

Markov Chain Monte Carlo Method without Detailed Balance

Hidemaro Suwa¹ and Syngge Todo^{1,2}

¹*Department of Applied Physics, University of Tokyo, Tokyo 113-8656, Japan*

²*CREST, Japan Science and Technology Agency, Kawaguchi 332-0012, Japan*

(Received 14 July 2010; published 17 September 2010)

We present a specific algorithm that generally satisfies the balance condition without imposing the detailed balance in the Markov chain Monte Carlo. In our algorithm, the average rejection rate is minimized, and even reduced to zero in many relevant cases. The absence of the detailed balance also introduces a net stochastic flow in a configuration space, which further boosts up the convergence. We demonstrate that the autocorrelation time of the Potts model becomes more than 6 times shorter than that by the conventional Metropolis algorithm. Based on the same concept, a bounce-free worm algorithm for generic quantum spin models is formulated as well.

DOI: [10.1103/PhysRevLett.105.120603](https://doi.org/10.1103/PhysRevLett.105.120603)

PACS numbers: 05.10.Ln, 02.50.-r, 02.70.Ss, 02.70.Tt

The Markov chain Monte Carlo (MCMC) method, which is a vital tool for investigating almost all kinds of statistical problems, especially systems with multiple degrees of freedom, is being applied extensively across the various disciplines, such as statistics, economics, and bioinformatics, not to mention physics [1–3]. In the MCMC method for a stationary distribution, the balance condition [BC, Eq. (2) below] and the ergodicity must be imposed; the former ensures the invariance of a target distribution, while the latter does the convergence to the invariant [4,5]. Although an MCMC method satisfying these two conditions guarantees unbiased results in infinite time in principle, a rapid convergence, that is, a short correlation time, is essential for the method to work in practice. The longer the autocorrelation time is, the larger the error bar becomes. In order to achieve weaker correlations in a Markov sequence, a number of variants have been invented so far, e.g., the Swendsen-Wang algorithm [6] and the loop algorithm [7]. The extended ensemble methods, such as the multicanonical method [8] and the exchange Monte Carlo method [9], have also been proposed and applied successfully to protein folding problems, spin glasses, etc.

In most practical implementations of the MCMC method, the detailed balance condition (DBC), the reversibility, is imposed, where every elementary transition is forced to balance with a corresponding inverse process. Thanks to the DBC, it becomes easy to find qualified transition probabilities. In the meantime, it has long been considered difficult to satisfy the BC without imposing the DBC, and attempts to reduce autocorrelations in Markov sequences have concentrated on optimizing transition probabilities within the DBC [10,11]. Here, we need to be reminded that the DBC is not a necessary condition for the invariance. The BC, which is a weaker condition than the DBC, is mathematically sufficient [4,5]. In fact, the DBC is often broken secretly, even though the DBC is used apparently to define the transition probabilities. The single spin update in a classical system is such an example. The

random update, where a spin to be flipped is chosen uniformly randomly among all spins, satisfies the DBC strictly. On the other hand, the DBC is broken in the sequential update, where spins are swept in a fixed order. The DBC is satisfied only locally, that is, only in each spin flip, and the BC is eventually fulfilled in one sweep [12].

In this Letter, we present a simple and versatile algorithm to find a set of transition probabilities in the MCMC method, which fulfills the BC but breaks the DBC even locally. As the BC is a weaker condition than the DBC, the solution space of transition probabilities is enlarged, and rejection rates can be reduced as the result. We show that by the present algorithm the average rejection rate is indeed minimized, and even reduced to zero in many relevant cases. Furthermore, breaking the DBC introduces a net stochastic flow in the configuration space. It will boost up the convergence further by suppressing random walk behavior [13–16].

In what follows, after describing our specific algorithm for finding a rejection-minimized solution, we will demonstrate its effectiveness in the single spin update of the Potts model [17] and the worm (directed-loop) update [18,19] of the quantum Heisenberg spin chain in a magnetic field. In both cases, it is established that our algorithm boosts up the convergence significantly in comparison with the conventional algorithms, such as the Metropolis (also called Metropolis-Hastings) algorithm [20,21], or the heat bath algorithm (Gibbs sampler) [22,23]. Especially, a bounce-free (rejection-free) worm algorithm can be formulated for generic quantum spin models, by which the autocorrelation time is often reduced by orders of magnitude as we will see below.

We start with considering an elementary update process, e.g., flipping a single spin in the Ising or Potts models, or choosing an exit at an operator in the worm algorithm (see below). Given an environmental configuration, we would have n candidates (including the current one) for the next configuration. The weight of each candidate configuration

(or state) is given by w_i ($i = 1, \dots, n$), to which the equilibrium probability distribution is proportional. Although in the formulation of the MCMC method, the BC (or the DBC) is usually expressed in terms of the weights $\{w_i\}$ and the transition probabilities $\{p_{i \rightarrow j}\}$, it is more convenient to introduce a quantity $v_{i \rightarrow j} := w_i p_{i \rightarrow j}$, which corresponds to the amount of (raw) stochastic flow from state i to j . The law of probability conservation and the BC are then expressed as

$$w_i = \sum_{j=1}^n v_{i \rightarrow j} \quad \forall i \quad (1)$$

$$w_j = \sum_{i=1}^n v_{i \rightarrow j} \quad \forall j, \quad (2)$$

respectively. The average rejection rate is written as $\sum_i v_{i \rightarrow i} / \sum_i w_i$. Also, it is straightforward to confirm that $\{v_{i \rightarrow j}\}$ satisfy

$$v_{i \rightarrow j} = \frac{1}{n-1} \min(w_i, w_j) \quad i \neq j \quad (3)$$

for the Metropolis algorithm, and

$$v_{i \rightarrow j} = \frac{w_i w_j}{\sum_{k=1}^n w_k} \quad \forall i, j \quad (4)$$

for the heat bath algorithm, where the detailed balance, i.e., the absence of net stochastic flow, is manifested by the symmetry under the interchange of the indices: $v_{i \rightarrow j} = v_{j \rightarrow i}$.

Our task is to find a set $\{v_{i \rightarrow j}\}$ that minimizes the average rejection rate while satisfying Eqs. (1) and (2). This procedure can be understood visually as weight landfill, where we move (or allocate) some amount of weight ($v_{i \rightarrow j}$) from state i to j keeping the entire shape of the weight boxes intact. For catching on this landfill picture, let us think at first the case with $n = 2$ as in the single spin update of the Ising model. Figure 1 shows the landfill when the Metropolis and heat bath algorithms [Eqs. (3) and (4)] are applied, where the average rejection rate ($\propto v_{1 \rightarrow 1} + v_{2 \rightarrow 2}$) clearly remains finite. Indeed, for $n = 2$ the Metropolis algorithm gives the best solution, i.e., the minimum average rejection rate even within the BC [see Eq. (9) below].

For $n \geq 3$, on the other hand, we can get ahead by breaking with the DBC. In Fig. 2, we show examples of weight landfill in the case with $n = 4$ by using the Metropolis and the heat bath algorithms together with a

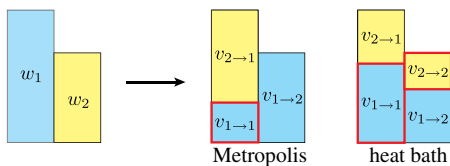


FIG. 1 (color online). Example of weight landfill by the Metropolis and heat bath algorithms for $n = 2$. The amount of landfilled weight, $v_{i \rightarrow j}$, is defined by Eqs. (3) and (4), respectively. The regions with thick frame denote the rejection rates.

solution that does not satisfy the DBC. The first two again have finite rejection rates, while the last is rejection free (i.e., $v_{i \rightarrow i} = 0 \quad \forall i$). Although a solution is not unique obviously, we propose the following procedure as a concrete algorithm to find a solution for general n . (i) Choose a configuration with maximum weight. If two or more configurations have the same maximum weight, choose one of them. In the following, we assume w_1 is the maximum without loss of generality. The order of the remaining weights does not matter. (ii) Allocate the maximum weight w_1 to the next box ($i = 2$). If the weight still remains after saturating the box, reallocate the remainder to the next ($i = 3$). Continue until the weight is all allocated. (iii) Allocate the weight of the first landfilled box (w_2) to the last partially filled box in step (ii). Continue the allocation likewise. (iv) Repeat step (iii) for w_3, w_4, \dots, w_n . Once all the boxes with $i \geq 2$ are saturated, landfill the first box ($i = 1$) afterward.

In the above procedure, all the boxes are landfilled without any space; that is, it satisfies the BC [Eq. (2)]. Since the BC is satisfied in each elementary transition, it is fulfilled in one sweep as well. It is also clear that the second and subsequent boxes must be already saturated when the allocation of its own weight is initiated, since w_1 is the maximum. By this procedure, $\{v_{i \rightarrow j}\}$ are determined as

$$v_{i \rightarrow j} = \max(0, \min(\Delta_{ij}, w_i + w_j - \Delta_{ij}, w_i, w_j)), \quad (5)$$

where

$$\Delta_{ij} := S_i - S_{j-1} + w_1 \quad 1 \leq i, \quad j \leq n, \quad (6)$$

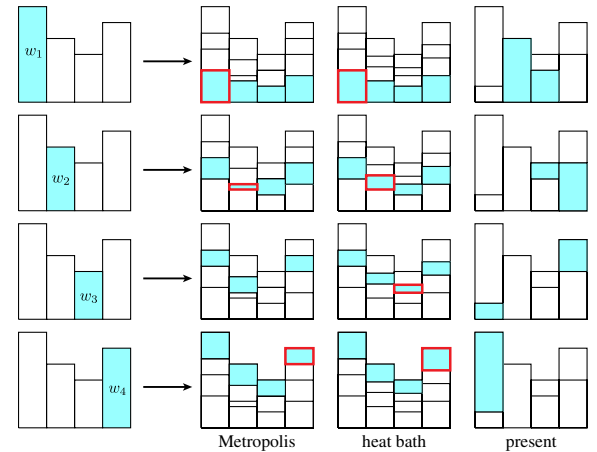


FIG. 2 (color online). Example of weight landfill by the Metropolis, heat bath, and the proposed algorithms for $n = 4$. In the present algorithm, first the maximum weight (w_1) is allocated to the second box. It saturates the second box, and the remainder is all put into the third one (first row). Next, w_2 is allocated to the partially filled box and the subsequent box (second row). The same procedure is repeated for w_3 and w_4 . This proposed algorithm is rejection free, while there remain finite rejection rates in the others as indicated by the thick frames.

$$S_i := \sum_{k=1}^i w_k \quad 1 \leq i \leq n, \quad (7)$$

$$S_0 := S_n. \quad (8)$$

Especially, from Eq. (5) we obtain

$$v_{i \rightarrow i} = \begin{cases} \max(0, 2w_1 - S_n) & i = 1 \\ 0 & i \geq 2. \end{cases} \quad (9)$$

That is, a rejection-free solution can be obtained, if

$$w_1 \leq \frac{S_n}{2} \equiv \frac{1}{2} \sum_{k=1}^n w_k \quad (10)$$

is satisfied. On the contrary, when inequality (10) is not satisfied, one has to necessarily assign the maximum weight to itself, as it is bigger than the sum of the rest. Thus, the present solution is optimal within the BC, in the sense that it minimizes the average rejection rate.

We close the introduction of our algorithm with a note about the ergodicity. It is far from trivial to prove that the present algorithm, as well as the Metropolis algorithm, satisfies the ergodicity, since many of the transition probabilities become zero exactly. In principle, however, one can always ensure the ergodicity by combining the present algorithm with the heat bath method. Another way to ensure the ergodicity is randomly choosing one of the sets of transition probabilities obtained by different landfill order, though we have not observed any glimpse of ergodicity breaking in the following simulations.

In order to assess the effectiveness of the present algorithm, we investigate the autocorrelations in the ferromagnetic q -state Potts models on the square lattice [17], which exhibit a continuous ($q \leq 4$) or first-order ($q > 4$) phase transition at $T = 1/\ln(1 + \sqrt{q})$. We calculate the autocorrelation time of the square of order parameter for $q = 4$ and 8 by several algorithms. The autocorrelation time τ_{int} is estimated through the relation: $\sigma^2 = (1 + 2\tau_{\text{int}})\sigma_0^2$, where σ_0^2 is the variance of the raw time series data and σ^2 is the

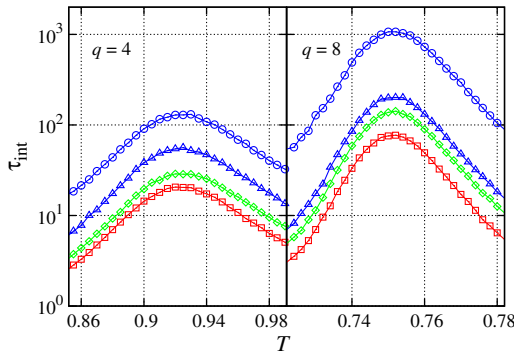


FIG. 3 (color online). Autocorrelation time of the square of order parameter near the transition temperature ($T \approx 0.910$ and 0.745 , respectively) in the 4-state (left) and 8-state (right) Potts models by the Metropolis (circles), heat bath (triangles), LOU (diamonds), and present (squares) methods. The system size is 16×16 . The error bars are the same order with the point sizes.

true variance calculated from the binned data using a bin size much larger than the τ_{int} [1]. In Fig. 3, it is clearly seen that our algorithm significantly boosts up the convergence in both models in comparison with the conventional methods. In the 4-state Potts model, the autocorrelation time becomes nearly 6.4 times shorter than that by the Metropolis algorithm, 2.7 times than the heat bath algorithm, and even 1.4 times than the locally optimal update (LOU) by Pollet *et al.* [11], which is considered as one of the best solutions within the DBC. Furthermore, the present algorithm is increasingly advantaged as q increases.

Next, we move onto the quantum Monte Carlo methods. The worm algorithm for quantum spin and lattice boson models is formulated based on either the Euclidean path integral or the high-temperature series [18,19]. One Monte Carlo sweep of the worm algorithm consists of the diagonal update, where operators are inserted or removed without changing the shape of worldlines, and the off-diagonal update, where the worldlines (and the type of operators) are updated with keeping the position of operators unchanged. In the latter process, a pair of creation and annihilation operators, which is called a worm, is inserted on a worldline (pair creation), and one of them (called the head) is moved stochastically until the head and the tail destroy each other (pair annihilation). As a thorny problem, a bounce process, where the head just backtracks and cancels the last update, has been generally inevitable within the DBC. Here, as an example, we consider the $S = 1/2$ antiferromagnetic XXZ model:

$$\mathcal{H} = \sum_{\langle i,j \rangle} (S_i^x S_j^x + S_i^y S_j^y + \Delta S_i^z S_j^z - C) - h \sum_i S_i^z, \quad (11)$$

where we introduce an arbitrary parameter C controlling the ratio between the diagonal and off-diagonal weights. In the head scattering process at an operator, only three among four exits have a nonzero weight due to the conservation of the total S^z [(a)–(c) in Fig. 4]. At the Heisenberg point ($\Delta = 1$), there remain finite bounce probabilities except at $h = 0$ within the DBC [11,19,24]. Unfortunately, the situation does not improve much even in the BC, because the total number of candidates is too small. However, the condition (10) provides us a clear prospect;

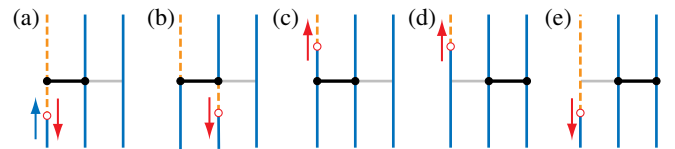


FIG. 4 (color online). Extension of the worm-going pathway in the $S = 1/2$ model. Here, dashed (solid) vertical lines denote spin up (down). The head of worm (open circle) moves on the worldline (a), and scatters at the operator (horizontal thick line). As candidate configurations, we introduce operator-flip updates (d)–(e) in addition to the conventional ones (a)–(c). Note that in (e) the position of the operator is shifted simultaneously in contrast to the simple bounce process (a).

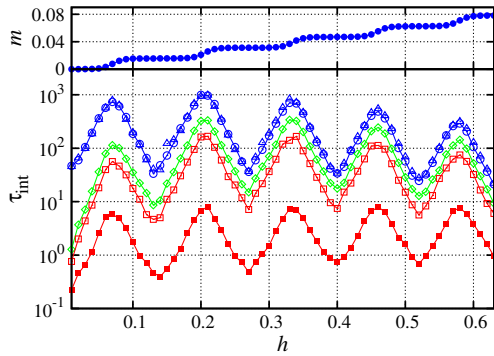


FIG. 5 (color online). Magnetic field dependence of magnetization (upper) and autocorrelation time (lower) of the $S = 1/2$ antiferromagnetic Heisenberg chain ($L = 64$, $T = 1/2L$). The maximum autocorrelation time is 1.0×10^3 by the worm update with Metropolis (circles), 9.8×10^2 by the worm with heat bath (triangles), 3.3×10^2 by the LOU (diamonds), an improvement of the directed loop, and 1.7×10^2 by the present algorithm (squares). By the bounce-free worm with the operator flip, τ_{int} is further reduced down to 8.1 (solid squares).

by increasing the number of candidates, a bounce-free algorithm will be realized. According to this strategy, we introduce *operator-flip* updates, where sites on which an operator acts are shifted simultaneously (Fig. 4). By the operator flip together with the constant C chosen as

$$C = \max[\frac{1}{4}(2\Delta + 3h - 1), \frac{1}{8}(\Delta + 3h + 1)], \quad (12)$$

we can actually eliminate the bounce process absolutely.

The autocorrelation data of the magnetization in the Heisenberg chain ($\Delta = 1$) are shown in Fig. 5. Amazingly, the bounce-free worm algorithm with the operator flip is faster by about 2 orders of magnitude than the Metropolis and the heat bath algorithms. Also in high- S spin systems, the bounce-free worms can be constructed by representing the partition function by decomposed $S = 1/2$ spins [25]. Our idea of breaking the DBC and operator-flip updates are also applied to general bosonic models effectively, because bosonic worms get bounce-minimized with more candidates.

In the present study, we have developed a simple and versatile MCMC algorithm that generally satisfies the BC without imposing the DBC. In our algorithm, the average rejection rate gets minimized, which reduces the autocorrelation time significantly in comparison with the conventional methods based on the DBC. We also have introduced operator-flip updates in the worm algorithm, yielding a bounce-free algorithm in generic spin models. The present concept can be naturally extended to systems with continuous state variables by replacing the weight landfill presented here with an asymmetric random cyclic shifting in a cumulative probability distribution function. Our approach beyond the DBC can be universally applied to any MCMC methods, even to ingenious established methods, such as the cluster algorithms and the extended ensemble

methods, and will undoubtedly improve the convergence in Markov sequences.

Most simulations were performed on T2K Supercomputer at University of Tsukuba. The program was developed based on the ALPS library [25,26]. We acknowledge support by Grant-in-Aid for Scientific Research Program (No. 20540364) from JSPS, and by Grand Challenges in Next-Generation Integrated Nanoscience, Next-Generation Supercomputer Project from MEXT, Japan.

- [1] D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, Cambridge, 2005), 2nd ed.
- [2] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, *Markov Chain Monte Carlo in Practice* (Chapman & Hall, London/New York, 1996).
- [3] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods* (Springer, New York, 2004), 2nd ed.
- [4] L. Tierney, *Ann. Stat.* **22**, 1701 (1994).
- [5] S.P. Meyn and R.L. Tweedie, *Markov Chains and Stochastic Stability* (Springer, New York, 1993).
- [6] R.H. Swendsen and J.S. Wang, *Phys. Rev. Lett.* **58**, 86 (1987).
- [7] H.G. Evertz, G. Lana, and M. Marcu, *Phys. Rev. Lett.* **70**, 875 (1993).
- [8] B.A. Berg and T. Neuhaus, *Phys. Rev. Lett.* **68**, 9 (1992).
- [9] K. Hukushima and K. Nemoto, *J. Phys. Soc. Jpn.* **65**, 1604 (1996).
- [10] J.S. Liu, *Stat. Comput.* **6**, 113 (1996).
- [11] L. Pollet, S.M.A. Rombouts, K. Van Houcke, and K. Heyde, *Phys. Rev. E* **70**, 056705 (2004).
- [12] V.I. Manoussouhakis and M.W. Deem, *J. Chem. Phys.* **110**, 2753 (1999).
- [13] S.L. Adler, *Phys. Rev. D* **23**, 2901 (1981).
- [14] S. Duane, A.D. Kennedy, B.J. Pendleton, and D. Roweth, *Phys. Lett. B* **195**, 216 (1987).
- [15] P. Diaconis, S. Holmes, and R.M. Neal, *Ann. Appl. Probab.* **10**, 726 (2000).
- [16] The overrelaxation method [13] and the hybrid Monte Carlo [14] both satisfy the DBC, mistakenly believed to break it.
- [17] F.Y. Wu, *Rev. Mod. Phys.* **54**, 235 (1982).
- [18] N.V. Prokov'ev, B.V. Svistunov, and I.S. Tupitsyn, *Sov. Phys. JETP* **87**, 310 (1998).
- [19] O.F. Syljuasen and A.W. Sandvik, *Phys. Rev. E* **66**, 046701 (2002).
- [20] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, *J. Chem. Phys.* **21**, 1087 (1953).
- [21] W.K. Hastings, *Biometrika* **57**, 97 (1970).
- [22] M. Creutz, *Phys. Rev. D* **21**, 2308 (1980).
- [23] S. Geman and D. Geman, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721 (1984).
- [24] F. Alet, S. Wessel, and M. Troyer, *Phys. Rev. E* **71**, 036706 (2005).
- [25] S. Todo and K. Kato, *Phys. Rev. Lett.* **87**, 047203 (2001).
- [26] A. Albuquerque *et al.*, *J. Magn. Magn. Mater.* **310**, 1187 (2007).