Class of Monte Carlo algorithms for dynamic problems leads to an adaptive method

Erwan Adam,* Luc Billard, and Frédéric Lançon
Département de Recherche Fondamentale sur la Matière Condensée, SP2M/NM, CEA-Grenoble, 38054 Grenoble Cedex 9, France
(Received 14 May 1998)

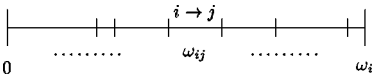
We introduce a class of Monte Carlo algorithms that solve a dynamic problem defined by the transition rates and the initial state of a discrete system. This class contains the method of Bortz, Kalos, and Lebowitz (BKL) [J. Comp. Phys. **17**, 10 (1975)] as a limit. We show that introducing a constant time step in a Metropolis algorithm leads to an approximation of the solution in which the system relaxation times are underestimated. This can be corrected if the time step is an adequate stochastic variable. Thus, we are able to define kinetic Metropolis algorithms and generalize them in a case of nonconstant numbers of attempt configurations. The algorithm class allows us to introduce a useful method in which the calculation of transition rates are exploited for the next step in an adaptive way. This method corresponds to a kinetic Metropolis algorithm when the rejection probability is reasonable and becomes similar to the BKL method otherwise. We describe and compare four different algorithms applied to a physical example about the diffusion in lattice gases. [S1063-651X(99)07101-9]

PACS number(s): 02.70.Lq, 05.20.Dd, 05.50.+q

I. INTRODUCTION

A dynamic problem in the case of a discrete system that can occupy $\mathcal{N}$ states is defined by giving a particular configuration as initial state and the transition rates of the different connected configurations. The transition rate $\omega_{ij} \geqslant 0$ is the probability per unit time for the system to transit from the state i to the state $j \neq i$.

The Bortz, Kalos, and Lebowitz method [1–3] (also called BKL or *n*-fold way) to treat this problem is based on the fact that the dynamical processes are in their formulation Poisson processes. The algorithm can be summarized with a diagram [4] where each segment represents a transition from i and its length is equal to the rate of this transition,



When the system is in state i, the method consists in steps of three operations. (i) The time is increased by a random number Δt distributed as $\omega_i \exp(-\omega_i \Delta t)$ where ω_i is the total transition rate from i, $\Sigma_{j \neq i} \omega_{ij}$. (ii) The system transits from i to $j \neq i$ with probability $p_{ij} = \omega_{ij} / \omega_i$. (iii) One has to update the new transition rates ω_{jk} and the new total transition rate ω_j. The main advantage of this algorithm is that there is no rejection but two difficulties remain. Operation (ii) can be time consuming and is the object of previous studies [4,5]. Because of operations (ii) and (iii), all the ω_{ij} are needed at each step. This can be a serious problem when the calculation of each ω_{ij} is complex, for instance, if it involves the whole system state.

On the other hand, the ''historical'' way for introducing time in a Monte Carlo simulation is to associate a constant time step at each step of a Metropolis algorithm. More precisely, a Metropolis algorithm consists in (i) choosing an attempt configuration j between a *constant* number N of configurations in a set containing at least those with $\omega_{ij} > 0$. Then, (ii) this configuration is accepted with probability $\alpha_{ij} = \omega_{ij} / \nu_0 \in [0,1]$ or rejected with probability $1 - \alpha_{ij}$ where ν_0 is an attempt frequency. In this algorithm, the dynamics are introduced by associating to each Monte Carlo step the constant time step $1/(\nu_0 N)$ [6]. We will show, in Sec. V, that this algorithm leads to an approximation of the solution in which the system relaxation times are underestimated. This can be corrected if the time steps are adequate stochastic variables.

We introduce, in Sec. III, a class of algorithms to treat the dynamic problem defined in the first paragraph. We show (Appendix) that this class satisfies the formal master equation solution given in Sec. II. Section IV is devoted to particular cases, i.e., BKL and kinetic Metropolis algorithms with or without constant number of attempt configuration. These kinetic Metropolis algorithms are compared to the ''historical'' algorithms in Sec. V. In Sec. VI, we describe our adaptive method. Finally, these different methods are tested in Sec. VII.

II. FORMAL SOLUTION

Let us recall the exact, but formal, solution of the dynamic problem of Sec. I. By defining the matrix $\Omega = (\Omega_{ij})$ with $\Omega_{ij} = \omega_{ij}$ if $j \neq i$ and $\Omega_{ii} = -\Sigma_{j \neq i} \omega_{ij}$, we write the master equation

$$\dot{Q}_i(t) = \sum_j Q_j(t) \Omega_{ji}, \tag{1}$$

where $Q_i(t)$ is the probability for the system to be in configuration i at time t. This relation gives the exact solution for the system state at time t as

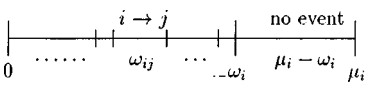
$$[Q_1(t) \cdots Q_{\mathcal{N}}(t)] = [Q_1(0) \cdots Q_{\mathcal{N}}(0)][\exp(\Omega t)], \tag{2}$$

*Electronic address: Erwan.Adam@cea.fr

where $[Q_1(0)\cdots Q_{\mathcal{N}}(0)]$ is the initial vector, i.e., $Q_i(0)=1$ for one i and $Q_{j\neq i}(0)=0$.

III. ALGORITHM CLASS

The set of algorithms we introduce is summarized by a diagram, which generalizes the BKL one presented in Sec. I,



They consist in successive steps of two operations. In the first one, the time is increased by a random number Δt distributed as

$$\rho_i(\Delta t)=\mu_i\exp(-\mu_i\Delta t). \tag{3}$$

In the second operation, the system transits in $j\neq i$ with probability p_{ij} or it stays in i with probability p_{ii} with

$$p_{ij}=\frac{\omega_{ij}}{\mu_i} \text{ for } j\neq i, \tag{4}$$

$$p_{ii}=1-\frac{\omega_i}{\mu_i} \text{ with } \omega_i=\sum_{j\neq i}\omega_{ij}. \tag{5}$$

The value of μ_i is not specified but must satisfy the obvious condition $p_{ii}\geqslant 0$,

$$\mu_i\geqslant\omega_i. \tag{6}$$

The previous algorithms form a class in the sense that the relation (6) gives a condition to be fulfilled by the set $\{\mu_i\}$ but does not specify their values. In fact, each value of μ_i gives different probabilities p_{ij} and p_{ii} and then defines a different Monte Carlo algorithm. A proof that this class of algorithm satisfies exactly the relation (1), independently of the choice of $\{\mu_i\}$, is given in the Appendix.

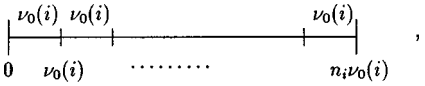
IV. KINETIC METROPOLIS ALGORITHMS

If we choose $\mu_i=\omega_i$, the generalized Monte Carlo algorithm is exactly the BKL one. As soon as $\mu_i>\omega_i$, the probability to stay in the configuration i is $p_{ii}\neq 0$ and we have another Monte Carlo algorithm. Introducing the range $[\omega_i,\mu_i]$ is very useful in order to understand the Metropolis-type algorithms in terms of kinetics. Indeed, they just correspond to a judicious subdivision of the interval $[0,\mu_i]$ into intervals of the same lengths as shown below.

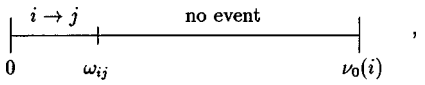
Let n_i be the number of configurations j *really connected* to i (i.e., $\omega_{ij}>0$). Note that n_i are not necessarily the same for all configurations. Let $\nu_0(i)$ be an attempt frequency for the configuration i satisfying

$$\nu_0(i)\geqslant\max_j(\omega_{ij}). \tag{7}$$

A kinetic Metropolis algorithm with n_i attempt configurations corresponds to the choice $\mu_i=n_i\nu_0(i)$, i.e., the range $[0,\mu_i]$ is divided into n_i subsegments



where there is one segment for each configuration j really connected to i and each interval contains two subintervals

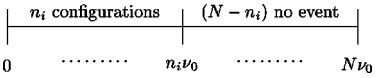


i.e., the algorithm consists in increasing the time of a random number Δt distributed as $n_i\nu_0(i)\exp[-n_i\nu_0(i)\Delta t]$, then choosing an attempt configuration j with equiprobability among the n_i that are really connected to i and finally, transiting in j with probability $\omega_{ij}/\nu_0(i)$. The main advantage of this algorithm is that only one ω_{ij} is computed in one step, but there is a rejection probability measured by $p_{ii}=1-\omega_i/[n_i\nu_0(i)]$.

To define a kinetic Metropolis algorithm with a constant number N of attempt configurations, we assume

$$N\geqslant\max_i(n_i) \text{ and } \nu_0\geqslant\max_{(i,j)}(\omega_{ij}) \tag{8}$$

and apply the generalized algorithm with $\mu_i=N\nu_0$. It corresponds to



where the n_i intervals corresponding to the really connected configurations are divided in two subintervals as described above. The algorithm is the same as the one described in Sec. I but the time increments are now random numbers Δt distributed as $\nu_0 N\exp(-\nu_0 N\Delta t)$ and not constant time steps $\Delta t=1/(\nu_0 N)$. We will see in the next section what this difference does imply. Note that there is *no assumption* here that the time steps should be smaller than a time resolution during which no events can occur simultaneously.

V. COMPARISON WITH THE CONSTANT TIME-STEP ALGORITHMS

If the time increment is a constant Δt, for instance, as in the ''historical'' algorithms described in the beginning of the text where $\Delta t=1/(\nu_0 N)$, the system state at time t is

$$[Q(t)]=[Q(0)][\mathbf{1}+\Omega\Delta t]^{\mathrm{int}(t/\Delta t)} \tag{9}$$

with the notations of Sec. II. Whereas the exact result (2) is $[Q(t)]=[Q(0)][\exp(\Omega t)]$. Let us study what relation (9) implies on the relaxation towards equilibrium. When Ω satisfies the detailed balance relation,

$$\omega_{ij}\exp(-\beta E_i)=\omega_{ji}\exp(-\beta E_j), \tag{10}$$

where E_i is the energy of the configuration i, and if for every pair (i,j), there exists a path $(i,i_1,\ldots,i_n,j)$ such that $\omega_{ii_1}\cdots\omega_{i_nj}>0$ (ergodicity condition), one shows [7] the matrix Ω can be diagonalized with the eigenvalues $\{-\lambda_k,k\in[1,\mathcal{N}]\}$ such that $\lambda_1=0$ and $\lambda_k>0$ for all $k\geqslant 2$. Those

eigenvalues define the system relaxation times $\tau_k = 1/\lambda_k$ for $k \geqslant 2$. The eigenvalue $\exp(-t/\tau_k)$ of the exponential matrix in Eq. (2) is replaced in Eq. (9) by

$$\left(1 - \frac{\Delta t}{\tau_k}\right)^{\mathrm{int}(t/\Delta t)}; \tag{11}$$

then, on one hand, for short relaxation times such that $\Delta t > \tau_k$, Eq. (11) is an oscillating function. This implies the time step is too large to study these relaxations. On the other hand, for $\Delta t < \tau_k$, Eq. (11) is written

$$\exp\left[\mathrm{int}(t/\Delta t)\ln\left(1 - \frac{\Delta t}{\tau_k}\right)\right] \sim \exp(-t/\tau_k') \quad \text{when} \quad t \gg \Delta t \tag{12}$$

with

$$\tau_k' = -\frac{\Delta t}{\ln(1 - \Delta t/\tau_k)} < \tau_k \tag{13}$$

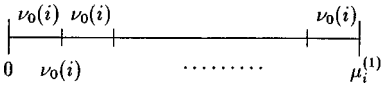
and therefore, the constant time-step algorithms underestimate the system relaxation times. For instance, for a time step $\Delta t = \tau_k/10$, the relaxation time τ_k is underestimated by 5%.

VI. THE ADAPTIVE METHOD

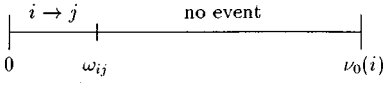
In a kinetic Metropolis algorithm introduced in Sec. IV, only one transition rate is computed per step but there is a rejection probability. In the BKL method, there is no rejection but all the transition rates are needed at each step. We describe here an adaptive algorithm that goes automatically and continuously from a kinetic Metropolis algorithm when the rejection probabilities are low to the BKL method otherwise.

The main idea is to start with the kinetic Metropolis algorithm with n_i attempt configurations described in Sec. IV, but to store the values of the transition probabilities of the rejected transitions in order to use them in the following steps.

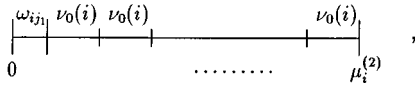
The first step of the adaptive method in configuration i corresponds to a step of the kinetic Metropolis algorithm with $\mu_i^{(1)} = n_i \nu_0(i)$,



where there is one interval for each configuration j directly connected to i and each interval contains two subintervals



i.e., the time is increased by a random number Δt distributed as $\mu_i^{(1)}\exp(-\mu_i^{(1)}\Delta t)$. Then an attempt configuration j_1 is chosen among the n_i that are directly connected to i and finally, the system transits in j_1 with probability $\omega_{ij_1}/\nu_0(i)$. If the system does transit in j_1, we have to start a new first step with $i = j_1$; otherwise, the next step is proceeded with $\mu_i^{(2)} = \omega_{ij_1} + (n_i - 1)\nu_0(i)$,

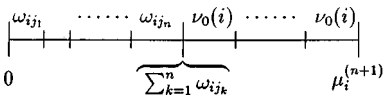


i.e., the time is increased by a random number Δt distributed as $\mu_i^{(2)}\exp(-\mu_i^{(2)}\Delta t)$. Then, with probability $\omega_{ij_1}/\mu_i^{(2)}$, j_1 is selected once again *and* the system does transit in j_1. Otherwise, a new attempt configuration j_2 is selected randomly between the $n_i - 1$ other ones and the system transits in j_2 with probability $\omega_{ij_2}/\nu_0(i)$.

More generally, the step number $n+1 (n \in [0, n_i])$ corresponds to

$$\mu_i^{(n+1)} = \sum_{k=1}^{n} \omega_{ij_k} + (n_i - n)\nu_0(i), \tag{14}$$

where $j_1, \ldots, j_n$ are the n different configurations, which have been successively attempted and rejected:



Precisely, the algorithm in step $n+1$ consists in increasing the time of a random number distributed as $\mu_i^{(n+1)}\exp(-\mu_i^{(n+1)}\Delta t)$. Then, with probability $\Sigma_{k=1}^{n}\omega_{ij_k}/\mu_i^{(n+1)}$, the system will transit in a configuration already attempted; the research of this configuration is similar to the research in a BKL algorithm [4,5]. Otherwise, a segment corresponding to a new configuration j_{n+1} is selected randomly amongst $n_i - n$ and the system transits in j_{n+1} with probability $\omega_{ij_{n+1}}/\nu_0(i)$.

In this algorithm, only one ω_{ij} is computed in one step, but we have the guarantee that the system transits in a number of steps less than or equal to $n_i + 1$. At each time the system transits, the process is started at a new first step with the new configuration. If the rejection probability is low, few transition rates will be calculated. Otherwise, almost all the ω_{ij} will be needed and the algorithm will be similar to the BKL method.

VII. A PHYSICAL APPLICATION: DIFFUSION IN LATTICE GASES

To illustrate the usefulness of the different algorithms on a physical application, we have used them in the case of a lattice gas or a binary alloy where the N sites of a regular simple cubic lattice are occupied by A particles and by V vacancies or particles. We consider that two configurations i and j are directly connected if the system can transit from i to j by exchanging only A and V particles in one *nearest-neighbor* pair. The Hamiltonian is taken as

$$\mathcal{H}(\{S_\alpha\}) = -\sum_{\langle \alpha,\beta \rangle} J S_\alpha S_\beta, \tag{15}$$

with $S_\alpha = -1\,(+1)$ if the site α is occupied by an A (V) particle. In practice the size of the system is $N = 32 \times 32$

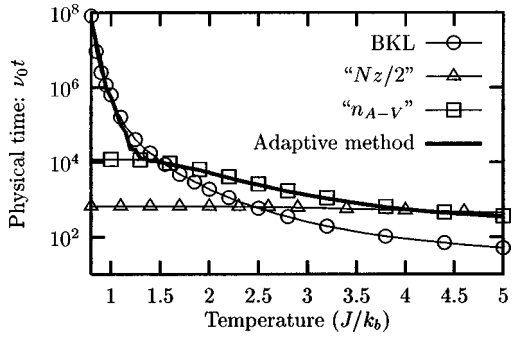


FIG. 1. Simulated physical time in a dynamical system versus temperature for a given computer time (CPU). See Sec. VII for the description of the Monte Carlo method corresponding to each curve. Each value is an average over five realizations.

$\times 32$ and the number of A particles is equal to $N/2$. The boundary conditions are periodic and the initial state, at each temperature $T=1/\beta$, is made with all the A particles together in one-half of the cubic box. The transition rates are

$$\omega_{ij}=\nu_0 \exp[-\beta \max(0,\mathcal{H}_j-\mathcal{H}_i)]. \tag{16}$$

Each configuration i is characterized by the number $n_{A-V}(i)$ of $A-V$ nearest-neighbor pairs. This number corresponds to the number of configurations j really connected to i.

For each of the following four algorithms, we have carried out five numerical simulations (corresponding to five different random number series) from the same initial state. The average energy curves $\mathcal{H}_T(t)$ are indistinguishable for all the algorithms. However, as seen below, the computation cost is different and depends on the temperature.

For the BKL algorithm, we have to calculate all the transition rates ω_{ij} and their sum ω_i. Then the time is increased by a random number Δt distributed as $\omega_i \exp(-\omega_i \Delta t)$. An $A-V$ nearest-neighbor pair is selected with the probability ω_{ij}/ω_i and the particles are exchanged. Figure 1 shows the total simulated physical time t after 10 sec of computer time (CPU time) versus temperature. The BKL method appears to be very efficient at low temperature and becomes less and less efficient when the temperature increases.

The second algorithm presented here is a kinetic Metropolis algorithm with a constant number of attempt configurations. We denote it $Nz/2$ in Fig. 1 since it corresponds to parameters $\mu_i=\nu_0 Nz/2$, where z is the coordination number of the lattice (six in the simple cubic crystal) and $Nz/2$ is the total number of nearest-neighbor pairs in the system. In this algorithm, one increases the time by Δt distributed as $\mu_i \exp(-\mu_i \Delta t)$. Then, a site and one of its z nearest neighbors are chosen randomly and if an $A-V$ pair is selected, the particles are exchanged with the probability ω_{ij}/ν_0. Whatever the configuration and the temperature, the parameter μ_i is a constant and, therefore, the simulated time is almost independent of the temperature. At high temperature, it is more efficient than the BKL method since the rejection rate is low and fewer calculations are needed.

However, in this second algorithm, the probability that the selected nearest-neighbor pair is an $A-V$ pair is low if there are large clusters of A, since they are typically located at the interfaces. This is the case here at low temperature. Thus, we introduce a third algorithm, called n_{A-V} in Fig. 1. It is a new kinetic Metropolis method in which the attempt pairs are chosen randomly only among the $n_{A-V}(i) A-V$ nearest-neighbor pairs of the configuration i, i.e., the number of attempt configurations is not constant. It corresponds to a parameter $\mu_i=n_{A-V}(i)\nu_0$. Since the rejection rate is lower than in the second algorithm, this third one is more efficient at all temperatures, despite the overhead due to the management of the $A-V$ pair list.

Finally, the adaptive method described in Sec. VI is equivalent to the previous n_{A-V} method at high temperature where it is the most efficient, and to the BKL one at low temperature where this one is the most efficient. Only near the crossover, the BKL could be significantly better, but in this method, the adapting is done automatically even if we do not know where the crossover is.

VIII. CONCLUSION

We have defined a class of Monte Carlo algorithms to solve dynamic problems on discrete systems. This class contains the well-known BKL method [1], and allows an analysis of the widely used Metropolis-type algorithm. We have demonstrated that the constant time step used in this method leads to an approximation of the solution in which the system relaxation times are underestimated. The use of distributed time steps corrects this method. We have also generalized it to the case where the number of reachable configurations is not a constant but depends on the current configuration. Depending on the system and the transition rates, one or the other of these methods can be more adapted. We have introduced an algorithm that automatically tends to the one that is the most suitable. This method is particularly useful when the transition rate calculations are computer time consuming.

ACKNOWLEDGMENT

We gratefully acknowledge Professor J. Villain for fruitful discussions.

APPENDIX: PROOF

Let i_0 be the initial configuration and $Q_i^{\mathrm{MC}}(t)$ the probability for the system to be in the state i at time t if we use one of the proposed algorithms. We write

$$\begin{aligned} Q_i^{\mathrm{MC}}(t) &= \delta_{i_0 i}\Pr[t_{i_0}>t]+p_{i_0 i}\Pr[(t_{i_0}\leqslant t),(t_{i_0}+t_i>t)] \\ &\quad +\sum_{n=1}^{\infty}\sum_{i_1,\ldots,i_n} p_{i_0 i_1}\cdots p_{i_n i} \\ &\quad \times \Pr[(t_{i_0}+\cdots t_{i_n}\leqslant t),(t_{i_0}+\cdots t_{i_n}+t_i>t)], \end{aligned} \tag{A1}$$

where $\Pr[t_{i_0}>t]$ is the probability for the system to have spent a time $t_{i_0}>t$ in the initial configuration. We recall that t_{i_0} is distributed as Eq. (3). The probabilities in Eq. (A1) can be separated in two parts as

$$Q_i^{\mathrm{MC}}(t)=\delta_{i_0 i}(1-\Pr[t_{i_0}\leqslant t])+p_{i_0 i}(\Pr[t_{i_0}\leqslant t]-\Pr[t_{i_0}+t_i\leqslant t])+\sum_{n=1}^{\infty}\sum_{i_1,\ldots,i_n} p_{i_0 i_1}\cdots p_{i_n i}\times(\Pr[t_{i_0}+\cdots+t_{i_n}\leqslant t]-\Pr[t_{i_0}+\cdots+t_{i_n}+t_i\leqslant t]). \tag{A2}$$

The last term in Eq. (A2) is

$$\Pr[t_{i_0}+\cdots+t_{i_n}+t_i\leqslant t]=\int_{t_{i_0}+\cdots t_{i_n}+t_i\leqslant t}\cdots\int dt_{i_0}\cdots dt_{i_n}dt_i\times\mu_{i_0}e^{-\mu_{i_0}t_{i_0}}\ldots\mu_{i_n}e^{-\mu_{i_n}t_{i_n}}\mu_i e^{-\mu_i t_i} \tag{A3}$$

and after a quite technical recurrence, we find

$$\frac{d}{dt}\Pr[t_{i_0}+\cdots+t_{i_n}+t_i\leqslant t]=\mu_i(\Pr[t_{i_0}+\cdots+t_{i_n}\leqslant t]-\Pr[t_{i_0}+\cdots+t_{i_n}+t_i\leqslant t]) \tag{A4}$$

and similar results for the other members of Eq. (A2). Hence, from Eq. (A2), the time derivative of $Q_i^{\mathrm{MC}}(t)$ is

$$\frac{dQ_i^{\mathrm{MC}}}{dt}=-\delta_{i_0 i}\mu_{i_0}(1-\Pr[t_{i_0}\leqslant t])+p_{i_0 i}\{\mu_{i_0}(1-\Pr[t_{i_0}\leqslant t])-\mu_i(\Pr[t_{i_0}\leqslant t]-\Pr[t_{i_0}+t_i\leqslant t])\}+\sum_{n=1}^{\infty}\sum_{i_1,i_2,\ldots,i_n} p_{i_0 i_1}\cdots p_{i_n i}\times\{\mu_{i_n}(\Pr[t_{i_0}+\cdots+t_{i_{n-1}}\leqslant t]-\Pr[t_{i_0}+\cdots+t_{i_n}\leqslant t])-\mu_i(\Pr[t_{i_0}+\cdots+t_{i_n}\leqslant t]-\Pr[t_{i_0}+\cdots+t_{i_n}+t_i\leqslant t])\}. \tag{A5}$$

By using again the relation (A2), we can express Eq. (A5) as

$$\frac{dQ_i^{\mathrm{MC}}}{dt}=-\mu_i Q_i^{\mathrm{MC}}(t)+\sum_j p_{ji}\mu_j Q_j^{\mathrm{MC}}(t). \tag{A6}$$

With the relation (5), this result becomes independent of $\{\mu_i\}$,

$$\frac{dQ_i^{\mathrm{MC}}}{dt}=-\left(\sum_{j\neq i}\omega_{ij}\right)Q_i^{\mathrm{MC}}(t)+\sum_{j\neq i}\omega_{ji}Q_j^{\mathrm{MC}}(t). \tag{A7}$$

This equation is the same as Eq. (1) and we have

$$[Q_1^{\mathrm{MC}}(t)\cdots Q_{\mathcal{N}}^{\mathrm{MC}}(t)]=[Q_1(0)\cdots Q_{\mathcal{N}}(0)][\exp(\Omega t)]. \tag{A8}$$

[1] A.B. Bortz, M.H. Kalos, and J.L. Lebowitz, J. Comput. Phys. **17**, 10 (1975).

[2] *Monte Carlo Methods in Statistical Physics*, edited by K. Binder, Topics in Current Physics, Vol. 7 (Springer, Berlin, 1979), Sec. 1.3.1.

[3] F. Lançon, L. Billard, W. Chambron, and A. Chamberod, J. Phys. F **15**, 1457 (1985).

[4] P.A. Maksym, Semicond. Sci. Technol. **3**, 594 (1988).

[5] J.L. Blue, I. Beichl, and F. Sullivan, Phys. Rev. E **51**, R867 (1995).

[6] K. Binder, Rep. Prog. Phys. **60**, 487 (1997), Sec. 3.4, and references therein.

[7] A.J.F. Siegert, Phys. Rev. **76**, 1708 (1949).