JCTC Journal of Chemical Theory and Computation

Article

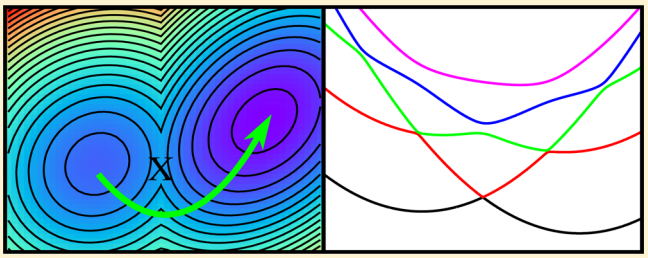
pubs.acs.org/JCTC

An Efficient, Augmented Surface Hopping Algorithm That Includes Decoherence for Use in Large-Scale Simulations

Amber Jain,* Ethan Alguire, and Joseph E. Subotnik

Department of Chemistry, University of Pennsylvania, 231 South 34th Street, Philadelphia, Pennsylvania 19104, United States

ABSTRACT: We propose and implement a highly efficient augmented surface hopping algorithm that (i) can be used for large simulations (with many nuclei and many electronic states) and (ii) includes the effects of decoherence without parametrization. Our protocol is based on three key modifications of the surface hopping methodology: (a) a novel separation of classical and quantum degrees of freedom that treats avoided and trivial crossings efficiently, (b) a multidimensional approximation of the time derivative matrix that avoids explicit construction of the derivative coupling at most time steps, and (c) an efficient approximation for the augmented fewest-switches surface hopping decoherence rate. We will show that this protocol can be several orders of magnitude more efficient than the traditional protocol for large multidimensional problems. Furthermore, the marginal cost for including decoherence effects is now negligible.



1. INTRODUCTION

The fewest switches surface hopping (FSSH)[1] algorithm is currently one of the most popular methods for simulating nonadiabatic dynamics, having been used in the past to simulate a host of processes relevant to scattering, proton and proton-coupled electron transfer, and photoexcited dynamics.[2−6] The successes of FSSH are due to the simple structure of the algorithm and the fact that the algorithm has proven fairly accurate for many problems of interest.[7−16] Moreover, there have been a host of publications in recent years aimed at improving upon the efficiency of the FSSH implementation,[7,17−22] such that large-scale simulations are now possible. Moreover, several programs (e.g., Newton-X,[8] PYXAID,[23,24] SHARC,[25,26] and JADE[27]) have emerged in the past decade that can combine surface hopping with *ab initio* electronic structure packages; surface hopping is also now directly available in Qchem.[28] That being said, combining FSSH with electronic structure packages still remains challenging for large-scale simulations in the presence of many (i.e., more than a handful of) electronic states with frequent crossings, for which FSSH can be computationally prohibitive.[29] Furthermore, if one wishes to include a nonempirical decoherence correction[30,31] for FSSH, the cost of the algorithm only increases, such that decoherence corrections are often included today only through some fixed, parametrized time scale.[32] Let us now discuss these limitations in more detail.

1.1. An Efficient Implementation of FSSH. While Tully's original manuscript in 1990[1] offered the essential equations corresponding to the FSSH algorithm, ref 1 did not offer a detailed prescription for how to implement the equations efficiently for large systems. Compared with standard molecular dynamics, FSSH dynamics can require much more computational effort because, with a naive implementation, the time step must be smaller, derivative couplings must be calculated at every time step, etc. With this in mind, in 1994, Hammes-Schiffer and Tully proposed the first efficient implementation of the algorithm as relevant to large scale simulations of proton transfer.[7] Reference 7 introduced two simple implementation tools for decreasing the cost of FSSH calculations, and these tricks are still essential today:

1.1.1. Enforcing the Separation of Time Scales between Classical and Quantum Degrees of Freedom. A separation of time scales is critical for large-scale simulations, for which computation of the nuclear forces is the most expensive step of the simulation. As recognized in ref 7, because the quantum subsystem always comprises higher frequency modes, that subsystem must be evolved using smaller time steps; vice versa, the classical subsystem can be evolved using large time steps. This separation of time scales leads to a massive speedup of the simulation.[7]

1.1.2. An Efficient Scheme to Compute the Time Derivative Matrix $\mathbf{T}$*.* Because the surface hopping algorithm is most natural in the adiabatic basis, the time derivative matrix $\mathbf{T}$ is absolutely essential in order to integrate the time-dependent Schrödinger equation. The $\mathbf{T}$ matrix quantifies the effect of classical motion on the quantum subsystem, and its elements are defined as

$$T_{jk} = \left\langle \phi_j^{\mathrm{ad}} \middle| \frac{\mathrm{d}\phi_k^{\mathrm{ad}}}{\mathrm{d}t} \right\rangle \tag{1}$$

where $|\phi_j^{\mathrm{ad}}\rangle$ is the wave function for the adiabatic state j, and t is the time.

Received: July 7, 2016
Published: October 7, 2016

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

Traditionally, the **T** matrix is calculated by first computing the nonadiabatic coupling vector $\vec{\mathbf{d}}$, whose elements are defined as

$$\vec{d}_{jk} = \langle \phi_j^{\mathrm{ad}} | \vec{\nabla} \phi_k^{\mathrm{ad}} \rangle \tag{2}$$

and then invoking the following equality:

$$\mathbf{T} = \vec{v} \cdot \vec{d} \tag{3}$$

where $\vec{v}$ is the nuclear velocity. Because derivative coupling vectors are expensive to calculate with *ab initio* electronic structure packages, and because the number of derivative couplings grows quadratically with the number of electronic states, computing derivative couplings can be the most expensive step in the calculation. In ref 7, Hammes-Schiffer and Tully argued that, rather than calculate derivative couplings, a better approach was to evaluate the **T** matrix directly using an efficient overlap-based scheme:

$$T_{jk}(t_0 + \mathrm{d}t_c/2) = \frac{1}{2\mathrm{d}t_c}[U_{jk}(\mathrm{d}t_c) - U_{kj}(\mathrm{d}t_c)] \tag{4}$$

where $\mathrm{d}t_c$ is the classical time step, and U_{jk} are the elements of the overlap matrix given by

$$U_{jk}(\tau) = \langle \phi_j^{\mathrm{ad}}(t_0) | \phi_k^{\mathrm{ad}}(t_0 + \tau) \rangle \tag{5}$$

1.1.3. Recent Developments: Time Steps and Trivial Crossings. Even with the speedups above, however, it is important to note that in the presence of conical intersections and/or "trivial crossings" (i.e., crossings with zero or nearly zero diabatic coupling), the required classical and quantum time steps can be arbitrarily small and, with a naive implementation, FSSH can be very expensive to converge.[19] In general, it would be impractical to compute the **T** matrix elements at every quantum time step, given how rapidly such matrix elements can change along a given trajectory. In recent years, there have been several prescriptions for accommodating this difficulty for the quantum time step:

- Tretiak and co-workers proposed a simple scheme to identify trivial crossings based on the values of the overlap matrix elements. If the overlap of the active state with some other state j exceeds a certain threshold, the crossing is considered to be trivial, and a hop to the state j occurs with unit probability.[33,34] The only ambiguity in this approach is choosing a parameter to distinguish real vs trivial crossings.
- Granucci and co-workers recommend using a local diabatic basis to evolve the quantum amplitudes (while evolving the classical degrees of freedom on the adiabatic surfaces). By comparing amplitudes, one can recover accurate hopping rates when only two surfaces cross on a time step.[18]
- Prezhdo and co-workers have suggested propagating the Schrödinger equation in a diabatic basis when possible, and using the final populations at the end of a time step to compute hopping rates.[19] This prescription can be very effective when a diabatic basis is available and when only two states cross at a time.
- Meek and Levine have extended the original Hammes-Schiffer/Tully implementation, arguing that one should time-average the **T** matrix element, such that one obtains a simple expression that will work for larger time steps and conserve normalization with ordinary integrators. For our purposes, we consider the Meek–Levine algorithm optimal for the two-state problem.[20]

Although it might be obvious, we emphasize that one of the problems facing any FSSH-like algorithm is how to enforce energy conservation efficiently for both real and trivial crossings.

1.2. Including Decoherence To Go Beyond Standard FSSH. Beyond advances in the implementation of the FSSH algorithm, since 1990 there have also been several structural changes proposed for the FSSH algorithm itself. In particular, since the early work of Rossky,[35] Hammes-Schiffer,[7] and Truhlar,[36] it has been known that the FSSH algorithm does not properly account for decoherence, that is, the splitting apart of wavepackets. Several research groups have proposed protocols for including decoherence into the FSSH (or Ehrenfest) algorithm[7,37−39] with minimal cost. For condensed phase simulations, these corrections have often invoked the temperature of the bath.[35,39]

Following a recent mapping of the FSSH algorithm into the quantum-classical Liouville Equation,[40] it is now clear that the correct decoherence rate cannot be calculated exactly with noninteracting trajectories. As a result, to estimate a meaningful decoherence rate, we have argued that the best solution is to propagate extra degrees of freedom compared with the standard FSSH method so as to let one trajectory have partial information about another. In particular, it becomes essential to integrate (in time) the difference in forces between different adiabatic surfaces and estimate how wavepackets move apart. With this in mind, Shenvi, Landry, and Subotnik developed an improved algorithm—augmented-FSSH (A-FSSH)—that requires evolving a new set of variables: the moments of the classical position and momentum.[30,40,41,42] Using these new variables, one can estimate a decoherence rate without invoking any parametrization, and the A-FSSH protocol has been shown to accurately correct for the overcoherence issue for a wide range of systems, including the nonadiabatic Marcus regime.[15,43]

Unfortunately, the exact computation of moments (within the A-FSSH formalism) comes with a reasonably large computational cost. For instance, the formal evolution of the position and momentum moments requires a small time step, that is, a quantum rather than classical time step, which leads to significant computational expense. Furthermore, as originally suggested in ref 30, the A-FSSH scheme requires the computation of the full nonadiabatic coupling vector $\vec{\mathbf{d}}$, rather than the **T** matrix. Lastly, the number of moments formally scales quadratically in the number of electronic states, which also leads to considerable expense. For this reason, to date, the largest *ab initio* A-FSSH simulations have involved only two electronic states and 33 atoms.[16]

1.3. Overcoming FSSH's Practical Limitations. Having outlined above the practical limitations of the FSSH algorithm, our objective here is to propose an efficient implementation of the FSSH algorithm that can be easily combined with *ab initio* electronic structure packages to treat many electronic states, while also approximately incorporating the A-FSSH decoherence correction[30] that avoids any unnecessary parametrization. Such a protocol should be able to treat avoided crossings, conical intersections, and trivial crossings efficiently and all on the same footing. To achieve such an objective, we will introduce several modifications of the original FSSH protocol:

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

1. Our first modification allows a large classical time step in the presence of trivial crossings (and conical intersections) that should enforce energy conservation.
2. Our second modification is an extension of the Meek–Levine approach[20] for the computation of the **T** matrix. For the case of multiple electronic states, the Meek–Levine approach decomposes the derivative coupling pairwise between states. While this approach will work well for most crossings with multiple electronic states, there may be failures for the rare case of three of more states crossing at one time step. Thus, we generalize the Meek–Levine norm-preserving interpolation to include such rare cases.
3. Our third modification is an approximation to the moment propagation in A-FSSH. In particular, we will propagate the moments of position and momentum approximately such that updates are required only at each classical time step (as opposed to a quantum time step). Furthermore, our new protocol will not require explicit calculations of the $\vec{\mathbf{d}}$ matrix, but rather only the **T** matrix. Lastly, we will propagate a limited number of moments (linear, not quadratic, in the number of electronic states). Altogether, these adjustments will allow for huge savings and, as we will show below, they seem to introduce only a minimal loss of accuracy for the effective decoherence rate (at least for the models tested thus far).

An outline of the paper is as follows: In section 2 we review the A-FSSH formalism and propose modifications to improve the algorithm's efficiency. Results for a wide range of model systems are presented in section 3. We conclude in section 4

Notation: A plethora of variable names are essential for our discussion below. To make the paper more accessible, all variable names are delineated in Table 1. In all cases, bold variables denote quantum mechanical operators, and arrows denote a vector for the classical degrees of freedom.

Table 1. Variable Names for the Present Paper[a]

variable	definition
$\vec{x}$, $\vec{v}$, $\vec{m}$	classical positions, velocities, and masses
c_j, V_j, $\|\phi_j^{\mathrm{ad}}\rangle$	quantum amplitude, eigenenergy, and eigenfunctions for electronic state j
j, k, n, λ	indices labeling adiabatic (electronic) quantum states, with λ denoting active surface
dt_c, dt_q	time steps for evolving the classical and quantum degrees of freedom, respectively
T	time derivative matrix with elements $T_{jk} = \langle\phi_j^{\mathrm{ad}}\|\frac{d\phi_k^{\mathrm{ad}}}{dt}\rangle$
$\vec{\mathbf{d}}$	nonadiabatic coupling vector with elements $\vec{d}_{jk} = \langle\phi_j^{\mathrm{ad}}\|\vec{\nabla}\phi_k^{\mathrm{ad}}\rangle$
$\mathbf{H}_0$	electronic Hamiltonian
$\vec{\mathbf{F}}$	force matrix with elements $\vec{F}_{jk} = -\langle\phi_j^{\mathrm{ad}}\|\vec{\nabla}H_0\|\phi_k^{\mathrm{ad}}\rangle$
$\delta\vec{\mathbf{R}}$, $\delta\vec{\mathbf{P}}$	moments of position and momentum that dictate decoherence rate

[a]Bold variables represent quantum mechanical operators, and arrows denote vectors representing classical degrees of freedom.

2. METHODS

2.1. Review of the A-FSSH Formalism. The A-FSSH method has been described extensively previously;[2] the algorithm is a fairly straightforward extension of the original FSSH algorithm.[1] Briefly, the coordinates of the total system are divided into classical and quantum coordinates. It is convenient to write the total Hamiltonian for the full system as

$$H = T_{\vec{x}} + H_0 \tag{6}$$

where $T_{\vec{x}}$ is the kinetic energy for the classical coordinates $\vec{x}$ and H_0 is the electronic Hamiltonian. We define the jth adiabatic eigenfunction of H_0 as $|\phi_j^{\mathrm{ad}}\rangle$ and the eigenenergy of H_0 as V_j where

$$H_0|\phi_j^{\mathrm{ad}}\rangle = V_j|\phi_j^{\mathrm{ad}}\rangle \tag{7}$$

There are three key components underlying the A-FSSH algorithm.

2.1.1. Standard FSSH Propagation of Variables. The first component is just the usual FSSH propagation.[1] Namely, the classical and quantum subsystems are evolved according to Newton's equations of motion and the Schrödinger equation, respectively. Specifically, the equations of motion for the classical coordinates are

$$\dot{\vec{x}} = \vec{v} \tag{8}$$

$$\dot{\vec{v}} = -\frac{1}{\vec{m}}\vec{\nabla}V_\lambda \tag{9}$$

where λ is the active surface, and $\vec{v}$ and $\vec{m}$ are the velocities and masses of the classical coordinates, respectively. The state of the quantum subsystem is specified by the electronic wave function, which is expanded as a linear combination of the adiabatic eigenfunctions defined in eq 7 as follows:

$$|\Psi\rangle = \sum_j c_j|\phi_j^{\mathrm{ad}}\rangle \tag{10}$$

The Schrödinger equation for the quantum amplitudes c_j is given by

$$i\hbar\dot{c}_j = V_j c_j - i\hbar\sum_k T_{jk}c_k \tag{11}$$

where the time derivative matrix elements T_{jk} are defined in eq 1.

2.1.2. Standard Tully-Style Hopping with Jasper and Truhlar's Velocity Reversal. The second component of the A-FSSH algorithm is just the standard FSSH hopping protocol, whereby trajectories switch adiabatic surfaces according to the evolution of the quantum probabilities $|c_j|^2$. The probability of hopping from state λ to j over a time step dt is

$$g_{\lambda j} = \frac{-2Re(c_j c_\lambda^* T_{j\lambda})}{|c_\lambda|^2}\,dt \tag{12}$$

If $g_{\lambda j}$ is negative, we set $g_{\lambda j} = 0$. As suggested by Herman[44−46] and Tully[1] (and confirmed by Kapral[47]), upon hopping the velocities of the classical coordinates are adjusted to conserve energy. The adjustment is made along the direction of the nonadiabatic coupling vector defined in eq 2.

In case of a frustrated hop (i.e., if the kinetic energy corresponding to the velocity along the direction of $\vec{d}_{\lambda,j}$ is not sufficient to maintain energy conservation) we closely follow the recommendations of Jasper and Truhlar.[48] The velocity along $\vec{d}_{\lambda,j}$ is reversed if $(\vec{F}_\lambda\cdot\vec{d}_{\lambda,j})(\vec{F}_j\cdot\vec{d}_{\lambda,j}) < 0$ and $(\vec{F}_j\cdot\vec{d}_{\lambda,j})(\vec{v}\cdot\vec{d}_{\lambda,j}) < 0$, where $\vec{F}_j$ is the force on the jth surface. We have shown previously that this velocity reversal is a necessary ingredient for the purposes of computing rates.[15,49] Recently, it has been suggested (based on empirical evidence

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

alone) that this velocity reversal improves detailed balance in surface hopping.[50]

2.1.3. Position and Momentum Moments Plus Decoherence. The third and final component of the A-FSSH algorithm is the treatment of decoherence whereby, following the work of Rossky,[35] Hammes-Schiffer,[7] and Truhlar,[36] we reset $c_\lambda = 1$ after a nonadiabatic crossing event occurs, whenever the resulting wavepackets separate. To compute a decoherence rate, A-FSSH propagates an extra set of variables along each trajectory.[30,40,51] These variables are the moments $\boldsymbol{\delta}\vec{\mathbf{R}}$ and $\boldsymbol{\delta}\vec{\mathbf{P}}$, defined as

$$\boldsymbol{\delta}\vec{\mathbf{R}} = \mathrm{Tr}_N[(\vec{R} - \vec{R}_{SH})\boldsymbol{\rho}] \tag{13}$$

$$\boldsymbol{\delta}\vec{\mathbf{P}} = \mathrm{Tr}_N[(\vec{P} - \vec{P}_{SH})\boldsymbol{\rho}] \tag{14}$$

where the trace is over the nuclear coordinates, $\boldsymbol{\rho}$ is the density matrix for the combined (classical and quantum) system, $\vec{\mathbf{R}}$ and $\vec{\mathbf{P}}$ are the position and momentum operators for the classical system and $(\vec{R}_{SH}, \vec{P}_{SH})$ are the phase space coordinates for a surface hopping trajectory. If we take the time derivative of eqs 13 and 14 and follow the steps of ref 42 (whereby we expand the potential to first order in position), we find[30]

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\vec{R}_{jk} = \left[-\frac{i}{\hbar}\mathbf{V} - \mathbf{T}, \boldsymbol{\delta}\vec{\mathbf{R}}\right]_{jk} + \frac{\delta\vec{P}_{jk}}{\vec{m}} \tag{15}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\vec{P}_{jk} = \left[-\frac{i}{\hbar}\mathbf{V} - \mathbf{T}, \delta\vec{\mathbf{P}}\right]_{jk} + \frac{1}{2}(\boldsymbol{\delta}\vec{\mathbf{F}}\boldsymbol{\sigma} + \boldsymbol{\sigma}\boldsymbol{\delta}\vec{\mathbf{F}})_{jk} \tag{16}$$

where the potential matrix is defined by $V_{jk} = V_j\delta_{jk}$, and $\sigma_{jk} = c_j c_k^*$ is the quantum density matrix. Here $\delta\vec{F}_{jk} = \vec{F}_{jk} - \vec{F}_{\lambda\lambda}\delta_{jk}$, where the force matrix elements are given by $F_{jk} = -\langle\phi_j^{ad}|\vec{\nabla}H_0|\phi_k^{ad}\rangle$ and, as usual, λ is the active surface. Note that by integrating eqs 15 and 16, we are effectively integrating the difference in forces.

The decoherence rate for electronic state n is then estimated to be[30]

$$\frac{1}{\tau_d^{n\lambda}} = -\frac{\frac{\mathrm{d}}{\mathrm{d}t}|\sigma_{n\lambda}|}{|\sigma_{n\lambda}|} \tag{17}$$

$$\simeq \frac{\delta\vec{F}_{nn}\cdot(\delta\vec{R}_{nn} - \delta\vec{R}_{\lambda\lambda})}{2\hbar} - \frac{2|\vec{F}_{\lambda n}\cdot(\delta\vec{R}_{nn} - \delta\vec{R}_{\lambda\lambda})|}{\hbar} \tag{18}$$

The derivation of eq 18 is rather involved and requires quite a few assumptions; see ref 30. Briefly, the first term is derived by assuming $\vec{d}_{n\lambda} = 0$ and then invoking a frozen Gaussian approximation. The second term arises naturally (but in a slightly different form) when the derivative coupling is nonzero. The artificial factor of 2 multiplying the second term, as well as the absolute value, were introduced to discourage collapses in the region where $\vec{d}_{n\lambda}$ is non-negligible. In practice, we collapse the amplitude for state n if $\frac{\mathrm{d}t_c}{\tau_d^{n\lambda}}$ is larger than a random number chosen with a uniform distribution between 0 and 1.

As just discussed, in order for eq 18 to be a meaningful rate of decoherence, one requires several approximations: linear approximations of the potential, frozen Gaussians, etc. These approximations are likely not valid for long times. With that in mind, Landry and Subotnik suggested resetting the moments for state n with the rate[30]

$$\frac{1}{\tau_r^{n\lambda}} = \frac{-\delta\vec{F}_{nn}\cdot(\delta\vec{R}_{nn} - \delta\vec{R}_{\lambda\lambda})}{2\hbar} \tag{19}$$

so that the final algorithm would at least be stable and, almost always, eq 18 functions as an effective lower bound for the true decoherence rate.

2.2. Algorithmic and Implementation Modifications. The A-FSSH algorithm described above has been benchmarked extensively in the literature and has usually performed well.[2,6] That being said, if either the equations above are propagated without approximation and/or the implementation is not optimized, the A-FSSH algorithm will be significantly slower than the standard FSSH approach. We will now outline a set of modifications that render an efficient, approximate A-FSSH algorithm.

2.2.1. Classical and Quantum Time Step. As discussed above and in the literature,[7] an efficient surface hopping algorithm requires a clear distinction between (long) classical and (short) quantum time steps. Let the large time step be called $\mathrm{d}t_c$ and the short time step $\mathrm{d}t_q$. Classical variables $\vec{x},\vec{v}$ are propagated over the large time step, and the quantum variables (c_j) are evolved over the short time step $\mathrm{d}t_q$; the adiabatic energies and time-derivative matrix are interpolated between classical time steps.

In choosing $\mathrm{d}t_q$, we require both that $\mathrm{d}t_q$ should be small enough to accurately represent fast quantum oscillations and also that $\mathrm{d}t_q$ should be a rational fraction of $\mathrm{d}t_c$. With this in mind, let $\mathrm{d}t_q'$ be the following:

$$\mathrm{d}t_q' = \min\begin{pmatrix}\mathrm{d}t_c\\ 0.02/\mathrm{Max}[\mathbf{T}]\\ 0.02\hbar/\mathrm{Max}[\mathbf{V} - \bar{V}]\end{pmatrix} \tag{20}$$

where $\mathrm{Max}[\mathbf{A}]$ refers to the maximum absolute value of the elements the matrix $\mathbf{A}$, and $\bar{V}$ is mean value of all adiabatic energies. We then choose $\mathrm{d}t_q$ to be

$$\mathrm{d}t_q \equiv \frac{\mathrm{d}t_c}{\mathrm{nint}(\mathrm{d}t_c/\mathrm{d}t_q')} \tag{21}$$

where $\mathrm{nint}(x)$ is the nearest integer function. Here, we have assumed $\mathbf{T}$ to be constant over the time period $\mathrm{d}t_c$ (see the next subsection for more details).

Armed with two different time steps, we propagate surface hopping dynamics as follows:

- We employ the velocity Verlet algorithm for numerical integration of the classical equations of motion [see eqs 8 and 9].[52]
- We employ the Runge−Kutta fourth [RK4] order method for evolving the quantum amplitudes [see eq 11].[52]
- As the quantum amplitudes are evolved, we compute the hopping probability to state j using eq 12. A hop to every state based on these probabilities is attempted (using a random number as described in ref 7). In case a hop is successful, the corresponding state j is saved until the end of the classical time step.
- Suppose a hop to a state j has occurred. For an efficient algorithm that treats sharp crossings correctly and efficiently (i.e., without requiring a very small time step), we must distinguish two cases:

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

- On the one hand, if there is only a small deviation in energy during the classical time step, the active state is changed to $\lambda = j$, and the velocity is adjusted to conserve energy.
- On the other hand, if one finds a large deviation in energy, we use the force derived from state j to evolve the velocity in the velocity Verlet algorithm as

$$\vec{v}(t_0 + \mathrm{d}t_c) = \vec{v}(t_0) + \frac{1}{2\vec{m}}\,\mathrm{d}t_c(\vec{F}_{\lambda\lambda}(t_0) + \vec{F}_{jj}(t_0 + \mathrm{d}t_c)) \tag{22}$$

Again, there are two possibilities: (a) On the one hand, if eq 22 leads to conservation of energy within the desired amount, then we switch the active state to $\lambda = j$, and set the moments $\delta\vec{\mathbf{R}}$, $\delta\vec{\mathbf{P}}$ to be zero. Since the energy is conserved, there is no need for any velocity adjustment. (b) On the other hand, if eq 22 still suffers from a large energy deviation, there is no alternative but to evolve from t_0 to $t_0 + \mathrm{d}t_c$ using a smaller classical time step until the energy deviation is below the desired threshold. Note that, for these smaller classical time steps, only the force on the active surface is needed—no hop is attempted during the smaller classical time steps.

2.2.2. Computation of T Matrix. As described above, the traditional (and slowest) approach for computing the $\mathbf{T}$ matrix is to calculate the nonadiabatic derivative coupling vector $\vec{\mathbf{d}}$. Alternatively, Hammes-Schiffer and Tully estimated the $\mathbf{T}$ matrix elements using an overlap-based scheme [see eq 4].[7]

Recently, Meek and Levine pursued a slightly different, and even better, approach, the norm-preserving interpolation (NPI) method, where the adiabatic wave functions are approximated to vary linearly between times t_0 and $t_0 + \mathrm{d}t_c$ while maintaining orthonormality at all times.[20] The $\mathbf{T}$ matrix elements in this approach are averaged over the classical time step as follows

$$T_{jk}(t_0 + \mathrm{d}t_c/2) = \frac{1}{\mathrm{d}t_c}\int_{t_0}^{t_0+\mathrm{d}t_c} \mathrm{d}\tau \left\langle \phi_j^{\mathrm{ad}}(t_0)\middle|\mathbf{U}^{\dagger}(\tau)\frac{\partial}{\partial\tau}\mathbf{U}(\tau)\middle|\phi_k^{\mathrm{ad}}(t_0)\right\rangle \tag{23}$$

Meek and Levine have derived an analytical expression for eq 23 in the case of two electronic states. For the case of three or more electronic states, the Meek–Levine approach interpolates the derivative coupling pairwise between adiabats, which is usually an excellent approximation.

Below, we will extend the Meek–Levine approach to include the possibility of three of more states crossing together at the same time. We begin by considering the time derivative of the overlap matrix defined in eq 5

$$\frac{\mathrm{d}U_{jk}}{\mathrm{d}\tau} = \left\langle \phi_j^{\mathrm{ad}}(t_0)\middle|\frac{\mathrm{d}\phi_k^{\mathrm{ad}}(t_0+\tau)}{\mathrm{d}\tau}\right\rangle \tag{24}$$

$$= \sum_m \langle\phi_j^{\mathrm{ad}}(t_0)|\phi_m^{\mathrm{ad}}(t_0+\tau)\rangle\left\langle \phi_m^{\mathrm{ad}}(t_0+\tau)\middle|\frac{\mathrm{d}\phi_k^{\mathrm{ad}}(t_0+\tau)}{\mathrm{d}\tau}\right\rangle \tag{25}$$

$$= \sum_k U_{jm}T_{mk} \tag{26}$$

This gives

$$\dot{\mathbf{U}} = \mathbf{U}\mathbf{T} \tag{27}$$

Solving the differential equation leads to

$$\mathbf{U}(\tau) \simeq \mathrm{e}^{\int_{t_0}^{t_0+\tau}\mathbf{T}\,\mathrm{d}t'} \tag{28}$$

and thus, when averaged over the classical time step $\mathrm{d}t_c$, the $\mathbf{T}$ matrix at the midpoint in time is

$$\mathbf{T}(t_0 + \mathrm{d}t_c/2) = \frac{1}{\mathrm{d}t_c}\log[\mathbf{U}(\mathrm{d}t_c)] \tag{29}$$

Equation 29 reduces to the Meek–Levine expression in the case of two electronic states.

We numerically compute the logarithm in eq 29 using a Schur decomposition.[53] There are two important numerical issues worth noting:

- The logarithm of a unitary matrix can be complex. When implementing eq 29, we choose the sign of the adiabatic eigenfunctions such that all the diagonal entries of the $\mathbf{U}$ matrix are positive. In so doing, the $\mathbf{U}$ matrix should correspond to a proper rotation with a real logarithm (at least if the rotation is not far from the identity).

  In the case that any diagonal entry is exactly zero, as is the case for a truly trivial crossing, we recompute the adiabatic eigenfunctions after adding a small value ϵ to all entries of the Hamiltonian

$$\langle i|H_0^{\epsilon}|j\rangle \equiv \langle i|H_0|j\rangle + \epsilon \tag{30}$$

  Here $|i\rangle$ represents the ith basis function and the new adiabats are

$$H_0^{\epsilon}|\phi_j^{\epsilon}\rangle = V_j^{\epsilon}|\phi_j^{\epsilon}\rangle \tag{31}$$

  ϵ can be arbitrarily small, for example, $\epsilon = 10^{-10}$ cm^{-1}. We now choose the sign of $|\phi_j^{\mathrm{ad}}(t_0 + \mathrm{d}t_c)\rangle$ in two steps. In the first step, the sign of $|\phi_j^{\epsilon}(t_0 + \mathrm{d}t_c)\rangle$ is chosen such that

$$\langle\phi_j^{\mathrm{ad}}(t_0)|\phi_j^{\epsilon}(t_0 + \mathrm{d}t_c)\rangle > 0 \tag{32}$$

  holds true for all j. In the second step, we choose the sign of $|\phi_j^{\mathrm{ad}}(t_o + \mathrm{d}t_c)\rangle$ by ensuring that its overlap with $|\phi_j^{\epsilon}(t_0 + \mathrm{d}t_c)\rangle$ is positive:

$$\langle\phi_j^{\epsilon}(t_0 + \mathrm{d}t_c)|\phi_j^{\mathrm{ad}}(t_0 + \mathrm{d}t_c)\rangle > 0 \tag{33}$$

- For purposes of stability, we orthogonalize the $\mathbf{U}$ matrix. We use the symmetric Löwdin orthogonalization scheme[54] for this purpose:

$$\mathbf{U} \equiv \mathbf{U}[(\mathbf{U})^{\mathrm{T}}\mathbf{U}]^{-1/2} \tag{34}$$

  This orthogonalization is necessary for the case in which we use an incomplete electronic basis (which is the usual case in practice).

2.2.3. Decoherence. As described above, the computation of the moments $\delta\vec{\mathbf{R}}$ and $\delta\vec{\mathbf{P}}$ is very expensive (in practice) relative to a standard FSSH algorithm: propagating eqs 15 and 16 requires small time steps (on the same order of magnitude as the quantum time step) as well as $N_{\mathrm{QM}}^3 \cdot N_{\mathrm{cl}}$ FLOPS for matrix multiplication (where N_{QM} is the number of quantum states included, and N_{cl} is the number of classical degrees of freedom). To achieve an enormous increase in numerical efficiency, we will consider a few approximations. To begin with, we will transform all moments into a diabatic

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256–5268

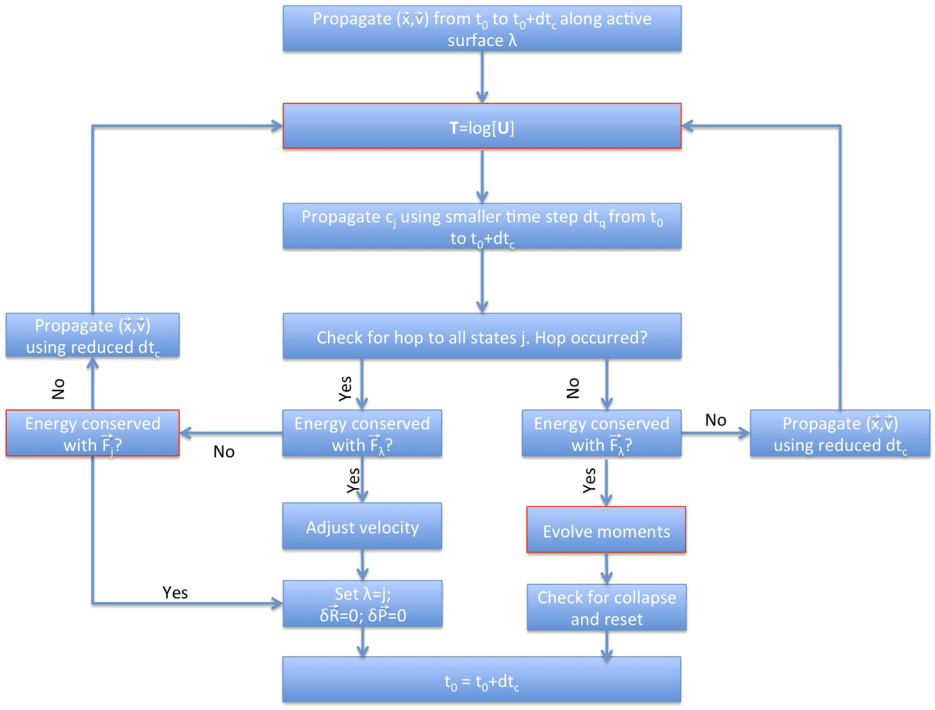


Figure 1. A flowchart depicting the algorithm. The three modifications proposed in this paper are highlighted in red boxes. For access to the open-source codes, see ref 55.

representation, which we define to be the adiabatic basis at some reference time t_0:

$$\delta\tilde{\tilde{\mathbf{R}}}(t) = \mathbf{W}^{\dagger}\delta\vec{\mathbf{R}}(t)\mathbf{W} \tag{35}$$

$$\delta\tilde{\tilde{\mathbf{P}}}(t) = \mathbf{W}^{\dagger}\delta\vec{\mathbf{P}}(t)\mathbf{W} \tag{36}$$

Here, $\mathbf{W}$ is the unitary matrix defined as

$$\mathbf{W} = \exp\left(\int_{t_0}^{t} \mathrm{d}t'\left(-\frac{i}{\hbar}\mathbf{V} - \mathbf{T}\right)\right) \tag{37}$$

Note that, if we take the time derivative of eqs 35 and 36 (and use eqs 15 and 16), several terms vanish simply because of this adiabatic-to-diabatic transformation. We now find

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\tilde{\tilde{R}}_{jk} = \frac{\delta\tilde{\tilde{P}}_{jk}}{\vec{m}} \tag{38}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\tilde{\tilde{P}}_{jk} = \frac{1}{2}(\delta\tilde{\tilde{\mathbf{F}}}\tilde{\sigma} + \tilde{\sigma}\delta\tilde{\tilde{\mathbf{F}}})_{jk} \tag{39}$$

with

$$\tilde{\sigma} = \mathbf{W}^{\dagger}\boldsymbol{\sigma}\mathbf{W} \tag{40}$$

$$\delta\tilde{\tilde{\mathbf{F}}} = \mathbf{W}^{\dagger}\delta\vec{\mathbf{F}}\mathbf{W} \tag{41}$$

At this point, we discard all off-diagonal matrix elements (for a massive speed up), and we assume that all of the matrix moments $\{\delta\tilde{\tilde{\mathbf{R}}}, \delta\tilde{\tilde{\mathbf{P}}}, \delta\tilde{\tilde{\mathbf{F}}}\}$ and $\{\delta\vec{\mathbf{R}}, \delta\vec{\mathbf{P}}, \delta\vec{\mathbf{F}}\}$ are diagonal. Thus, the equations of motions further simplify, such that

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\tilde{\tilde{R}}_{jj} = \frac{\delta\tilde{\tilde{P}}_{jj}}{\vec{m}} \tag{42}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\tilde{\tilde{P}}_{jj} = \delta\tilde{\tilde{F}}_{jj}\tilde{\sigma}_{jj} \tag{43}$$

Finally, we set the reference time t_0 to be the starting time for a classical time step and the time t to be the final time for a classical time step, so that we initialize:

$$\delta\tilde{\tilde{R}}_{jj}(t_0) = \delta\vec{R}_{jj}(t_0) \tag{44}$$

$$\delta\tilde{\tilde{P}}_{jj}(t_0) = \delta\vec{P}_{jj}(t_0) \tag{45}$$

$$\delta\tilde{\tilde{F}}_{jj}(t_0) = \delta\vec{F}_{jj}(t_0) \tag{46}$$

The velocity Verlet algorithm is used to evolve eqs 42 and 43, such that

$$\delta\tilde{\tilde{R}}_{jj}(t_0 + \mathrm{d}t_c) = \delta\tilde{\tilde{R}}_{jj}(t_0) + \frac{\delta\tilde{\tilde{P}}_{jj}(t_0)}{\vec{m}}\mathrm{d}t_c + 0.5\frac{\delta\tilde{\tilde{F}}_{jj}(t_0)}{\vec{m}}\sigma_{jj}(t_0)\,\mathrm{d}t_c^2 \tag{47}$$

$$\delta\tilde{\tilde{P}}_{jj}(t_0 + \mathrm{d}t_c) = \delta\tilde{\tilde{P}}_{jj}(t_0) + \frac{1}{2}(\delta\tilde{\tilde{F}}_{jj}(t_0) + \delta\tilde{\tilde{F}}_{jj}(t_0 + \mathrm{d}t_c))\sigma_{jj}(t_0)\,\mathrm{d}t_c \tag{48}$$

Assuming the classical time step is not too large, we approximate

$$\mathbf{W} \simeq \exp\left(\int_{t_0}^{t}\mathrm{d}t'\left(-\frac{i}{\hbar}\mathbf{V}\right)\right)\exp(-\mathbf{T}(t - t_0)) = \exp\left(\int_{t_0}^{t}\mathrm{d}t'\left(-\frac{i}{\hbar}\mathbf{V}\right)\right)\mathbf{U}$$

so that using eq 41 (and again dropping off-diagonal contributions), we get:

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

$$\delta\tilde{\vec{F}}_{jj}(t_0 + \mathrm{d}t_c) = \sum_k |U_{jk}|^2 \delta\vec{F}_{kk}(t_0 + \mathrm{d}t_c) \tag{49}$$

The moments are then transformed back into the instantaneous adiabatic basis in a similar fashion (using eqs 35 and 36):

$$\delta\vec{R}_{jj}(t_0 + \mathrm{d}t_c) = \sum_k |U_{kj}|^2 \delta\tilde{\vec{R}}_{kk}(t_0 + \mathrm{d}t_c) \tag{50}$$

$$\delta\vec{P}_{jj}(t_0 + \mathrm{d}t_c) = \sum_k |U_{kj}|^2 \delta\tilde{\vec{P}}_{kk}(t_0 + \mathrm{d}t_c) \tag{51}$$

Equations 44−51 provide a very simplified (and much faster) scheme for propagating moments approximately. Using eq 18, we may now compute a decoherence rate. However, there remains a practical problem in evaluating the second term of eq 18, whereby the nonadiabatic couplings at each classical time step are required. Note, though, that this additional term was effectively introduced only to discourage collapses in regions of strong nonadiabatic coupling. Thus, here, we make yet another approximation for the decoherence rates by considering only the component of $\delta\vec{\mathbf{R}}$ along the direction of the nuclear velocity $\vec{v}$:

$$\delta\vec{F}_{\lambda n}\cdot(\delta\vec{R}_{nn} - \delta\vec{R}_{\lambda\lambda}) \simeq \delta\vec{F}_{\lambda n}\cdot\vec{v}\left(\frac{\delta\vec{R}_{nn}\cdot\vec{v} - \delta\vec{R}_{\lambda\lambda}\cdot\vec{v}}{\vec{v}\cdot\vec{v}}\right) \tag{52}$$

$$= T_{\lambda n}\cdot(V_\lambda - V_n)\left(\frac{\delta\vec{R}_{nn}\cdot\vec{v} - \delta\vec{R}_{\lambda\lambda}\cdot\vec{v}}{\vec{v}\cdot\vec{v}}\right) \tag{53}$$

Substituting eq 53 into eq 18 gives the final approximation to the decoherence rate

$$\frac{1}{\tau_{n\lambda}} = \frac{\delta\vec{F}_{nn}\cdot(\delta\vec{R}_{nn} - \delta\vec{R}_{\lambda\lambda})}{2\hbar} - \frac{2|T_{\lambda n}\cdot(V_\lambda - V_n)(\delta\vec{R}_{nn} - \delta\vec{R}_{\lambda\lambda})\cdot\vec{v}|}{\hbar\vec{v}\cdot\vec{v}} \tag{54}$$

The expression for the reset rate remains unchanged (as in eq 19).

In arriving at eq 54, we have made two key approximations: (1) $\{\delta\vec{\mathbf{R}}, \delta\vec{\mathbf{P}}, \delta\vec{\mathbf{F}}\}$ are assumed to be diagonal, and (2) we consider only the component of $\delta\vec{\mathbf{R}}$ along $\vec{v}$ for computation of the second term. Both these approximations are made as an ansatz to gain enormous speedups in time, and as we will show that they still capture the decoherence rates qualitatively. Note that the modified algorithm remains parameter-free.

2.3. Algorithm. In summary, for a simplified but maximally efficient A-FSSH algorithm, we have proposed the following concrete steps (see the flowchart in Figure 1; ref 55 provides the link to the open-source codes):

Step 1. Initialize $\vec{x},\vec{v},\lambda,c_j,\delta\vec{\mathbf{R}},\delta\vec{\mathbf{P}}$ based on the requirements of the simulation (see Table 1 for notation).

Step 2. Nuclear Classical Evolution: Evolve $\vec{x},\vec{v}$ from current time t_0 using a classical time step $\mathrm{d}t_c$ (see eqs 8 and 9). We use velocity Verlet for this step.

Step 3. Compute the overlap matrix $\mathbf{U}$ using eq 5. The signs of the adiabatic eigenfunctions are chosen such that all the diagonal entries of $\mathbf{U}$ are positive. In case any diagonal entry is zero, we follow the protocol described in section 2.2.2 (see eqs 31−33). If the $\mathbf{U}$ matrix is not exactly orthogonal (as would be true in any incomplete electronic basis), we orthogonalize $\mathbf{U}$ using eq 34. The time derivative matrix $\mathbf{T}$ is then evaluated using eq 29.

Step 4. Quantum Electronic Evolution: The quantum time step $\mathrm{d}t_q$ is chosen as in eq 20. To propagate from time t_0 to $t_0 + \mathrm{d}t_c$ in intervals of $\mathrm{d}t_q$, $\mathrm{d}t_q$ is minimally altered such that $\mathrm{d}t_c$ will be an integer multiple of $\mathrm{d}t_q$ [see eq 21]. Over the classical time step, we will assume that $\mathbf{T}$ is constant, and the adiabatic energies can be interpolated linearly in time. We now loop from $k = 1$ to $k = \mathrm{d}t_c/\mathrm{d}t_q$:

Step 4a Evolve the quantum amplitudes c_j from $t_0 + (k-1)\,\mathrm{d}t_q$ to $t_0 + k\,\mathrm{d}t_q$ according to eq 11. For this evolution, we use the RK4 method in the adiabatic basis.

Step 4b Compute the hopping probability $h_{\lambda,j} = -2\,\mathrm{d}t_q\,Re[c_j/c_\lambda]T_{j,\lambda}$ from the active surface λ to all other states j. Thereafter, generate a random number, and attempt a hop. If a hop is successful, the corresponding state j is saved, and no more hop attempts are made until the end of this loop over k.

Step 5. For energy conserving calculations (i.e., without a thermostat), pick a tolerance for energy conservation. The energy deviation is computed as the absolute value of the difference of energy obtained after the nuclear evolution in Step 2 and the energy at the previous time step.

Step 5a If this energy deviation is less than the chosen threshold, go to Step 6.

Step 5b If the energy deviation is too large, check if a hop to state j was attempted during the evolution of quantum amplitudes in Step 4.

i. If yes, then use the force of the hopped-to surface at time $t_0+\mathrm{d}t_c$ (and the force of the current surface λ at time t_0) for evolution of velocity from time t_0 to $t_0 + \mathrm{d}t_c$ [see eq 22]. If the new energy is now conserved to within the predefined threshold, set the active state to $\lambda = j$, set the moments to zero, and go to Step 9.

(ii) If the energy deviation is still larger than the desired value, revert back to time t_0 and use a smaller classical time step to evolve until $t_0 + \mathrm{d}t_c$; return to Step 3.

Step 6. If a successful hop to state j was attempted at Step 4 (and energy was conserved to within the desired threshold if applicable), the velocity is adjusted along the direction of $\vec{d}_{\lambda,j}$. *Note that this is the only step requiring computation of* $\vec{d}_{\lambda,j}$.

- In the case of a frustrated hop, following the work of Truhlar et al., the velocity is reversed along the direction of $\vec{d}_{\lambda,j}$ if two conditions are satisfied: (a) $(\vec{F}_\lambda\cdot\vec{d}_{\lambda,j})(\vec{F}_j\cdot\vec{d}_{\lambda,j}) < 0$ and (b) $(\vec{F}_j\cdot\vec{d}_{\lambda,j})(\vec{v}\cdot\vec{d}_{\lambda,j}) < 0$.
- For A-FSSH, all the moments are reset to zero in the case of a successful hop. The moments are not adjusted for frustrated hops.

Step 7. Evolution of the Moments for Decoherence: In the case of A-FSSH, $\delta\vec{\mathbf{R}}$ and $\delta\vec{\mathbf{P}}$ are evolved using eqs 44−51.

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

Step 8. In the case of A-FSSH, check if a collapse or reset for all states $n \neq \lambda$ is required using eqs 54 and 19 and a random number η.

Step 8a A collapse occurs if $\eta < dt_c/\tau_{n\lambda}$ (see eq 54). The new amplitudes become

$$c_j^{\text{new}} = \begin{cases} c_j & j \neq n, j \neq \lambda \\ \dfrac{c_\lambda}{|c_\lambda|}\sqrt{|c_\lambda|^2 + |c_n|^2} & j = \lambda \\ 0 & j = n \end{cases}$$

Step 8b We reset the moments (and set them to zero) if either $\eta < dt_c/\tau_r^{n\lambda}$ (see eq 19) or $\eta < dt_c/\tau_{n\lambda}$:

$$\delta\vec{R}_{nn} = 0 \quad (55)$$

$$\delta\vec{P}_{nn} = 0 \quad (56)$$

Step 9. Finally, change the current time step from t_0 to $t_0 + dt_c$. Return to Step 2 and iterate.

3. RESULTS AND DISCUSSION

We now benchmark the accuracy and efficiency of the above algorithm for a number of model problems. Henceforth, we will refer to the protocol presented in section 2.1 as the "old" A-FSSH protocol, and the protocol presented in section 2.3 as the new A-FSSH protocol. We will show results for five model problems: (i) the spin-Boson problem, (ii) two of the Tully problems, (iii) a nonseparable two-dimensional model problem, and (iv) a linear chain model with five classical and quantum degrees of freedom (which is a rough model of a molecular crystal).

For a thorough benchmark of the computational time, we have done our best to ensure that all simulations be performed under similar environments. For example, comparable simulations are performed on the same cluster (and usually the same node), and we parallelized the simulations by running different trajectories on different nodes. The average time per node per trajectory is reported as the final simulation wall time.

3.1. Spin-Boson Hamiltonian. We start by benchmarking our new protocol for the spin-Boson Hamiltonian, one of the most studied systems modeling electron transfer.[56] The details of the Hamiltonian employed and the corresponding parameters are described in ref 15. Briefly, the model comprises a two-level system coupled linearly to a classical degree of freedom representing the reaction coordinate. This reaction coordinate is further linearly coupled to a harmonic bath; this coupling is assumed to be Ohmic in this work. In practice, we treat the harmonic bath (with couplings given by the Ohmic spectral density) implicitly, and the effects of the bath are simulated using the Langevin equation.

To begin our discussion, consider the severe case of a trivial crossing and zero friction where a trajectory starting on the left diabat should never switch and move along the right diabat. We show such a representative trajectory in Figure 2. For this trajectory, the initial conditions are chosen with enough energy (E = 2000 cm^{-1}) in the left well to reach the diabatic crossing. A time step of 0.5 fs is employed for a total simulation time of 10 ps. Figure 2 shows that the trajectory is highly regular; energy is conserved to within 0.1 cm^{-1}, and the phase space shown in Figure 2a shows no sign of chaos. Note that an accurate computation of the **T** matrix is critical for this computation so that the trajectory never switches diabats to reach the right well. Furthermore, Step 5 in the new protocol is crucial to capture the diabatic curve crossing here; even though our surface hopping dynamics move along the adiabatic surfaces, the classical time step does not need to be reduced for purposes of energy conservation for the entire duration of the trajectory. Note that we have not introduced any parameter for identification of the trivial crossing.[33]

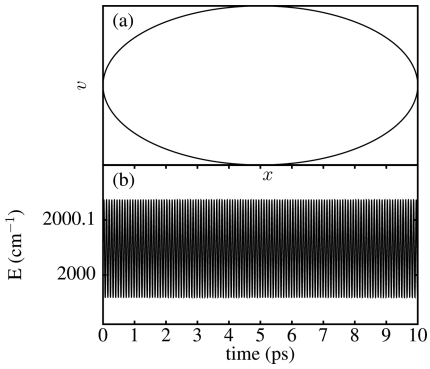


Figure 2. A representative trajectory for the spin-Boson Hamiltonian with diabatic coupling V_c = 0 cm^{-1}, and without any friction at total energy of 2000 cm^{-1}. (a) The trajectory is plotted in phase space. (b) The energy of the trajectory as a function of time. Our new protocol gives a regular trajectory, with a classical time step dt_c = 0.5 fs. Note that, because of Step 5 in our protocol, the classical time step did not need to be reduced to maintain energy conservation for this trajectory.

Next, we compute rate constants with a small diabatic coupling of V_c = 50 cm^{-1} and moderate friction (η/ω_0 = 1). The forward rate constants are computed by choosing the initial conditions in the left well using a Boltzmann distribution, and thereafter fitting the decay of population in the left well for 10 ps averaged over 10000 trajectories.

Figure 3a shows the computed forward rate constants as a function of the classical time step along with the standard Marcus results.[56,57] For small enough time steps, both protocols

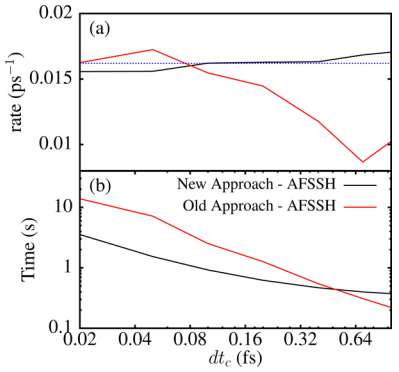


Figure 3. (a) Forward rate constant as a function of the classical time step dt_c for the spin-Boson problem. The parameters of the Hamiltonian are exothermicity ϵ = 400 cm^{-1}, diabatic coupling V_c = 50 cm^{-1}, mass m = 1836 au, frequency ω_0 = 200 cm^{-1}, reorganization energy λ = 6374 cm^{-1}, temperature T = 575.5 K, and friction strength η/ω_0 = 1. See ref 15 for details. The dotted blue line shows the converged results, which matches well with the Marcus result of 0.016 ps^{-1}. The new protocol presented is accurate for the entire range of dt_c shown above, while the old protocol is accurate only for dt_c < 0.1 fs. (b) Simulation wall time per trajectory as dt_c is varied. The new protocol is 2 to 10 times faster than the old protocol for dt_c < 0.1 fs.

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

agree with Marcus theory. However, the new protocol gives highly accurate results for $dt_c < 1$ fs, while the old protocol is accurate only for $dt_c < 0.1$ fs. For $dt_c \in [0.02, 0.1]$ fs, the new protocol is 10 to 2 times faster. The fact that the new protocol is more efficient than the old protocol even for this reduced dimensional model problem is unexpected and perhaps not that important. The modifications presented above were designed to target large scale simulations with many classical degrees of freedom and multiple quantum states. Gains in efficiency (without loss of accuracy) for small systems are unexpected benefits.

3.2. Tully Problems. The next systems we investigate are two of the Tully model problems:[1] the simple avoided crossing (Tully 1) and the extended coupling model (Tully 3). These model problems have been chosen with two goals in mind: (a) Tully 1 was chosen to benchmark the accuracy and efficiency of our modified protocol for treating avoided crossings, and (b) Tully 3 was chosen to benchmark our new simplified scheme for decoherence.

For both systems, the incoming momentum is fixed at $\hbar k = 20$ au, and the results are averaged over 100 000 trajectories. (We require many trajectories here to resolve any subtle changes that may occur as dt_c is varied.) For the new protocol, we choose the energy conservation threshold to be 0.05 cm^{-1} for Tully 1 and 1 cm^{-1} for the Tully 3. Different energy conservation thresholds are used for Tully 1 and Tully 3 because, in the latter case, the region of strong coupling is flat so that one need not integrate changes in potential energy as accurately as in the former case (which is a sloped avoided crossing).

We begin with the results for the Tully 1 model problem. Figure 4 shows (a) the transmission on lower surface (P_L^1), (b)

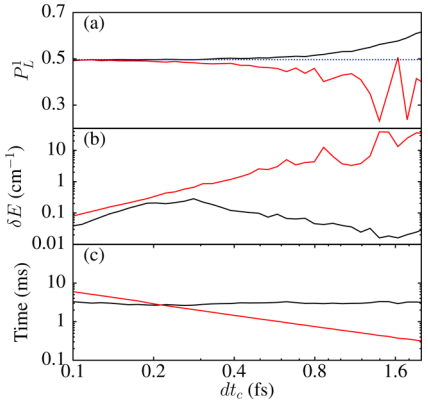


Figure 4. Effect of the classical time step for Tully model system no. 1. (a) The probability of transmission on the lower surface, (b) the standard deviation in energy, and (c) the simulation wall time taken per trajectory in milliseconds. In all cases, the solid black and red lines plot results for the new and the old protocol, respectively. The dotted blue line in part (a) shows the converged transmission probability. The new protocol is accurate for $dt_c < 0.8$ fs, while the old protocol is accurate for $dt_c < 0.4$ fs.

the standard deviation in energy (δE), and (c) the simulation wall time as the time step (dt_c) is varied. For the transmission probability on the lower surface, the new protocol achieves accuracy for a classical time step $dt_c < 0.8$ fs, while the old protocol converges only for $dt_c < 0.4$ fs. The energy is conserved to within 0.1 cm^{-1} for the new protocol over the entire range of dt_c. Since there was never a dynamic check for energy conservation in the old protocol, not surprisingly there is a monotonic increase in the standard deviation of the energy with increasing dt_c. Note that there is no speed-up in the total required time—this is not surprising since we are working with a 1D problem and the old protocol does not even obey strict energy conservation.

Let us now compare results for the Tully 3 model system. Figure 5a compares the reflection probability on the excited

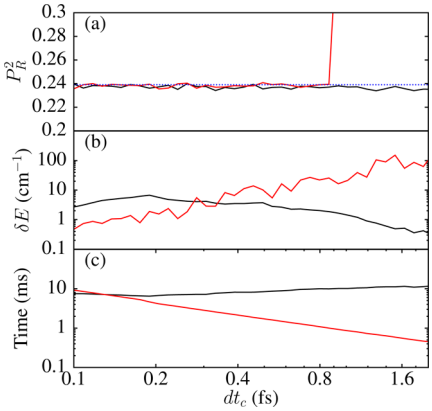


Figure 5. Effect of the classical time step for Tully's extended crossing model problem (i.e., Tully 3). (a) The probability of reflection on the upper surface, (b) the standard deviation in energy, and (c) the simulation wall time taken per trajectory in milliseconds. Both protocols return the correct transmission probability for $dt_c < 0.8$ fs. The solid red and black lines plot the results corresponding to the old and the new protocol, respectively. The dashed blue line in part a shows the converged result.

surface (P_R^2) from our new protocol versus our old protocol. We obtain very good agreement between the two protocols. Note that decoherence plays an important role for this problem;[30,42] without any decoherence, the reflected probability on the excited state is roughly 0.06 (compared to 0.24 with decoherence). These decoherence effects are correctly captured by the new protocol. Note, however, that our new protocol does not achieve computational savings for the same reasons as for Tully 1 above.

So far we have shown the results for a fixed value of the incoming momentum. For a thorough benchmark, we also compare results for a wide range of incoming momentum values. For these calculations, an average over 5000 trajectories is taken, with a time step of 0.5 fs for the new protocol and 0.1 fs for the old protocol. Figure 6 shows the results for (a) the transmission probability on the lower surface for Tully 1, (b) the reflection probability on the lower surface for Tully 3, and (c) the reflection probability on the upper surface for Tully 3 in this figure. The new protocol is in excellent agreement with the old protocol for the whole range of the incoming momentum.

3.3. 2D Model Hamiltonian. Now we investigate a model problem with two classical degrees of freedom (rather than one) for which we will perform on-the-fly diagonalization of the electronic Hamiltonian. Recall that, for our new approximation of the decoherence rate, we consider only the component of $\delta\vec{R}$ along the direction of $\vec{v}$. Thus, a multidimensional model problem serves as an important test for this new approach; such a problem will also measure any gains in computational efficiency.

We define the potential for the system as

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

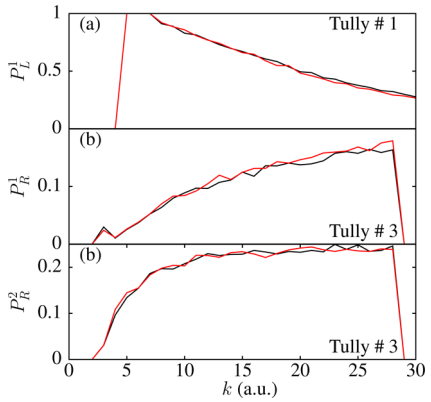


Figure 6. Comparison of the new and the old protocol for the transmission and the reflection probabilities as a function of the incoming momentum k. (a) Probability of transmission on the lower surface for Tully 1, (b) probability of reflection on the lower surface for Tully 3, and (c) probability of reflection on the excited surface for Tully 3. The red and black lines plot the results for the old and the new protocol, respectively. The new protocol is in excellent agreement with the old protocol for the entire range of k.

$$V(q, x_1, x_2) = V_0(q) + V_1(x_1, x_2) + V_c(q, x_1, x_2) \quad (57)$$

with

$$V_0(q) = A^0 q^2 + B^0 q^4 \quad (58)$$

$$V_1(x_1, x_2) = \sum_{i=1}^{2} \frac{1}{2} m\omega_i^2 x_i^2 + k_3 x_1 x_2 \quad (59)$$

$$V_c(q, x_1, x_2) = k_1 q x_1 + k_2 q^2 x_2 \quad (60)$$

Equations 57−60 were inspired by the proton transfer process within the Azouz-Borgis picture.[7,58] Here, q represents the proton coordinate, x_1 represents a low-frequency reaction coordinate, and x_2 represents the donor−acceptor distance. We treat the proton coordinate q quantum mechanically, while x_1 and x_2 are treated classically. The parameters of the model are as follows: $A^0 = -3367.6$ cm^{-1} Å^{-2}, $B^0 = 133501.2$ cm^{-1} Å^{-4}, mass $m = 20$ a.m.u., frequencies $\omega_1 = 200$ cm^{-1}, and $\omega_2 = 287$ cm^{-1}, $k_1 = 1.5 \times 10^{-4}$ cm^{-1} Å^{-2}, and $k_3 = -7 \times 10^{-3}$ cm^{-1} Å^{-2}. As the parameter k_2 is varied, we investigate the rate constant at temperature $T = 300$ K. The mass of the quantum mode q is taken to be 1 a.m.u. Figure 7 panels (a) and (b) show the ground and the first excited adiabatic energy surface as a function of the classical coordinates x_1 and x_2 (with q treated quantum mechanically) for $k_2 = -5 \times 10^{-4}$ cm^{-1} Å^{-3}. We also plot the magnitude of the derivative coupling in Figure 7c, and we show a cut of the first five adiabatic energy surfaces as a function of x_1 (with a fixed value of $x_2 = 0.18$ Å) in Figure 7d. Note that there are multiple crossings within the manifold of excited states so that we must include multiple quantum states to converge our surface hopping dynamics.

For this model, we define $x_1 = 0$ to be the dividing surface for the proton transfer reaction. As far as the ground adiabat is concerned, the transition state (TS) is located at $(x_1,x_2) = (0,0.1)$ Å (shown as 'X' in Figure 7a) with a barrier height of 443 cm^{-1}. Beyond the primary reaction mode x_1, an interesting feature of this model is the role of the donor−acceptor mode x_2. Physically, for proton transfer to occur, the donor and acceptor moieties must approach each other. The role of the x_2 mode is to modulate the potential energy surface near the

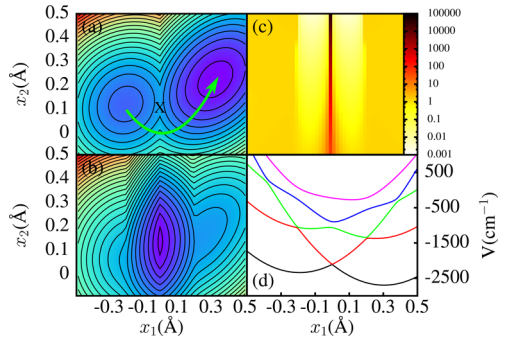


Figure 7. Plots of potential energy surfaces and the nonadiabatic coupling for the 2D model described in section 3.3. (a) A contour plot of the ground adiabatic potential energy surface as a function of the coordinates x_1 and x_2 with $k_2 = -5 \times 10^{-4}$ cm^{-1} Å^{-3}. The contours are plotted with isosurfaces of 1000 cm^{-1}. The 'X' shows the transition state on the ground adiabatic surface, and the green arrow shows a representative path for a reactive trajectory. (b) A contour plot of the first excited state. (c) A contour plot of the magnitude of the nonadiabatic coupling vector $\vec{d}_{12}$. (d) A 1D slice of the first five adiabatic surfaces as a function of x_1 for $x_2 = 0.18$ Å. The dividing surface for this model is defined to be the $x_1 = 0$ surface. The x_2 mode strongly modulates the nonadiabatic coupling at the crossing surface, resulting in the curved green path for a reactive trajectory in part a.

crossing, including the magnitude of the d_{12} vector between the first two adiabats (shown in Figure 7c). More specifically, as x_2 decreases from the TS, $|\vec{d}_{12}|$ decreases and the energy difference between the adiabats (i.e., the tunneling splitting) increases. Hence, in practice, for trajectories to be reactive and relax into the product basin in the $x_1 > 0$ region, they must access smaller values of x_2 (as shown by the arrow in Figure 7a). Because the same trajectory will access regions of strong diabatic coupling (for example around $(x_1,x_2) = (0,0)$ Å, where the tunneling splitting is roughly 50 cm^{-1}) and weak diabatic coupling (for example around $(x_1,x_2) = (0,0.5)$ Å, where the tunneling splitting is roughly 0.1 cm^{-1}), this model problem does not allow a single and static classical time step that is both efficient and accurate. As such, this problem tests the dynamical choice of the quantum time step (dt_q) we have proposed above in section 2.2.1.

We first analyze a trajectory with no friction. We choose the initial conditions in the left well to be such that the particle has sufficient energy to reach the dividing surface for a large range of values of x_2. We choose $k_2 = -7 \times 10^{-4}$ cm^{-1} Å^{-3} for this trajectory, and the energy threshold is set to 0.05 cm^{-1}. The trajectory is run with a classical time step of 0.5 fs for a duration of 10 ps. Figure 8 plots the coordinates of the trajectory as well as the total energy. As was seen for the spin-Boson trajectory (see Figure 2), the trajectory is highly regular with no signs of chaos. Every time the trajectory crosses the dividing surface $x_1 = 0$, either we decrease the time step (to conserve energy) or we find that using the force of the new surface (in case of a hop as described in section 2.2.1), energy conservation is automatic. Without this protocol, we would require $dt_c \leq 0.005$ fs to conserve energy within a 0.05 cm^{-1} threshold. In such a case, choosing dt_c dynamically, that is, initially setting dt_c to be reasonably large but reducing dt_c occasionally as given by Step 5(b)ii for reasons of energy conservation, leads to a speedup by a factor of 100.

Next we compute the thermal rate constant for this model problem as a function of the parameter k_2. This parameter

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

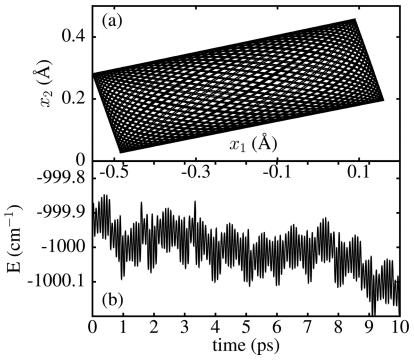


Figure 8. A representative trajectory for the 2D model Hamiltonian (see section 3.3) with $k_2 = -7 \times 10^{-4}$ cm^{-1} Å^{-3} and in the absence of friction. (a) The trajectory plotted in the coordinate-space, and (b) the energy of the trajectory as a function of time (which shows only a very small drift). Our new protocol gives a regular trajectory, with a classical time step of 0.5 fs. The simulation time is greatly reduced by using our modified protocol to treat the separation of classical and quantum degrees of freedom.

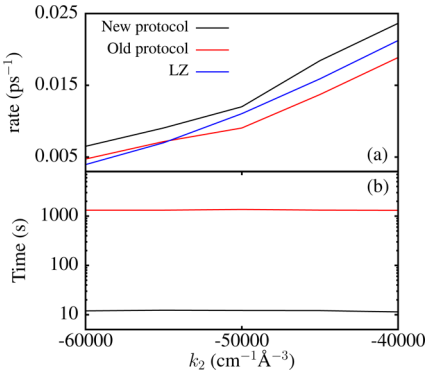


Figure 9. (a) The forward rate constant as a function of the potential parameter k_2 (see eq 60). Results from the old and the new protocol are compared with the Landau−Zener estimate. (b) The simulation wall time per trajectory as a function of k_2. The new protocol obtains results in close agreement with both the old protocol and the Landau−Zener results, while being roughly 100 times faster than the old protocol.

primarily changes the nonadiabatic coupling between the first two adiabatic surfaces at the $x_1 = 0$ surface (along with small changes to exothermicity and the location of the minima). We consider the left well ($x_1 < 0$) as the reactant region and the right well ($x_1 > 0$) as the product region. The coordinates x_1 and x_2 are coupled to a harmonic bath, and its effects are simulated using the Langevin equation. The friction strength is set to $\eta = \omega_1$. The trajectories are first equilibrated in the reactant region for a period of 1 ps. Thereafter, all trajectories that are still in the reactant region after the equilibration period are evolved for 50 ps to obtain the long-time decay of the reactant population. Any trajectory that reacts during the equilibration time is discarded. The long-time decay is then fit to an exponential to obtain the rate constant. The results are compared with the thermally averaged Landau−Zener[59] (LZ) result (computed as the flux across the dividing surface $x_1 = 0$ using the transmission probability given by the LZ result perpendicular to the dividing surface).

We average over 1000 trajectories. The adiabatic eigenfunctions are computed on the fly using 31 discrete variable representation (DVR) grid points corresponding to the quantum mode q, equally spaced from −2 Å to 2 Å. The details of the DVR representation are given in Appendix A [eq A7] of ref 60. A trajectory is identified as a reactant if $x_1 < 0$. Converged results are obtained with a classical time step of 0.5 fs for the new protocol and 0.01 fs for the old protocol. Three adiabatic quantum states are included for both the new and the old protocol.

Figure 9 shows the rate constants and simulation wall times as a function of k_2. The rate constants from the new protocol match well both with the results from the old protocol as well as with the LZ results, as can be seen in Figure 9a. However, the new protocol is roughly 100 times more efficient than the old protocol.

The 2D model in eqs 57−60 also presents a good model system to benchmark our new decoherence protocol against the old protocol. To this purpose, we run a short trajectory for 100 fs for $k_2 = -5 \times 10^{-4}$ cm^{-1} Å^{-3} using both the new and the old protocol at constant energy such that the trajectory starts as a reactant, crosses the dividing surface at around 10 fs, hops to the excited surface, and then recrosses the dividing surface at around 50 fs finally hopping back to the ground state. In Figure 10, for this representative trajectory, we compare (a) the decoherence times given by the new protocol (eq 54) versus

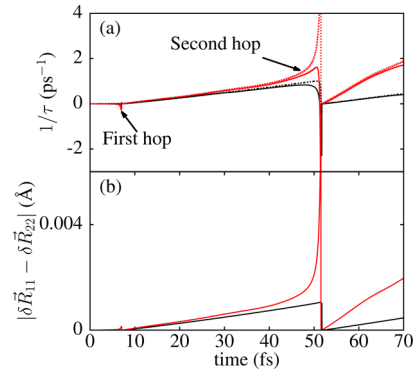


Figure 10. Various quantities for a representative trajectory for the 2D model (see section 3.3): (a) The total decoherence time computed using eq 54 (black line) and eq 18 (red line). The dashed line shows the first term of eq 54 (black) and eq 18 (red). (b) $|\delta\vec{R}_{11}-\delta\vec{R}_{22}|$. Note that the decoherence times and the moments computed using the new and the old protocol are in qualitative agreement up through 70 fs.

the old protocol (eq 18), and (b) $|\delta\vec{R}_{11}-\delta\vec{R}_{22}|$. Figure 10a also compares the first term of eq 54 and eq 18. The decoherence times and the moments computed using the old and the new protocol are in qualitative agreement up through 70 fs, though the new protocol is underestimating the decoherence rates (and the values of the moments). Given the massive speedups gained by the additional approximations introduced here, this new scheme for decoherence should prove useful for a wide range of applications, even if there is a slight underestimate of the approximate decoherence rate.

One additional point regarding decoherence merits comment here. Without decoherence (rates not shown for clarity in Figure 9), the results are roughly an order of magnitude larger, and the reactant populations show spurious nonexponential behavior at short times. This erroneous and unphysical behavior is expected from earlier works of Jain and Subotnik for the spin-Boson problem.[61] Furthermore, without decoherence one needs to include eight quantum states to obtain converged results for FSSH (as opposed to three quantum

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

states for A-FSSH method). This difference is a consequence of the intruder state problem[62] whereby high energy excited state crossings, even if not inaccessible, can significantly impact the results by changing the quality of the electronic amplitudes. These intruder states can be visualized in Figure 7d, for example the crossing of the green and the blue curve around $x_1 = -0.4$ Å. These excited state crossings become irrelevant with decoherence, however, and the quantum amplitudes for these excited states collapse before the trajectory can access diabatic crossing because all physical dynamics occur on the lowest two states.

3.4. Chain of Coupled Oscillators. From the previous example, we see that new protocol is far more efficient than the older protocol for larger systems, as was desired. To highlight this gain in efficiency, the final example we consider is the Holstein model, whereby we consider a chain of five harmonic oscillators, each of which is coupled diabatically to its neighbors. (Note that the old protocol becomes computationally unfeasible with longer chains.) This exact model problem was studied earlier by Prezhdo in ref 19 to highlight the difficulty with standard FSSH and the advantage of propagation in a diabatic basis. The Hamiltonian of the system is

$$H = \sum_{i=1}^{5}\left(\frac{p_i^2}{2m} + \frac{1}{2}m\omega^2 x_i^2\right) + \sum_{i=1}^{5} g x_i |i\rangle\langle i| + \sum_{i=1}^{4} V_c(|i\rangle\langle i+1| + |i+1\rangle\langle i|) \tag{61}$$

Here the set $\{ |i\rangle \}$ is a diabatic basis. The parameters are chosen as mass $m = 1836$ au, frequency $\omega = 200$ cm^{-1}, bath coupling $g = 3091.8$ cm^{-1} Å^{-1}, temperature $T = 575.5$ K, and the diabatic coupling $V_c = 50$ cm^{-1}.

All of the modes x_i are treated classically, and all the states $|i\rangle$ are treated quantum mechanically. The classical coordinates are initialized around $(x_i,p_i) = (0,0)$ for all (x_i,p_i) and sampled from the normalized, undisplaced Boltzmann distribution:

$$P(x) = \sqrt{\frac{m\omega^2}{2\pi k_B T}}\, e^{-m\omega^2 x^2/2k_B T} \tag{62}$$

$$P(p) = \sqrt{2\pi m k_B T}\, e^{-p^2/2mk_B T} \tag{63}$$

The quantum coefficients are then initialized corresponding to diabatic state $|1\rangle$. We average over 1000 trajectories for the new protocol and 100 trajectories for the old protocol (due to the much higher computational demand, we average over fewer trajectories for the old protocol). Each trajectory is evolved at constant energy for a duration of 10 ps. The population of the various diabats are assigned using the density matrix method described in ref 41. Converged results are obtained with a time step of 0.5 fs for the new protocol; this performance should be compared against 0.0001 fs for the old protocol.

Figure 11 shows the diabatic population for the first three diabats. The results from the old and the new protocol are in close agreement. However, the new protocol is now substantially faster than the old protocol. With decoherence, while the old protocol requires 6.35 h per trajectory, the new protocol requires only 1.65 s per trajectory. Furthermore, the effects of decoherence can also be clearly seen in Figure 11: the diabatic populations show significantly faster dynamics without decoherence. The decoherence component presented here adds a negligible computational cost. If we ignore decoherence,

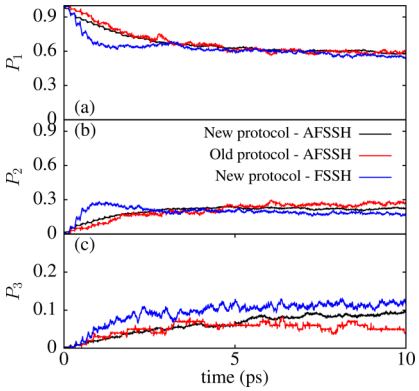


Figure 11. Evolution of the diabatic populations corresponding to the first three diabats (P_1, P_2, and P_3) as a function of time for the Hamiltonian given in eq 61. Results are provided using the new A-FSSH protocol (black), the old A-FSSH protocol (red), and the standard FSSH with the new protocol (blue). (The old FSSH protocol results are in close agreement with the new FSSH protocol, and are not shown for clarity.) While the new protocol requires 1.65 s per trajectory, the old protocol requires 6.35 h per trajectory when decoherence is included. Without decoherence, the new protocol requires 1.5 s per trajectory, compared to 53 min per trajectory for the old protocol. Results with the new protocol are fully converged (using 1000 trajectories) while results with the old protocol are only partially converged (using 100 trajectories) because of the enormous computational cost.

the new protocol requires about 1.5 s per trajectory, while the old protocol requires 53 min. Similar improvements were also found in ref 19 (without decoherence). Thus, our new protocol represents a massive improvement in this case.

4. CONCLUSIONS

We have presented several modifications to the A-FSSH algorithm. These modifications can drastically increase the efficiency of the A-FSSH method and allow large-scale simulations that include decoherence without any parametrization. To achieve this goal, we have made three broad modifications: (a) we have proposed a practical scheme for efficient classical evolution in the presence of trivial and narrowly avoided crossings, (b) we have implemented a new approach (based on the Meek−Levine scheme) for calculating a time-averaged multidimensional time derivative matrix ($\mathbf{T}$) through a matrix logarithm, and (c) we have made several approximations for the propagation of moments to allow an efficient computation of a decoherence rate. Free and open-source codes are provided for the public; see ref 55 for details. An implementation of this algorithm with an *ab initio* electronic structure package will be forthcoming as well.[63]

The new protocol has been compared against the old protocol for a number of model problems. Five model problems have been investigated: the spin-Boson Hamiltonian, Tully model problem no. 1 (simple avoided crossing), Tully model problem no. 3 (extended coupling), a 2D system modeling proton transfer, and a linear chain composed of five oscillators (each of which is coupled to its neighbor). For all of these problems, the new protocol retains the accuracy of the old protocol. However, for the multidimensional problems, the new protocol is significantly more efficient (between 100 and 10000 times more). For one-dimensional problems, there is no advantage to the new protocol.

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268

Overall, these results are highly encouraging and should pave the path for large-scale surface hopping nonadiabatic simulations that include decoherence.

AUTHOR INFORMATION

Corresponding Author

*E-mail: ambjain@sas.upenn.edu.

Funding

This work was supported by NSF CAREER Grant CHE-1150851. J.E.S. acknowledges a Cottrell Research Scholar Fellowship and a David and Lucille Packard Fellowship.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

J.E.S. thanks Ben Levine for very helpful discussions.

REFERENCES

(1) Tully, J. C. *J. Chem. Phys.* **1990**, *93*, 1061.
(2) Subotnik, J. E.; Jain, A.; Landry, B.; Petit, A.; Ouyang, W.; Bellonzi, N. *Annu. Rev. Phys. Chem.* **2016**, *67*, 387−417.
(3) Doltsinis, N. L. Nonadiabatic dynamics: mean-field and surface hopping. In *Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms*; Grotendorst, J., Marx, D., Muramatsu, A., Eds.; NIC: FZ Jülich, 2002; www.fz-juelich.de/nic-series/volume10/doltsinis.pdf.
(4) Müller, U.; Stock, G. *J. Chem. Phys.* **1997**, *107*, 6230−6245.
(5) Barbatti, M. *WIREs: Comput. Mol. Sci.* **2011**, *1*, 620−633.
(6) Wang, L.; Akimov, A.; Prezhdo, O. V. *J. Phys. Chem. Lett.* **2016**, *7*, 2100−2112.
(7) Hammes-Schiffer, S.; Tully, J. C. *J. Chem. Phys.* **1994**, *101*, 4657.
(8) Barbatti, M.; Granucci, G.; Persico, M.; Ruckenbauer, M.; Vazdar, M.; Eckert-Maksić, M.; Lischka, H. *J. Photochem. Photobiol., A* **2007**, *190*, 228−240.
(9) Sterpone, F.; Bedard-Hearn, M. J.; Rossky, P. J. *J. Phys. Chem. A* **2009**, *113*, 3427−3430.
(10) Shenvi, N.; Roy, S.; Tully, J. C. *Science* **2009**, *326*, 829−832.
(11) Nelson, T.; Fernandez-Alberti, S.; Chernyak, V.; Roitberg, A. E.; Tretiak, S. *J. Phys. Chem. B* **2011**, *115*, 5402−5414. PMID: 21218841.
(12) Nelson, T.; Fernandez-Alberti, S.; Chernyak, V.; Roitberg, A. E.; Tretiak, S. *J. Chem. Phys.* **2012**, *136*, 054108.
(13) Schwerdtfeger, C. A.; Soudackov, A. V.; Hammes-Schiffer, S. *J. Chem. Phys.* **2014**, *140*, 034113.
(14) Nelson, T.; Fernandez-Alberti, S.; Roitberg, A. E.; Tretiak, S. *Acc. Chem. Res.* **2014**, *47*, 1155−1164.
(15) Jain, A.; Subotnik, J. E. *J. Chem. Phys.* **2015**, *143*, 134107.
(16) Landry, B. R.; Subotnik, J. E. *J. Chem. Theory Comput.* **2014**, *10*, 4253−4263.
(17) Fabiano, E.; Keal, T.; Thiel, W. *Chem. Phys.* **2008**, *349*, 334−347.
(18) Plasser, F.; Granucci, G.; Pittner, J.; Barbatti, M.; Persico, M.; Lischka, H. *J. Chem. Phys.* **2012**, *137*, 22A514.
(19) Wang, L.; Prezhdo, O. V. *J. Phys. Chem. Lett.* **2014**, *5*, 713−719.
(20) Meek, G. A.; Levine, B. G. *J. Phys. Chem. Lett.* **2014**, *5*, 2351−2356.
(21) Spörkel, L.; Thiel, W. *J. Chem. Phys.* **2016**, *144*, 194108.
(22) Pal, S.; Trivedi, D. J.; Akimov, A. V.; Aradi, B.; Frauenheim, T.; Prezhdo, O. V. *J. Chem. Theory Comput.* **2016**, *12*, 1436−1448.
(23) Akimov, A. V.; Prezhdo, O. V. *J. Chem. Theory Comput.* **2013**, *9*, 4959−4972.
(24) Akimov, A. V.; Prezhdo, O. V. *J. Chem. Theory Comput.* **2014**, *10*, 789−804.
(25) Richter, M.; Marquetand, P.; González-Vázquez, J.; Sola, I.; González, L. *J. Chem. Theory Comput.* **2011**, *7*, 1253−1258.
(26) Mai, S.; Marquetand, P.; González, L. *Int. J. Quantum Chem.* **2015**, *115*, 1215−1231.
(27) Du, L.; Lan, Z. *J. Chem. Theory Comput.* **2015**, *11*, 1360−1374.
(28) Shao, Y.; Gan, Z.; Epifanovsky, E.; Gilbert, A. T.; Wormit, M.; Kussmann, J.; Lange, A. W.; Behn, A.; Deng, J.; Feng, X.; et al. *Mol. Phys.* **2015**, *113*, 184−215.
(29) Shenvi, N.; Roy, S.; Tully, J. C. *J. Chem. Phys.* **2009**, *130*, 174107.
(30) Landry, B. R.; Subotnik, J. E. *J. Chem. Phys.* **2012**, *137*, 22A513.
(31) Falk, M. J.; Landry, B. R.; Subotnik, J. E. *J. Phys. Chem. B* **2014**, *118*, 8108−8117.
(32) Granucci, G.; Persico, M. *J. Chem. Phys.* **2007**, *126*, 134114.
(33) Fernandez-Alberti, S.; Roitberg, A. E.; Nelson, T.; Tretiak, S. *J. Chem. Phys.* **2012**, *137*, 014512.
(34) Nelson, T.; Fernandez-Alberti, S.; Roitberg, A. E.; Tretiak, S. *Chem. Phys. Lett.* **2013**, *590*, 208−213.
(35) Schwartz, B. J.; Bittner, E. R.; Prezhdo, O. V.; Rossky, P. J. *J. Chem. Phys.* **1996**, *104*, 5942−5955.
(36) Hack, M. D.; Truhlar, D. G. *J. Chem. Phys.* **2001**, *114*, 2894−2902.
(37) Nelson, T.; Fernandez-Alberti, S.; Roitberg, A. E.; Tretiak, S. *J. Chem. Phys.* **2013**, *138*, 224111.
(38) Granucci, G.; Persico, M.; Zoccante, A. *J. Chem. Phys.* **2010**, *133*, 134111.
(39) Bedard-Hearn, M. J.; Larsen, R. E.; Schwartz, B. J. *J. Chem. Phys.* **2005**, *123*, 234106.
(40) Subotnik, J. E.; Ouyang, W.; Landry, B. R. *J. Chem. Phys.* **2013**, *139*, 214107.
(41) Landry, B. R.; Falk, M. J.; Subotnik, J. E. *J. Chem. Phys.* **2013**, *139*, 211101.
(42) Subotnik, J. E.; Shenvi, N. *J. Chem. Phys.* **2011**, *134*, 024105.
(43) Chen, H.-T.; Reichman, D. R. *J. Chem. Phys.* **2016**, *144*, 094104.
(44) Herman, M. F. *J. Chem. Phys.* **1984**, *81*, 754.
(45) Arce, J. C.; Herman, M. F. *J. Chem. Phys.* **1994**, *101*, 7520−7527.
(46) Herman, M. F. *J. Chem. Phys.* **1995**, *103*, 8081−8097.
(47) Kapral, R.; Ciccotti, G. *J. Chem. Phys.* **1999**, *110*, 8919−8929.
(48) Jasper, A. W.; Truhlar, D. G. *Chem. Phys. Lett.* **2003**, *369*, 60−67.
(49) Jain, A.; Herman, M. F.; Ouyang, W.; Subotnik, J. E. *J. Chem. Phys.* **2015**, *143*, 134106.
(50) Sifain, A. E.; Wang, L.; Prezhdo, O. V. *J. Chem. Phys.* **2016**, *144*, 211102.
(51) Landry, B. R.; Subotnik, J. E. *J. Chem. Phys.* **2011**, *135*, 191101.
(52) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Clarendon Press: New York, NY, USA, 1989.
(53) Loring, T. A. *Numer. Linear Algebr. Appl.* **2014**, *21*, 744−760.
(54) Löwdin, P. O. *Adv. Quantum Chem.* **1970**, *5*, 185−199.
(55) AFSSH Modifications. https://github.com/subotnikgroup/AFSSH_modifications (accessed Oct. 3, 2016)
(56) Nitzan, A. *Chemical Dynamics in Condensed Phases: Relaxation, Transfer and Reactions in Condensed Molecular Systems: Relaxation, Transfer and Reactions in Condensed Molecular Systems*; Oxford University Press: Great Clarendon Street, Oxford OX2 6DP, United Kingdom, 2006.
(57) Marcus, R. *Annu. Rev. Phys. Chem.* **1964**, *15*, 155−196.
(58) Azzouz, H.; Borgis, D. *J. Chem. Phys.* **1993**, *98*, 7361−7374.
(59) Zener, C. *Proc. R. Soc. London, Ser. A* **1932**, *137*, 696−702.
(60) Colbert, D. T.; Miller, W. H. *J. Chem. Phys.* **1992**, *96*, 1982.
(61) Jain, A.; Subotnik, J. E. *J. Phys. Chem. Lett.* **2015**, *6*, 4809−4814.
(62) Nazarewicz, W. *Nucl. Phys. A* **1993**, *557*, 489−514.
(63) Medders, G. R., Jain, A., Alguire, E., Subotnik, J. E. Unpublished.

DOI: 10.1021/acs.jctc.6b00673
J. Chem. Theory Comput. 2016, 12, 5256−5268