Unbiased computation of transition times by pathway recombination

J. Kuipers[1,a)] and G. T. Barkema[1,2]
[1]*Institute for Theoretical Physics, Utrecht University, Leuvenlaan 4, Utrecht 3584 CE, The Netherlands*
[2]*Instituut-Lorentz for Theoretical Physics, Leiden University, Niels Bohrweg 2, Leiden 2333 CA, The Netherlands*

(Received 3 December 2007; accepted 26 March 2008; published online 5 May 2008)

In many systems, the time scales of the microscopic dynamics and macroscopic dynamics of interest are separated by many orders of magnitude. Examples abound, for instance, nucleation, protein folding, and chemical reactions. For these systems, direct simulation of phase space trajectories does not efficiently determine most physical quantities of interest. The past decade has seen the advent of methods circumventing brute force simulation. For most dynamical quantities, these methods all share the drawback of systematical errors. We present a novel method for generating ensembles of phase space trajectories. By sampling small pieces of these trajectories in different phase space domains and piecing them together in a smart way using equilibrium properties, we obtain physical quantities such as transition times. This method does not have any systematical error and is very efficient; the computational effort to calculate the first passage time across a free energy barrier does not increase with the height of the barrier. The strength of the method is shown in the Ising model. Accurate measurements of nucleation times span almost ten orders of magnitude and reveal corrections to classical nucleation theory. © *2008 American Institute of Physics*. [DOI: 10.1063/1.2911689]

I. INTRODUCTION

The average time it takes a protein to fold, an undercooled liquid to crystallize, or a chemically active molecule to react can, in principle, be obtained from brute force computer simulations by simply starting several times in the unfolded, liquid, or prereaction state and then integrating the dynamical equations in time until folding, crystallization, or reaction takes place. However, the typical time scales of the microscopic dynamics and macroscopic dynamics of interest are often separated by many orders of magnitude; for such systems, direct simulation of relevant phase space trajectories is very inefficient, if at all possible. In the past decade, methods have been developed that sample transition pathways while circumventing brute force simulation, such as transition path sampling,[1] transition interface sampling[2] and milestoning,[3] but most dynamical quantities, e.g., the average transition time, are not provided by those methods or are provided with systematical errors. We present a method to determine such dynamical quantities, free from systematic errors and very efficient. Our method is generally applicable to systems with known equilibrium properties, consisting of two regions with locally stable states, separated from each other by a barrier, which may be very high.

II. METHOD

A. Problem

A typical question that can be addressed by our method is the following: if a system is in equilibrium in region A of the phase space, what is the average time of first arrival in another region B? One should typically think of A and B as attracting basins in phase space, separated by a barrier. Examples of structures where such phenomena are found include nucleation, protein folding, and chemical reactions. The simplest approach to answer such questions is by direct simulation: The system is started in A and evolved in time until B is reached, and this procedure is repeated many times to collect statistics. If, however, after leaving A returns to it are much more frequent than traversals to B, this direct simulation approach becomes very inefficient since most of the computational effort is invested in dynamical trajectories from A back to A rather than to B. Our method more efficiently distributes the computational effort, spending more time on actual traversals.

B. Sampling subtrajectories

The main idea behind our method is to sample different relatively small pieces of the phase space trajectories, which we call *subtrajectories*, and combine them with an appropriate weighing into complete trajectories. To classify the different subtrajectories, one should identify a slice M in phase space, such that every path connecting A and B has to pass through it. In the case of nucleation, for instance, regions A, B, and M could be the set of states in which the size of the largest nucleus is smaller than half, larger than twice, or equal to the critical nucleus size (or a good estimate of this). A long simulation trajectory can then be divided into subtrajectories, which are classified according to (1) their initial and final states (A or B) where the system resides at least the correlation time τ_c,[4] (2) whether M has been crossed (denoted by an "M" between the initial and final states), and (3) whether the path crosses the other region without residing

a)Electronic mail: jkuipers@phys.uu.nl.

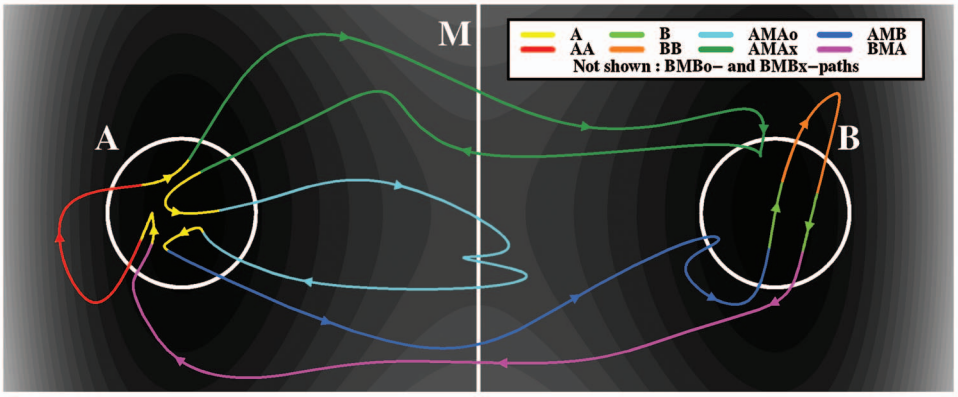

FIG. 1. (Color) Division of a trajectory in subtrajectories. A cut is made in the trajectory every time that the system resides in region A or B longer than some time τ_c, which is the time in which the system loses its memory of when, how, and where it entered. Random recombinations of these subtrajectories are equally valid trajectories.

there longer than τ_c continuously or not (denoted by an "x" or "o," respectively). All these different types of subtrajectories are illustrated in Fig. 1. Figure 1 also shows A and B subtrajectories that account for the time spent in A or B in excess of τ_c.

The different types of subtrajectories are generated by performing three different simulations. First, by starting in A, staying there for a time τ_c, and evolving from then, ignoring the paths that go through M, the A and AA subtrajectories are sampled. Next, analogous simulations around B are performed to find the B and BB subtrajectories. Finally, simulations starting in M are performed to generate the subtrajectories through M. These subtrajectories through M are sampled by starting in M and evolving both forward and backward in time until a correlation time τ_c is spent in A or B continuously at both ends of the path. As shown in Refs. 5 and 6, the starting points should be chosen from the set of points of first arrival in M from either A or B to sample the trajectories with the correct frequency. Because of time reversal symmetry, this can be achieved by sampling points on M from its equilibrium distribution, generating the trajectory by going forward and backward in time, and ignoring all paths that encounter M again on its backward part.

C. Recombination of subtrajectories

Now that the different types of subtrajectories are sampled, they should be recombined to generate complete phase space trajectories. These are concatenations of the subtrajectories, where the ones of the types AA, AMAo, AMAx, AMB, BB, BMBo, BMBx, and BMA are intertwined with A and B subtrajectories. Recombination is based on the notion that in the stable regions in phase space A and B, the system will mostly wander for a long time. If this time exceeds some value τ_c, the system has lost memory of where it entered. Since after this time there is no correlation between the entrance and exit point of region A or B, any random recombination of these subtrajectories would constitute a valid trajectory. The subtrajectories of different simulations can be recombined if given the proper weights; that is why the method is efficient. In a straightforward simulation, the subtrajectories crossing M typically are extremely rare, which strongly reduces statistical accuracy, but we generate additional subtrajectories through M by starting there and these subtrajectories can be combined with the AA and BB subtrajectories to sample long trajectories.

The weights for the different sets of subtrajectories can be determined from the condition that in the resulting long trajectories the system must be found in A, B, or M with the correct equilibrium probabilities, $p^{(\mathrm{A})}$, $p^{(\mathrm{B})}$, and $p^{(\mathrm{M})}$. We assume these are known, either analytically or from other kinds of simulations, for instance, parallel tempering,[7] methods involving umbrella sampling,[8,9] cluster algorithms,[10,11] or the Wang–Landau method.[12] For each subtrajectory, we measure the time it spends in regions A, B, and M. From these measurements, we determine for each class C of subtrajectories (where C can be AA, AMAo, AMAx, AMB, BB, BMBo, BMBx, and BMA) the average time spent in each region X (where X can be A, B, or M) and these average times are also determined for the class M of all subtrajectories crossing M; these are the quantities $T_C^{(X)}$. The notation T_C without superscript will denote the average total time spent on one subtrajectory of class C. Since these different subtrajectories are always preceded and succeeded by either an A or B subtrajectory, the time these latter subtrajectories take is also included in the times $T_C^{(X)}$. The frequency with which a long trajectory enters subtrajectories of class C is called n_C. With these definitions, it immediately follows that the probabilities of being in the three regions satisfy the equalities

$$p^{(\mathrm{A})} = n_{\mathrm{M}} T_{\mathrm{M}}^{(\mathrm{A})} + n_{\mathrm{AA}} T_{\mathrm{AA}}^{(\mathrm{A})}, \tag{1a}$$

$$p^{(\mathrm{B})} = n_{\mathrm{M}} T_{\mathrm{M}}^{(\mathrm{B})} + n_{\mathrm{BB}} T_{\mathrm{BB}}^{(\mathrm{B})}, \tag{1b}$$

$$p^{(\mathrm{M})} = n_{\mathrm{M}} T_{\mathrm{M}}^{(\mathrm{M})}. \tag{1c}$$

From these, the subtrajectory frequencies may be expressed as follows:

$$n_{\mathrm{AA}} = (p^{(\mathrm{A})} - n_{\mathrm{M}} T_{\mathrm{M}}^{(\mathrm{A})})/T_{\mathrm{AA}}^{(\mathrm{A})}, \tag{2a}$$

$$n_{\mathrm{BB}} = (p^{(\mathrm{B})} - n_{\mathrm{M}} T_{\mathrm{M}}^{(\mathrm{B})})/T_{\mathrm{BB}}^{(\mathrm{B})}, \tag{2b}$$

$$n_{\mathrm{M}} = p^{(\mathrm{M})}/T_{\mathrm{M}}^{(\mathrm{M})}. \tag{2c}$$

During the simulations through M, the number of times

N_C that the specific classes of subtrajectories through M are encountered are counted. These lead to the following relationships:

$$n_C = \frac{N_C}{N_M} n_M, \tag{3}$$

and these determine the remaining frequencies.

From the frequencies of the different subtrajectories, we immediately obtain the probabilities for subtrajectories leaving A or B to be of specific type

$$p_{AA} = n_{AA}/\mathcal{N}_A, \tag{4a}$$

$$p_{AMAo} = n_{AMAo}/\mathcal{N}_A, \tag{4b}$$

$$p_{AMAx} = n_{AMAx}/\mathcal{N}_A, \tag{4c}$$

$$p_{AMB} = n_{AMB}/\mathcal{N}_A, \tag{4d}$$

with $\mathcal{N}_A = n_{AA} + n_{AMAo} + n_{AMAx} + n_{AMB}$ and the analogous relations for the paths starting in B. Knowing these probabilities, we can randomly recombine subtrajectories with the proper weights to generate complete trajectories.

D. Determining transition times

Depending on the exact quantity of interest, often the explicit recombination process can be skipped and replaced by a combination of appropriate averages over the subtrajectories. Here, we specifically want to address the average transition time from A to B. Note that a traversal to B has to end with either an AMB or AMAx subtrajectory. To calculate the average transition time, we need to know the probabilities of finishing with an AMB or AMAx subtrajectory, the average numbers of times the AA and AMAo subtrajectories are traversed before this happens, and the average times these subtrajectories take. This results in the following equation for the transition time:

$$T^*_{A\to B} = \frac{p_{AA}T_{AA} + p_{AMAo}T_{AMAo} + p_{AMAx}T^{first}_{AMAx} + p_{AMB}T^{first}_{AMB}}{p_{AMAx} + p_{AMB}}. \tag{5}$$

The labels "first" are added to T_{AMAx} and T_{AMB} since the first time that region B is reached is relevant for the transition time, instead of the total time of the subtrajectory. These can also be measured during the simulations. We added an asterisk to distinguish these times from the first arrival time of B for a system starting Boltzmann distributed in A; these times are the first arrival times of B for a system starting in A with another distribution, namely, the distribution of points after being in A for a time τ_c. To obtain the real first arrival time $T_{A\to B}$, we must perform final simulations that start Boltzmann distributed in A until either τ_c time is spent in A or arrival in B occurs. We call the average time until a time τ_c in A is spent $T^{start}_{A\to A}$, and the average time until arrival in B occurs $T^{start}_{A\to B}$; these events happen with the probabilities $p^{start}_{A\to A}$ and $p^{start}_{A\to B}$. The transition time from A to B, starting Boltzmann distributed in A, is then

$$T_{A\to B} = p^{start}_{A\to A}(T^{start}_{A\to A} + T^*_{A\to B}) + p^{start}_{A\to B}T^{start}_{A\to B}. \tag{6}$$

E. Overview

Since the determination of the transition time from A to B involves multiple simulations in which a lot of quantities are measured, this section provides an overview of the different simulations including all the measured quantities; these are presented in Table I. With these quantities measured, Eqs. (2)–(6) yield the average time of first arrival in B for a system starting Boltmann distributed in A; other dynamical quantities can be obtained as well from different related equations.

III. SIMPLE TOY MODEL

As a first test, we applied this method to a system consisting of a 10×10 lattice with a potential energy assigned to each site.[13] The dynamics consist of jumps of a single particle to the neighboring sites with Metropolis[14] jump rates. "Regions" A and B are two opposing corners of the lattice, which are the minima of the potential. M is the diagonal in between, which forms a ridge in the potential landscape. The simulations are performed for different values of the temperature. Their results are shown in Fig. 2, together with the results of brute force simulations. Both simulations lasted for 1 min of CPU time. Also plotted is the exact transition time that is calculated by solving a set of linear equations.

The results are as expected: both our method and the brute force method sample the transition time without any systematical error. However, with our method the statistical error is constant as a function of the transition time, while the statistical errors of brute force simulation increase proportional to the inverse square root of the transition time, as the law of large numbers dictates (see inset of Fig. 2).

IV. ISING MODEL

To show that the method does not only work efficiently in low-dimensional toy systems, we apply it to determine the nucleation time of a 64×64 Ising model with spin-flip dynamics and Metropolis acceptance probabilities for a large range of parameter values for βJ and βh, i.e., the coupling constant and the external magnetization in units of $k_B T$.

TABLE I. The measured quantities in the different simulations.

Simulations starting with a time τ_c in A	
T_{A}	The total time of an A trajectory.
T_{AA}	The total time of an AA trajectory.
$T_{\mathrm{AA}}^{(\mathrm{A})}$	The time an AA trajectory spends in A
Simulations starting with a time τ_c in B	
T_{B}	The total time of a B trajectory.
T_{BB}	The total time of a BB trajectory.
$T_{\mathrm{BB}}^{(\mathrm{B})}$	The time a BB trajectory spends in B
Simulations starting in M	
T_{M}	The total time of an M trajectory
$T_{\mathrm{M}}^{(\mathrm{A})}$, $T_{\mathrm{M}}^{(\mathrm{B})}$, $T_{\mathrm{M}}^{(\mathrm{M})}$	The time an M trajectory spends in A, B, or M, respectively
$T_{\mathrm{AMAx}}^{\mathrm{first}}$, $T_{\mathrm{AMB}}^{\mathrm{first}}$	The first time that an AMAx or AMB trajectory arrives in B
N_{M}	The total number of M trajectories
N_{AMAx}, N_{AMAo}, N_{AMB}	The number of AMAx-, AMAo- and AMB-trajectories, respectively
N_{BMBx}, N_{BMBo}, N_{BMA}	The number of BMBx-, BMBo- and BMA-trajectories, respectively
Simulations starting Boltzmann distributed in A	
$T_{\mathrm{A}\to\mathrm{A}}^{\mathrm{start}}$	The time until a time τ_c is spent in A continuously
$T_{\mathrm{A}\to\mathrm{B}}^{\mathrm{start}}$	The time to reach B without spending a time τ_c in A continuously
$P_{\mathrm{A}\to\mathrm{A}}^{\mathrm{start}}$, $P_{\mathrm{A}\to\mathrm{B}}^{\mathrm{start}}$	The probabilities of starting with A→A and A→B

Details of the simulations are as follows. We start in a metastable state consisting of spins antiparallel to the external magnetization. The coordinate used to characterize regions A, B, and M is the number n_4 of spins that are parallel to the external magnetization and have four neighbors which are also parallel to it. This turns out to be a much better reaction coordinate than, for instance, the size of the largest cluster of parallel spins. The convenient property of n_4 is that it can only change by a maximum of five; if the size of the largest cluster is taken as a reaction coordinate, it can undergo large changes due to the merging or splitting of clusters. To obtain the free energy as a function of n_4, we use successive umbrella sampling:[15] by restricting n_4 to either i or $i+1$, the free energy difference between $n_4=i$ and $n_4=i+1$ is determined; this process is repeated for increasing values of i, until the free energy (as a function of n_4) returns to the value at $n_4=0$. A typical result of this free energy sampling is shown in Fig. 3.

Next, regions A, B, and M are characterized in terms of n_4. Region M is chosen at the top of the barrier and has a width of 5, so that every nucleation trajectory intersects it. Regions A and B are such that the free energy barriers to its boundaries are $5k_BT$. The motivation behind this is as fol-

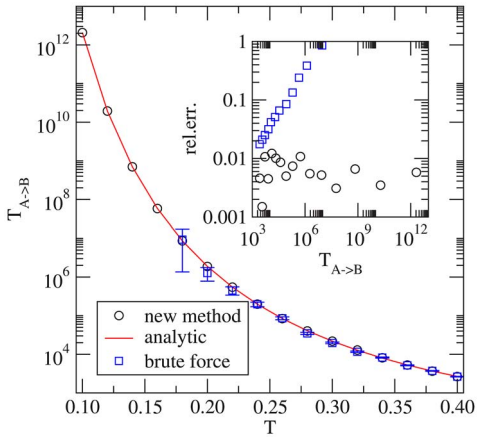

FIG. 2. (Color online) Results of our method applied to the toy model. Shown are the results of our method, compared to brute force simulations and an analytic result. Error bars of our method are omitted since they are very small. The inset shows the relative error of both our method and brute force simulation.

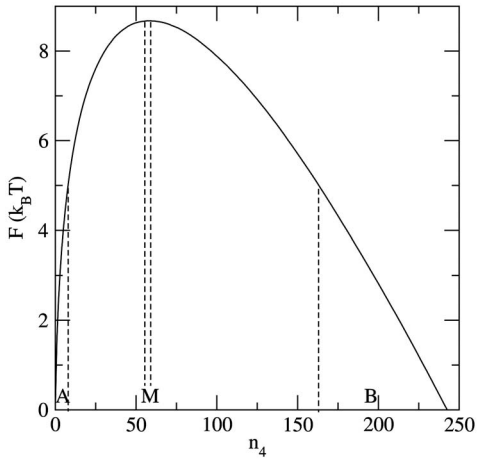

FIG. 3. Free energy of the two-dimensional Ising model with 64×64 sites, a coupling constant of $\beta J=0.60$ and an external field of $\beta h=0.06$, as function of the number n_4 of spins parallel to the external field which have four aligned neighbors. Also the prenucleation region A, the barrier region M, and the postnucleation region B are indicated.

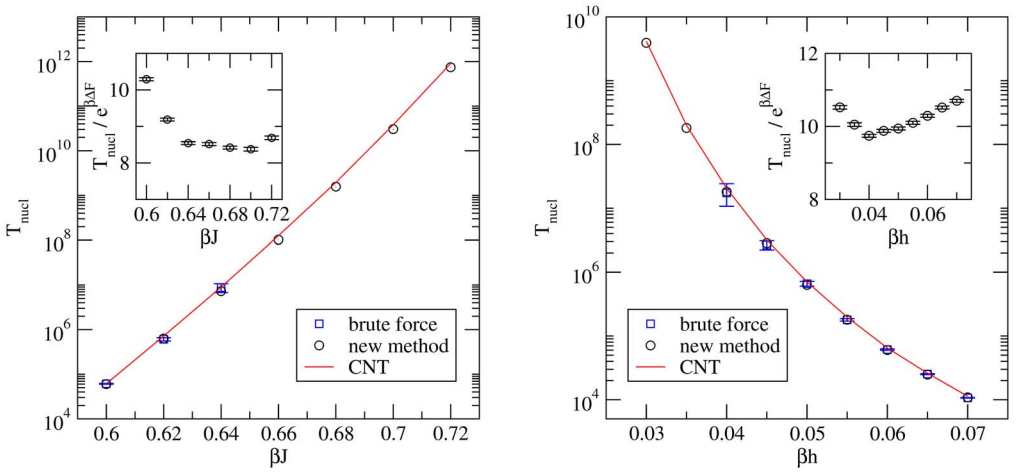

FIG. 4. (Color online) Nucleation times in the two-dimensional Ising model with spin-flip dynamics, with a system of 64×64 sites. Left: as function of the coupling constant βJ, for a fixed strength of the external field $\beta h=0.06$. Right: as function of the strength of the external field βh, for fixed coupling constant $\beta J=0.6$. Error bars of our method are omitted since they are very small. The insets show the deviations from classical nucleation theory.

lows: the barriers should be high enough so that the system will linger in A and B long enough to lose correlation, but, on the other hand, low enough that it can be crossed by thermal activation to sample AA and BB trajectories; $5k_BT$ seems a reasonable choice for that. We will call region A as the prenucleation state and B as the nucleated state since when the system arrives in B, it is extremely likely to continue to a stable state in which most spins are aligned with the external field. Regions A, B, and M are also indicated in Fig. 3. With regions A, B, and M defined and the probabilities of being there are known, the method can be applied to determine the nucleation times. In our simulations, we took a correlation time τ_c of 100 attempted spin-flips per site. Equilibration inside regions A and B is fast, and the system is certainly statistically uncorrelated within this time. We also verified this in simulations with $\tau_c=200$ and 500.

The resulting nucleation times are presented in Fig. 4. Note that the nucleation times span ten orders of magnitude with constant relative statistical errors. For comparison, results of brute force simulations (if possible) and classical nucleation theory (CNT)[16] are also shown. The general trend is well captured by CNT. However, as shown in the insets, our method is accurate enough to reveal the shortcomings of CNT.

The computational effort in this calculation of the average nucleation time is approximately 13 h of CPU time on an AMD-64 single-processor workstation for each set of temperature and field strength; 1 h is spent for the determination of the free energy as a function of cluster size and 12 h for the generation of the various subtrajectories and the nucleation time. An equal amount of computational effort was invested in the brute force computations.

ACKNOWLEDGMENTS

We would like to thank Henk van Beijeren for useful discussion.

[1] C. Dellago, P. G. Bolhuis, and P. L. Geissler, Adv. Chem. Phys. **123**, 1 (2002).
[2] T. S. van Erp *et al.*, J. Chem. Phys. **118**, 7762 (2003).
[3] A. K. Faradjian and R. Elber, J. Chem. Phys. **120**, 10880 (2004).
[4] The correlation time can be different for regions A and B, but for simplicity, we take a single value for τ_c.
[5] R. L. Jaffe, J. M. Henry, and J. B. Anderson, J. Chem. Phys. **59**, 1128 (1973).
[6] J. B. Anderson, J. Chem. Phys. **60**, 2566 (1974).
[7] D. J. Earl and M. W. Deem, Phys. Chem. Chem. Phys. **7**, 3910 (2005).
[8] G. Torrie and J. Valleau, Chem. Phys. Lett. **28**, 578 (1974).
[9] G. Torrie and J. Valleau, J. Comput. Phys. **23**, 187 (1977).
[10] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **58**, 86 (1987).
[11] U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).
[12] F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).
[13] The potential energy of the lattice sites is given by the function $V=((x+2y)(x+2y-2.6))^2+(x-y-0.1)^2$ evaluated in the points $x,y=0,\ldots,0.9$ with steps of 0.1. The sites (0,0) and (0.9,0.9) form potential minima, while the diagonal in between is around the potential maximum.
[14] N. Metropolis *et al.*, J. Chem. Phys. **21**, 1087 (1953).
[15] P. Virnau and M. Mueller, J. Chem. Phys. **120**, 10925 (2004).
[16] Classical nucleation theory predicts that the nucleation time equals $\text{const}\cdot\exp(\beta\Delta F_{\max})$. The constant is fitted to the data.