

Army Ants Algorithm for Rare Event Sampling of Delocalized Nonadiabatic Transitions by Trajectory Surface Hopping and the Estimation of Sampling Errors by the Bootstrap Method

Shikha Nangia, Ahren W. Jasper, Thomas F. Miller III, and Donald G. Truhlar
*Department of Chemistry and Supercomputing Institute, University of Minnesota,
Minneapolis, MN 55455-0431*

Abstract

The most widely used algorithm for Monte Carlo sampling of electronic transitions in trajectory surface hopping (TSH) calculations is the so-called anteatr algorithm, which is inefficient for sampling low-probability nonadiabatic events. We present a new sampling scheme (called the army ants algorithm) for carrying out TSH calculations that is applicable to systems with any strength of coupling. The army ants algorithm is a form of rare event sampling whose efficiency is controlled by an input parameter. By choosing a suitable value of the input parameter the army ants algorithm can be reduced to the anteatr algorithm (which is efficient for strongly coupled cases), and by optimizing the parameter the army ants algorithm may be efficiently applied to systems with low-probability events. To demonstrate the efficiency of the army ants algorithm, we performed atom-diatom scattering calculations on a model system involving weakly coupled electronic states. Fully converged quantum mechanical calculations were performed, and the probabilities for nonadiabatic reaction and nonreactive de-excitation (quenching) were found to be on the order of 10^{-8} . For such low-probability events the anteatr sampling scheme requires a large number of trajectories ($\sim 10^{10}$) to obtain good statistics and converged semiclassical results. In contrast by using the new army ants algorithm converged results were obtained by running 10^5 trajectories. Furthermore, the results were found to be in excellent agreement with the quantum mechanical results. Sampling errors were estimated using the bootstrap method, which is validated for use with the army ants algorithm.

I. INTRODUCTION

The most accurate way to describe a chemical system theoretically is to treat the entire system quantum mechanically. Currently, however, exact quantum mechanical calculations on chemical systems have been restricted to small chemical systems (involving two to six atoms for electronically adiabatic processes and two or three atoms for electronically nonadiabatic processes) because of the computational cost involved. On the other hand, classical mechanics may be used to model much larger systems but this is inadequate for systems where quantum effects play an important role. “Semiclassical” dynamical methods attempt to find an effective compromise between an entirely quantum mechanical treatment and completely classical treatment. In the present paper we are concerned with semiclassical trajectory methods, in which quantum mechanics is used to treat the electronic degrees of freedom, and the nuclear degrees of freedom are modeled as an ensemble of classical trajectories. This kind of semiclassical method has been widely used for electronically nonadiabatic collisions and photochemical reactions, and several reviews are available.¹⁻¹³

Trajectory surface hopping methods¹⁴⁻⁵⁴ are one group of semiclassical trajectory methods which incorporate electronic transitions into the overall dynamics by allowing the classical trajectories in the ensemble to make sudden hops (also called switches) between the coupled potential energy surfaces. Specifically, each trajectory in the ensemble is propagated independently, and at small time intervals along each trajectory, a hopping probability is computed. Tully proposed a fewest-switches prescription for the hopping probability such that the nuclear and electronic degrees of freedom evolve self-consistently.²² In the widely-used anteatr implementation of TSH, trajectories hop between states according to the hopping probability.

In general, the strength of the coupling between the potential energy surfaces governs the probability of nonadiabatic events and also the number of trajectories required in the ensemble to obtain converged results using the anteatr implementation of TSH. For example, if the probability of a nonadiabatic event is on the order of 10^{-1} or 10^{-2} , then three to five thousand trajectories are required to obtain good statistics., whereas in cases where the potential energy surfaces are very weakly coupled, and nonadiabatic events are rare, e.g., on the order of 10^{-8} , the anteatr algorithm requires

on the order of 10^8 trajectories to sample even a single nonadiabatic event, and sampling with good statistics is impractical. In fact, adequate sampling is already impractical for nonadiabatic probabilities on the order of 10^{-5} . Therefore, it has not been possible to model polyatomic systems with weakly coupled surfaces using the available TSH algorithms. In this paper, we present a new algorithm (called the army ants algorithm) that is designed to efficiently handle weakly coupled systems. This new algorithm may be considered a form of rare event sampling for the nonadiabatic processes. Although rare event sampling has been widely studied, and many algorithms are available,^{55–63} essentially all previous work has been based on transition state concepts where the sampling occurs at a reasonably well-localized dynamical bottleneck, whereas the present algorithm can treat rare and delocalized nonadiabatic events that may occur at any point along a trajectory.

One motivation for our recent studies of TSH methods has been to test them against accurate quantum dynamics.^{25,27,32–34,36,41–44,46,53} Because it has been impractical to study dynamics for systems with very small semiclassical transition probabilities, these tests have been carried out for systems with nonadiabatic probabilities of 3×10^{-4} and larger. The army ants algorithm allows us to extend these tests down to much lower probabilities; for example, in the present paper we present well-converged calculations for a system with a nonadiabatic transition probability of 1×10^{-8} . We will test not only the TFS surface hopping method, but also a variant of the TFS method called the fewest-switches with time uncertainty (FSTU) surface hopping method⁴⁶ that was previously shown^{46,53} to be more accurate than the TFS method for nonadiabatic probabilities in the range 1×10^{-2} to 3×10^{-4} . In particular we test the original version of the TFS method (TFS with reflection at frustrated hops, called TFS–) and three versions of the FSTU method (FSTU–, FSTU+, and FSTU ∇ V). All of these methods can be applied with either the anteaters scheme (which was first denoted “anteater” by Tully²²) or the new army ants sampling scheme.

In addition, we validate the bootstrap method for estimating Monte Carlo sampling errors. Although formulas for sampling errors can be derived for many of the quantities and algorithms employed in trajectory calculations,³ there are many other cases where error formulas are hard to derive. The army ants algorithm provides an

example of such a problem. The bootstrap method^{64–66} provides a general solution to the problem of estimating sampling errors, and in the present article we validate it and use it successfully for this purpose.

We summarize the existing sampling algorithms in Sec. II, and present the details of the army ants algorithm in Sec. III. The model system used for the calculations is described in Sec. IV. Section V contains the formulas for analysis of final product states. The bootstrap method of error analysis is presented in Sec. VI. Section VII provides details of all calculations performed and the results obtained. A discussion is provided in Sec. VIII and conclusions are presented in Sec. IX.

II. SAMPLING ALGORITHMS FOR TSH

In the TSH method, an ensemble of trajectories is used to model the nuclear dynamics, and each trajectory evolves classically under the influence of a single potential energy surface. The single surface propagation is interrupted, at small time intervals by decision points at which electronic transitions may occur. At each decision point (which we label by their times t_n) the electronic transition probability $P_{a \rightarrow b}(t_n)$ is computed, where a is the occupied potential energy surface and b is the target potential energy surface. The TFS method defines $P_{a \rightarrow b}(t_n)$ based on the local net flux of probability density such that the self-consistency of electronic and nuclear motions is maintained.^{22,23,30} (Actually, when “frustrated hops” are encountered, this self-consistency is also frustrated. We defer consideration of this aspect to Sec. III.) Tully^{15,22} proposed two schemes for sampling $P_{a \rightarrow b}(t_n)$ along the classical trajectory, which he called the ants and the anteater algorithms.

Before we discuss the ants and the anteater algorithms, it is useful to introduce the concept of extended trajectory space. For electronically adiabatic processes, trajectories are specified by a sequence of points in phase space. One can sample trajectory space by sampling initial conditions of the trajectories, i.e., by sampling phase space. Surface hopping trajectories in contrast, are specified not only by their initial phase points and initial surface but also by the times on phase points at which the hops occur. The space of all surface hopping trajectories will be called extended trajectory space, and the ants

algorithm, anteater algorithm, and new algorithm are all methods for sampling extended trajectory space.

II.A. Ants algorithm

In the ants algorithm¹⁵ each trajectory in the ensemble begins the simulation on a particular potential energy surface and is integrated to the first decision point t_1 , at which it splits into two branches. One branch continues to follow the initial potential energy surface and is called the nonhopping branch, whereas the other branch hops to follow the unoccupied potential surface and is called the hopping branch. Each of these resulting branches is assigned a weight according to the transition probability $P_{a \rightarrow b}(t_1)$ such that the total weight of both branches adds up to one, i.e.,

$$\begin{aligned} w_{\text{hop}} &= P_{a \rightarrow b}(t_n) \\ w_{\text{non}} &= 1 - P_{a \rightarrow b}(t_n) \end{aligned} \quad (1)$$

where w_{hop} is the weight assigned to the hopping branch, and w_{non} is the weight assigned to the nonhopping branch. The branches are independent, and each of them proceeds to additional decision points. The weight of each branch is the product of all weights assigned at every decision point in that branch's history. As a result, the weights assigned to each branch get smaller and smaller as the number of branches gets larger and larger. The repeated branching process results into a swarm of trajectories that is analogous to a swarm of ants—hence the name, ants method. The advantage of this method is that it allows a trajectory to follow all nonadiabatic events independent of the weights of the resulting branches. However, this is also the major disadvantage of the ants method. A trajectory with N_{dec} decision points would result in $2^{N_{\text{dec}}}$ branches. When the ants method was first proposed, the primitive trajectory surface hopping algorithms then in use restricted surface transitions to predefined seams, thus keeping N_{dec} small. However the modern algorithms based on fewest-switches considerations allow hopping decisions after every time step.

II.B. Anteater algorithm

The anteater method is the most widely used sampling algorithm for TSH simulations. In the anteater algorithm, the branching event is replaced by a stochastic event. Specifically, the nonadiabatic transition probability $P_{a \rightarrow b}(t_n)$ at each decision point t_n is compared to a random number λ , between 0 and 1. If $P_{a \rightarrow b}(t_n)$ is greater than λ , the hopping branch is followed and is assigned weight 1. The nonhopping branch is not followed and is assigned a weight of 0, i.e.,

$$\begin{aligned} w_{\text{hop}} &= 1 \\ w_{\text{non}} &= 0 \end{aligned} \quad (2)$$

If $P_{a \rightarrow b}(t_n)$ is less than λ , the nonhopping branch is followed, and the trajectory remains on the initial potential surface with weights

$$\begin{aligned} w_{\text{hop}} &= 0 \\ w_{\text{non}} &= 1 \end{aligned} \quad (3)$$

This scheme gets the name anteater from the analogy that an anteater is most likely to follow the path where the probability of finding ants is greatest.

Each anteater trajectory finishes with a weight of unity on one of the two potential energy surfaces and final results are obtained by averaging over many anteater trajectories. The anteater implementation of TSH is widely used and is entirely satisfactory for systems where the coupling between the potential energy surfaces is large enough that the probability of nonadiabatic events is on the order of 10^{-1} – 10^{-3} .

II.C. Generalization to more than two electronic states

In cases with more than two coupled potential energy surfaces, the ants and the anteater algorithms are slightly more complicated. Consider a system with S coupled potential energy surfaces, where surface 1 is occupied initially. At the first decision point, the transition probabilities from surface 1 to each of the other target surfaces are $P_{1 \rightarrow 2}, P_{1 \rightarrow 3}, \dots, P_{1 \rightarrow g}, \dots, P_{1 \rightarrow S}$. For the case of multiple potential energy surfaces the variable w_{hop} is the sum $P_{i \rightarrow j}$ over all $i \neq j$, where i is the current surface.

In the anteater algorithm, the transition probabilities are compared to a random number λ (between 0 and 1) to determine the surface on which to continue the trajectory.

A hop to surface 2 occurs if $\lambda < P_{1 \rightarrow 2}$, a hop to surface 3 if $P_{1 \rightarrow 2} < \lambda < P_{1 \rightarrow 2} + P_{1 \rightarrow 3}$, and so on. If no hop occurs, the trajectory remains on surface 1. In any event the trajectory then moves on to the next decision point.

In the ants algorithm, branching is allowed at every decision point from potential energy surface 1 to all the other unoccupied potential energy surfaces. The weight of a hopping branch $w_{\text{hop}}^{1 \rightarrow g}$ from surface 1 to surface g , shown explicitly by the superscript $1 \rightarrow g$, is determined by the transition probability for that surface, i.e., $w_{\text{hop}}^{1 \rightarrow g} = P_{1 \rightarrow g}$. The total weight of all of the hopping branches and the nonhopping branch is one, i.e., the weight of the nonhopping branch is $w_{\text{non}}^{1 \rightarrow 1} = 1 - (P_{1 \rightarrow 2} + P_{1 \rightarrow 3} + \dots + P_{1 \rightarrow g} + \dots + P_{1 \rightarrow S})$. Each of the hopping branches and nonhopping branch propagate independently, branching further at decision points. Thus the total number of branches would be $S^{N_{\text{dec}}}$.

III. ARMY ANTS ALGORITHM

Consider a weakly coupled system with a nonadiabatic reaction probability of $\sim 10^{-8}$. Since it requires on the order of 100 trajectories to obtain reasonable final-state statistics for a given final electronic state, the anteat algorithm would require a minimum of 10^{10} trajectories to obtain reasonably converged results. In the ants algorithm, every trajectory would sample the low-probability events, but the large number of resultant branches makes the ants method computationally expensive to implement, as described earlier. We propose a new algorithm, called the army ants algorithm, that is capable of performing calculations for systems with weakly coupled electronic states.

The army ants algorithm is designed to incorporate the strengths both the ants and the anteat algorithms. Specifically, by allowing branches to propagate with fractional weights, the ants algorithm is able to sample the critical regions of extended trajectory space; including those associated with low-probability events that the anteat algorithm “misses” when the number of trajectories is too small; therefore this aspect was crucial to incorporate in the new algorithm. Also, it is important to keep the algorithm tractable;

therefore, the ants aspect of branching at every time step was replaced by the stochastic anteater scheme of randomly choosing to follow only one of the possible branches.

In essence, an army ants trajectory is propagated using stochastic elements of the anteater algorithm at every time step along with the feature of fractional weights as in the case of the ants algorithm. In fact the army ants algorithm reduces in certain limits to the ants or anteater algorithms, as described later in this section.

The army ants algorithm is defined in terms of a parameter η such that $0 \leq \eta \leq 1$. The value of η is compared to the nonadiabatic probability $P_{a \rightarrow b}$ at each decision point t_n . The greater of the two values is called γ_n :

$$\gamma_n = \max \left\{ \begin{array}{l} \eta \\ P_{a \rightarrow b}(t_n) \end{array} \right. \quad (4)$$

In order to determine whether branching is allowed at that decision point a random number λ between 0 and 1 is drawn and compared to γ_n with the following consequences:

$$\lambda > \gamma_n : \text{no branching} \quad (5)$$

$$\lambda < \gamma_n : \text{branching} \quad .$$

In a non-branching case, the trajectory moves on to the next decision point while remaining on the current surface. If, on the other hand, branching occurs, then the branch weights w_{hop} for the hopping branch and w_{non} for the nonhopping branch are calculated as follows:

$$\begin{aligned} w_{\text{hop}} &= \frac{P_{a \rightarrow b}(t_n)}{\gamma_n} \\ w_{\text{non}} &= 1 - \frac{P_{a \rightarrow b}(t_n)}{\gamma_n} \quad . \end{aligned} \quad (6)$$

Another random number is drawn and the hopping branch is propagated if the random number is greater than 0.5, otherwise the non-hopping branch is followed, i.e., one follows each branch 50% of the time, even though they have different weights. We can summarize a successful branching event as follows:

Step 1. Initiate a trajectory from the ensemble on the appropriate potential energy surface, and at each decision point t_n compute $P_{a \rightarrow b}(t_n)$.

Step 2. Obtain $\gamma_n = \max [\eta, P_{a \rightarrow b}(t_n)]$

Step 3. Generate a random number λ_1 between 0 and 1.

Step 4. Compare γ_n and λ_1 . Branch if $\lambda_1 < \gamma_n$ and calculate w_{hop} and w_{non} .

Step 5. Generate another random number λ_2 between 0 and 1.

Step 6. Choose the hopping branch if $\lambda_2 > 0.5$, and choose the nonhopping branch otherwise.

It should be noted that the army ants algorithm reduces to the anteatr algorithm for $\eta = 0$, since the maximum of $[0, P_{a \rightarrow b}(t_n)]$ at every decision point yields $\gamma_n = P_{a \rightarrow b}(t_n)$ which on substitution in Eq. (6) results in branch weights for the anteatr algorithm as in Eqs. (2) and (3). On the other hand the army ants algorithm can be reduced to the ants algorithm by choosing $\eta = 1$. In this case, the value of γ_n is equal to 1 (since $\gamma_n = \max [1, P_{a \rightarrow b}(t_n)]$) at every time step, and substitution in Eq. (6) then results in ants algorithm weights as in Eq. (1). The parameter η therefore plays a role in the efficiency of the calculation, and in fact η may be optimized for this purpose. Depending upon the magnitude of coupling, the amount of branching character can be regulated by choosing the most appropriate value of η . This attribute makes the algorithm universally applicable to any kind of system, irrespective of the strength of coupling between the electronic states.

It is important to notice that all three trajectory surface hopping sampling algorithms (ants ($\eta = 1$), anteatr ($\eta = 0$), and army ants (non integer η)) achieve the same results in the limit of infinite sampling, i.e., the choice of η does not affect the results for a large sample.

The army ants algorithm can be implemented in two different ways, depending on how η is chosen. We label the first implementation as “fixed- η mode” and the second method as “ k mode,” the reasons for which are given below.

(1) In fixed- η mode, η at every step is set equal to η_0 , which is an input parameter in this mode. This parameter is the target value of the fraction of decision points at which a branch occurs. For example, in the system considered in this paper, a typical trajectory encounters about $N_{\text{dec}} = 900$ decision points. If one’s target is for every trajectory to branch (ants method) at 6 time steps and not branch (anteatr method) at the remaining

steps, one should set η_0 equal to 6/900 or about 7×10^{-3} . Depending upon the amount of branching desired, any value can be chosen for η_0 , provided only that it is a number between 0 and 1.

(2) In k mode, the distribution of branching points is independent of the time step taken by the integrator. The input parameter in this implementation is a constant k that has units of inverse time (and can be considered analogous to a first-order rate constant). The input k value is then multiplied by the instantaneous time step Δt_n at each decision point n , to obtain a unitless time-dependent variable given by

$$\eta = k\Delta t_n. \quad (7)$$

By allowing η to vary in this way at each decision point, we can regulate the branching because an integrator with variable step size can take small steps on the potential energy surface where the potential is steep, but a smaller value of η in this region will prevent excessive branching events. Conversely, k mode allows for more branching in the areas on the potential energy surface where the potential is flat and the integrator takes large steps.

The input parameter k can have any value, but a good value for k can be obtained from

$$k = \frac{\eta_{\text{opt}}}{\Delta t_{\text{avg}}}. \quad (8)$$

where η_{opt} is an optimal value for η , and Δt_{avg} is the average time step of the integrator. For the present paper, this approach was used to obtain the k parameter for the k mode army ants calculations.

It should be noted that decision points occur all along the classical trajectory, even when the system is far from the region of maximum coupling. When this is the case, w_{hop} may be several orders of magnitude smaller than probabilities of interest. In the army ants algorithm, the hopping branch is followed 50% of the time independent of the magnitudes of weights, but the branch may have a very small weight such that it will not contribute significantly to the final results. We therefore introduce a cutoff parameter w_{cut} such that if

$w_{\text{hop}} < w_{\text{cut}}$ at t_n , the hopping decision is ignored at t_n . For all calculations in the present article we set w_{cut} equal to 1×10^{-20} .

Since the new algorithm is more evolved and more efficient than the previous ones, we named it the army ants algorithm in recognition of a highly organized species of ants called army ants inhabiting the equatorial forest of planet Earth. In particular, a collection of army ants, taken as a whole, functions as a well integrated social entity with the extraordinary ability of forging into unknown territory, and we can hope that our collection of trajectories is equally adept at sampling an unknown extended-trajectory-space and discovering its most significant features.

The extension of army ants algorithm to more than two surfaces is straightforward. For example, for three surfaces one would follow each surface one third of the time (at random) at each branching point. Actually, one will stay unbiased even if one changes the fraction of the time that each surface is followed. If one were especially interested in the detailed product distribution on surface 2, one could follow surface 2 at 70% of the branches (chosen at random) and surfaces 1 and 3 at 15% each. In the present paper we have two surfaces, and we follow each surface at 50% of the branches.

We have discussed three sampling schemes (ants, anteater, and army ants) for TSH. We next discuss several variants of the TSH approach that differ in their treatment of frustrated hops. Any of the sampling schemes can be combined with any of these variants, and in the present article we will illustrate the new army ants algorithm with four of the variants, namely TFS–, FSTU–, FSTU+, and FSTU ∇ V.

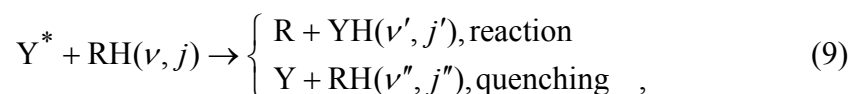
In trajectory surface hopping calculations, trajectories make sudden hops from an occupied surface to a target potential energy surface, and the potential energy of the system changes discontinuously when the system hops. To conserve the total energy of the system, the kinetic energy of the system on the new surface is adjusted by changing the nuclear momentum along the hopping vector. (In the present paper, the hopping vector is always a unit vector parallel to the nonadiabatic coupling vector \mathbf{d} , a choice that has been previously been justified by theoretical arguments^{19,24} and by testing³³ against accurate quantum mechanical calculations.) At certain points along a trajectory, a hopping attempt from a lower-energy to a higher-energy electronic state may occur such that the kinetic energy

associated with the component of nuclear momentum along the hopping vector \mathbf{h} is less than the potential energy gap between the occupied and the target electronic states. Such hops are classically forbidden, and are called “frustrated hops.” Frustrated hops are common in semiclassical trajectory calculations, and various prescriptions have been proposed to treat frustrated hops. Earlier treatments include ignoring the frustrated hop, denoted by “+”, or reflecting the nuclear momentum along \mathbf{h} , denoted by “-”. When implemented with the TFS method, these choices are labeled TFS+ and TFS-. The TFS- scheme is the original version of TFS,^{22,67} and TFS+ was introduced later,²⁶ although a combination of + and - was used even earlier in a general surface hopping scheme.¹⁸ In our group, we compared the performance of these variants systematically⁴⁴ and then introduced a new method of treating the frustrated hops, called the fewest-switches time uncertainty (FSTU) method.⁴⁶ The FSTU method is like TFS except that where frustrated hops are encountered the system may hop nonlocally. In the FSTU method, some hops remain frustrated, and these can be ignored (+) or cause reflection (-), yielding FSTU+ and FSTU-. Another FSTU prescription proposed recently is the FSTU ∇V ⁵³ scheme that uses the gradient information of the target potential surface to determine how momentum will be treated at frustrated hops.

We performed calculations on a realistic model system called the YRH system using the above variants of the TSH approach along with the anteatr and army ants sampling algorithms. Details of the YRH system are provided in the following section.

IV. THE YRH MODEL SYSTEM

The YRH model system⁴⁴ is a three-body system that has been developed in our group to study weakly coupled systems. The model reaction is an electronically nonadiabatic scattering process between an excited atom Y^* and a ground-electronic state diatomic molecule RH in a specific quantum state (ν, j) , where ν the vibrational quantum number, and j is the rotational quantum number. The collision can result in two possible outcomes as shown in the following equations:



where Y, R and H are model atoms, the asterisk denotes electronic excitation, and the primes and double primes denote the quantum numbers of the diatomic molecules associated with the reactive and quenched molecular arrangements, respectively.

The probability of the scattering process resulting in reaction is called P_R , whereas the probability of a quenching process is represented as P_Q . The sum of these probabilities is the total nonadiabatic probability P_N for a system to emerge in the ground electronic state in a scattering event, i.e.,

$$P_N = P_R + P_Q. \quad (10)$$

The details of the model YRH system have been reported in earlier work⁴⁴ in which a family of four YRH potential energy matrices (PEMs) was introduced. Briefly, the masses of Y, R, and H are taken as 10, 6, and 1.00783 amu, respectively. The model Y atom is electronically excited with energy equal to 0.36 eV, and the equilibrium bond energies for of RH and YH molecules are 3.9 and 4.3 eV, respectively. The zero point energies of RH and YH are 0.18 and 0.19 eV, respectively. The coupled potential energy surfaces are defined in the diabatic representation to have qualitatively similar shapes to those for the $\text{Br}^* + \text{H}_2$ system.³² Figure 1 provides the plots for the elements of the diabatic potential energy matrix. The energy gap between the two potential energy surfaces U_{11} and U_{22} remains almost constant at 0.36 eV, and the diabatic coupling U_{12} is small and localized in the interaction region. Adiabatic potential energy surfaces were obtained by diagonalizing the diabatic potential energy matrix, as described elsewhere.^{2,11,44}

Each member of the family differs from the others only in the magnitude of the diabatic coupling surface, and each may be labeled by the maximum value of its diabatic coupling U_{12}^{\max} . For the present work, we have extended the YRH family of four surfaces ($U_{12}^{\max} = 0.2, 0.10, 0.03, \text{ or } 0.01$ eV) to include a very weakly coupled system with the maximum diabatic coupling $U_{12}^{\max} = 0.0001$ eV. We will focus specifically for the present work on the set of coupled potential energy surfaces with $U_{12}^{\max} = 0.0001$ eV

in order to demonstrate the efficiency of the army ants algorithm, although we also report some preliminary calculations with $U_{12}^{\max} = 0.2$ eV.

V. FINAL STATE ANALYSIS

The final product analysis for army ants trajectories requires an ensemble of trajectories, for which information about the final arrangement and the final weight is known for each trajectory. Each trajectory in the ensemble, denoted by index i , finishes the simulation with some weight W_i that is the product of the weights assigned to it at every decision point along the propagation of the trajectory. By using the histogram method,^{3,27} each electronically nonadiabatic trajectory is also assigned values μ_{ri} for three of the applicable final quantum number μ_r , where $\mu_2 = v'$, $\mu_3 = j'$, $\mu_4 = v''$, $\mu_5 = j''$, and μ_1 is the final electronic-arrangement quantum number α , which is assigned as 1 for $Y^* + RH$, 2 for $R + YH$ and 3 for $Y + RH$. Note that μ_{1i} , μ_{2i} , and μ_{3i} are assigned if $\alpha = 2$, and μ_{1i} , μ_{4i} , and μ_{5i} are assigned if $\alpha = 3$.

If the total number of trajectories is N_{traj} , then the probability of a given electronic arrangement is

$$P_\alpha = \frac{\sum_i^{N_{\text{traj}}} W_i \delta_{\mu_{1i}\alpha}}{W_{\text{tot}}} , \quad (11)$$

where

$$W_{\text{tot}} = \sum_i^{N_{\text{traj}}} W_i . \quad (12)$$

We also label P_2 as P_R (R denotes reaction) and P_3 as P_Q (Q denotes quenching). The total probability of an electronically nonadiabatic outcome is P_N and is defined in Eq. (10). The final quantum states of the diatomic products are calculated according to the following equations using the energy non-conserving histogram method, as discussed elsewhere.²⁷ The first moments of the final vibrational and rotational quantum numbers μ_r are given by

$$\langle \mu_r \rangle = \frac{\sum_i^{N_{\text{traj}}} \mu_{ri} W_i \delta_{\mu_{1i} \alpha_r}}{W_{\text{tot}} P_{\alpha_r}}, \quad (13)$$

where $\alpha_2 = \alpha_3 = 2$ and $\alpha_4 = \alpha_5 = 3$. Note that P_{α} is the mean over all the trajectories, and $\langle \mu_r \rangle$ is the mean over the relevant subset of trajectories. To estimate the sampling errors in all the above observable quantities, we use a new method described in the following section.

VI. BOOTSTRAP RESAMPLING: METHOD OF ERROR ANALYSIS

The bootstrap method⁶⁴⁻⁶⁶ of error analysis is a resampling technique that can be used to estimate the sampling distribution of any well-defined function of sampled data. In general, resampling techniques are widely used statistical tools that are favored by virtue of their robustness and simplicity. In cases where there is no information about the underlying distribution of the sample and no analytical formulas are available, this method proves to be very useful.

In the army ants method, branching is a stochastic process, but due to the fractional weight carried by the trajectories the usual Monte Carlo error formulas³ cannot be applied. Error analysis was therefore carried out by the bootstrap resampling method.

The bootstrap method was first introduced by Efron and was named with the notion of pulling oneself out of the mud by one's own bootstraps.⁶⁴ In particular, in cases where the knowledge of the distribution is lacking, the sample itself may be taken as the best guide to the sampling distribution. The bootstrap method is applicable to our problem because our sample is unbiased and also is uncorrelated. In the bootstrap method, the initial sample is resampled by creating large number of bootstrap samples. The bootstrap estimation procedure consists of the following steps:

1) Take the original data set with N data points: $(x_1, x_2, \dots, x_i, \dots, x_N)$ and call it B_0 . Calculate the statistic of interest, which in this example is the mean \bar{x} .

2) Draw a sample of N data points at random "with replacement" from the initial set B_0 and name the new set bootstrap sample 1 (B_1). All data points for B_1 are selected from B_0 at random, using a random number generator, in such a way that once a data point has been drawn its value is recorded in B_1 , and it is replaced back in B_0 to assure

that in the next draw all the data points again have equal probability of being drawn. It is therefore likely that some data points in the new set will occur more than once. Calculate the statistic of interest (\bar{x}) for B_1 just as it was done for B_0 . Call this \bar{x}_1 .

3) Repeat, the second step M times, where M is a large number, to produce $B_2, \dots, B_m, \dots, B_M$. Calculate the statistic of interest (\bar{x}) for each of them. Label these \bar{x}_m .

4) Calculate the average value of the statistic of interest over all the bootstrap cycles by

$$\mu_{\text{bootstrap}} = \frac{\sum_{m=1}^M \bar{x}_m}{M} \quad (14)$$

5) Calculate the standard deviation of the calculated value of the statistic using

$$\sigma_{\text{bootstrap}} = \sqrt{\frac{\sum_{m=1}^M (\bar{x}_m - \mu_{\text{bootstrap}})^2}{M - 1}} \quad (15)$$

The bootstrap method is a very general. We expect, for example, that the bootstrap method will be very useful for calculations employing smooth sampling methods,³ because standard error formulae are not applicable to smooth sampling results. In both histogram and smooth sampling calculations the final observables are weighted means, but the bootstrap analysis does not require this, and it can be used to estimate the sampling distribution of any well-defined function of the sample data.

VII. CALCULATIONS AND RESULTS

The semiclassical calculations were done using version 7.0 of the nonadiabatic trajectory surface hopping code NAT.⁶⁸ Our first objective was to confirm that both anteatr and army ants results converge to the same semiclassical result. Also, we wanted to compare the army ants bootstrap error estimates with the anteatr error estimates that were obtained using the analytical³ formula. Therefore, we applied both the anteatr and army ants sampling schemes to the YRH system with relatively strong coupling

$U_{12}^{\max} = 0.2$ eV for which anteatr results have already been reported.⁵³ Calculations were performed using the FSTU ∇V method for the original, unmodified, anteatr algorithm and for the new army ants algorithm in the anteatr limit in the fixed- η implementation with $\eta_0 = 0$; in both cases we propagated 100000 trajectories, and we used the adiabatic representation. The results obtained by both of the methods are shown in Table I. The nonadiabatic probabilities, P_R and P_Q , with their respective vibrational and rotational moments, show very good agreement between the methods. The error analysis for the observable quantities in the army ants case was done by the bootstrap technique using 10000 bootstrap cycles. (A study of convergence with respect to the number of bootstrap cycles is given in Appendix A.) The bootstrap error estimates of army ants run, shown in Table I, match well with the anteatr analytical error estimates. This confirms that in the limit of $\eta_0 = 0$, the bootstrap method can be successfully applied to obtain the same error estimates as from the analytical formula.

To check the bootstrap method for the army ants calculations for values of η_0 other than 0, we performed calculations on larger sets of army ants trajectories for the optimum η_0 value of 1×10^{-2} . The aim of this calculation was to validate that the bootstrap method is applicable for the army ants algorithm. The error estimates are presented in Table II, and it can be seen that in most cases the bootstrap error estimates for the observable quantities decrease approximately by the inverse of the square root of the number of trajectories used for the calculations, which is the Monte Carlo result that is expected on general principles. This further validates the use of the bootstrap method for analyzing the results of army ants calculations.

To test the merit of the new algorithm we applied the army ants algorithm to a system for which anteatr calculations are not computationally affordable. In particular, the army ants algorithm was used for $Y^* + RH$ ($\nu = 0, j = 0$) with a scattering energy of 1.1 eV and with $U_{12}^{\max} = 0.0001$ eV. Table III shows the results obtained and the computer time taken for the calculations using TFS-, FSTU-, FSTU+, and FSTU ∇V method in the fixed- η mode of army ants and a value of η_0 value of 1×10^{-2} . The

criterion used for the choice of η_0 is discussed in the Appendix B. The army ants results are very converged well for 100000 trajectories. In contrast, the anteatr calculations failed to provide any electronically nonadiabatic final states in 100000 trajectories. The error estimates on the observable quantities, i.e., P_R , P_Q , $\langle \nu' \rangle$, $\langle j' \rangle$, $\langle \nu'' \rangle$, and $\langle j'' \rangle$ were calculated using the bootstrap resampling technique described in Sec. VI. For the purpose of comparison the quantum mechanical scattering results at scattering energy 1.10 eV are also shown in Table III.

In order to demonstrate the alternative k mode implementation of the army ants algorithm, we performed the FSTU ∇V calculations with same initial conditions as those for the fixed- η mode in Table IV. We used a variable-step-size Bulirsch-Stoer^{33,73} integrator, and the value of the input parameter was $k = 7.95 \times 10^{12} \text{ s}^{-1}$, which was obtained using Eq. (8) with the optimum value of $\eta_{\text{opt}} = 1 \times 10^{-2}$ and an average integrator time step, $t_{\text{avg}} = 1.26 \text{ fs}$ (obtained by taking an average over a small set of trajectories). The results for both fixed- η mode and k mode calculations are summarized in Table IV. Both methods require about the same computer time and converge to the same results, therefore confirming that the two implementations can be used interchangeably.

Since quantum mechanical scattering results sometimes oscillate as a function of scattering energy,⁴⁴ we carried out quantum mechanical calculations at seven energies. Appendix C shows that the results vary systematically with energy without significant oscillations so for testing the semiclassical methods we need not be concerned with oscillations. Therefore, we compared the semiclassical army ants algorithm results and quantum mechanical scattering calculations at three different scattering energies centered at 1.10 eV. In particular, for both methods, the results were obtained for $Y^* + \text{RH}$ ($\nu = 0$, $j = 0$) with $U_{12}^{\text{max}} = 0.0001 \text{ eV}$ at three values of the total energy, namely 1.07, 1.10, and 1.13 eV. The army ants calculations were performed using the FSTU ∇V method in the fixed- η mode with η_0 equal to 1×10^{-2} . All quantum mechanical calculations were obtained by the outgoing wave variational principle⁶⁹⁻⁷¹ using version 18.8 of the VP

computer code.⁷² The calculations involve 18934 basis functions (13884 square-integrable functions and 5050 non-square-integrable functions) in 334 channels (73 channels corresponding to $Y^* + RH$, 107 corresponding to $Y + RH$, and 154 corresponding to $R + YH$); other details of the calculations have been reported earlier.⁴⁴ The accurate quantum calculations are well converged to at least the number of significant figures shown in the tables, as demonstrated by their stability to increasing the number of basis functions and channels in each arrangement and increasing the number of quadrature points. The comparison of quantal and semiclassical results is presented in Table V.

VIII. DISCUSSION

The results obtained by the army ants calculations indicate a very significant improvement in efficiency as compared to the existing surface hopping algorithms. It was found that 100000 army ants trajectories running in parallel on four 375 MHz Power 3 WinterHawk+ processors of IBM SP supercomputer take about 10 hours to complete. There are three slave processors running trajectories and one master processor in control which is not load balanced; therefore the computer time is between 30 and 40 processor hours. For the purpose of demonstration we hypothetically assume propagating trajectories by the existing anteatr and ants methods and compare the time required for each of them to give a converged result for the YRH system with $U_{12}^{\max} = 0.0001$ eV. In the ants method a single initial trajectory taking to N_{dec} number of decision points equal to 900 results in $2^{900} \approx 10^{271}$ branches. In order to average over the initial conditions we will need a minimum of 500 trajectories which leads to an extraordinarily large number of resultant branches, $500 \times 10^{271} \approx 5 \times 10^{273}$, each of which (on an average) would be integrated for half as long as an army ants or anteatr trajectory. The time taken to complete this ants simulation is shown in Table VI. Consider now running anteatr trajectories for the system with the nonadiabatic probability of the order of magnitude $\sim 10^{-8}$. This also requires a large number of trajectories because a single nonadiabatic event will be experienced in approximately 10^8 trajectories and in order to get good statistics we would need a total of at least hundred nonadiabatic events which leads to

$\sim 10^{10}$ trajectories. (Recall that we want enough reactive and quenched trajectories to converge the quantum number moments.) The calculated time required for this hypothetical simulation is also reported in Table VI.

The huge computational requirements of the ant eater ($\sim 10^6$ hours) and ants ($\sim 3 \times 10^{269}$ hours) methods were an insuperable impediment to carrying out semiclassical trajectory surface hopping calculations on weakly coupled systems. In contrast, the new and flexible army ants algorithm can be adapted to systems with any kind of coupling, ranging from weak to strong.

The availability of well converged surface hopping calculations for a system with such a small probability of electronically nonadiabatic events allows us to test the semiclassical simulations in a new dynamical regime where they have never before been able to be tested. Table VII shows the mean unsigned relative errors. The average absolute error in the nonadiabatic reaction probability is only 15%, and that in the nonreactive quenching probability is only 8%. The average errors in the moments range from 7 to 58%. Considering the highly quantal character of these weakly allowed processes, the semiclassical methods are surprisingly accurate.

IX. CONCLUSIONS

The army ants algorithm is an efficient method for computing the probabilities of nonadiabatic events in weakly coupled systems. Since all trajectory surface hopping algorithms, i.e., ant eater, ants, and army ants, give the same converged results in the limit of infinite sampling, one may choose the algorithm that is most efficient. The present study shows that the army ants algorithm is useful and accurate for systems that are intractable by the two other sampling algorithms that have been proposed.

The army ants algorithm successfully captures the most desirable aspects of both the ants and ant eater algorithms. The new army ants algorithm retains the ants feature of assigning fractional weights to the daughter trajectories, and it also incorporates the stochastic nature of the ant eater algorithm. The method is designed in such a way that it can be applied to systems irrespective of the strength of the coupling between the

potential energy surfaces, thus providing a general algorithm for performing trajectory surface hopping calculations.

The present article also provides the first application of the bootstrap method for error estimation in molecular trajectory calculations. The method is quite successful, and it should be useful for error estimation in general, not just for army ants calculations.

Finally, the new algorithm allows us to test the trajectory surface hopping method for much weaker transition probabilities than has been ever before been possible. For a transition probability of the order 10^{-8} , the mean unsigned relative error in the six observables that were calculated is only 26%.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grant no. CHE00-92019.

APPENDIX A: CONVERGENCE WITH RESPECT TO NUMBER OF BOOTSTRAP CYCLES

The bootstrap method of error analysis resamples the original data set by randomly selecting data points from the original sample, with replacement, to generate a large number of bootstrap cycles, as explained in Sec. VI. To determine the number of bootstrap cycles required for the error analysis of the army ants runs, convergence studies of the error estimates with respect to the number of bootstrap cycles were performed. Bootstrap analyses were carried out for various numbers of cycles using the results obtained by the FSTU ∇V method with the army ants algorithm in the fixed- η mode with η_0 equal to 1×10^{-2} for $Y^* + RH$ ($\nu = 0, j = 0$) with $U_{12}^{\max} = 0.0001$ eV at scattering energy 1.10 eV. Table A-I shows the values of the bootstrap averages and error estimates for 5000, 10000, and 20000 bootstrap cycles and the computer time taken by

each of the runs. Notice that the computer times for analyses are less than 1% of the computer time required to run the trajectories, which is between 30 and 40 processor hours for 10^5 trajectories. It was found that 10000 bootstrap cycles yields good convergence, and therefore 10000 bootstrap cycles are used for the bootstrap analyses in this paper.

APPENDIX B: CHOICE OF INPUT PARAMETER FOR BRANCHING

In the fixed- η mode of the army ants algorithm, the value of the input parameter η_0 controls the number of branching events that occur along each trajectory. The amount of branching that is desired can depend upon the strength of the coupling between the potential energy surfaces. As mentioned in Sec. III, the value of η_0 can be any number such that $0 \leq \eta_0 \leq 1$, but the efficiency of the calculation can depend strongly on η_0 . An efficient value of η_0 is one that requires the least number of trajectories N_{traj} to obtain results for the observable quantities P_α and $\langle \mu_r \rangle$ that are converged to some small errors $\Delta \varepsilon_r$. To determine the most efficient value of η_0 for the weakly coupled model systems studied here [specifically for $Y^* + \text{RH}$ ($\nu = 0, j = 0$) at scattering energy 1.10 eV with $U_{12}^{\text{max}} = 0.0001$ eV], calculations were performed for a range of η_0 values using the FSTUVV method, as shown in Table B-I. The results for the probabilities, i.e., P_R , P_Q , and for the moments, i.e., $\langle \nu' \rangle$, $\langle j' \rangle$, $\langle \nu'' \rangle$, and $\langle j'' \rangle$, along with the error estimates, are shown in Table B-I. Error analyses were carried out using the bootstrap method with 10000 bootstrap cycles for each η_0 value.

With $\eta_0 = 1 \times 10^{-4}$, good convergence was obtained with 500000 trajectories, which is a fairly large number due to the fact that at small values of η_0 the number of branching events encountered by each trajectory is small. Since $\eta_0 = 1 \times 10^{-4}$ provided the largest sampled space, we used the values in row 1 along with the Monte Carlo error formula

$$\Delta\mathcal{E} \propto \frac{1}{\sqrt{N_{\text{traj}}}} . \quad (\text{B1})$$

to estimate the number of trajectories needed to obtain the at least as small as in row 1 for P_R and P_Q , and the larger of the two values is listed as N_{traj}^P in Table B-I. Similarly, the maximum value of the estimated number of trajectories needed for good convergence of (at least as good as row 1) $\langle \nu' \rangle$, $\langle j' \rangle$, $\langle \nu'' \rangle$, and $\langle j'' \rangle$ is listed as N_{traj}^M in Table B-I. From Table B-I, it can be concluded that the least number of trajectories required to obtain the same relative errors as obtained in the run with 500,000 trajectories and $\eta_0 = 1 \times 10^{-4}$ is obtained using $\eta_0 = 2 \times 10^{-2}$ for the probabilities and using $\eta_0 = 1 \times 10^{-2}$ for the moments. Since using $\eta_0 = 1 \times 10^{-2}$ performs best on average for both probabilities and moments, it was concluded that this is the most efficient η_0 value, that is, $\eta_{\text{opt}} = 1 \times 10^{-2}$. With η_0 equal to 1×10^{-2} , only 25000 trajectories are required to obtain good convergence.

APPENDIX C

Quantum mechanical scattering calculations for $Y^* + \text{RH}$ ($\nu = 0, j = 0$) with $U_{12}^{\text{max}} = 0.0001$ eV were performed at seven different values of the total energy centered at 1.10 eV, as shown in Table C-I. The average values for the observable quantities over the entire set of scattering energies are also shown in Table C-I.

References

- ¹W. H. Miller, *Adv. Chem. Phys.* **30**, 77 (1975).
- ²J. C. Tully, in *Dynamics of Molecular Collisions*, edited by W. H. Miller (Plenum, New York, 1976), Part B, pp. 217–267.
- ³D. G. Truhlar and J. T. Muckerman, in *Atom-Molecule Collision Theory*, edited by R. B. Bernstein (Plenum, New York, 1979), pp. 505–566.
- ⁴E. E. Nikitin and S. Ya. Umanski, *Theory of Slow Atomic Collisions* (Springer-Verlag, Berlin, 1984).
- ⁵H. Nakamura, *Adv. Chem. Phys.* **82**, 243 (1992).
- ⁶E. A. Gislason, G. Parlant, and M. Sizun, *Adv. Chem. Phys.* **82**, 321 (1992).
- ⁷S. Chapman, *Adv. Chem. Phys.* **82**, 423 (1992).
- ⁸M. F. Herman, *Annu. Rev. Phys. Chem.* **45**, 83 (1994).
- ⁹H. Nakamura, in *Dynamics of Molecules and Chemical Reactions*, edited by R. E. Wyatt and J. Z. H. Zhang (Dekker, New York, 1996), pp. 473–529.
- ¹⁰E. E. Nikitin, *Annu. Rev. Phys. Chem.* **50**, 1 (1999).
- ¹¹J. C. Tully, in *Modern Methods for Multidimensional Dynamics Computations in Chemistry*, edited by D. L. Thompson (World Scientific, Singapore, 1998), pp. 34–72.
- ¹²M. D. Hack and D. G. Truhlar, *J. Phys. Chem. A* **104**, 7917 (2000).
- ¹³A. W. Jasper, B. K. Kendrick, C. A. Mead, and D. G. Truhlar, in *Modern Trends in Chemical Reaction Dynamics*, edited by X. Yang and K. Liu (World Scientific, Singapore, 2003), in press.
- ¹⁴A. Bjerre and E. E. Nikitin, *Chem. Phys. Lett.* **1**, 179 (1967).
- ¹⁵R. K. Preston and J. C. Tully, *J. Chem. Phys.* **54**, 4297 (1971).
- ¹⁶A. Komornicki, T. F. George, and K. Morokuma, *J. Chem. Phys.* **65**, 48 (1976).
- ¹⁷P. J. Kuntz, J. Kendrick, and W. N. Whitten, *Chem. Phys.* **38**, 147 (1979).
- ¹⁸N. C. Blais and D. G. Truhlar, *J. Chem. Phys.* **79**, 1334 (1983).
- ¹⁹M. F. Herman, *J. Chem. Phys.* **81**, 754 (1984).
- ²⁰N. C. Blais, D. G. Truhlar, and C. A. Mead, *J. Chem. Phys.* **89**, 6204 (1988).
- ²¹G. Parlant and E. A. Gislason, *J. Chem. Phys.* **91**, 4416 (1989).
- ²²J. C. Tully, *J. Chem. Phys.* **93**, 1061 (1990).
- ²³J. C. Tully, *Int. J. Quantum Chem., Symp.* **25**, 299 (1991).

- ²⁴D. F. Coker and L. Xiao, *J. Chem. Phys.* **102**, 496 (1995).
- ²⁵M. S. Topaler, M. D. Hack, T. C. Allison, Y.-P. Liu, S. L. Mielke, D. W. Schwenke, and D. G. Truhlar, *J. Chem. Phys.* **106**, 8699 (1997).
- ²⁶U. Müller and G. Stock, *J. Chem. Phys.* **107**, 6230 (1997).
- ²⁷M. S. Topaler, T. C. Allison, D. W. Schwenke, and D. G. Truhlar, *J. Phys. Chem. A* **102**, 1666 (1998).
- ²⁸M. S. Topaler, T. C. Allison, D. W. Schwenke, and D. G. Truhlar, *J. Chem. Phys.* **109**, 3321 (1998).
- ²⁹M. Sizun, J.-B. Song, and E. A. Gislason, *J. Chem. Phys.* **109**, 4815 (1998).
- ³⁰J. C. Tully, *Faraday Discuss.* **110**, 407 (1998).
- ³¹W. Jakubetz, J. N. L. Connor, and P. J. Kuntz, *Phys. Chem. Chem. Phys.* **1**, 1213 (1999).
- ³²Y. L. Volobuev, M. D. Hack, and D. G. Truhlar, *J. Phys. Chem. A* **103**, 6225 (1999).
- ³³M. D. Hack, A. W. Jasper, Y. L. Volobuev, D. W. Schwenke, and D. G. Truhlar, *J. Phys. Chem. A* **103**, 6309 (1999).
- ³⁴M. D. Hack, A. W. Jasper, Y. L. Volobuev, D. W. Schwenke, and D. G. Truhlar, *J. Phys. Chem. A* **104**, 217 (2000).
- ³⁵D. Babikov, E. A. Gislason, M. Sizun, F. Aguillon, and V. Sidis, *J. Chem. Phys.* **112**, 7032 (2000).
- ³⁶Y. L. Volobuev, M. D. Hack, M. S. Topaler, and D. G. Truhlar, *J. Chem. Phys.* **112**, 9716 (2000).
- ³⁷M. Y. Niv, B. Bargheer, and R. B. Gerber, *J. Chem. Phys.* **113**, 6660 (2000).
- ³⁸S. K. Gray, G. G. Balint-Kurti, and G. C. Schatz, *J. Chem. Phys.* **113**, 7330 (2000).
- ³⁹V. G. Ushakov, K. Nobusada, and V. I. Osherov, *Phys. Chem. Chem. Phys.* **3**, 63 (2001).
- ⁴⁰A. L. Kaledin, M. C. Heaven, and K. Morokuma, *J. Chem. Phys.* **114**, 215 (2001).
- ⁴¹M. D. Hack and D. G. Truhlar, *J. Chem. Phys.* **114**, 2894 (2001).
- ⁴²M. D. Hack and D. G. Truhlar, *J. Chem. Phys.* **114**, 9305 (2001).
- ⁴³M. D. Hack, A. M. Wensmann, D. G. Truhlar, M. Ben-Num, and T. J. Martinez, *J. Chem. Phys.* **115**, 1172 (2001).
- ⁴⁴A. W. Jasper, M. D. Hack, and D. G. Truhlar, *J. Chem. Phys.* **115**, 1804 (2001).

- ⁴⁵C. Zhu, H. Kamisaka, and H. Nakamura, *J. Chem. Phys.* **115**, 11036 (2001).
- ⁴⁶A.W. Jasper, S. N. Stechmann, and D. G. Truhlar, *J. Chem. Phys.* **116**, 5424 (2001), **117**, 10427(E) (2002).
- ⁴⁷J. M. C. Marques, A. I. Voronin, and A. J. C. Varandas, *J. Phys. Chem. A* **106**, 3673 (2002).
- ⁴⁸N. L. Doltsinis and D. Marx, *Phys. Rev. Lett.* **88**, 166402 (2002).
- ⁴⁹G. A. Jones, M. N. Paddon-Row, B. K. Carpenter, and P. Piotrowiak, *J. Phys. Chem. A* **106**, 5011 (2002).
- ⁵⁰T. Takayanagi, *J. Phys. Chem. A* **106**, 4914 (2002).
- ⁵¹F. Santoro, C. Petrongolo, and G. C. Schatz, *J. Phys. Chem. A* **106**, 8276 (2002).
- ⁵²Ph. Halvick, M. Boggio-Pasqua, L. Bonnet, A. I. Voronin, and J.-C. Rayez, *Phys. Chem. Chem. Phys.* **4**, 2560 (2002).
- ⁵³A.W. Jasper and D. G. Truhlar, *Chem. Phys. Lett.* **369**, 60 (2003).
- ⁵⁴A. Bastida, C. Cruz, J. Zúñiga, and A. Requena, *J. Chem. Phys.* **119**, 6489 (2003).
- ⁵⁵J. P. Valleau, in *Computer Simulations in Materials Science*, edited by M. Meyer and V. Pontikis (Kluwer, Dordrecht, 1991), pp. 67–84.
- ⁵⁶J. B. Anderson, *Adv. Chem. Phys.* **91**, 381 (1995).
- ⁵⁷S. Melchionna, *Phys. Rev. E* **62**, 8762 (2000).
- ⁵⁸G. Ciccotti and M. Ferrario, *J. Mol. Liq.* **89**, 1 (2000).
- ⁵⁹T. Simonson, in *Computational Biochemistry and Biophysics*, edited by O. M. Becker, A. D. Mackerell, Jr., B. Roux, and M. Watanabe (Marcel Dekker, New York, 2001), pp. 169–197.
- ⁶⁰P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, *Ann. Rev. Phys. Chem.* **53**, 291 (2002).
- ⁶¹A. F. Voter, F. Montalenti, and T. C. Germann, *Annu. Rev. Mat. Rec.* **32**, 321 (2002).
- ⁶²U. Röthlisberger, M. Spink, and J. Hutter, in *Bridging Time Scales: Molecular Simulations for the Next Decade*, edited by P. Nielaba, M. Mareschal, and G. Ciccotti (Springer, Berlin, 2002), pp. 414–442.
- ⁶³J. A. Rahman and J. C. Tully, *Chem. Phys.* **285**, 277 (2002).
- ⁶⁴B. Efron, *Ann. Statist.* **7**, 1 (1979).

- ⁶⁵B. Efron and R. Tibshirani, *An Introduction to the bootstrap* (Chapman and Hall, London, 1993).
- ⁶⁶C. Z. Mooney and R. D. Duval, *Bootstrapping: A Nonparametric Approach to Statistical Inference* (Sage, Newbury Park, 1993).
- ⁶⁷S. Hammes-Schiffer and J. C. Tully, *J. Chem. Phys.* **101**, 4657 (1994).
- ⁶⁸S. Nangia, A. W. Jasper, Y. L. Volobuev, M. S. Topaler, T. C. Allison, M. D. Hack, Y.-P. Liu, S. N. Stechmann, and D. G. Truhlar, NAT-version 7.0, University of Minnesota, Minneapolis, MN, 2003.
- ⁶⁹Y. Sun, D. J. Kouri, D. G. Truhlar, and D. W. Schwenke, *Phys. Rev. A* **41**, 4857 (1990).
- ⁷⁰D. W. Schwenke, S. L. Mielke, and D. G. Truhlar, *Theor. Chim. Acta* **79**, 241 (1991).
- ⁷¹G. J. Tawa, S. L. Mielke, D. G. Truhlar, and D. W. Schwenke, *Adv. At. Molec. Vibrations Collision Dynamics* **2B**, 45 (1994).
- ⁷²Y. L. Volobuev, G. J. Tawa, S. L. Mielke, D. W. Schwenke, T. C. Allison, M. D. Hack, and D. G. Truhlar, VP-version 18.8, University of Minnesota, Minneapolis, MN, 2000.
- ⁷³R. Bulirsch and J. Stoer, *Numer. Math* **8**, 1 (1966).

Table I. Results for anteater and army ants ($\eta_0 = 0$) calculations using the FSTU ∇V method, for the $Y^* + RH(\nu = 0, j = 0)$ system with a total energy of 1.10 eV and with $U_{12}^{\max} = 0.2$ eV using 100000 trajectories.

Method	η_0	P_R	$\langle \nu' \rangle$	$\langle j' \rangle$	P_Q	$\langle \nu'' \rangle$	$\langle j'' \rangle$
Anteater ^a	-	$(1.38 \pm 0.03) \times 10^{-2}$	1.11 ± 0.02	12.8 ± 0.13	$(4.39 \pm 0.07) \times 10^{-2}$	0.98 ± 0.01	6.31 ± 0.07
Army ants ^b	0.0	$(1.41 \pm 0.03) \times 10^{-2}$	1.12 ± 0.02	12.8 ± 0.13	$(4.42 \pm 0.06) \times 10^{-2}$	0.97 ± 0.01	6.30 ± 0.07

^aAnteater calculations using the original, unmodified method and the Monte Carlo error formula of Ref. 3.

^bAnteater calculations were done using the army ants algorithm with $\eta = 0$. The errors were estimated using 10000 bootstrap cycles.

Table II. Bootstrap analysis results^a for army ants calculations.^b

N_{traj}	P_R	$\langle \nu' \rangle$	$\langle j' \rangle$	P_Q	$\langle \nu'' \rangle$	$\langle j'' \rangle$
100000	$(1.36 \pm 0.034) \times 10^{-8}$	1.33 ± 0.019	12.5 ± 0.115	$(3.28 \pm 0.046) \times 10^{-8}$	1.13 ± 0.010	4.83 ± 0.071
200000	$(1.34 \pm 0.023) \times 10^{-8}$	1.30 ± 0.014	12.7 ± 0.09	$(3.31 \pm 0.031) \times 10^{-8}$	1.14 ± 0.007	4.83 ± 0.059
300000	$(1.35 \pm 0.020) \times 10^{-8}$	1.31 ± 0.011	12.7 ± 0.067	$(3.33 \pm 0.025) \times 10^{-8}$	1.13 ± 0.006	4.84 ± 0.041
400000	$(1.36 \pm 0.019) \times 10^{-8}$	1.31 ± 0.008	12.6 ± 0.051	$(3.30 \pm 0.021) \times 10^{-8}$	1.13 ± 0.006	4.87 ± 0.037

^aError estimates were calculated using 10000 bootstrap cycles.

^bArmy ants calculations were performed in the fixed- η mode with $\eta_0 = 1 \times 10^{-2}$ using the FSTU ∇ V method for $Y^* + \text{RH}$

($\nu = 0, j = 0$), for the total energy 1.10 eV with $U_{12}^{\text{max}} = 0.0001$.^a

Table III. Results for $Y^* + RH$ ($\nu = 0, j = 0$) system for the total energy 1.10 eV with $U_{12}^{\max} = 0.0001$ eV in the fixed- η mode of army ants algorithm, for 100000 trajectories.^a

Method	η_0	P_R	$\langle \nu' \rangle$	$\langle j' \rangle$	P_Q	$\langle \nu'' \rangle$	$\langle j'' \rangle$	Time ^b (hours)
Quantum	-	1.21×10^{-8}	0.90	11.6	3.35×10^{-8}	0.93	3.28	-
Anteater ^c	0.0	0	-	-	0	-	-	9.4
TFS-	1.0×10^{-2}	$(1.35 \pm 0.04) \times 10^{-8}$	1.30 ± 0.02	13.4 ± 0.11	$(3.59 \pm 0.05) \times 10^{-8}$	1.19 ± 0.02	5.03 ± 0.07	10.2
FSTU-	1.0×10^{-2}	$(1.39 \pm 0.03) \times 10^{-8}$	1.23 ± 0.02	12.3 ± 0.09	$(3.39 \pm 0.05) \times 10^{-8}$	1.07 ± 0.01	5.17 ± 0.09	9.9
FSTU+	1.0×10^{-2}	$(1.38 \pm 0.03) \times 10^{-8}$	1.24 ± 0.02	12.2 ± 0.10	$(3.38 \pm 0.05) \times 10^{-8}$	1.07 ± 0.01	5.17 ± 0.09	10.0
FSTU $\nabla\nabla$	1.0×10^{-2}	$(1.33 \pm 0.03) \times 10^{-8}$	1.33 ± 0.02	12.5 ± 0.11	$(3.28 \pm 0.05) \times 10^{-8}$	1.13 ± 0.02	4.83 ± 0.07	10.0

^aError estimates were calculated using 10000 bootstrap cycles.

^bOn four processors.

^cAnteater calculations for 100000 trajectories fail to show any statistics.

Table IV. Results for calculations using FSTU ∇V method for $Y^* + \text{RH}$ ($\nu = 0, j = 0$), for the scattering energy 1.10 eV with $U_{12}^{\text{max}} = 0.0001$, using 100000 trajectories for both modes of army ants algorithm.^a

mode	Input parameter	P_R	$\langle \nu' \rangle$	$\langle j' \rangle$	P_Q	$\langle \nu'' \rangle$	$\langle j'' \rangle$	Time (hours)
fixed- η	1.0×10^{-2}	$(1.33 \pm 0.03) \times 10^{-8}$	1.33 ± 0.02	12.5 ± 0.11	$(3.28 \pm 0.05) \times 10^{-8}$	1.13 ± 0.02	4.83 ± 0.07	10.0
k^b	7.95×10^{12}	$(1.34 \pm 0.03) \times 10^{-8}$	1.34 ± 0.02	12.3 ± 0.11	$(3.34 \pm 0.05) \times 10^{-8}$	1.15 ± 0.01	4.65 ± 0.07	10.2

^aError estimates were calculated using 10000 bootstrap cycles.

^bOn four processors

^cIn units of s^{-1} .

Table V. Results for quantum mechanical scattering and semiclassical army ants calculations for $Y^* + RH$ ($\nu = 0, j = 0$) with $U_{12}^{\max} = 0.0001$ eV. The results are compared at the three energies and also averaged over three values of scattering energy.

Method	Energy (eV)	P_R	$\langle \nu' \rangle$	$\langle j' \rangle$	P_Q	$\langle \nu'' \rangle$	$\langle j'' \rangle$
Quantum	1.07	8.64×10^{-9}	1.01	11.4	2.87×10^{-8}	0.97	2.24
	1.10	1.21×10^{-8}	0.90	11.6	3.35×10^{-8}	0.93	3.28
	1.13	1.33×10^{-8}	0.78	12.4	3.28×10^{-8}	0.90	3.60
	Average	1.13×10^{-8}	0.89	11.8	3.16×10^{-8}	0.93	3.04
Army ants ^a	1.07	$(1.16 \pm 0.03) \times 10^{-8}$	1.41 ± 0.02	12.0 ± 0.12	$(3.64 \pm 0.05) \times 10^{-8}$	1.16 ± 0.01	4.48 ± 0.06
	1.10	$(1.33 \pm 0.03) \times 10^{-8}$	1.33 ± 0.02	12.5 ± 0.11	$(3.28 \pm 0.05) \times 10^{-8}$	1.13 ± 0.02	4.83 ± 0.07
	1.13	$(1.29 \pm 0.03) \times 10^{-8}$	1.40 ± 0.02	12.5 ± 0.12	$(3.04 \pm 0.05) \times 10^{-8}$	1.15 ± 0.01	4.84 ± 0.08
	Average	$(1.26 \pm 0.03) \times 10^{-8}$	1.38 ± 0.02	12.3 ± 0.12	$(3.32 \pm 0.05) \times 10^{-8}$	1.15 ± 0.01	4.72 ± 0.07

^aCalculations were performed in the fixed- η mode of the army ants algorithm with $\eta_0 = 1 \times 10^{-2}$ using the FSTU ∇ V method for 100000 trajectories.

Table VI. Time required for calculations on $Y^* + RH$ ($\nu = 0, j = 0$), for the scattering energy 1.10 eV with $U_{12}^{\max} = 0.0001$ eV using either the fixed- η mode or the k mode implementation of army ants algorithm, and estimated time requirement to obtain similar converged results with the anteatr and the ants methods.

Method	Number of trajectories		Time required
	Initial	Final	Hours
Army ants ^a	10^5	10^5	~ 10
Anteatr ^b	10^{10}	10^{10}	$\sim 10^6$
Ants ^c	500	5×10^{273}	$\sim 3 \times 10^{269}$

^aTime required to run in parallel on four 375 MHz Power 3 WinterHawk+ processors on the IBM SP supercomputer.

^bEstimated time required to finish the anteatr calculations using the same number of processors as in the army ants calculations.

^cEstimated time required to finish the initial ants trajectories and all the resulting branches, using the same number and type of processors as in the army ants calculations.

Table VII. Mean unsigned relative errors (in %) for reaction probability, reactive moments, quenching probability, and quenching moments.

Cases	P_R	$\langle \nu' \rangle$	$\langle j' \rangle$	P_Q	$\langle \nu'' \rangle$	$\langle j'' \rangle$
All 6 ^a	15	48	7	8	21	58

^aResults are averaged over the four cases in Table III and the three cases in Table V (for a total of six cases since FSTU ∇ V at 1.10 eV occurs in both tables).

Table A-I. Convergence test with respect to number of bootstrap cycles. The army ants calculations were carried out in the fixed- η mode with $\eta_0 = 1.0 \times 10^{-2}$, using the FSTU ∇ V method, for $Y^* + \text{RH}$ ($\nu = 0, j = 0$), with scattering energy 1.10 eV and $U_{12}^{\text{max}} = 0.0001$.^a

Number of bootstrap cycles	P_R	$\langle \nu' \rangle$	$\langle j' \rangle$	P_Q	$\langle \nu'' \rangle$	$\langle j'' \rangle$	Time ^b (mins)
5000	$(1.33 \pm 0.032) \times 10^{-8}$	1.33 ± 0.02	12.5 ± 0.11	$(3.28 \pm 0.048) \times 10^{-8}$	1.13 ± 0.02	4.83 ± 0.08	5.3
10000	$(1.33 \pm 0.034) \times 10^{-8}$	1.33 ± 0.02	12.5 ± 0.11	$(3.28 \pm 0.046) \times 10^{-8}$	1.13 ± 0.02	4.83 ± 0.07	9.8
20000	$(1.33 \pm 0.034) \times 10^{-8}$	1.33 ± 0.02	12.5 ± 0.11	$(3.28 \pm 0.046) \times 10^{-8}$	1.13 ± 0.02	4.83 ± 0.07	21.2

^a100000 trajectories used for the bootstrap analysis.

^bComputer time for the bootstrap analysis on a single processor.

Table B-I. Effect of the parameter η_0 in the fixed- η mode army ants algorithm calculations, for $Y^* + \text{RH}$ ($\nu = 0, j = 0$), with scattering energy 1.10 eV and $U_{12}^{\text{max}} = 0.0001$, using the FSTU ∇V method.^a

η_0	$N_{\text{traj}}^{\text{b}}$	P_R	$\langle \nu' \rangle$	$\langle j' \rangle$	P_Q	$\langle \nu'' \rangle$	$\langle j'' \rangle$	$N_{\text{traj}}^{\text{P c}}$	$N_{\text{traj}}^{\text{M d}}$
1×10^{-4}	500000	$(1.16 \pm 0.101) \times 10^{-8}$	1.24 ± 0.05	13.1 ± 0.33	$(3.51 \pm 0.121) \times 10^{-8}$	1.04 ± 0.02	5.47 ± 0.18	500000	500000
5×10^{-4}	300000	$(1.23 \pm 0.091) \times 10^{-8}$	1.30 ± 0.05	13.2 ± 0.29	$(3.23 \pm 0.110) \times 10^{-8}$	1.05 ± 0.02	5.12 ± 0.16	247116	300000
1×10^{-3}	200000	$(1.21 \pm 0.092) \times 10^{-8}$	1.33 ± 0.05	12.8 ± 0.27	$(2.83 \pm 0.111) \times 10^{-8}$	1.10 ± 0.02	5.28 ± 0.21	167029	272222
5×10^{-3}	100000	$(1.25 \pm 0.045) \times 10^{-8}$	1.30 ± 0.03	13.0 ± 0.16	$(3.12 \pm 0.057) \times 10^{-8}$	1.13 ± 0.01	5.10 ± 0.10	22429	30864
8×10^{-3}	100000	$(1.28 \pm 0.034) \times 10^{-8}$	1.32 ± 0.02	12.5 ± 0.12	$(3.18 \pm 0.048) \times 10^{-8}$	1.15 ± 0.01	4.67 ± 0.08	15554	30250
1×10^{-2}	100000	$(1.33 \pm 0.034) \times 10^{-8}$	1.33 ± 0.02	12.5 ± 0.11	$(3.28 \pm 0.046) \times 10^{-8}$	1.13 ± 0.01	4.83 ± 0.07	14280	25000
2×10^{-2}	150000	$(1.36 \pm 0.022) \times 10^{-8}$	1.32 ± 0.01	12.4 ± 0.07	$(3.31 \pm 0.032) \times 10^{-8}$	1.17 ± 0.01	4.30 ± 0.05	10261	37500
5×10^{-2}	400000	$(1.19 \pm 0.021) \times 10^{-8}$	1.24 ± 0.01	11.9 ± 0.08	$(3.08 \pm 0.033) \times 10^{-8}$	1.16 ± 0.01	4.50 ± 0.06	29475	100000
1×10^{-1}	500000	$(1.29 \pm 0.032) \times 10^{-8}$	1.31 ± 0.06	12.1 ± 0.11	$(3.21 \pm 0.033) \times 10^{-8}$	1.13 ± 0.02	4.77 ± 0.07	50821	500000

^aError estimates were calculated using 10000 bootstrap cycles.

^bNumber of trajectories used for this row of results.

^cMaximum of the two values of the number of trajectories required to converge P_R and P_Q to the same levels as row 1, as estimated by using Eq. (B1).

^dMaximum number of trajectories required for convergence of all moment quantities ($\langle \nu' \rangle, \langle j' \rangle, \langle \nu'' \rangle$, and $\langle j'' \rangle$) to the levels of row 1, as estimated by using Eq. (B1).

Table C-I. Results for quantum mechanical calculations as a function of scattering energy for the initial condition $Y^* + RH$ ($\nu = 0, j = 0$) with $U_{12}^{\max} = 0.0001$ eV.

Energy (eV)	P_R	$\langle \nu' \rangle$	$\langle j' \rangle$	P_Q	$\langle \nu'' \rangle$	$\langle j'' \rangle$
1.07	8.64×10^{-9}	1.01	11.4	2.87×10^{-8}	0.97	2.24
1.08	8.63×10^{-9}	0.80	12.5	3.00×10^{-8}	0.95	2.52
1.09	9.77×10^{-9}	0.85	12.1	3.25×10^{-8}	0.93	3.00
1.10	1.21×10^{-8}	0.90	11.6	3.35×10^{-8}	0.93	3.28
1.11	1.32×10^{-8}	0.83	11.9	3.32×10^{-8}	0.91	3.54
1.12	1.32×10^{-8}	0.77	12.3	3.29×10^{-8}	0.90	3.61
1.13	1.33×10^{-8}	0.78	12.4	3.28×10^{-8}	0.90	3.60
Average	1.12×10^{-8}	0.85	12.0	3.19×10^{-8}	0.93	3.11