Fast Monte Carlo algorithm for nonequilibrium systems

Heinz-Peter Breuer, Wolfgang Huber, and Francesco Petruccione
Albert-Ludwigs-Universität, Fakultät für Physik, Hermann-Herder-Straße 3, D-79104 Freiburg im Breisgau, Federal Republic of Germany

(Received 9 August 1995)

A fast method for the Monte Carlo solution of the balance equations that arise in nonequilibrium thermodynamics is suggested. It is applicable to chemical kinetics, reaction-diffusion processes, fluid dynamics, and heat conduction. The method is based on a multivariate master equation that is constructed in such a way that the simulation algorithm avoids time-consuming transition selection procedures, and thus becomes particularly efficient. For the example of a large chemical reaction scheme, the proposed method is shown to perform significantly faster than conventional methods, which rely on a birth-and-death master equation in discrete occupation number space.

PACS number(s): 02.70.Lq, 05.70.Ln, 82.20.Wt

Monte Carlo methods are a standard tool for the study of equilibrium properties (see, e.g., [1]) and are also widely used to investigate nonequilibrium situations, such as chemical reactions and reaction-diffusion processes [2–5]. These systems are most conveniently described by multivariate master equations, which describe reactions as a birth-and-death process and diffusion as a collective random walk [6–10]. Recently similar master equations were also shown to offer a systematic and numerically efficient approach to fluctuating hydrodynamics, fluid turbulence, and heat conduction [11–13].

For the purpose of this letter, it is important to recall two fundamental aspects of the master equation approach to nonequilibrium simulations: First, the master equation must be consistent with the macroscopic description from nonequilibrium thermodynamics, which is usually given in terms of balance equations for the conserved quantities, e.g., mass, momentum, energy, and particle number density. The consistency condition is that in the macroscopic limit the expectation values of the corresponding stochastic quantities obey these balance equations. Furthermore, it is possible to demand consistency of the fluctuations obtained through the master equation with those predicted by the theory of fluctuating hydrodynamics by Landau and Lifshitz [14]. Second, rather than solving the master equation directly as a differential equation for a probability density, the simulation algorithm generates a sample of realizations (or paths) of the stochastic process from which the quantities of interest are estimated as ensemble averages.

Most applications of the master equation approach have up to now been on chemical systems and reaction-diffusion processes, where the stochastic variables are discrete particle numbers. Here, the selection of the next event (or state transition) is the time-critical part of the simulation algorithms, and much effort has been dedicated to its optimization [1–3,15–17]. On the other hand, the introduction of continuous instead of discrete stochastic variables may result in a significant speedup, as it has been already demonstrated for the case of two-dimensional turbulence simulations [18]. The basic idea is to circumvent time-consuming selection and bookkeeping procedures by assigning equal rates to all transitions, but giving them appropriately varying transition step widths.

Fast simulations can thus be performed whenever it is possible to construct a stochastic process whose simulation algorithm assigns equal rates to all transitions and which is consistent with nonequilibrium thermodynamics. The purpose of the present report is to outline the method in general, and to compare its numerical performance to methods that are based on a birth-and-death master equation. Since we are going to present a simulation example from chemical kinetics, notation and terminology will be adapted to chemical kinetics.

Let us briefly recall the usual birth-and-death master equation description of chemical reactions [10]: The state of the system is described by a set of integer stochastic occupation numbers N_s. If the system is not homogeneous in space, then a discrete partition of space can be introduced in the usual way [7]. N_s is interpreted as the number of particles of a species in a space cell, s labeling both species and cells. The connection with the macroscopic description is made through interpreting the expectation values

$$c_s = \frac{1}{\Omega}\langle N_s \rangle \tag{1}$$

as the macroscopic concentrations, Ω denoting the volume of the space cells.

Now we generalize this approach and consider a system that is described by a set of real-valued stochastic variables $(x_1, \ldots, x_n) \equiv x$, where $x_s \in \mathbb{R}$. The probability of finding the system in a state within the n-dimensional interval $[x, x+dx]$ at time t is given by $P(x,t)dx$. In analogy to (1) the macroscopic variables are obtained through

$$c_s = \langle x_s \rangle$$

and the time evolution of the probability density $P(x,t)$ is described by a master equation

$$\frac{\partial}{\partial t} P(x,t) = \int dx' \{ w(x,x') P(x',t) - w(x',x) P(x,t) \}. \tag{2}$$

The transition rates $w(x,x')$ are specified as follows: We require that all realizations of the stochastic process are piecewise constant and that there is only a finite number R of transitions that *all have the same rate*:

$$w(x,x') = \frac{1}{\alpha\tau} \sum_{\lambda=1}^{R} \delta(x - b_\lambda(\alpha,x')). \tag{3}$$

Here, τ is the typical time scale of the system, and it defines the time unit in which all times are measured throughout this paper. α is a dimensionless scaling parameter that we are free to introduce in this ansatz. Its role is similar to that of the volume Ω in the birth-and-death master equation. The index λ labels the different possible transitions, and the functions b_λ specify these transitions as follows: If at time t_0 the transition λ occurs, and the system has been in state x' just until t_0, it is in state $b_\lambda(\alpha,x')$ immediately afterwards [13,18]. The choice of the b_λ depends on the specific system to be simulated and is guided by two requirements: (i) thermodynamic consistency, as mentioned above, and (ii) numerical efficiency. In the following we shall see in detail *how* these requirements can be employed to choose suitable transitions b_λ.

The thermodynamic consistency conditions are derived using an expansion in powers of the parameter α which is completely analogous to van Kampen's well-known system size expansion [10]. Thus, the first requirement on b_λ is that it depends on α such that

$$b_\lambda(\alpha,x) = x + \alpha\beta_\lambda(x) + O(\alpha^2), \tag{4}$$

where β_λ is a differentiable function that does not depend on α. The basic idea of the expansion is to split the stochastic dynamics of x into a large deterministic part c and a smaller stochastic part η, i.e., to set $x = c + \sqrt{\alpha}\,\eta$. Inserting this ansatz into the master equation, expanding the b_λ functions according to (4), and sorting the terms with respect to their order in α, we obtain the following results: The leading order terms are of order $\alpha^{-1/2}$, and thus they are required to cancel out if the expansion is to converge in the limit $\alpha \to 0$. This leads to the following ordinary system of differential equations ("the macroscopic equation") for $c = (c_1, \ldots, c_n)$:

$$\frac{d}{dt} c(t) = \frac{1}{\tau} \sum_{\lambda=1}^{R} \beta_\lambda(c(t)). \tag{5}$$

The next-to-leading order is α^0 and the corresponding terms constitute a linear Fokker-Planck equation for the probability density $\Pi(\eta,t)$ of the new stochastic variable η

$$\frac{\partial \Pi}{\partial t} = \frac{1}{\tau} \sum_{\lambda=1}^{R} \sum_{r,s=1}^{n} \left(-\frac{\partial \beta_{\lambda,s}(c)}{\partial c_r} \frac{\partial(\eta_r \Pi)}{\partial \eta_s} + \frac{1}{2} \beta_{\lambda,r}(c) \beta_{\lambda,s}(c) \frac{\partial^2 \Pi}{\partial \eta_r \partial \eta_s} \right).$$

All remaining higher order terms of the expansion become arbitrarily small as α goes to zero.

The following conclusions can be drawn from the expansion: First, the macroscopic equation and the Fokker-Planck equation depend on the transitions b_λ only via the β_λ, i.e., terms of order α^2 in (4) have no effect and $b_\lambda(\alpha,x)$ might as well be chosen such that it only depends linearly on α. Second, any macroscopic equation can be constructed by an appropriate choice of the transitions b_λ. Moreover, the choice of these transitions is not uniquely determined by the macroscopic equation: as is seen from Eq. (5), any set of functions b_λ for which the right-hand side of (5) is the same is equally admissible. Third, under certain conditions this arbitrariness can be employed to obtain the desired fluctuating behavior, which is described by the diffusion matrix in the Fokker-Planck equation [13].

These results are strictly valid only in the limit $\alpha \to 0$, but simulations are always performed with finite α. This is no serious problem however, since Monte Carlo methods have some statistical error anyway, due to the finiteness of the sample of realizations that is generated. Thus, α can always be chosen sufficiently small such that the systematic error caused by the finiteness of α is negligible in relation to the statistical error. This statement will be illustrated in the example.

Let us now turn to the numerical treatment of the master equation. The simulation algorithm which generates a realization of the Markov process defined by (2) and (3) is as follows:

(i) Set the state variables $(x_1, \ldots, x_n)$ to their initial values and set the time step counter $z = 0$.

(ii) Draw a uniformly distributed random number λ from the set $\{1, \ldots, R\}$.

(iii) Perform the transition $x \mapsto b_\lambda(\alpha,x)$.

(iv) Increment z by 1, and if not finished, go to (ii).

(v) The elapsed time t is the sum of z independent, exponentially distributed random time steps with mean $\alpha\tau/R$.

Since z typically is very large, setting $t = z\alpha\tau/R$ is a good approximation according to the central limit theorem, and no random numbers need to be generated for the time step. We see that the selection of the next transition simply amounts to drawing a uniformly distributed random number, and after the transition no further auxiliary variables need to be updated. Since typically the state vector x has many components, it is computationally advantageous if each single b_λ works only on a few components of x, and leaves the rest of them unchanged.

In order to compare the method's numerical performance with that of conventional schemes that are based on a discrete birth-and-death master equation, let us now consider a simple example from chemical kinetics, namely, an irreversible polymerization (or coagulation) reaction [19]. Each chain length or particle size r constitutes a separate species M_r, and reactions of the type

$$M_r + M_s \xrightarrow{k_{rs}} M_{r+s} \tag{6}$$

are possible for any pair (r,s). Here, k_{rs} denotes the reaction rate.

In the proposed method, the stochastic variable x_s now stands for the concentration of M_s. The thermodynamical consistency condition is that its expectation value $c_s = \langle x_s \rangle$ obeys the macroscopic rate equations corresponding to the reaction scheme (6). One can easily verify that the condition is satisfied if the transitions $b_\lambda \equiv b_{rs}$ are chosen as

$$b_{rs}:\begin{cases} x_r & \mapsto & x_r-\frac{1}{2}\alpha\tau k_{rs}x_rx_s \\ x_s & \mapsto & x_s-\frac{1}{2}\alpha\tau k_{rs}x_rx_s \\ x_{r+s} & \mapsto & x_{r+s}+\frac{1}{2}\alpha\tau k_{rs}x_rx_s \end{cases}$$

for $r\neq s$, and

$$b_{rr}:\begin{cases} x_r & \mapsto & x_r-\alpha\tau k_{rr}x_r^2 \\ x_{2r} & \mapsto & x_{2r}+\frac{1}{2}\alpha\tau k_{rr}x_r^2. \end{cases}$$

The discrete birth-and-death master equation corresponding to the reaction scheme (6) can be found explicitely in reference [4]. It depends on the volume Ω of the reaction vessel. Choosing an appropriate length unit, Ω equals the total number of monomers. In the simulation of the discrete process, the selection of the next transition is the time-critical part, and it can be implemented in several ways. First, there are the rather simple linear search [2] and the rejection method [1]. A modification of the rejection method is the null-process method [3]. Furthermore, there is a variety of more refined methods where the set of possible transitions is arranged in classes, or more generally in a tree, and the selection of the transition is performed by searching through the tree, beginning at its root [15–17]. Each method requires a set of variables that depend on the transitions rates, and have to be updated after each transition. We implemented the following schemes: linear search [2], rejection method [1], and a multilevel tree search method [15] with 2, 4, and 10 levels.

In order to have an exact analytical solution of the rate equations that belong to the reaction scheme (6) as reference, we employed the trivial kernel $k_{rs}=2\times10^5\tau^{-1}$ together with the initial condition $c_s(0)=\delta_{1,s}$ [19]. For the benchmarking, the simulation was run from time $t=2.5\times10^{-4}\tau$ to $t=10^{-3}\tau$. The simulation programs using the different algorithms were written as similar as possible and run on the same machine (IBM RS/6000).

In order to make a fair comparison between the different algorithms one must take into account that they do not solve the master equation exactly, but rather generate a sample of realizations of the stochastic process, from which the concentrations c_s are estimated through the sample averages $\hat{c}_s$. Consequently, the CPU time consumption of an algorithm must be related to the accuracy of its results. Using the exact solution $c_s^{ex}(t)$ of the rate equations, the error can be defined as

$$\varepsilon=\sqrt{\frac{1}{n}\sum_{s=1}^{n}[c_s^{ex}(t)-\hat{c}_s(t)]^2}. \tag{7}$$

Then there are two sources of error: (i) A systematic error, which stems from the fact that in the simulations α respectively Ω are finite, whereas the rate equations assume the macroscopic limit $\alpha\to0$, respectively $\Omega\to\infty$. (ii) A statistical error, which is caused by the finiteness of the sample generated in the simulation. Here we are interested in the case where the systematic error is much smaller than the statistical error. Then the more realizations are generated, the smaller becomes the error, and generically the reciprocal of the error will be proportional to the root of the sample size. This can in fact be verified in Fig. 1, where the total CPU time (which is proportional to the number of realizations) is plotted against the reciprocal of the squared error at $t=10^{-3}\tau$. We can see that the curves are roughly straight lines. Their slope is a measure of the "CPU time consumption per accuracy," and thus the lower the slope, the higher is the efficiency of the algorithm.

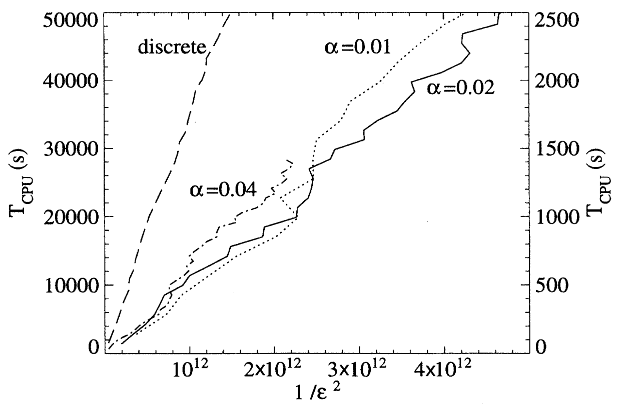

FIG. 1. CPU time consumption T_{CPU} that is required to achieve an accuracy ε^{-2} of the Monte Carlo simulation results [see Eq. (7)]. The slopes of the lines are used to measure the efficiencies of the different algorithms. The solid, dotted, and dash-dotted lines correspond to the suggested algorithm with different values of α, and their time axis is on the *right* side of the plot. The dashed line is the result of simulations of the birth-and-death master equation using the linear search selection method ($\Omega=2.5\times10^5$). Its time axis is on the left-hand side of the plot.

The fact that the lines for different α have nearly the same slope can easily be understood: When varying α, the CPU time *per realization* increases proportionally to $1/\alpha$, but the number of realizations necessary to achieve a given accuracy decreases by the same factor, such that the total CPU time remains constant. An analogous trade-off is found in the choice of Ω in the discrete master equation. This trade-off may be used for efficient load-balancing when the realizations are generated in parallel on multiple CPUs [20].

Noting the different ordinate scales in Fig. 1, we see that the method proposed here is 65 times faster than simulations of the discrete master equation using the linear search selection method. In exactly the same manner the other selection schemes were benchmarked, which lead to the results summarized in Table I. It is no question that such numbers depend very much on the type of problem, as well as on the implementation. Important factors are the inhomogeneity of the distribution of transition rates (respectively widths), the number of variables that have to be updated after each transition, and the speed of the random number generator. However, the presented results are meaningful beyond the particular numerical example that we have chosen here. In the following we shall see that a significant speedup can be achieved with the suggested method for a large class of simulation problems.

TABLE I. CPU times required for the different algorithms to solve the test problem to a given accuracy (see text). Shown are the factors by which the other methods are slower than the method proposed in this report.

	CPU time
Proposed method	1
Linear search	65
Rejection method	325
Two-level tree	94
Six-level tree	225
Ten-level tree	394

First, consider the relative performance of the different selection schemes in the discrete simulations. The rejection [1] and null-process [3] method are good for homogeneous situations, whereas they become extremely slow when the transition rates vary in a wide range, since a lot of time is wasted in rejection steps or null processes. This is confirmed by our results. Generally, the linear search method [2] is slow. However, in the present example, it performs very well, since the transition rates are, for the given initial condition and the considered time range, sharply peaked near the origin (r,s small), and the linear search starts at the right end. When the distribution of transition rates is quite inhomogeneous, tree methods are reported to perform the fastest [15]. How does this comply with our results? Besides the time consumption of the actual search for the next transition one must as well consider the amount of bookkeeping for maintaining the data structure. Therefore at least two cases must be distinguished:

(i) After a transition, only a small number of transition rates is changed. This is typical of reaction-diffusion systems with a small number of species and a large number of spatial cells. Clearly, a reaction within one cell does not affect the rates in the others, and a diffusion step only affects the rates of the two cells involved. In these cases the updating even of a complicated data structure like a tree is not computationally expensive, and the bookkeeping time is in general negligible compared to the search time.

(ii) A transition changes a large number of transition rates. This is typical of complex chemical reactions systems where the number of species is large and each species can react with all or many of the others. The presented aggregation or polymerization reaction is of this type. Our results indicate that tree or classification methods become inefficient for systems of case (ii), since the bookkeeping requires much time.

In contrast, the method proposed here is by construction not diverted by time-consuming selection and bookkeeping procedures, and for the given problem proves to be about 65 times faster than the fastest discrete algorithm.

How is the performance of our method effected by the distribution of the transition widths? Obviously it is expected to work best if the transitions are all of the same order of magnitude. On the other side, if there are a few large transitions and many small ones, a lot of CPU time is wasted on simulation steps that only make insignificant changes in the variables. Furthermore, the CPU time consumption is directly proportional to the number of possible transitions. Therefore, the set of possible transitions should not contain transitions that do not practically occur in the simulation. Qualitatively, the situation resembles very much the one encountered with the rejection method.

Note that the presented simulation example is not biased towards the proposed method: indeed, the transition rates, resp. widths, do vary in a wide range. This statement is verified by the good performance of the linear search compared to the other traditional methods, as mentioned above. The suggested method performs remarkably well even under these conditions.

Concluding, the method proposed in this report can be expected to significantly outperform the conventional algorithms whenever the simulated system is of case (ii). It offers an efficient and uncomplicated approach to nonequilibrium simulations.

[1] K. Binder, *Monte Carlo Methods in Statistical Physics* (Springer-Verlag, Berlin, 1979).

[2] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).

[3] P. Hanusse and A. Blanché, J. Chem. Phys. **74**, 6148 (1981).

[4] H. P. Breuer, J. Honerkamp, and F. Petruccione, Chem. Phys. Lett. **190**, 199 (1992).

[5] D. Dab, J.-P. Boon, and Y.-X. Li, Phys. Rev. Lett. **66**, 2535 (1991).

[6] D. A. McQuarrie, J. Appl. Prob. **4**, 413 (1967).

[7] G. Nicolis and I. Prigogine, *Self-Organization in Nonequilibrium Systems* (Wiley, New York, 1977).

[8] G. Nicolis and M. Malek-Mansour, J. Stat. Phys. **22**, 495 (1980).

[9] C. van den Broeck, Phys. Lett. **90A**, 119 (1982).

[10] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*, 2nd ed. (North-Holland, Amsterdam, 1992).

[11] H. P. Breuer and F. Petruccione, Phys. Rev. E **47**, 1803 (1993).

[12] H. P. Breuer and F. Petruccione, J. Phys. A **26**, 7563 (1993).

[13] H. P. Breuer and F. Petruccione, Phys. Lett. A **185**, 385 (1994).

[14] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics* (Pergamon Press, London, 1959).

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

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

[17] T. Fricke and J. Schnakenberg, Z. Phys. B **83**, 277 (1991).

[18] H. P. Breuer and F. Petruccione, Phys. Rev. E **50**, 2795 (1994).

[19] J. A. Biesenberger and D. H. Sebastian, *Principles of Polymerization Engineering* (Wiley, New York, 1983).

[20] H. P. Breuer and F. Petruccione, Continuum Mech. Thermodyn. **7**, 439 (1995).