

## A Dynamical Monte Carlo Algorithm for Master Equations with Time-Dependent Transition Rates

A. Prados,<sup>1</sup> J. J. Brey,<sup>1</sup> and B. Sánchez-Rey<sup>2</sup>

*Received November 19, 1996; final June 13, 1997*

---

A Monte Carlo algorithm for simulating master equations with time-dependent transition rates is described. It is based on a waiting time image, and takes into account that the system can become frozen when the transition rates tend to zero fast enough in time. An analytical justification is provided. The algorithm reduces to the Bortz–Kalos–Lebowitz one when the transition rates are constant. Since the exact evaluation of waiting times is rather involved in general, a simple and efficient iterative method for accurately calculating them is introduced. As an example, the algorithm is applied to a one-dimensional Ising system with Glauber dynamics. It is shown that it reproduces the exact analytical results, being more efficient than the direct implementation of the Metropolis algorithm.

---

**KEY WORDS:** Monte Carlo simulation; master equations; time-dependent transition rates; waiting times.

### I. INTRODUCTION

In the past years, model systems whose dynamics is based in a master equation formulation have been extensively used in different fields of statistical physics. In particular, many master equation models have been introduced in order to study the basic mechanisms of glass transition phenomena,<sup>(1)</sup> and quite a relevant part of this behavior has been qualitatively understood.<sup>(2)</sup> Also, in the research of stochastic resonance and related problems, some master equation models have been proposed.<sup>(3)</sup>

Although the transition rates appearing in a Markovian master equation do not depend on time strictly speaking,<sup>(4)</sup> time dependent transition

---

<sup>1</sup> Física Teórica, Facultad de Física, Universidad de Sevilla, E-41080, Sevilla, Spain.

<sup>2</sup> Escuela Politécnica Superior, Universidad de Huelva, E-21819 La Rábida, Huelva, Spain.

rates have been considered in many of the above cited models. In general, the transition rates depend on some parameters that can be externally controlled, such as the temperature, external forces, etc. These parameters are considered as constants in time in the usual derivations of the master equation for a Markov process. When they are forced to vary in time following a given law, the simplest possibility is to assume that the master equation remains valid, substituting the constant parameters by the corresponding functions of time. Of course, this is an hypothesis that may not be necessarily fulfilled by any stochastic description.<sup>(5)</sup> However, the above simple approach has been used in many of the papers in refs. 1, 3 and found to be useful to understand the dynamics of driven systems. For the glassy systems in ref. 1, the externally controlled parameter is the temperature, which appears in the transition rates because of the detailed balance condition.

Master equations are often difficult to solve by analytical methods. Then, numerical approaches are frequently introduced. For time independent transition rates, two Monte Carlo procedures are usually used, the Metropolis algorithm<sup>(6)</sup> and the Bortz–Kalos–Lebowitz (BKL) algorithm.<sup>(7)</sup> Both of them provide a numerical solution to the master equation.<sup>(8)</sup> Nevertheless, both approaches are quite different in spirit. In the Metropolis algorithm a trajectory of the system is built as an equidistant in time chain of configurations. For a given configuration, a possible transition is considered and it is accepted with probability equal to its corresponding transition rate. Thus rejections of transitions are possible, and two consecutive configurations can coincide. The probability of rejection increases when the transition rate decreases, giving rise to many failed attempts of doing a transition and slowing down the algorithm.

On the other hand, the BKL algorithm is based on a waiting time image, i.e. the system spends a waiting time in a given state before making a transition to another one. A trajectory of the system is built by assigning a waiting time to the initial configuration. Then, the new state to which the system goes is chosen with a probability proportional to the corresponding transition rate. Afterwards, another waiting time for the new configuration is given, etc. Therefore, one trajectory of the system is not a chain of configurations at equidistant times, since the waiting times are random functions of the transition rates. As a consequence, we construct the probability distribution at any time, and no rejections are present. In the small transition rates regime, waiting times are large and the algorithm becomes very efficient.

The Metropolis procedure is easily extended to the time-dependent case, by approximating the transition rates by piecewise constant functions through time intervals of width  $\Delta t$ . Over each of those intervals one works

as in the case of constant transition rates, and the final configuration of an interval is taken as the initial state for the following one, in which the transition rates are up-dated. This procedure should be exact in the limit  $\Delta t \rightarrow 0$ .

The extension of the BKL algorithm requires some care. The piecewise approximation is not enough by itself, because the random function of the transition rates giving the waiting times is not bounded. It is possible that the time of arrival to a given state and the time of departure from it, which differ by the waiting time, correspond to different values of the transition rates. From a theoretical point of view, the BKL assignment of waiting times is based on the fact that the waiting time distribution is purely exponential. This simple property is not valid when the transition rates depend on time, but, in principle, one can try to modify the BKL algorithm so as to numerically solve master equations with time dependent rates.

When trying to extend the BKL algorithm to master equations with time dependent rates, the main point is to introduce the correct distribution of waiting times and to generalize the way of choosing the specific transitions actually made by the system. The result would be a generalized waiting time (GWT) algorithm. The effort of developing it, which is the principal purpose of this paper, is worth because this GWT algorithm is expected to be more efficient than the Metropolis one for small values of the transition rates. For instance, this is the case when studying glassy behavior with model systems described by master equations. In fact, we have already applied the GWT algorithm developed in this paper to a simple model with purely entropic barriers showing glass-like behavior.<sup>(9)</sup>

The plan of the paper is as follows. In Sec. II the waiting time distribution for the case of time dependent rates is introduced. In general, this distribution is only normalized to unity if the possibility of the system being frozen is taken into account. The latter appears when the transition rates vanish in the long time limit. The GWT Monte Carlo algorithm is introduced in Sec. III, where it is shown that it reduces to the BKL algorithm for constant rates. A proof showing that the GWT algorithm provides a numerical solution to the master equation is given in the Appendix. The exact evaluation of waiting times may be rather involved and, therefore, an accurate procedure is developed in Sec. IV. It is based on approximating the time dependent transition rates by piecewise constant functions. Section V deals with the application of the GWT algorithm to a one-dimensional Ising model with Glauber dynamics. Thermal cycles of cooling up to low temperatures and subsequent reheating are considered. It is found that the GWT algorithm is faster than the Metropolis one when the cycle reaches the low temperature region. Finally, a short summary of the paper is given in Sec. VI.

## II. THE WAITING TIME DISTRIBUTION

We consider a system with a discrete range of states and whose dynamics is described by a master equation with time-dependent transition rates, i.e. the probability  $p_i(t)$  of finding the system in state  $i$  at time  $t$  obeys the equation

$$\frac{dp_i(t)}{dt} = \sum_j [W_{ij}(t) p_j(t) - W_{ji}(t) p_i(t)] \quad (1)$$

where  $W_{ji}(t)$  is the transition rate from state  $i$  to state  $j$  at time  $t$ . Its time dependence is assumed to be given. The total transition rate from state  $i$  is

$$\Omega_i(t) = \sum_{j \neq i} W_{ji}(t) \quad (2)$$

Let  $q_i(t + \tau | t)$  be the probability that the system remains in state  $i$  at time  $t + \tau$ , given it was in the same state at time  $t$ , so that no transition has occurred in the time interval between  $t$  and  $t + \tau$ . The time evolution of this quantity is obtained directly from Eq. (1),

$$\frac{\partial}{\partial \tau} q_i(t + \tau | t) = -\Omega_i(t + \tau) q_i(t + \tau | t). \quad (3)$$

This equation must be solved with the initial condition  $q_i(t | t) = 1$ ; then

$$q_i(t + \tau | t) = \exp \left[ - \int_t^{t+\tau} dt' \Omega_i(t') \right] \quad (4)$$

The distribution  $f_i(t + \tau | t)$  of waiting times  $\tau$  that the system stays in state  $i$ , after arriving to it at time  $t$  and before jumping to another state, is

$$f_i(\tau | t) = -\frac{\partial}{\partial \tau} q_i(t + \tau | t) = \Omega_i(t + \tau) \exp \left[ - \int_t^{t+\tau} dt' \Omega_i(t') \right] \quad (5)$$

For time-independent transition rates,  $\Omega_i$  is a constant and  $f_i$  reduces to an exponential distribution for  $\tau$ , being independent of  $t$ . On the other hand, when the transition rates are functions of  $t$ , the waiting time distribution can be quite complicated. It is then useful to formally introduce a set of different time scales, one for each possible state of the system. For state  $i$  we define the dimension-less time scale

$$s_i = \int_0^t dt' \Omega_i(t') \quad (6)$$

When  $\Omega_i(t)$  is positive and finite for all finite  $t$ , the above scale transformation can be inverted. This is the case for the processes we are interested in and we will restrict ourselves to it in the following. It is important to differentiate between two rather different physical situations. Let us consider

$$\lim_{t \rightarrow \infty} \int_0^t dt' \Omega_i(t') \equiv s_i^{(\infty)} \tag{7}$$

If the above limit is finite,  $s_i^{(\infty)}$  provides an upper bound for  $s_i$ . The meaning of  $s_i^{(\infty)}$  is clear from Eq. (4). Putting in it  $t = 0$ , we have

$$q_i^*(0) \equiv \lim_{\tau \rightarrow \infty} q_i(\tau | 0) = e^{-s_i^{(\infty)}} \tag{8}$$

i.e.  $\exp[-s_i^{(\infty)}]$  is the probability that, if the system is initially in state  $i$ , it will remain frozen in this state for ever. There is no transition, even in the limit  $t \rightarrow \infty$ . This phenomenon occurs in many physical systems and, in particular, it is responsible of the glassy behavior observed when a system is continuously cooled to low temperatures. According to Eq. (7), the condition for  $s_i^{(\infty)}$  to be finite and, consequently, for the system to become frozen in state  $i$  is

$$\lim_{t \rightarrow \infty} t\Omega_i(t) = 0 \tag{9}$$

This condition depends, in general, on the particular state under consideration, therefore being possible that only a part of the range of states become frozen in a given experiment. Besides, it is clear that no freezing can be obtained if the transition rates are constant, since Eq. (9) can not be fulfilled.

The waiting time distribution in the  $s_i$  scale reads

$$g_i(\sigma_i | s_i) = f_i(\tau | t) \left| \frac{\partial \tau}{\partial \sigma_i} \right| = e^{-\sigma_i} \tag{10}$$

where  $\sigma_i$  is the scaled waiting time, i.e.

$$\sigma_i = \int_t^{t+\tau} dt' \Omega_i(t') \tag{11}$$

Note that, in principle,  $\sigma_i$  depends on both  $t$  and  $\tau$ . The expression given by Eq. (10) holds for values  $\sigma_i$  such that

$$s_i + \sigma_i \equiv \int_0^{t+\tau} dt' \Omega_i(t') \leq s_i^{(\infty)} \tag{12}$$

as seen from Eq. (7). Moreover, the probability that the system be frozen in state  $i$  after arriving to it at time  $t$  will be

$$q_i^*(s_i) = \lim_{\tau \rightarrow \infty} q_i(t + \tau | t) = e^{-(s_i^{(\infty)} - s_i)} \quad (13)$$

Of course, the normalization condition

$$\int_0^{s_i^{(\infty)} - s_i} d\sigma_i g_i(\sigma_i | s_i) + q_i^*(s_i) = 1 \quad (14)$$

is verified. The first term in the above expression is the probability that the system remains any arbitrarily large waiting time in state  $i$ , after having arrived there at time  $t$ . The upper limit in the integral follows from Eq. (12). The second term accounts for the probability of the system getting frozen in state  $i$  at time  $t$ . Let us note that, if  $s_i^{(\infty)}$  is finite,  $q_i^*(s_i)$  approaches unity as  $t \rightarrow \infty$ . This reflects the fact that the probability of the system getting frozen in state  $i$  grows in time, since the transition probability  $\Omega_i(t)$  goes to zero as  $t \rightarrow \infty$  for finite  $s_i^{(\infty)}$ .

The consequence of the above discussion is that the distribution of waiting times for state  $i$  at time  $t$ , including the possibility that the system be frozen in that state for ever, can be generated in the following way. A random number  $x$  uniformly distributed in the interval  $(0, 1)$  is generated. If

$$-\ln x \leq s_i^{(\infty)} - s_i(t) \quad (15)$$

a waiting time  $\tau$  given by

$$-\ln x = \int_t^{t+\tau} dt' \Omega_i(t') = s_i(t + \tau) - s_i(t) \quad (16)$$

is assigned. On the other hand, if

$$-\ln x > s_i^{(\infty)} - s_i(t) \quad (17)$$

the system is considered as frozen in state  $i$ .

### III. MONTE CARLO ALGORITHM

Now we proceed to formulate a Monte Carlo algorithm to solve Eq. (1). It is a generalization of the one introduced by Bortz, Kalos, and Lebowitz<sup>(7)</sup> for master equations with time independent transition rates. Of course, the basic idea is to generate a set of trajectories of the system

distributed according to the stochastic process described by Eq. (1). The results should be exact in the limit of an infinite number of trajectories. In this section we are going to describe the rules defining the algorithm at a intuitive level. A more detailed mathematical justification is given in the Appendix.

We introduce dimension-less transition rates  $P_{ji}(t)$  by

$$P_{ji}(t) = \frac{W_{ji}(t)}{\Omega_i(t)} \tag{18}$$

for  $i \neq j$ . From this definition and Eq. (2) it follows that

$$\sum_{j \neq i} P_{ji}(t) = 1 \tag{19}$$

It is also convenient to define the quantities

$$P_{ji}^{(+)}(t) = \frac{\sum_{k > j, k \neq i} W_{ki}(t)}{\Omega_i(t)} \tag{20}$$

and

$$P_{ji}^{(-)}(t) = \frac{\sum_{k < j, k \neq i} W_{ki}(t)}{\Omega_i(t)} \tag{21}$$

which assume that the states accessible to the system have been ordered following an arbitrary criterion. Eqs. (18)–(21) lead to the relationship

$$P_{ji}^{(+)}(t) + P_{ji}^{(-)}(t) + P_{ji}(t) = 1 \tag{22}$$

In the simulation a trajectory of the system is constructed in the following way:

1. An initial state is generated according to the initial probability distribution  $p_i(0)$ . Suppose that  $i$  is the initial state of the system for the trajectory we are considering. A random number  $x_1$  uniformly distributed in the interval  $(0, 1)$  is generated. If  $x_1 < q_i^*(0)$ , where  $q_i^*(0)$  is given by Eq. (8), the trajectory is finished and the system remains frozen in state  $i$  for ever. Otherwise, a residence time  $\tau_1$  in state  $i$  is assigned by means of Eq. (16) with  $t = 0$ , i.e.

$$-\ln x_1 = \int_0^{\tau_1} dt' \Omega_i(t') \tag{23}$$

and we proceed to the next step.

2. At  $t = \tau_1$ , the system is moved from state  $i$  to state  $j$ . The latter is determined by using that, once we know that a transition from state  $i$  occurs, the relative probability of each of them is given by the reduced transition rates  $P_{ji}(\tau_1)$ . Then, another random number  $y_1$ , again between 0 and 1, is generated and the transition takes place to the state  $j$  for which

$$P_{ji}^{(-)}(\tau_1) < y_1 < 1 - P_{ji}^{(+)}(\tau_1) \quad (24)$$

i.e. the interval  $(0, 1)$  is divided into segments of lengths  $P_{ki}(\tau_1)$ ,  $k = 1, 2, 3, \dots, i-1, i+1, \dots$ , and the transition which is actually realized is the one corresponding to the segment inside which  $y_1$  lies.

3. For  $t = \tau_1^+$ , the system will be in state  $j$  picked out as described in the preceding step. A new random number  $x_2$  is chosen. If  $x_2 < q_j^*(t)$ , the system is considered frozen in state  $j$ , and the simulation process finished, while if  $x_2 \geq q_j^*(t)$  the residence time  $\tau_2$  in state  $j$  is given by

$$-\ln x_2 = \int_{\tau_1}^{\tau_1 + \tau_2} dt' \Omega_j(t') \quad (25)$$

4. Next, step 2 is repeated, now for time  $\tau_1 + \tau_2$ , i.e. the distribution probability for the possible final states  $k$  is given by  $P_{kj}(\tau_1 + \tau_2)$ , with  $k \neq j$ . Once the final state has been determined, step 3 is again applied, and so on.

Unless the time integrals of the total transition rates have a simple analytical expression, the calculation of the residence times, as indicated by Eqs. (23) and (25), can be quite complicated in practice. This would largely increase the computer simulation time. In the next section, we will discuss an efficient and accurate approximation which can be used as long as the transition rates are smooth functions of time.

When the transition rates are time independent, the above prescription for the assignment of residence times reduces to the BKL one.<sup>(7)</sup> For constant  $\Omega_i$ , Eq. (11) reads

$$\sigma_i = \tau \Omega_i \quad (26)$$

so that Eq. (16) is easily solved for the residence time  $\tau$ ,

$$\tau = -\frac{\ln x}{\Omega_i} \quad (27)$$

This is precisely the expression for the waiting times used in ref. 7. Besides, in the basic BKL algorithm, the state to which the transition takes place is determined as described in step 2. However, when there is a great number



of possible transitions from a given state, a simplification can be introduced, because it is usually found that many of the transitions are equiprobable. Then, it is useful to classify the transitions according to their rate and slightly modify the procedure for choosing the transition that is going to be carried out. The situation is similar for time dependent transition rates. The only difference is that they must be up-dated for each transition, as indicated in steps 2 and 4. Since the details of the modification have been discussed in detail by Bortz, Kalos, and Lebowitz they will not be given here. This alternative method for determining the final state of a transition is described and used in the example discussed in Sec. V.

#### IV. AN EFFICIENT SCHEME FOR THE CALCULATION OF WAITING TIMES

The determination of the waiting time  $\tau$  in state  $i$  after arriving to it at time  $t$  requires solving Eq. (16). Here we discuss a numerical method to obtain  $\tau$  which is useful when an analytical expression can not be found. We introduce a discretization of time by defining intervals  $I_n$ , all of them of the same amplitude  $\Delta t$ ,

$$I_n = [t_{n-1}, t_n) \tag{28}$$

with  $t_n = n\Delta t$ ,  $n = 1, 2, 3, \dots$ . The extension of the method to intervals of different amplitude is straightforward. Notice that for the sake of precision the intervals have been defined as open to the right. It is assumed that  $\Delta t$  is chosen such that the total transition rates  $\Omega_i$  can be considered as approximately constant inside each of the intervals. If they are smooth continuous functions of time, this implies that

$$\frac{\partial \Omega_i(t)}{\partial t} \Delta t \ll \Omega_i(t) \tag{29}$$

for all states and times inside the region of interest. Of course, the method also holds if the transitions rates are a succession of steps taking place at regular intervals  $\Delta t$ , becoming then exact. We assign to each interval a constant value of the transition rates, i.e.

$$\Omega_i(t) = \Omega_i^{(n)} \tag{30}$$

for  $t \in I_n$ . There are many possible criteria to fix the value of  $\Omega_i^{(n)}$ . Some of them are:

$$\Omega_i^{(n)} = \Omega_i(t_n) \quad (31a)$$

$$\Omega_i^{(n)} = \Omega_i\left(\frac{t_{n-1} + t_n}{2}\right) \quad (31b)$$

$$\Omega_i^{(n)} = \frac{1}{\Delta t} \int_{t_{n-1}}^{t_n} dt' \Omega_i(t') \quad (31c)$$

In the limit  $\Delta t \rightarrow 0$ , all the above expressions coincide and, therefore, the accuracy of the numerical calculation must not depend on the particular choice used for  $\Omega_i^{(n)}$ , if  $\Delta t$  is small enough.

Let us suppose, without loss of generality, that  $t \in I_n$ . It is

$$\int_t^{t+\tau} dt' \Omega_i(t') = \tau \Omega_i^{(n)} \Theta(t_n - t - \tau) + \left[ (t_n - t) \Omega_i^{(n)} + \int_{t_n}^{t+\tau} dt' \Omega_i(t') \right] \Theta(t + \tau - t_n) \quad (32)$$

where  $\Theta(t)$  is the Heaviside step function defined by  $\Theta(t) = 0$  for  $t < 0$ ,  $\Theta(t) = 1$  for  $t > 0$ , and  $\Theta(0) = 1/2$ . We can continue the process and by iteration get

$$\begin{aligned} & \int_t^{t+\tau} dt' \Omega_i(t') \\ &= \tau \Omega_i^{(n)} \Theta(t_n - t - \tau) + (t_n - t) \Omega_i^{(n)} \Theta(t + \tau - t_n) \\ & \quad + \Theta(t + \tau - t_n) \left\{ (t + \tau - t_n) \Omega_i^{(n+1)} \Theta(t_{n+1} - t - \tau) \right. \\ & \quad \left. + \left[ \Delta t \Omega_i^{(n+1)} + \int_{t_{n+1}}^{t+\tau} dt' \Omega_i(t') \right] \Theta(t + \tau - t_{n+1}) \right\} \\ &= \dots \\ &= \tau \Omega_i^{(n)} \Theta(t_n - t - \tau) + (t_n - t) \Omega_i^{(n)} \Theta(t + \tau - t_n) \\ & \quad + \sum_{m=0}^{m_0} (t + \tau - t_{n+m}) \Omega_i^{(n+m+1)} \Theta(t + \tau - t_{n+m}) \Theta(t_{n+m+1} - t - \tau) \\ & \quad + \sum_{m=0}^{m_0} \Delta t \Omega_i^{(n+m+1)} \Theta(t + \tau - t_{n+m+1}) \\ & \quad + \Theta(t + \tau - t_{n+m_0+1}) \int_{t_{n+m_0+1}}^{t+\tau} dt' \Omega_i(t') \end{aligned} \quad (33)$$

This equation, although apparently complicated, directly suggests a method to compute  $\tau$  given  $t$  and the value of the integral, i.e.  $-\ln x$  [see Eq. (16)]. The key point is to realize that the contribution to the integral from each time interval  $I_{n+m}$  is proportional to  $\Omega_i^{(n+m)}$ . First, we define a number  $u_0$  as

$$u_0 = -\frac{\ln x}{\Omega_i^{(n)}} \tag{34}$$

If  $u_0 \leq t_n - t$  we take  $\tau = u_0$ . Otherwise, we define

$$u_1 = [u_0 - (t_n - t)] \frac{\Omega_i^{(n)}}{\Omega_i^{(n+1)}} \tag{35}$$

The meaning of the above prescription can be easily understood as sketched in Fig. 1. Equation (34) defines the length of the base of a rectangle

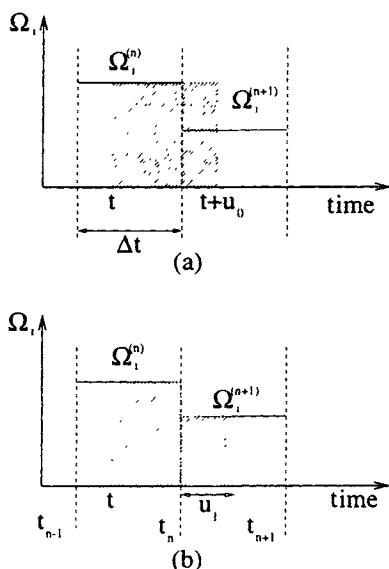


Fig. 1. Sketch of a waiting time assignment in which the rescaling procedure is necessary. In the time intervals  $I_n$  and  $I_{n+1}$  the value of the involved transition rate  $\Omega_i$  is equal to  $\Omega_i^{(n)}$  and  $\Omega_i^{(n+1)}$ , respectively. The shadowed areas of parts (a) and (b) are both equal to the value of the integral, i.e.  $-\ln x$ . In part (a) the quantity  $u_0$ , given by Eq. (34), is such that the final time  $t + u_0$  belongs to  $I_{n+1}$ . Therefore, the “excess” of area outside  $I_n$  is given by  $[u_0 - (t_n - t)] \Omega_i^{(n)}$ . In part (b) the quantity  $u_1$  is the length of the base of the rectangle of height  $\Omega_i^{(n+1)}$  with the same area. As  $u_1$  is smaller than  $\Delta t$ , the iterative method would have finished, and the waiting time is given by Eq. (36).

of height  $\Omega_i^{(n)}$  having the same area as the given value of the integral. Nevertheless, the maximum area that can be accounted for the interval  $I_n$  is  $(t_n - t) \Omega_i^{(n)}$ . If this quantity is smaller than the value of the integral, there is an "excess" of area  $[u_0 - (t_n - t)] \Omega_i^{(n)}$ , which must be assigned to the following intervals. To find out how much of the next interval is needed we have to divide this area by the value of the function  $\Omega_i^{(n+1)}$ . This defines the quantity  $u_1$ .

If now  $u_1 \leq \Delta t$ , as it is the case of Fig. 1, we have

$$\tau = (t_n - t) + u_1 \quad (36)$$

while if  $u_1 > \Delta t$  we have to consider the value of  $\Omega_i(t)$  in the next interval  $I_{n+2}$ . Again, the excess of area  $(u_1 - \Delta t) \Omega_i^{(n+1)}$  is divided by  $\Omega_i^{(n+2)}$ , introducing

$$u_2 = (u_1 - \Delta t) \frac{\Omega_i^{(n+1)}}{\Omega_i^{(n+2)}} \quad (37)$$

If  $u_2 \leq \Delta t$  it is

$$\tau = (t_{n+1} - t) + u_2 \quad (38)$$

while if  $u_2 > \Delta t$  a quantity  $u_3$  is defined, and so on. In general,  $u_f$  for  $f > 1$  is generated through the recursive relation

$$u_f = (u_{f-1} - \Delta t) \frac{\Omega_i^{(n+f-1)}}{\Omega_i^{(n+f)}} \quad (39)$$

and the iterative procedure is stopped when a value  $u_l \leq \Delta t$  is obtained, and

$$\tau = (t_{n+l-1} - t) + u_l, \quad (l \geq 1) \quad (40)$$

Thus the final time  $t + \tau$  belongs to the interval  $I_{n+l}$ , while  $t$  belonged to  $I_n$ . In general, the residence time is then given by

$$\tau = u_l + (1 - \delta_{l,0})(t_{n+l-1} - t) \quad (41)$$

It is easy to check analytically that the above value of  $\tau$  is really the solution of Eq. (16), when the transition rates are considered as constant in each of the intervals. For  $\tau$  given by Eq. (41) we have

$$\begin{aligned}
 \int_t^{t+\tau} dt' \Omega_i^{(n)}(t') &= \Omega_i^{(n)}(t_n - t) + \Omega_i^{(n+1)} \Delta t + \dots + \Omega_i^{(n+l-1)} \Delta t + \Omega_i^{(n+l)} u_l \\
 &= \Omega_i^{(n)}(t_n - t) + \Omega_i^{(n+1)} \Delta t + \dots + u_{l-1} \Omega_i^{(n+l-1)} \\
 &= \dots \\
 &= \Omega_i^{(n)}(t_n - t) + u_1 \Omega_i^{(n+1)} = u_0 \Omega_i^{(n)} = -\ln x \tag{42}
 \end{aligned}$$

where use has been made of Eqs. (34) and (39). In the above calculation we have considered  $l > 1$ , but the cases  $l = 0$  and  $l = 1$  are trivial.

The numerical algorithm for the evaluation of  $\tau$  can be formulated in a slightly different, although equivalent, form, which is simpler to implement in actual simulations. We start from

$$\tau^{(0)} = -\frac{\ln x}{\Omega_i^{(n)}} \tag{43}$$

and generate the succession ( $f \geq 1$ )

$$t + \tau^{(f)} - t_{n+f-1} = \frac{\Omega_i^{(n+f-1)}}{\Omega_i^{(n+f)}} (t + \tau^{(f-1)} - t_{n+f-1}) \tag{44}$$

The iteration is concluded when a value  $t + \tau^{(l)} < t_{n+l}$  is obtained, and the waiting time  $\tau$  is given by  $\tau^{(l)}$ .

In the preceding discussion we have supposed that a quantity  $u_l < \Delta t$  is reached for large enough  $l$ . Nevertheless, we have seen in Sec. II that such  $u_l$  might not exist when the limit in Eq. (7) converges, i.e. if in the discretized approximation it is

$$\lim_{n \rightarrow \infty} n \Omega_i^{(n)} = 0 \tag{45}$$

When a value  $u_l < \Delta t$  can not be obtained, even in the limit  $l \rightarrow \infty$ , the system is frozen in state  $i$ . In practice, a lot of simulation time may be needed to verify whether the system is frozen, because it would require to extrapolate to  $t \rightarrow \infty$ . Therefore, it is more efficient to proceed as follows. A maximum simulation time,  $t_{sim}$ , is fixed from the beginning, and whenever a waiting time  $\tau$  is obtained such that  $t + \tau > t_{sim}$ , the trajectory of the system is stopped, with the system staying in the same state it was at time  $t$ . This determines the evolution up to time  $t_{sim}$ , although no identification of the frozen trajectories is made. In this way, the parameters  $s_i^{(\infty)}$  defined by Eq. (24) play no explicit role in the simulation of the system.

Let us emphasize that the time discretization saves much computer time, not only because of the simpler evaluation of the waiting time at each state of the system, but also because the time intervals are fixed, and the transition rates associate to each of them are constant along the simulation. Therefore, they do not need to be reevaluated every time there is a transition in the system.

## V. APPLICATION TO THE ONE-DIMENSIONAL ISING MODEL

Here we are going to apply the GWT algorithm developed in the previous sections to the one-dimensional Ising model with nearest neighbor interactions. The Hamiltonian is

$$\mathcal{H} = -J \sum_{j=-\infty}^{+\infty} \sigma_j \sigma_{j+1} \quad (46)$$

where  $J > 0$ , and  $\sigma_j = \pm 1$ . The dynamics of the model is given by single-spin-flip Glauber dynamics.<sup>(10)</sup> Thus the probability  $p(\sigma, t)$  of finding the system in state  $\sigma$  satisfies the master equation

$$\frac{d}{dt} p(\sigma, t) = \sum_{j=-\infty}^{+\infty} [W_j(R_j\sigma) p(R_j\sigma, t) - W_j(\sigma) p(\sigma, t)] \quad (47)$$

where  $R_j\sigma$  is the configuration obtained from  $\sigma$  by flipping the  $j$ th spin, i.e. by changing  $\sigma_j$  into  $-\sigma_j$ . The transition rates are given by

$$W_j(\sigma; T) = \frac{\alpha(T)}{2} \left[ 1 - \frac{\gamma(T)}{2} \sigma_j (\sigma_{j-1} + \sigma_{j+1}) \right] \quad (48)$$

with

$$\gamma(T) = \tanh \frac{2J}{k_B T} \quad (49)$$

This choice of the transition rates guarantees that they verify the detailed balance condition for arbitrary  $\alpha(T)$ . A reasonable physical choice for this function is<sup>(11)</sup>

$$\alpha(T) = \alpha_0 \exp \left( -\frac{B}{k_B T} \right) \quad (50)$$

which introduces an activation energy barrier  $B$  that is to be surmounted by any spin in order to flip. The constant  $\alpha_0$  will be used to set up the time scale and, therefore, it will be taken equal to unity in the following.

Attention will be focused on the behavior of the adimensional average energy per particle,

$$e = -\langle \sigma_j \sigma_{j+1} \rangle \quad (51)$$

when the system is submitted to a thermal cycle, i.e. it is cooled down to low temperatures, and afterwards reheated. The equilibrium value of the energy is<sup>(10)</sup>

$$e_0 = -\tanh \frac{J}{k_B T} \quad (52)$$

Along the thermal cycle the transition rates, given by Eq. (48), depend on time through the temperature, which is assumed to be externally controlled. An analytical study of this system can be found in ref. 11. Here we are going to consider that the temperature varies linearly in time with a rate  $\rho$ , i.e.

$$\frac{dT}{dt} = \mp \rho \quad (53)$$

where the minus and plus signs correspond to the cooling and heating processes, respectively.

We have performed Monte Carlo simulation of the system, using the GWT algorithm. We have considered a lattice of  $N = 500$  spins with periodic boundary conditions. The external barrier has been taken  $B = 4J$ , as in ref. 11. We start the cooling program from a temperature  $T_0$ . First, we put the system in the all-spin-up state, and let it relax towards equilibrium at constant temperature  $T_0$ . Once equilibrium is reached, time is initialized for each run and the cooling program is started. The cooling program is finished when the energy reaches a constant value at low temperatures. This defines in a natural way the maximum time  $t_{sim}$  as the time corresponding to the lowest temperature during the cooling process. As the temperature at which the system departs from equilibrium is an increasing function of  $\rho$ ,<sup>(11)</sup> the initial temperature  $T_0$  of the cycle should be lowered with decreasing  $\rho$ , so as not to spend too much time over the equilibrium curve. Then the system is reheated, until the initial temperature  $T_0$  is recovered.

As the transition rates depend on time through the temperature, the piecewise approximation consists in considering them constant along intervals of small width  $\Delta T$ . A reasonable value of this parameter is  $k_B \Delta T / J = 0.01$ , which we have used in our simulation. We have checked that smaller values of  $\Delta T$  lead to the same results. Since the temperature

varies linearly in time with a rate  $\rho$ , the width  $\Delta t$  of the intervals  $I_n$  defined by Eq. (30) is  $\Delta t = \Delta T / \rho$ . We have chosen for the transition rates the approximation in Eq. (31a), i.e. we gave them the values corresponding to the final temperature of the interval.

Besides, we have taken advantage of the  $n$ -fold way procedure introduced by Bortz, Kalos and Lebowitz.<sup>(7)</sup> Let us consider a given configuration of spins  $\sigma$ . The possible transitions consist in the flip of one spin, and they have transition rates depending only on the flipping spin and its two nearest neighbors, through the combination

$$\lambda_i = \frac{1}{2} \sigma_i (\sigma_{i-1} + \sigma_{i+1}) \quad (54)$$

If we introduce

$$W_\lambda(T) = \alpha(T) [1 - \lambda\gamma(T)] \quad (55)$$

the transition rate for the  $i$ th spin can be expressed as

$$W_i(\sigma; T) = W_{\lambda=\lambda_i}(T) \quad (56)$$

Thus we classify the spins in our system in three groups, depending on the two neighbors being (a) both parallel to the considered spin,  $\lambda_i = 1$ , (b) one parallel and one antiparallel,  $\lambda_i = 0$ , or (c) both antiparallel,  $\lambda_i = -1$ . All the spins belonging to the same class have the same transition rate, as indicated in Fig. 2.

We introduce two arrays LOOK( $N$ ) and IBLOQ( $N$ ). The array IBLOQ has the spins ordered according to their class. Let  $n_\lambda$  be the number of spins in class  $\lambda$  and  $m_\lambda$  the number of spins whose class number is less than  $\lambda$ . Then IBLOQ( $m_\lambda + 1$ ) through IBLOQ( $m_{\lambda+1}$ ) contains the spins of class  $\lambda$ . The second array LOOK gives the position of the spins in

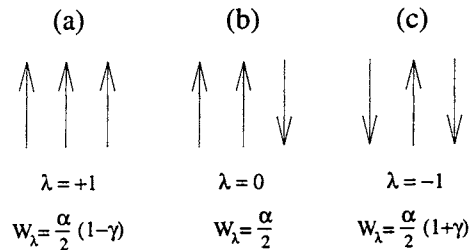


Fig. 2. The three classes of spins in the one-dimensional Ising model, attending to the value of its transition rate. The index of the class and the corresponding value of the transition rate are also shown.



IBLOQ, i.e. if  $\text{IBLOQ}(i) = j$  it is  $\text{LOOK}(j) = i$ . Both arrays are necessary because when the  $i$ th spin flips not only its class changes, but also the class of its nearest neighbors. These changes can be summarized in two simple rules: (1) the class of the  $i$ th spin reverses its sign. Of course, class  $\lambda = 0$  is invariant. (2) The class of the adjacent spins changes by  $+1$  or  $-1$  depending whether they are antiparallel or parallel to the flipping one. The global transition rate from a given configuration  $\sigma$  is given by

$$\Omega(\sigma; T) = \sum_i W_i(\sigma; T) = \sum_\lambda n_\lambda W_\lambda(T) \quad (57)$$

To be concrete, let us consider the cooling process. According to the general GWT algorithm, the steps to generate a trajectory starting from a given configuration  $\sigma$  are:

1. We evaluate  $\Omega(\sigma; T_1)$ , where  $T_1$  is the temperature in the first time interval,  $T_1 = T_0 - \rho \Delta t$ . A waiting time  $\tau_1$  is assigned according to the method discussed in the previous section, i.e.

$$\tau_1^{(0)} = -\frac{\ln x_1}{\Omega(\sigma; T_1)} \quad (58)$$

This time must be rescaled by the procedure given by Eq. (44) when it exceeds  $\Delta t$ , giving rise to the waiting time  $\tau_1$  in the configuration  $\sigma$ . Of course, if the waiting time  $\tau_1$  exceeds the maximum time  $t_{\text{sim}}$  the trajectory is finished.

2. The transition made by the system at  $t = \tau_1$  is selected in two steps. The idea is similar to the constant temperature case,<sup>(7)</sup> but we have to update the transition rates if  $t > \Delta t$ . Let us suppose that  $t$  belongs to the interval  $I_n$ , and then  $T = T_n$ . A class of spins  $\lambda$ , with probability

$$P_\lambda(T_n) = \frac{n_\lambda W_\lambda(T_n)}{\Omega(\sigma; T_n)} \quad (59)$$

is chosen drawing a random number in the interval  $(0, 1)$ .

Afterwards, the spin of this class which actually flips is taken at random, with a random integer  $q_1$  uniformly distributed in the interval  $[1, n_\lambda]$ . The particular spin which flips is  $i = \text{IBLOQ}(m_\lambda + q_1)$ . Thus we have to rearrange the two arrays IBLOQ and LOOK, reversing the sign of the class of the  $i$ th spin, without varying the class of the remainder. Besides, the class of the nearest neighbors also changes as indicated above. In order to do this, we look for the positions of the neighbors in IBLOQ, that are given by  $\text{LOOK}(i \pm 1)$ , and make the proper rearrangements. It is important to remark that whenever IBLOQ is reordered we have to change

LOOK, according to its definition. Of course,  $n_\lambda$  and  $m_\lambda$  also change after each transition. From here the algorithm proceed as discussed in Sec. III.

In Table I we compare the efficiency of the proposed GWT algorithm, measured by CPU time, with the usual implementation of Metropolis. In the Metropolis program we have performed  $\Delta t$  Monte Carlo steps (MCS) at each temperature  $T_n$ , before changing the temperature by an amount  $\pm \Delta T$ . This simulates a process with a rate of variation of the temperature  $\rho$ . We have used the Berkeley random number generator (RANF) for both algorithms and carried out the simulation in a Digital DEC-3000 AXP workstation. For the interval of values of the cooling rate under consideration, the GWT algorithm is faster than the Metropolis one, the ratios of the respective CPU time ranging from 1.5 to 30, approximately. This ratio increases as the rate  $\rho$  decreases. Of course, its specific value may depend on the random number generator and the computer used. The rates considered are not artificially small, but laying in the range usually considered in the literature.<sup>(12, 13)</sup>

As an example, in Fig. 3 a hysteresis cycle is plotted. Both the numerical results from the GWT Monte Carlo simulation and the analytical expressions in ref. 11 are shown. It is seen that they agree, showing that the GWT algorithm leads to the correct result. Of course, by means of the Metropolis procedure the curves are practically the same. It must be noticed that the values of the rate  $\rho$  are somehow restricted in the Metropolis algorithm. The temperature steps  $\Delta T$  must be small in order to can consider the transition rates accurately constant in each of the intervals. On the other hand, the corresponding time discretization  $\Delta t$  cannot be smaller than one MCS. Therefore it is not possible to have arbitrarily large

**Table I. Comparison of CPU Time (in seconds) for the Metropolis and GWT Algorithms, along 10 Runs<sup>a</sup>**

$k_B T_0/J$	$k_B \rho/J$	Metropolis	GWT
4.0	$10^{-2}$	24.6	16.2
3.0	$10^{-3}$	169.2	59.3
2.0	$10^{-4}$	1037.5	110.4
1.5	$10^{-5}$	7607.5	234.8

<sup>a</sup> A very good average is obtained with 1000 trajectories. Both the initial value and the rate of variation of the temperature in the thermal cycle are also indicated. The efficiency of the GWT algorithm increases when  $\rho$  decreases, since the cycle reaches lower temperatures and waiting times are larger.

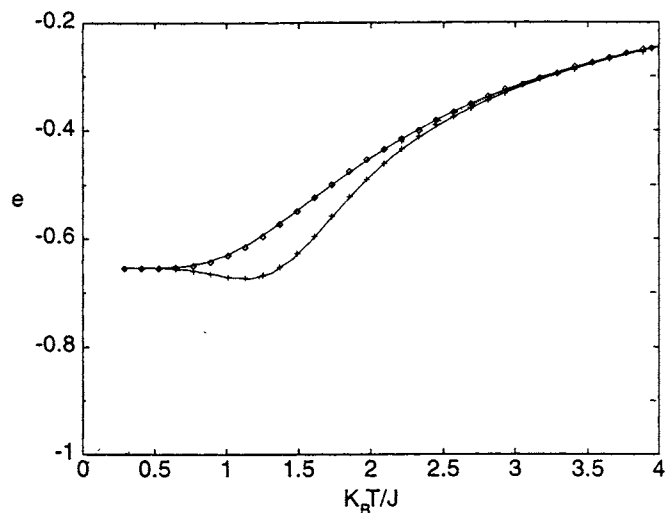


Fig. 3. Comparison of the thermal cycles of energy obtained from GWT Monte Carlo simulation (diamonds) and the evaluation of the analytical expressions in ref. 11 (solid line). The law of variation of the temperature is given by Eq. (53), with a rate  $k_B \rho/J = 10^{-2}$ . The dotted line is the equilibrium energy.

rates  $\rho$ , while the proposed GWT algorithm is free of this restriction. The temperature step  $\Delta T$  must be also small, but it is possible to take an arbitrarily small time discretization  $\Delta t$ .

## VI. CONCLUSIONS

We have developed a Monte Carlo algorithm for simulating model systems whose dynamics is described in terms of master equations with time-dependent transition rates. The algorithm is based on a waiting time description, i.e. each realization of the stochastic process is obtained by giving to the system a waiting time  $\tau_1$  in the initial state  $i$ . At  $t = \tau_1$  it goes to another state  $j$ , where it remains a waiting time  $\tau_2$ , etc. This GWT algorithm reduces to the well-known BKL procedure when the transition rates are time independent, and therefore it can be considered as its “generalization” to the case of time dependent transition rates.

As compared with the Metropolis algorithm this approach has some advantages, both from a theoretical and a practical point of view. First, as it is shown in the Appendix, this waiting time description appears naturally when the master equation is transformed into an integral equation. Besides, the GWT algorithm gives relevant information about the peculiar

dynamics of systems described by master equations with time-dependent transition rates. When the transition rates vanish in the long time limit, the algorithm is able to identify which trajectories are frozen at a given time  $t$ , i.e. those realizations which would have no more transitions even after waiting an infinite time. This may be used to measure the loss of ergodicity of the system, leading to the departure from equilibrium and to phenomena related with a glass-like transition.

The exact calculation of waiting times is rather involved in general, and an accurate and efficient procedure is discussed in Sec. IV. It is based on approximating the transition rates by piecewise constant functions through time intervals of small width  $\Delta t$ . Afterwards, an iterative method is shown to generate the correct waiting time within the above approximation, becoming exact in the limit  $\Delta t \rightarrow 0$ . Besides, the procedure is easily implemented in the computer.

We have applied the GWT algorithm to the one-dimensional Ising model with Glauber dynamics. Attention has been focused on the evolution of the energy along thermal cycles of cooling and reheating. The temperature evolves linearly with time, being  $\rho$  its rate of variation. In order to evaluate the waiting times with the iterative procedure of Sec. IV, the temperature is approximated by a piecewise constant function through intervals of width  $\Delta T$ . The results from the GWT algorithm agree with the analytical expressions obtained in ref. 11. Besides, for the interval of values of  $\rho$  considered, the GWT simulation is faster than the corresponding implementation of Metropolis, in which the temperature is kept constant for  $\Delta t = \Delta T/\rho$  MCS steps, after which the temperature is varied by  $\pm \Delta T$ .

In other systems, the GWT algorithm is also expected to be faster than the Metropolis one under the adequate conditions, when the waiting times are large. This happens when the transition rates are small, as it is the case in systems displaying a laboratory glass-like transition when cooled up to low temperatures. An example can be seen in ref. 9. Thus one step of the GWT algorithm corresponds to many MCS in Metropolis, making the former more efficient. This is not surprising, since the same situation shows up when the BKL algorithm is compared to the Metropolis one for time independent rates.

## APPENDIX A: THEORETICAL DERIVATION

Here we are going to show that the GWT Monte Carlo algorithm presented in Sec. III provides a numerical solution of the master equation. It is easy to show that Eq. (1), with an arbitrary initial condition

$$p_i(t=0) = p_i(0) \tag{A1}$$

is equivalent to the integral equation

$$p_i(t) = p_i(0) e^{-\int_0^t dt' \Omega_i(t')} + \sum_{j \neq i} \int_0^t dt_1 W_{ij}(t_1) p_j(t_1) e^{-\int_1^t dt' \Omega_i(t')} \quad (A2)$$

Our starting point will be the above integral representation of the master equation. It is convenient for the proof to introduce an iterative solution,

$$\begin{aligned} p_i(t) &= p_i(0) e^{-\int_0^t dt' \Omega_i(t')} \\ &+ \sum_{j \neq i} p_j(0) \int_0^t dt_1 e^{-\int_0^{t_1} dt' \Omega_j(t')} W_{ij}(t_1) e^{-\int_1^t dt' \Omega_i(t')} \\ &+ \sum_{k \neq j} \sum_{j \neq i} p_k(0) \int_0^t dt_1 \int_0^{t_1} dt_2 \\ &\times e^{-\int_0^{t_2} dt' \Omega_k(t')} W_{jk}(t_2) e^{-\int_2^{t_1} dt' \Omega_j(t')} W_{ij}(t_1) e^{-\int_1^t dt' \Omega_i(t')} + \dots \\ &= p_i^{(0)}(t) + p_i^{(1)}(t) + p_i^{(2)}(t) + \dots \end{aligned} \quad (A3)$$

obtained by successive substitution of the integral equation (A2) on its right hand side. Eq. (A3) for the probability  $p_i(t)$  of finding the system in state  $i$  at time  $t$  can be understood in a simple way. The first term,  $p_i^{(0)}$ , accounts for the probability that the system remains in state  $i$  at time  $t$  because there has been no transitions in the time interval  $(0, t)$ . The second term,  $p_i^{(1)}$ , gives the contribution corresponding to only one transition in the time interval  $(0, t)$ , the system stays in state  $j$  until an instant  $t_1$ , at which it goes into state  $i$ , remaining afterwards there until  $t$ . In general, the term  $p_i^{(v)}$  describes a “trajectory” with  $v$  transitions, the last one ending in state  $i$ .

Now we are going to prove that the proposed GWT algorithm simulate the master equation, concretely it leads to the iterative solution, Eq. (A3). In our numerical simulation we will have  $N_T$  trajectories of the system, indexed by a parameter  $\alpha = 1, \dots, N_T$ . Let us define

$$n_{i\alpha}(t) = \begin{cases} 1 & \text{if } \alpha\text{th run is in state } i \text{ at time } t, \\ 0 & \text{any other case} \end{cases} \quad (A4)$$

The total number of trajectories in state  $i$  at time  $t$  is

$$N_i(t) = \sum_{\alpha=1}^{N_T} n_{i\alpha}(t) \quad (A5)$$

and the probability of finding the system in state  $i$  at time  $t$  is given by

$$p_i(t) = \frac{N_i(t)}{N_T} \quad (\text{A6})$$

in the limit of a large number of trajectories, formally  $N_T \rightarrow \infty$ . In order to make compatible Eq. (A6) with the initial condition  $p_i(0)$ , we must impose

$$p_i(0) = \frac{N_i(0)}{N_T} \quad (\text{A7})$$

We write the number of trajectories in state  $i$  as

$$N_i(t) = \sum_{\nu=0}^{\infty} N_i^{(\nu)}(t) \quad (\text{A8})$$

where  $N_i^{(\nu)}(t)$  is the number of runs in state  $i$  at time  $t$ , after  $\nu$  transitions in the time interval  $(0, t)$ . We are going to show that the series in Eq. (A8) coincides with the iterative solution of the master equation of Eq. (A3), i.e. each term  $N_i^{(\nu)}$  leads to  $p_i^{(\nu)}$  in the limit of a large number of trajectories. Taking into account Eqs. (A4)–(A5), we have

$$N_i^{(\nu)}(t) = \sum_{\alpha=1}^{N_T} n_{i\alpha}^{(\nu)}(t) \quad (\text{A9})$$

where  $n_{i\alpha}^{(\nu)}$  is defined as  $n_{i\alpha}$ , but restricted to  $\nu$  transitions in the time interval  $(0, t)$ .

Let us begin writing  $n_{i\alpha}^{(0)}$  for the  $\alpha$ th run. By definition, there is no transition in the time interval  $(0, t)$ . According to step 1 of the proposed GWT algorithm, it will be equal to unity if  $x_{1\alpha} < q_i^*(0)$  (the trajectory is frozen) or if the waiting time  $\tau_{1\alpha} > t$ , when  $x_{1\alpha} \geq q_i^*(0)$ . Therefore,

$$n_{i\alpha}^{(0)}(t) = n_{i\alpha}(0) \{ \Theta[q_i^*(0) - x_{1\alpha}] + \Theta[x_{1\alpha} - q_i^*(0)] \Theta(\tau_{1\alpha} - t) \} \quad (\text{A10})$$

where  $\Theta(x)$  is Heaviside's step function. We have written the index of the trajectory  $\alpha$  in the random variables explicitly to remark that they are different for each run. From Eq. (A9), particularized for  $\nu=0$ , we obtain the number of realizations in state  $i$  at time  $t$  with no transitions,

$$N_i^{(0)}(t) = \sum_{\alpha=1}^{N_T} n_{i\alpha}(0) \{ \Theta[q_i^*(0) - x_{1\alpha}] + \Theta[x_{1\alpha} - q_i^*(0)] \Theta(\tau_{1\alpha} - t) \} \quad (\text{A11})$$

In the limit of a large (infinite) number of realizations  $N_T$ , we will obtain the average of the above expression over the random variable  $x_1$ , uniformly distributed in the interval  $(0, 1)$ ,

$$N_i^{(0)}(t) = N_i(0) A_i(t) \tag{A12}$$

where  $A_i(t)$  is the average of the expression in brackets of Eq. (A11),

$$\begin{aligned} A_i(t) &= \langle \Theta[q_i^*(0) - x_1] + \Theta[x_1 - q_i^*(0)] \Theta(\tau_1 - t) \rangle \\ &= \int_0^1 dx_1 \{ \Theta[q_i^*(0) - x_1] + \Theta[x_1 - q_i^*(0)] \Theta(\tau_1 - t) \} \\ &= q_i^*(0) + \int_{q_i^*(0)}^1 dx_1 \Theta(\tau_1 - t) \end{aligned} \tag{A13}$$

Due to the relationship between  $\tau_1$  and  $x_1$ , Eq. (16), it is easy to evaluate the integral, obtaining

$$A_i(t) = e^{-\int_0^t dt' \Omega_i(t')} \tag{A14}$$

which introduced into Eq. (A12), leads to

$$N_i^{(0)}(t) = N_i(0) e^{-\int_0^t dt' \Omega_i(t')} \tag{A15}$$

By using Eqs. (A6)–(A7) we arrive at the simulation result for the contribution to  $p_i(t)$  with no transitions

$$p_i^{(0)}(t) = p_i(0) e^{-\int_0^t dt' \Omega_i(t')} \tag{A16}$$

in agreement with the *zero*-th order term in the iterative solution of the master equation, Eq. (A3).

Let us consider now the case of only one transition in the time interval  $(0, t)$ . For the  $\alpha$ th realization we have, taking into account steps 1–3 of the algorithm,

$$\begin{aligned} n_{i\alpha}^{(1)}(t) &= \sum_{j \neq i} n_{j\alpha}(0) \Theta[x_{1\alpha} - q_j^*(0)] \Theta(t - \tau_{1\alpha}) \Theta[y_{1\alpha} - P_{ij}^{(-)}(\tau_{1\alpha})] \\ &\quad \times \Theta[1 - P_{ij}^{(+)}(\tau_{1\alpha}) - y_{1\alpha}] \\ &\quad \times \{ \Theta[q_i^*(\tau_{1\alpha}) - x_{2\alpha}] + \Theta[x_{2\alpha} - q_i^*(\tau_{1\alpha})] \Theta(\tau_{1\alpha} + \tau_{2\alpha} - t) \} \end{aligned} \tag{A17}$$

The system starts in a state  $j \neq i$ , the first two Heaviside's functions account for the condition that the first transition time  $\tau_{1\alpha}$  belongs to the interval  $(0, t)$ , the following two select that the final state of the transition at  $t = \tau_{1\alpha}$

is state  $i$ , and the last three (in brackets) account for the second transition time  $\tau_{1\alpha} + \tau_{2\alpha}$  being greater than the considered time  $t$  or the system being frozen for ever in state  $i$ .

The total number of runs in state  $i$  at time  $t$ , with only one transition in the time interval  $(0, t)$  is

$$N_i^{(1)}(t) = \sum_{\alpha=1}^{N_T} n_{i\alpha}^{(1)}(t) \quad (\text{A18})$$

and in the limit  $N_T \rightarrow \infty$  we obtain the average of the product of Heaviside's functions in Eq. (A17) over the random variables  $x_1$ ,  $x_2$  and  $y_1$ , uniformly distributed in  $(0, 1)$ . Therefore,

$$N_i^{(1)}(t) = \sum_{j \neq i} N_j(0) B_{ij}(t) \quad (\text{A19})$$

with

$$\begin{aligned} B_{ij}(t) &= \langle \Theta[x_1 - q_j^*(0)] \Theta(t - \tau_1) \Theta[y_1 - P_{ij}^{(-)}(\tau_1)] \Theta[1 - P_{ij}^{(+)}(\tau_1) - y_1] \\ &\quad \times \{ \Theta[q_i^*(\tau_1) - x_2] + \Theta[x_2 - q_i^*(\tau_1)] \Theta(\tau_1 + \tau_2 - t) \} \rangle \\ &= \int_0^1 dx_1 \int_0^1 dy_1 \int_0^1 dx_2 \Theta[x_1 - q_j^*(0)] \Theta(t - \tau_1) \Theta[y_1 - P_{ij}^{(-)}(\tau_1)] \\ &\quad \times \Theta[1 - P_{ij}^{(+)}(\tau_1) - y_1] \\ &\quad \times \{ \Theta[q_i^*(\tau_1) - x_2] + \Theta[x_2 - q_i^*(\tau_1)] \Theta(\tau_1 + \tau_2 - t) \} \end{aligned} \quad (\text{A20})$$

The integral over  $x_2$  is similar to the one performed in Eq. (A13), yielding

$$\int_0^1 dx_2 \{ \Theta[q_i^*(\tau_1) - x_2] + \Theta[x_2 - q_i^*(\tau_1)] \Theta(\tau_1 + \tau_2 - t) \} = e^{-\int_{\tau_1}^t dt' \Omega_i(t')} \quad (\text{A21})$$

It is straightforward to carry out the integration over  $y_1$ ,

$$\begin{aligned} &\int_0^1 dy_1 \Theta[y_1 - P_{ij}^{(-)}(\tau_1)] \Theta[1 - P_{ij}^{(+)}(\tau_1) - y_1] \\ &= 1 - P_{ij}^{(-)}(\tau_1) - P_{ij}^{(+)}(\tau_1) = P_{ij}(\tau_1) \end{aligned} \quad (\text{A22})$$

where we have made use of Eq. (22). Therefore, coming back to Eq. (A20),

$$B_{ij}(t) = \int_{q_j^*(0)}^1 dx_1 \Theta(t - \tau_1) P_{ij}(\tau_1) e^{-\int_{\tau_1}^t dt' \Omega_i(t')} \quad (\text{A23})$$



and taking into account Eq. (16) to change the integration variable,

$$\begin{aligned} B_{ij}(t) &= \int_0^\infty d\tau_1 \Omega_j(\tau_1) e^{-\int_0^{\tau_1} dt' \Omega_j(t')} \Theta(t - \tau_1) P_{ij}(\tau_1) e^{-\int_{\tau_1}^t dt' \Omega_j(t')} \\ &= \int_0^t d\tau_1 e^{-\int_0^{\tau_1} dt' \Omega_j(t')} W_{ij}(\tau_1) e^{-\int_{\tau_1}^t dt' \Omega_j(t')} \end{aligned} \quad (\text{A24})$$

Thus, by using Eq. (A19), together with Eq. (A7), we arrive at the simulation value for the probability of finding the system in state  $i$ , with only one transition in the time interval  $(0, t)$ ,

$$\begin{aligned} p_i^{(1)}(t) &= \sum_{j \neq i} p_j(0) B_{ij}(t) \\ &= \sum_{j \neq i} p_j(0) \int_0^t d\tau_1 e^{-\int_0^{\tau_1} dt' \Omega_j(t')} W_{ij}(\tau_1) e^{-\int_{\tau_1}^t dt' \Omega_j(t')} \end{aligned} \quad (\text{A25})$$

This expression agrees with the corresponding term in the iterative solution of the master equation, Eq. (A3).

In a similar way, all the terms  $p_i^{(v)}$  from the simulation can be built and it can be shown that they coincide with the result from the iterative solution. Aside from notation problems, the proof is not different from the case  $v = 1$ , and it will not be given here.

## ACKNOWLEDGMENTS

We want to thank J. Casado-Pascual and D. Cubero for helpful discussions. Partial support from Dirección General de Investigación Científica y Técnica through Grant No. PB96-0534 is gratefully acknowledged.

## REFERENCES

1. S. A. Brawer, *J. Chem. Phys.* **81**:149 (1988); G. H. Fredrickson and H. C. Andersen, *Phys. Rev. Lett.* **53**:1244 (1984); D. A. Huse and D. S. Fisher, *Phys. Rev. Lett.* **57**:2203 (1986); J. C. Dyre, *Phys. Rev. Lett.* **58**:792 (1987); R. Schilling, *J. Stat. Phys.* **53**:1227 (1988); J. J. Brey and M. J. Ruiz-Montero, *Phys. Rev. B* **43**:585 (1991).
2. J. J. Brey, A. Prados, and M. J. Ruiz-Montero, *J. Non-Cryst. Solids* **172-174**:371 (1994).
3. J. E. Fletcher, S. Havlin, and G. H. Weiss, *J. Stat. Phys.* **51**:215 (1988); M. Gitterman and G. H. Weiss, *J. Stat. Phys.* **74**:941 (1994); J. J. Brey and J. Casado-Pascual, *Phys. Rev. E* **50**:116 (1994); J. J. Brey, J. Casado-Pascual, and B. Sánchez, *Phys. Rev. E* **52**:6071 (1995).
4. N. G. van Kampen, *Stochastic Processes in Physics and Chemistry* (North-Holland, Amsterdam, 1981).
5. J. J. Brey and J. Casado, *J. Stat. Phys.* **61**:713 (1991).
6. N. C. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, and E. Teller, *J. Chem. Phys.* **21**:1087 (1953).

7. A. B. Bortz, M. H. Kalos, and J. L. Lebowitz, *Comput. Phys.* **17**:10 (1975).
8. K. Binder and D. W. Heerman, *Monte Carlo Simulation in Statistical Physics* (Springer-Verlag, Berlin Heidelberg, 1988).
9. A. Prados, J. J. Brey, and B. Sánchez-Rey, *Phys. Rev. B* **55**:6343 (1997).
10. R. J. Glauber, *J. Math. Phys.* **4**:294 (1963).
11. J. J. Brey and A. Prados, *Phys. Rev. B* **49**:984 (1994).
12. F. Ritort, *Phys. Rev. Lett.* **75**:1190 (1995); S. Franz and F. Ritort, *Europhys. Lett.* **31**:507 (1995).
13. B. J. Kim, G. S. Jeon, and M. Y. Choi, *Phys. Rev. Lett.* **76**:4648 (1996).