ is the mean squared error of an estimator \hat{O} , $v_{\text{asyp},\hat{O}}$ is the asymptotic variance of \hat{O} , and M is the renormalized number of Monte Carlo steps. Here we assume \hat{O} to be an unbiased estimator of a physical quantity O : $\langle \hat{O} \rangle = O$. Then the asymptotic variance is represented by

$$v_{\text{asyp},\hat{O}} = 2\tau_{\text{int},\hat{O}} v_{\hat{O}}, \quad (20)$$

where $v_{\hat{O}} = \langle \hat{O}^2 \rangle - \langle \hat{O} \rangle^2$ is the variance of \hat{O} .

In the present paper, according to Eqs. (17), (18), (19), and (20), we estimate the variances using the jackknife method [36] and the following relations:

$$v_{\text{asympt},\hat{O}} = M' \frac{\sigma_{\hat{O}}^2 \langle \ell_{\text{worm}} \rangle}{\mu_{\hat{O}}^2 N}, \quad (21)$$

$$v_{\hat{O}} = M' \frac{\bar{\sigma}_{\hat{O}}^2}{\mu_{\hat{O}}^2}, \quad (22)$$

where M' is the original number of Monte Carlo steps used for sampling in a simulation, $\sigma_{\hat{O}}^2$ and $\bar{\sigma}_{\hat{O}}^2$ are the mean squared errors of an estimator \hat{O} with and without binning, and $\mu_{\hat{O}}$ is the average of the samples, respectively. We here use the squared coefficient of variation (σ^2/μ^2) to remove a trivial dependence on the definition of the estimator: for example, the variances of the total energy and the energy density are identical.

The renormalization of the number of Monte Carlo steps is necessary also for the Wolff algorithm. We simply replace the mean worm length with the mean cluster size in Eqs. (16), (18), and (21).

V. RESULTS

We investigate the performance of our worm algorithm for the simple cubic lattice Ising model, focusing on critical slowing down at the transition temperature. We compare the present algorithm with the classical worm [15] and the Wolff algorithms [10]. The ensemble used in the simulations is represented by Eq. (5) at the critical temperature, $1/T_c \approx 0.221\,654\,55$ [32]. Periodic boundaries were used in all the spatial directions. We optimize the worm scattering probability, as illustrated in Fig. 4. More than 2^{24} Monte Carlo samples were taken, in total, after 2^{16} thermalization steps.

For a fair comparison, we adopt N local worm processes in the worm algorithms and N spin updates in the Wolff algorithm as the unit of time in the Monte Carlo dynamics. Here N is the number of sites of the system. The autocorrelation times were rescaled as shown in Eqs. (16) and (18). The mean worm length in the classical worm update is proportional to the magnetic susceptibility: $\langle \ell_{\text{classical worm}} \rangle = \chi/\beta \propto L^{\gamma/\nu}$, where γ and ν are the critical exponents of the susceptibility and the correlation length, respectively [1]. We found a relation between the worm lengths in the present and the classical worm updates: $\langle \ell_{\text{present worm}} \rangle \approx 1.765 \langle \ell_{\text{classical worm}} \rangle$ for $L \geq 16$.

The integrated autocorrelation time, the variance, and the asymptotic variance of the energy estimator are shown in Fig. 7. We calculated these quantities in the manner explained in Sec. IV. Using the Wolff algorithm, we calculated the total energy from the spin configuration. The present algorithm produces the shortest integrated autocorrelation time and the smallest asymptotic variance. The shorter correlation time in the present worm update allowed us to run simulations for the larger system size.

Fitting a power law to data, we estimate the exponents of $\tau_{\text{int},\hat{E}}$ to be 0.28, 0.31, and 0.27, and those of $v_{\text{asympt},\hat{E}}$ to be -2.44 , -2.35 , and -2.46 in the Wolff cluster (triangles), the classical worm (circles), and the present worm (squares)

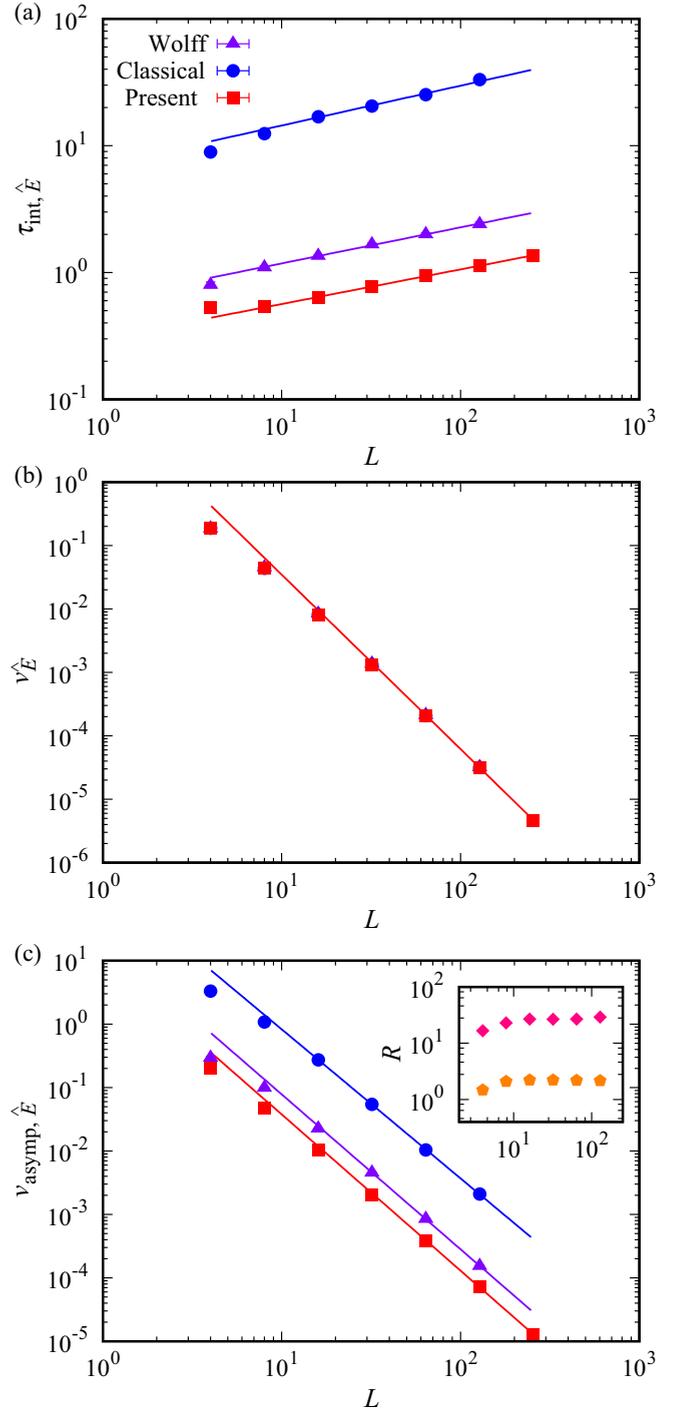


FIG. 7. (a) The integrated autocorrelation time, (b) the variance, and (c) the asymptotic variance of the total energy estimator as a function of the system length of the simple cubic lattice Ising model at the critical temperature. The exponents of $\tau_{\text{int},\hat{E}}$ are estimated to be 0.28, 0.31, and 0.27, and those of $v_{\text{asympt},\hat{E}}$ are to be -2.44 , -2.35 , and -2.46 in the Wolff cluster (triangles), the classical worm (circles), and the present worm (squares) updates, respectively; the exponent of $v_{\hat{E}}$ is estimated to be -2.75 in all the updates. The inset of panel (c) shows the ratios of the asymptotic variance in the classical worm (diamonds) and the Wolff cluster (pentagons) updates to the one in the present worm update. They are approximately 27 and 2.2 for large system sizes, respectively.

updates, respectively; we estimate the exponent of $v_{\hat{E}}$ to be -2.75 in all the updates. We expect the three algorithms to produce the same exponent asymptotically. Nevertheless, as shown in the inset of Fig. 7(c), the asymptotic variance in the present worm update is approximately 27 and 2.2 times as small as in the classical worm and the Wolff cluster updates, respectively.

In the worm algorithm, the weight of the loop configuration is not the Boltzmann distribution, as shown in Eq. (5): $\pi_i \propto (\tanh K)^{\ell_i}$. Thus, estimators for a physical quantity naturally depend on the representation. We can construct an estimator in the worm algorithm the mean value of which is identical to a physical quantity of the original spin system. Nevertheless, the variances of the estimators are generally different. The variance of the energy estimator is the same for the classical and the present worm algorithms simply because the same estimator is used. However, the variance is different from the one in the Wolff algorithm: the estimators are different, although their mean values are identical. Nevertheless, the difference is small, and both the variances show almost the same exponent (≈ -2.75) of the power-law decay, as shown in Fig. 7(b). The variance of the energy estimator in the Wolff algorithm is nothing but the energy variance of the original spin system, which is proportional to the specific heat. The asymptotic scaling, therefore, should be $v_{\hat{E}} \propto L^{\alpha/v-d}$ with the exponent $\alpha/v - d \approx -2.826$ [32]. The present estimate is slightly larger by 2.7%. Data of larger system sizes seem to be needed for a single power-law fit we use to match the exponent estimated from the more sophisticated analysis.

The quantities of the magnetic susceptibility estimators are shown in Fig. 8 like the energy estimator. In the Wolff algorithm, we test two estimators: $\hat{\chi} = \beta M_z^2/N$ (here dubbed Wolff spin), where M_z is the total magnetization of spins, and $\hat{\chi} = \beta \ell_{cl}$ (Wolff cluster), where ℓ_{cl} is the cluster size. We estimate the exponents of $\tau_{\text{int},\hat{\chi}}$ to be 0.150(9), $-0.50(1)$, $-0.731(7)$, and $-0.679(4)$; those of $v_{\hat{\chi}}$ to be 0.01, 0.58, 0.92, and 0.85; and those of $v_{\text{asympt},\hat{\chi}}$ to be 0.18, 0.18, 0.22, and 0.18 in the Wolff spin, in the Wolff cluster, in the classical worm, and in the present worm updates, respectively. The numbers in the parentheses indicate the statistical error, one standard deviation, in the preceding digit. Interestingly, while the exponents of $\tau_{\text{int},\hat{\chi}}$ and $v_{\hat{\chi}}$ are different for each estimator and algorithm, the exponent of $v_{\text{asympt},\hat{\chi}}$ is most likely identical. Particularly, $v_{\text{asympt},\hat{\chi}}$ is almost the same for the two estimators in the Wolff algorithm. Nonetheless, the asymptotic variance in the present worm update is approximately 23 and 1.6 times as small as in the classical worm and the Wolff cluster updates, respectively, as shown in the inset of Fig. 8(c).

We note that the susceptibility estimator is different for each case. Although the variance of the Wolff-spin estimator (simply using spins) includes four spin correlations, the variances of the estimators in the worm algorithms do not. As we mentioned above, this is because the estimators in the worm updates [Eqs. (7) and (13)] are different from the Wolff-spin estimator. In practice, the variances were measured using Eq. (22).

We emphasize that the sampling efficiency of the Monte Carlo method should be quantified by the asymptotic variance. As shown in Fig. 8, the exponent of the integrated autocorrelation time in the classical worm update is much smaller than

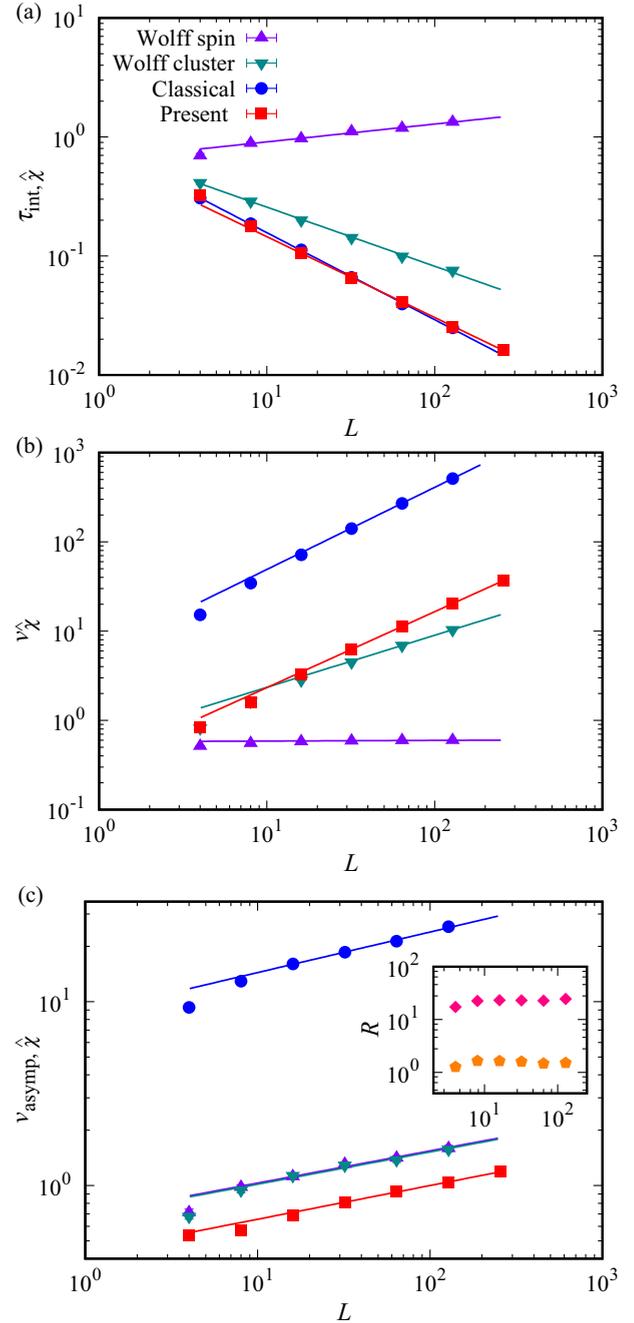


FIG. 8. (a) The integrated autocorrelation time, (b) the variance, and (c) the asymptotic variance of the magnetic susceptibility estimator as a function of the system length of the simple cubic lattice Ising model at the critical temperature in the Wolff (triangles), the classical (circles), and the present worm (squares) algorithms. In the Wolff algorithm, we test two estimators using the spins (Wolff spin) and the cluster size (Wolff cluster) (see the main text for the detail of the estimators). The exponents of $\tau_{\text{int},\hat{\chi}}$ are estimated to be 0.150(9), $-0.50(1)$, $-0.731(7)$, and $-0.679(4)$; those of $v_{\hat{\chi}}$ are to be 0.01, 0.58, 0.92, and 0.85; and those of $v_{\text{asympt},\hat{\chi}}$ are to be 0.18, 0.18, 0.22, and 0.18 in the Wolff spin, in the Wolff cluster, in the classical worm, and in the present worm updates, respectively. The inset of panel (c) shows the ratios of the asymptotic variance in the classical worm (diamonds) and the Wolff cluster (pentagons) updates to the one in the present worm update, which are approximately 23 and 1.6 for large system sizes, respectively.

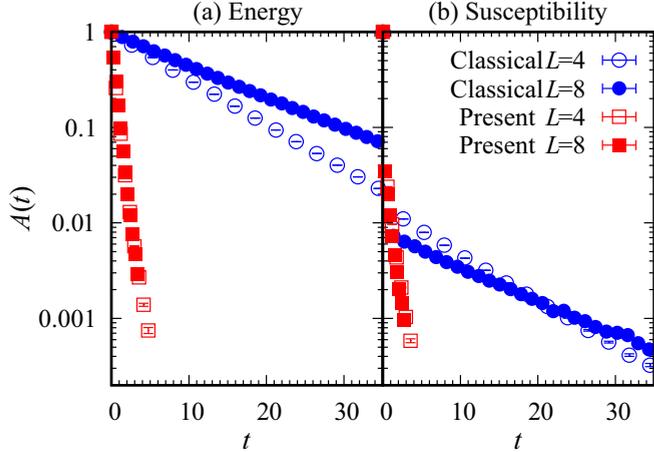


FIG. 9. Autocorrelation functions of (a) the total energy and (b) the magnetic susceptibility estimators in the classical (circles) and the present (squares) worm updates for the $L = 4$ (open) and 8 (solid) simple cubic lattice Ising model. The horizontal axis is the rescaled time of the Monte Carlo dynamics in units of L^3 worm shifting or scattering steps.

in the Wolff cluster update, but the exponent of the variance in the classical worm update is much larger than in the Wolff cluster update. Interestingly, the exponent of the asymptotic variance is almost the same for the two algorithms. Indeed, the asymptotic variance in the classical worm update is much larger than in the Wolff cluster update.

The present worm update successfully reduces the variance of the susceptibility estimator. Because the worm length is proportional to the susceptibility exactly in the classical worm update as shown in Eq. (7) and approximately in the present worm update as shown in Eq. (15), the variance of the worm length is also significantly reduced by the present algorithm. We expect the overall performance improvement to be attributed to the variance reduction of the worm length.

We investigate the relaxation rate as well as the sampling efficiency of the present update. The autocorrelation functions (1) of the total energy and the magnetic susceptibility estimators for $L = 4$ and 8 are shown in Fig. 9, calculated from more than 2^{30} independent Markov chains (sample paths). Each chain was sampled after thermalization steps that are much longer than the obtained exponential autocorrelation time, so the dependence on the initial state in the simulations is negligible in the present results. The function of the energy estimator shows an almost single exponential decay; that of the susceptibility estimator shows some fast and slow decays. While $\tau_{\text{int},\hat{\chi}}$ in the classical worm update decreases with L as shown in Fig. 8(a), $\tau_{\text{exp},\hat{\chi}}$ for $L = 8$ is larger than for $L = 4$ as shown in Fig. 9(b). The reason why $\tau_{\text{int},\hat{\chi}}$ decreases with L in contrast to $\tau_{\text{exp},\hat{\chi}}$ is that the prefactor of the slow mode decreases with L , which is also seen in Fig. 9(b).

We show the exponential autocorrelation times as a function of L in Fig. 10. The bootstrap method was used in the estimation of the fitting parameter as mentioned in Sec. IV. We found $\tau_{\text{exp},\hat{\chi}} \approx \tau_{\text{exp},\hat{E}}$, which is most likely the maximum exponential autocorrelation time among all the estimators. In addition, the autocorrelation function of the energy estimator

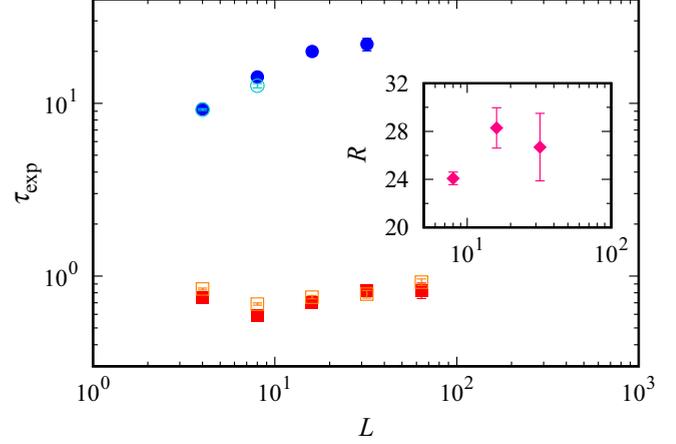


FIG. 10. Exponential autocorrelation times of the energy (solid) and the magnetic susceptibility (open) estimators as a function of the system length in the classical (circles) and the present (squares) worm updates for the simple cubic lattice Ising model. The inset shows the ratio of the autocorrelation time of the energy estimator in the classical algorithm to the one in the present algorithm.

is well approximated by a single exponential function, as shown in Fig. 9(a). Thus, the exponential and the integrated autocorrelation times should be almost the same: $\tau_{\text{exp},\hat{E}} \approx \tau_{\text{int},\hat{E}}$, which we indeed confirmed in the present results. We hence found the asymptotic scaling: $\tau_{\text{exp},\hat{\chi}} \approx \tau_{\text{exp},\hat{E}} \approx \tau_{\text{int},\hat{E}} \propto L^{0.27}$, the exponent of which was estimated from the plots in Fig. 7. We therefore estimate the dynamic critical exponent of the simple cubic lattice Ising model to be $z \approx 0.27$ in the worm update.

The exponential autocorrelation time in the present worm update is approximately 26 times as small as in the classical worm update, as shown in the inset of Fig. 10, which is consistent with the asymptotic variances of the energy and the magnetic susceptibility estimators. Note that the summation of the autocorrelation function in the rescaled time is somewhat different from the rescaled integrated autocorrelation time (18) because of the existence of the constant $1/2$ in the definition (3). Nonetheless, the asymptotic scaling is the same for the two quantities.

VI. SUMMARY AND DISCUSSION

We have proposed a modified worm algorithm for the Ising model and demonstrated performance improvement over the conventional worm algorithm at the critical temperature. The kinks of the present worm are located on bonds instead of sites of a lattice as shown in Figs. 1 and 2. The worm scattering probabilities are optimized using the directed worm framework and the geometric allocation approach as illustrated in Figs. 3 and 4. We minimize the backscattering (rejection) probability and reduce it to zero in a wide range of temperatures, including the critical point. Moreover, we maximize the forward scattering probability to enhance further the diffusivity, or the diffusion constant, of the kink. Successful enhancement of the diffusivity is confirmed by observing the increased variance of the distribution of the distance between the two kinks, as displayed in Figs. 5 and 6. As a result,

the variance of the worm length, which is approximately proportional to the variance of the susceptibility estimator, is significantly reduced, as shown in Fig. 8(b).

We have discussed how to quantify the computational efficiency of the MCMC method and measure relevant quantities in the present approach. The relaxation rate is quantified by the exponential autocorrelation time, and the sampling efficiency is by the asymptotic variance, which is the prefactor of the asymptotic scaling of the statistical error squared.

The exponential autocorrelation times and the asymptotic variances in the present worm update are approximately only 4% as large as in the classical (conventional) worm update for the simple cubic lattice Ising model as shown in Figs. 7, 8, 9, and 10. The present worm update is surprisingly even more efficient than the Wolff cluster update, although the exponent of the asymptotic variance is most likely the same. We expect the improvement over the classical algorithm to be attributed to the variance reduction of the worm length.

The dynamic critical exponent of the simple cubic lattice Ising model is estimated to be $z \approx 0.27$ from fitting to a single power law aL^z , in which a and z are the fitting parameters. The resultant fit is statistically reasonable, producing a plausible mean square error of the regression $\chi^2/N_{\text{dof}} \approx 1$, where χ^2 is the sum of the squared residuals and N_{dof} is the number of degrees of freedom in the regression.

The estimate of the dynamic critical exponent is somewhat larger than the previous estimate: $z = \alpha/\nu \approx 0.174$, which was proposed in the Wolff cluster update [12] and supported in the classical worm update [16]. This relation between the critical exponents was inferred from a numerical observation that the integrated autocorrelation time of the energy estimator is approximately proportional to the specific heat ($\propto L^{\alpha/\nu}$ asymptotically) at the critical temperature. We checked the ratio (not shown) of the autocorrelation time to the specific heat more precisely than the previous works did and found a slight but systematic increase as a function of L . This increase indicates $z > \alpha/\nu$, which is consistent with the direct fitting of the autocorrelation time. Note that although the total energy was measured in the extended state space in Ref. [16], the exponent of the autocorrelation time of the energy is expected to be the same for the original and the extended state space.

Our estimate $z \approx 0.27$ is interestingly consistent with an estimate for the Wolff update, $z = 0.24(2)$ [8]. This agreement suggests that the worm and the Wolff

algorithms share the same exponent not only of the asymptotic variance but also of the exponential autocorrelation time.

We have estimated the exponents of the autocorrelation times: $L^{0.27} \propto \tau_{\text{int},\hat{E}} \approx \tau_{\text{exp},\hat{E}} \approx \tau_{\text{exp},\hat{\chi}} \gg \tau_{\text{int},\hat{\chi}} \sim L^{-0.7}$.

A lesson to learn from the present analysis is that we must be careful to estimate τ_{exp} and needed thermalization (burn-in) steps. Because τ_{int} is usually easier to estimate than τ_{exp} , in some (or probably many) cases, people roughly estimate τ_{exp} assuming $\tau_{\text{exp}} \sim \tau_{\text{int}}$. This assumption is correct if the autocorrelation function is well approximated by a single exponential term and $\tau_{\text{exp}} \gg 1$. If the autocorrelation function has more than one exponential term, the integrated autocorrelation time is approximately given by $\tau_{\text{int}} \sim c \tau_{\text{exp}}$, where c is the prefactor of the slowest decay. Therefore, τ_{exp} can be much larger than τ_{int} possibly in orders of magnitude as we have estimated $\tau_{\text{int},\hat{\chi}} \propto L^{-0.73}$ but $\tau_{\text{exp},\hat{\chi}} \propto L^{0.27}$ in the classical worm update. It is interesting that the prefactor decreases with the system length: $c \propto L^{-1.0}$.

The present approach can be generalized to a wide range of physical models to which the conventional worm algorithm has been applied, such as the $|\phi|^4$ model [15], the Potts model [20], the $O(n)$ loop model [17,21,22], and lattice QCD [23]. The geometric allocation approach is expected to improve the computational efficiency of the directed worm update also for these systems. Our approach can be applied to frustrated models as well in combination with the dual worm formalism [27]. In the meantime, an application of the lifting technique, which is another way to break the detailed balance, to the worm algorithm was recently proposed for the Ising model [37]. It is of interest to further combine our approach and the lifting technique. The performance of the present worm algorithm for other models needs to be investigated in the future.

ACKNOWLEDGMENTS

The author is grateful to Syngye Todo for the discussion on estimators in the worm algorithm. Some simulations were performed using computational resources of the Supercomputer Center at the Institute for Solid State Physics, the University of Tokyo. The author acknowledges support by KAKENHI under Grant No. 16K17762 from Japan Society for the Promotion of Science.

-
- [1] D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, 2nd ed. (Cambridge University, Cambridge, England, 2005).
- [2] M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University, New York, 1999).
- [3] P. C. Hohenberg and B. I. Halperin, Theory of dynamic critical phenomena, *Rev. Mod. Phys.* **49**, 435 (1977).
- [4] A. Sokal, Monte Carlo methods in statistical mechanics: Foundations and new algorithms, in *Functional Integration: Basics and Applications*, edited by C. DeWitt-Morette, P. Cartier, and A. Folacci (Springer, New York, 1997).
- [5] F.-G. Wang and C.-K. Hu, Universality in dynamic critical phenomena, *Phys. Rev. E* **56**, 2310 (1997).
- [6] M. P. Nightingale and H. W. J. Blöte, Monte Carlo computation of correlation times of independent relaxation modes at criticality, *Phys. Rev. B* **62**, 1089 (2000).
- [7] Y. Murase and N. Ito, Dynamic critical exponents of three-dimensional Ising models and two-dimensional three-states Potts models, *J. Phys. Soc. Jpn.* **77**, 014002 (2008).
- [8] C.-W. Liu, A. Polkovnikov, and A. W. Sandvik, Dynamic scaling at classical phase transitions approached through nonequilibrium quenching, *Phys. Rev. B* **89**, 054307 (2014).
- [9] R. H. Swendsen and J. S. Wang, Nonuniversal Critical Dynamics in Monte Carlo Simulations, *Phys. Rev. Lett.* **58**, 86 (1987).
- [10] U. Wolff, Collective Monte Carlo Updating for Spin Systems, *Phys. Rev. Lett.* **62**, 361 (1989).

- [11] P. Tamayo, R. C. Brower, and W. Klein, Single-cluster Monte Carlo dynamics for the Ising model, *J. Stat. Phys.* **58**, 1083 (1990).
- [12] P. D. Coddington and C. F. Baillie, Empirical Relations Between Static and Dynamic Exponents for Ising Model Cluster Algorithms, *Phys. Rev. Lett.* **68**, 962 (1992).
- [13] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn, Exact, complete, and universal continuous-time world-line Monte Carlo approach to the statistics of discrete quantum systems, *Sov. Phys. JETP* **87**, 310 (1998).
- [14] O. F. Syljuasen and A. W. Sandvik, Quantum Monte Carlo with directed loops, *Phys. Rev. E* **66**, 046701 (2002).
- [15] N. Prokof'ev and B. Svistunov, Worm Algorithms for Classical Statistical Models, *Phys. Rev. Lett.* **87**, 160601 (2001).
- [16] Y. Deng, T. M. Garoni, and A. D. Sokal, Dynamic Critical Behavior of the Worm Algorithm for the Ising Model, *Phys. Rev. Lett.* **99**, 110601 (2007).
- [17] Q. Liu, Y. Deng, and T. M. Garoni, Worm Monte Carlo study of the honeycomb-lattice loop model, *Nucl. Phys. B* **846**, 283 (2011).
- [18] H. Suwa and S. Todo, Markov Chain Monte Carlo Method Without Detailed Balance, *Phys. Rev. Lett.* **105**, 120603 (2010).
- [19] H. Suwa, *Geometrically Constructed Markov Chain Monte Carlo Study of Quantum Spin-phonon Complex Systems*, Springer Theses (Springer, New York, 2014).
- [20] Y. D. Mercado, H. G. Evertz, and C. Gatteringer, Worm algorithms for the 3-state Potts model with magnetic field and chemical potential, *Comput. Phys. Commun.* **183**, 1920 (2012).
- [21] W. Janke, T. Neuhaus, and A. M. J. Schakel, Critical loop gases and the worm algorithm, *Nucl. Phys. B* **829**, 573 (2010).
- [22] H. Shimada, J. L. Jacobsen, and Y. Kamiya, Phase diagram and strong-coupling fixed point in the disordered $O(n)$ loop model, *J. Phys. A: Math. Theor.* **47**, 122001 (2014).
- [23] D. H. Adams and S. Chandrasekharan, Chiral limit of strongly coupled lattice gauge theories, *Nucl. Phys. B* **662**, 220 (2003).
- [24] H. Suwa and S. Todo, Generalized Moment Method for Gap Estimation and Quantum Monte Carlo Level Spectroscopy, *Phys. Rev. Lett.* **115**, 080601 (2015).
- [25] P. Hitchcock, E. S. Sørensen, and F. Alet, Dual geometric worm algorithm for two-dimensional discrete classical lattice models, *Phys. Rev. E* **70**, 016702 (2004).
- [26] J.-S. Wang, Worm algorithm for two-dimensional spin glasses, *Phys. Rev. E* **72**, 036706 (2005).
- [27] G. Rakala and K. Damle, Cluster algorithms for frustrated two-dimensional Ising antiferromagnets via dual worm constructions, *Phys. Rev. E* **96**, 023304 (2017).
- [28] H. A. Kramers and G. H. Wannier, Statistics of the two-dimensional ferromagnet. Part I, *Phys. Rev.* **60**, 252 (1941).
- [29] J. B. Kogut, An introduction to lattice gauge theory and spin systems, *Rev. Mod. Phys.* **51**, 659 (1979).
- [30] K. Fukui and S. Todo, Order- N cluster Monte Carlo method for spin systems with long-range interactions, *J. Comput. Phys.* **228**, 2629 (2009).
- [31] T. Horita, H. Suwa, and S. Todo, Upper and lower critical decay exponents of Ising ferromagnets with long-range interaction, *Phys. Rev. E* **95**, 012143 (2017).
- [32] Y. Deng and H. W. J. Blöte, Simultaneous analysis of several models in the three-dimensional Ising universality class, *Phys. Rev. E* **68**, 036125 (2003).
- [33] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. (Springer, New York, 2004).
- [34] A. C. Davison and D. V. Hinkley, *Bootstrap Methods and their Application* (Cambridge University, Cambridge, England, 1997).
- [35] A. Sen, H. Suwa, and A. W. Sandvik, Velocity of excitations in ordered, disordered and critical antiferromagnets, *Phys. Rev. B* **92**, 195145 (2015).
- [36] B. A. Berg, *Markov Chain Monte Carlo Simulations and their Statistical Analysis* (World Scientific, Singapore, 2004).
- [37] E. M. Elçi, J. Grimm, L. Ding, A. Nasrawi, T. M. Garoni, and Y. Deng, Lifted worm algorithm for the Ising model, *Phys. Rev. E* **97**, 042126 (2018).