Simultaneous-trajectory surface hopping: A parameter-free algorithm for implementing decoherence in nonadiabatic dynamics

Neil Shenvi,[1,a)] Joseph E. Subotnik,[2] and Weitao Yang[1]
[1]*Department of Chemistry, Duke University, Durham, North Carolina 27708*
[2]*Department of Chemistry, University of Pennsylvania, Philadelphia, Pennsylvania 19104*

(Received 23 December 2010; accepted 21 March 2011; published online 8 April 2011)

In this paper, we introduce a trajectory-based nonadiabatic dynamics algorithm which aims to correct the well-known overcoherence problem in Tully's popular fewest-switches surface hopping algorithm. Our simultaneous-trajectory surface hopping algorithm propagates a separate classical trajectory on each energetically accessible adiabatic surface. The divergence of these trajectories generates decoherence, which collapses the particle wavefunction onto a single adiabatic state. Decoherence is implemented without the need for any parameters, either empirical or adjustable. We apply our algorithm to several model problems and find a significant improvement over the traditional algorithm.
 [doi:10.1063/1.3575588]

I. INTRODUCTION

In the last few decades, it has been observed that many processes necessarily depend on the coupling of nuclear and electronic motion.[1,2] For instance, scattering at metal surfaces,[3] dynamics near conical intersections,[4] proton transfer,[5] proton-coupled electron transfer,[6,7] and the dynamics of photoexcitation[8] all involve quantum transitions that are induced by nuclear motion. Consequently, the Born–Oppenheimer approximation, which assumes that electrons adjust instantaneously to the motion of the nuclei, will be unable to capture the dynamics of such systems. One of the most popular methods for performing dynamics simulations beyond the Born–Oppenheimer approximation is Tully's fewest-switches surface hopping (FSSH) algorithm.[9]

The FSSH algorithm captures several theoretical features which are crucial for any realistic dynamics algorithm. First, FSSH allows transitions between electronic states due to the motion of the nuclei so that nonadiabatic effects can be taken into account. Second, FSSH includes feedback between the nuclear and electronic degrees of freedom such that the total energy is conserved. Third, FSSH ensures that the nuclei always propagate on a single electronic adiabatic surface in contrast to mean-field dynamics which allow the nuclei to propagate on an average potential energy surface even in the asymptotic limit. Finally, FSSH roughly maintains a detailed balance between the populations on different adiabats through the imposition of forbidden hops to energetically inaccessible electronic states.[10]

Although there are quite a few nonadiabatic dynamics methods which capture many or all of these features,[8,11–17] the main appeal of FSSH comes from its ease of implementation and its intuitive interpretation. Because surface hopping is a mixed quantum-classical method, the nuclei are treated classically, yielding real trajectories which can be followed over the course of a simulation. Nonadiabatic transitions are easily identified as "surface hops" which occur at discrete points along each trajectory. Because the trajectories are propagated independently, computational resources scale modestly with increasing nuclear dimensionality. FSSH requires only a knowledge of the adiabatic forces and the derivative couplings, both of which can be calculated straightforwardly from most *ab initio* electronic structure packages.

Despite the FSSH algorithm's great popularity and success in simulating nonadiabatic dynamics, it suffers from several known problems. First, because the nuclei are treated classically, quantum mechanical effects such as tunneling are completely ignored. Second, in original FSSH, hops between surfaces rescale the nuclear momenta but not the nuclear positions. This requirement is not necessary and is not always realistic.[16,18,19] Third, the nuclear momenta are rescaled along the direction of the nonadiabatic coupling vector during a hop,[1] a stipulation which is only one of many suggested rescaling schemes.[20] Finally, FSSH is known to suffer from problems of overcoherence because the electronic wavefunction is propagated coherently over the entire course of the trajectory. As a result, a particle can remain in a coherent mixture of two adiabatic states indefinitely, even in the asymptotic regime in which the wavefunction should consist of only a single adiabatic state.

This last shortcoming of FSSH will be the main focus of this paper. Although the overcoherence of surface hopping is usually not a problem for one-dimensional systems with a single region of coupling, it is far more troublesome in higher dimensions where the system can cross coupling regions repeatedly and where nuclear trajectories on different surfaces can diverge rapidly. Numerous works have attempted to overcome this well-known deficiency. The most straightforward approach is to introduce a decoherence term which collapses the coherent wavefunction onto a single adiabatic state.[17,21–26] One difficulty of these approaches is that they almost always require the use of some parameter which controls

a)Author to whom correspondence should be addressed. Electronic mail: nashenvi@gmail.com.

the timescale of the decoherence. Although the performance of the algorithm depends critically on this parameter, its value is usually set based on empirical considerations (like the thermal wavelength) or is adjusted to maximize the accuracy.

One important alternative to surface hopping is the multiple-spawning algorithm and its variants introduced by Martinez *et al.*[8,11,16] These algorithms also make use of parameters, albeit in a different way. By lowering the threshold for spawning new trajectories, the exact quantum limit can be obtained. Unfortunately, the computational cost can grow exponentially large for small values of the spawning parameter. Related to this multiple-spawning methodology is the recent work of Granucci, which also makes use of several parameters and a few approximations to lower the computational cost.[27] Multiple-spawning methods have the advantage of approximating an actual quantum nuclear wavefunction. Because many Gaussians are propagated simultaneously, the issue of decoherence disappears. The disadvantage of these approaches is that they must propagate multiple trajectories on each surface and that they must rely on one or more cutoff parameters to guarantee that the number of trajectories propagated does not grow too rapidly.

Given the obvious difficulties inherent in any methodology which makes use of extrinsic parameters, we have a clear motivation to obtain a parameter-free method for introducing decoherence. The recent work of Subotnik *et al.* was an initial step in this direction, in which the authors construct a parameter-free algorithm which introduces decoherence into FSSH by propagating a set of auxiliary variables.[28] These auxiliary variables roughly represent the relative positions of wavepackets on different surfaces. This augmented fewest-switches surface hopping (A-FSSH) algorithm bears many similarities to our present work and was the inspiration for our current approach. The main difference is that in A-FSSH, only a single trajectory is propagated along with auxiliary variables which approximate the position and momenta of trajectories on neighboring electronic surfaces. In contrast, our algorithm will explicitly propagate a separate trajectory on each electronic surface.

The goal of the simultaneous-trajectory surface hopping (STSH) algorithm presented in this paper will be three-fold: first, we will attempt to correct the overcoherence of FSSH by introducing decoherence which collapses the wavefunction onto a single electronic state. Second, our algorithm will be parameter-free so that no empirical or adjustable parameters are needed. Third, we will modify the structure of the FSSH algorithm as little as possible so that our modified algorithm will retain the simplicity and intuitive appeal of the FSSH algorithm. In the concluding section, we consider how the STSH algorithm may also address some of the other shortcomings or quirks of FSSH, including the issue of momentum rescaling.

Our paper is organized as follows: Sec. II presents a derivation of the STSH algorithm. Sec. III describes the algorithm in a step-by-step fashion for comparison with traditional FSSH. In Sec. IV, we make a brief digression into time-independent scattering theory to show how wavepacket dynamics relate to the exact time-independent answer for 1D multichannel problems. Sec. V presents the results of numerical simulations on a variety of one-dimensional model problems. Discussions and conclusions are presented in Sec. VI.

II. DERIVATION

The central motivation for building the STSH algorithm is the desire to add decoherence to the FSSH algorithm. To see the origin of decoherence, we imagine a two-dimensional electronic Hilbert space consisting of the adiabatic states $|1\rangle$ and $|2\rangle$ which are eigenstates of the 2×2 electronic Hamiltonian $H(x)$. Let us assume that the exact quantum wavefunction $|\Psi\rangle$ consists of Gaussian wavepackets on the two adiabatic surfaces,

$$|\Psi\rangle = c_1 |\psi_1(x)\rangle |1\rangle + c_2 |\psi_2(x)\rangle |2\rangle \tag{1}$$

$$= c_1 |g_1(x)\rangle |1\rangle + c_2 |g_2(x)\rangle |2\rangle , \tag{2}$$

where

$$g_i(x) = \left(\frac{2\mathrm{Re}\,\alpha_i}{\pi}\right)^{1/4} \exp\left(-\alpha_i (x - x_i)^2 + i k_i (x - x_i)\right). \tag{3}$$

Initially, let us assume that the wavepackets are centered at identical positions $x_1 = x_2$ with identical widths $\alpha_1 = \alpha_2$ but with different momenta $k_1 \neq k_2$. If we are in free space, these two wavepackets will immediately begin to move apart as time progresses due to their different momenta. This divergence is one clear example of decoherence, as the nuclear positions become entangled with the electronic state of the particle.

To define decoherence more rigorously, we can construct the electronic density matrix σ_{ij}, where

$$\sigma_{ij} = \int dx\, c_i c_j^* g_i(x) g_j(x)^*. \tag{4}$$

Decoherence is related to the decay of the off-diagonal component of this density matrix. Assuming our Gaussian wavepacket *ansatz*, we can perform this integration explicitly to obtain[23,28,29]

$$|\sigma_{ij}| = \left|\frac{(\mathrm{Re}\,2\alpha_i)^{1/4}(\mathrm{Re}\,2\alpha_j)^{1/4}}{(\alpha_i + \alpha_j^*)^{1/2}}\right| \times \exp\left[-\frac{1}{4}\mathrm{Re}\,\frac{\Delta k_{ij}^2 + 4\alpha_i \alpha_j^* \Delta x_{ij}^2 + 2i(\alpha_i - \alpha_j^*)\Delta k_{ij}}{\alpha_i + \alpha_j^*}\right], \tag{5}$$

where

$$\Delta x_{ij} = x_i - x_j, \tag{6}$$

$$\Delta k_{ij} = k_i - k_j. \tag{7}$$

Given this explicit form for σ_{ij}, it is tempting to immediately identify the decay of $|\sigma_{ij}|$ as the decoherence rate. However, we should notice that the overlap specified by Eq. (5) includes a contribution from the difference in momenta between the two wavepackets, Δk_{ij}, even if their spatial separation Δx_{ij} is negligible. If the two adiabats differ in energy

by ΔE but are otherwise uncoupled, then the two momenta $\hbar k_1$ and $\hbar k_2$ must necessarily be different so that the total energy is conserved (i.e., $\hbar^2 k_2^2 - \hbar^2 k_1^2 = -2m\Delta E$). Therefore, it seems unfair to attribute a difference in momenta to decoherence when this difference is mandated by energy conservation. We can actually formalize this logic by performing a gauge transformation on each adiabat.

Let us define a gauge $\exp(iA_j(x))$ for adiabat $|j\rangle$ such that $\phi_j(x) = \psi_j(x)\exp(-iA_j(x))$, where $A_j(x)$ is a real-valued gauge function. If we let the gauge be

$$A_j(x) = k_i(x - x_i), \tag{8}$$

we are left with a Gaussian

$$\phi_j(x) = \psi_j(x)\exp(-iA_j(x)) \tag{9}$$

$$= \left(\frac{2\mathrm{Re}\,\alpha_i}{\pi}\right)^{1/4} \exp(-\alpha_i(x - x_i)^2). \tag{10}$$

We can then rewrite σ_{ij} in terms of Gaussians in the new gauge-transformed basis to obtain

$$|\sigma_{ij}| = \left|\frac{(\mathrm{Re}\,2\alpha_i)^{1/4}(\mathrm{Re}\,2\alpha_j)^{1/4}}{(\alpha_i + \alpha_j^*)^{1/2}}\right| \exp\left[-\mathrm{Re}\,\frac{\alpha_i\alpha_j^*\Delta x_{ij}^2}{\alpha_i + \alpha_j^*}\right]. \tag{11}$$

Using this gauge transformation, we have effectively removed the momentum dependence of the overlap term. If we assume that decoherence events can be described by a Poisson process, then we can define the decoherence rate $1/\tau$ as the rate of decay of $|\sigma_{ij}|$ in the new gauge:

$$1/\tau_{ij} = -d\ln|\sigma_{ij}|/dt \tag{12}$$

$$\approx \mathrm{Re}\,[\alpha_{ij}\Delta x_{ij}\Delta\dot{x}_{ij}], \tag{13}$$

where $\Delta\dot{x}_{ij} = d\Delta x_{ij}/dt$, $\alpha_{ij} = \alpha_i\alpha_j^*/(\alpha_i + \alpha_j^*)$, and where we have assumed that the width parameters α_i (and indeed the wavepacket momenta) change slowly enough with time that their time derivatives can be neglected.

Eq. (13) is important because it defines a decoherence rate that depends only on the centers of the Gaussians x_j, their widths α_j, and their velocities $\dot{x}_j$. Unfortunately, this formula cannot be applied to traditional FSSH because FSSH propagates only one trajectory $x(t)$ at a time. Furthermore, there is no explicit width α_j associated with the particle, since FSSH treats the nuclei as purely classical.

The solution to these problems is straightforward. Rather than propagate a single trajectory $x(t)$ on the active surface, we will simultaneously propagate a separate trajectory $x_j(t)$ on each adiabatic surface j. Furthermore, we will also dynamically propagate a width α_j associated with the trajectory on each adiabatic surface. Thus, our STSH algorithm dynamically propagates at most N "thawed" Gaussians, where N is the number of adiabatic surfaces. The Gaussian parameters $(x_j(t), k_j(t), \alpha_j(t))$ are propagated according to the equations of motion originally derived in Ref. 30.

At this point, it would be extremely tempting to identify the STSH algorithm as a kind of multiple-spawning algorithm. However, this identification would be incorrect for one very important reason: in the STSH algorithm the auxiliary trajectories and widths that we are propagating will serve *only* to calculate the decoherence rate between state i and j, not to represent the dynamics themselves as in multiple-spawning approaches. We are not expanding the time-dependent Schrödinger equation in the time-dependent basis of the moving Gaussians, as we would within a multiple-spawning framework. The propagation of the wavefunction and hops between surfaces occur just as in the traditional FSSH algorithm. Because the extra dynamical variables are used only to calculate the decoherence rate, we will ensure that our algorithm adheres as closely as possible to FSSH and makes minimal adjustments. Furthermore, the STSH algorithm propagates at most one trajectory per adiabatic surface, preventing any possible exponential growth in the number of trajectories.

A few other features need to be specified before we can implement the STSH algorithm.

First, we need to specify when new trajectories are spawned. At every instant in time, the surface on which the particle is moving is specified by the label i. Trajectories with position, momentum, and width specified by x_i, k_i, and α_i are being propagated on surface i and on some subset of the remaining surfaces. According to FSSH and multiple spawning, the term $\mathbf{p}_i \cdot \mathbf{d}_{ji}$ determines the rate at which the amplitude is pumped from the current surface i to the surface j, where $\mathbf{d}_{ji}$ is the nonadiabatic coupling matrix:

$$d_{ij}^{\alpha}(\mathbf{x}) = \langle i|\frac{d}{dx^{\alpha}}|j\rangle. \tag{14}$$

Because this term is almost always nonzero, there should be some finite rate for spawning a new trajectory on surface j at almost any time. This fact immediately leads to a computational problem, since we might be forced to constantly spawn new trajectories leading to an exponential growth in the number of trajectories being propagated. For this reason, we take our inspiration from the original surface hopping proposal of Preston and Tully in 1971 (Ref. 31) and spawn a new trajectory on surface j only when the quantity $|\mathbf{p}_i \cdot \mathbf{d}_{ji}|$ reaches a local maximum. This restriction will partially solve the spawning problem, since new trajectories will be spawned only occasionally. Moreover, this approximation seems reasonable on physical grounds, since regions of strong nonadiabatic coupling will mostly be well-localized near curve crossings (indeed, this fact is what allows multiple-spawning algorithms to overcome the exponential proliferation of trajectories). It must be emphasized that spawning in STSH is a distinct process from hopping. While spawning occurs only at local maxima of $|\mathbf{p}_i \cdot \mathbf{d}_{ji}|$, hops are treated stochastically just as before.

Second, we need to specify how new trajectories are spawned. At any instant in time, the surface on which the particle is moving is specified by the label i and its position, momentum, and width are specified by x_i, k_i, and α_i. When the quantity $|\mathbf{p}_i \cdot \mathbf{d}_{ji}|$ reaches a local maximum (which can easily be determined by recording the values of $\mathbf{p}_i \cdot \mathbf{d}_{ji}$ and calculating its curvature via a finite difference formula), a new trajectory is spawned on surface j with the same position x_i and width α_i as the trajectory on surface i and with the

momentum is rescaled along the direction $\hat{\mathbf{d}}_{ji}$ such that energy is conserved.

One difficulty arises if there is already a spawned trajectory on the surface in question. In this case, we let a stochastic process determine whether we retain the old trajectory or replace it with a new trajectory. We generate a random number ξ and compare it to the ratio $|\mathbf{p}\cdot\mathbf{d}|_{\text{old}}/(|\mathbf{p}\cdot\mathbf{d}|_{\text{old}}+|\mathbf{p}\cdot\mathbf{d}|_{\text{new}})$. If the random number is smaller than this ratio, then the old trajectory is retained. If the random number is larger than this ratio, then the old trajectory is overwritten by the new one. This stochastic overwriting procedure appears to work quite well for the model problems we tested. It is not necessarily optimal but it allows us to avoid the problem of having to propagate multiple trajectories on each surface at the same time. It also ensures that the algorithm is numerically stable even in the presence of a nonadiabatic coupling function with multiple local maxima of extremely small magnitude.

When we consider whether to spawn on surface j, it may be possible that the particle will not possess sufficient kinetic energy in the direction $\hat{\mathbf{d}}_{ij}$ to conserve energy. In this case, we spawn a trajectory on surface j with all the kinetic energy along the direction $\hat{\mathbf{d}}_{ji}$ removed (i.e., we rescale the momentum such that $\mathbf{p}_j\cdot\hat{\mathbf{d}}_{ji}=0$). Although the newly spawned trajectory does not have the same total energy as the active trajectory on surface i, this fact is unimportant since the spawned trajectory will only be used to calculate coherence properties, rather than to represent the real dynamics of the active particle. If fact, this procedure seems intuitive, since in practice it results in very fast decoherence between the particle on surface i and surface j whenever a hop between i and j would be forbidden.

Third, we need to specify how decoherence events are implemented. Eq. (13) defines the decoherence rate between adiabatic states i and j. Assuming that decoherence occurs as a Poisson process, we can use a random number generator to determine whether the coherence between state i and j should be destroyed during a given timestep. If we are moving on adiabatic surface i and our random number generator indicates that the coherence between state i and j should be destroyed, then we project the adiabatic state j out of our current wavefunction, such that

$$|\psi_{\text{new}}\rangle = |\psi_{\text{old}}\rangle - |j\rangle\langle j|\psi_{\text{old}}\rangle\,, \tag{15}$$

where the new state $|\psi_{\text{new}}\rangle$ must be subsequently normalized. Having specified how and when trajectories are spawned and the mechanism for introducing decoherence, we can now sketch the STSH algorithm.

III. THE STSH ALGORITHM

1. Set the initial conditions of the simulation. The initial surface is set to i and the electronic wavefunction $|\psi\rangle$ is initialized to $(0,\ldots,1,\ldots,0)$ so that the amplitude of state i is 1 and the amplitude of all other states is 0. The position, momentum, and width of the trajectory are determined by the desired initial conditions. Only trajectory i is initially active, and no other trajectories are spawned initially.
2. Propagate the position, momentum, and width of each trajectory j according to Heller's thawed Gaussian propagation scheme,[30]

$$\dot{x}_j = \frac{\hbar}{m}k_j, \tag{16}$$

$$\dot{k}_j = -dV_j/dx, \tag{17}$$

$$\dot{\alpha}_j = -(2/m)\alpha_j^2 - \frac{1}{2}\frac{d^2V_j}{dx^2}. \tag{18}$$

3. Integrate the electronic wavefunction according to the time-dependent Schrödinger equation using the electronic Hamiltonian at x_i,

$$d\,|\psi\rangle\,/dt = -\frac{i}{\hbar}H(x_i)\,|\psi\rangle\,. \tag{19}$$

4. Test whether the quantity $p=|\mathbf{p}_i\cdot\mathbf{d}_{ji}|$ has reached a local maximum. If so, then spawn a new trajectory on surface j. If there is already a spawned trajectory on surface j, then generate a random number $\xi\in[0,1]$ and compare it to the ratio $p_{\text{old}}/(p+p_{\text{old}})$. If ξ is smaller than this ratio, then retain the old spawned trajectory. Otherwise, let $x_j=x_i$, $\alpha_j=\alpha_i$ and the momentum k_j be rescaled to conserve total energy. If there is insufficient kinetic energy in the direction $\hat{d}$ for a hop to occur, then spawn a trajectory on surface j with $x_j=x_i$, $\alpha_j=\alpha_i$, and $k_j=0$ along the direction $\hat{d}$.
5. Perform surface hopping. Calculate the quantities

$$\dot{a}_{jj} = \sum_{l\neq j} b_{ji}, \tag{20}$$

$$b_{ji} = \frac{2}{\hbar}\text{Im}\,a_{ji}^*H_{ji} - 2\text{Re}\,a_{ji}^*\dot{x}\cdot\mathbf{d}_{ji}. \tag{21}$$

The rate of surface hopping from state i to state j is then given by b_{ji}/a_{ii}. Given a random number $\zeta=[0,1]$, then a hop is made if $\Delta t\,b_{ji}/a_{ii}>\zeta$ and if the hop is not energetically forbidden. If a hop occurs, then respawn the trajectory on surface j with the momentum rescaled in the direction $\hat{d}_{ji}$.
6. Test for decoherence. For all active trajectories j, calculate the decoherence rate $1/\tau_{ji}$ between trajectory i and trajectory j according to Eq. (13). Given a random number $\zeta=[0,1]$, decoherence occurs if $\Delta t/\tau_{ji}>\zeta$. If decoherence occurs between states j and i, then project the adiabatic state $|j\rangle$ out of the electronic wavefunction $|\psi\rangle$ according to Eq. (15).
7. Return to step 2 and repeat until the trajectory $x_i(t)$ leaves the region of interest.

This algorithm closely follows the original FSSH algorithm except in its introduction of decoherence (steps 4 and 6). The simultaneous trajectories act only to calculate the decoherence rate and otherwise do not influence the propagation of the particle. Having described the STSH algorithm, we are now in a position to apply it to several model problems for which decoherence is a necessary ingredient. But first, we

will make a brief digression into time-independent scattering theory to show how the exact, time-independent scattering solutions relate to time-dependent wavepacket scattering solutions.

IV. TIME-INDEPENDENT SCATTERING

Obviously, the goal of any approximate dynamics method is to reproduce as accurately as possible the exact quantum dynamics. However, there are some important distinctions between time-dependent and time-independent scattering calculations that need to be emphasized. The goal of an exact time-independent quantum scattering calculation is to obtain the scattering matrix S, where $S_{ji}(\mathbf{k}, \mathbf{k}')$ is the scattering matrix element from the incoming plane wave state $\mathbf{k}'$ on surface i to the outgoing plane wave state $\mathbf{k}$ on the surface j. For one-dimensional problems, this simplifies to determining the branching probabilities $P_{j\pm}(k) = |S_{ji}(\pm k, k)|^2$ for forward/backward scattering (i.e., transmission/reflection) from surface i onto surface j. This exact time-independent solution does not have an immediate classical analogue since the quantum time-independent solutions will involve nuclear plane waves which are maximally delocalized with respect to position. A classical analogue must instead be provided by time-dependent scattering theory.

In contrast to time-independent scattering theory, time-dependent scattering theory follows the evolution of a quantum wavepacket. By forming a superposition of a large number of eigenstates, a wavepacket can be localized in both position and momentum space so that it behaves roughly like a classical particle. In particular, by convoluting the eigenstates with a Gaussian envelope function in k-space, we obtain a Gaussian wavepacket in position-space which has a fairly well-defined average position and momentum. Thus, we can never compare our trajectory-based mixed quantum-classical results directly to the scattering probabilities $P_{j\pm}(k)$. Instead, we need to compare our trajectory-based mixed quantum-classical results to the corresponding quantum wavepacket scattering probabilities $\tilde{P}_{j\pm}(k)$, which are obtained by convoluting the exact time-independent scattering probability $P_{j\pm}(k)$ with a Gaussian envelope function of the appropriate width. In other words, if our initial Gaussian wavepacket in k-space is given by the function $g(k; k_0)$, where $g(k; k_0)$ is a Gaussian centered on the average momentum k_0, then our time-dependent scattering probability $\tilde{P}_{j\pm}(k_0)$ is given by

$$\tilde{P}_{j\pm}(k_0) = \int dk' P_{j\pm}(k') |g(k'; k_0)|^2, \tag{22}$$

where $P_{j\pm}(k)$ is the time-independent scattering probability. It is this convoluted time-dependent scattering probability $\tilde{P}_{j\pm}(k_0)$ that we are trying to approximate when performing mixed quantum-classical dynamics.

This observation should be obvious to many readers, and indeed the link between time-dependent and time-independent scattering theory is well-known. However, as we will see momentarily, its implications are often forgotten when it comes to nonadiabatic dynamics because it is often assumed that the branching coefficients $P_{j\pm}(k)$ vary so slowly in k-space that they are essentially independent of the initial wavepacket width. Or equivalently, it is assumed that the wavepackets used in our exact time-dependent scattering calculations are sufficiently wide in position space (i.e., narrow in k-space) that the wavepacket results approach the exact, time-independent infinite-width limit. In either case, the time-dependent scattering probabilities would be almost identical to the time-independent scattering probabilities.

What this assumption forgets is that there may exist resonances which manifest as peaks in the scattering matrix. These peaks can be so narrow (corresponding to extremely long-lived resonances) that they will be substantially blurred in any realistic wavepacket calculation. Thus, the calculated scattering probabilities will be extremely sensitive to the initial width of the quantum wavepacket and the infinite-width limit will be nearly impossible to achieve through a wavepacket-based time-dependent calculation. At a more mundane level, even in the absence of resonances, the time-independent scattering probabilities might be rapidly oscillatory as a function of k, rendering the width of the convolution function extremely important. This width dependence is especially crucial when evaluating new algorithms because any comparison to the "exact results" must be a comparison to a wavepacket calculation *of the appropriate width*. The difficulty of wavepacket calculations is that resonance states may be exceptionally long-lived, necessitating long propagation times and appropriate absorbing boundary conditions.

To show how prevalent these resonance states are, in this paper we recalculate the exact branching coefficients $P_{j\pm}(k)$ for several popular model problems using time-independent scattering theory. From the time-independent results, we can obtain time-dependent scattering probabilities through convolution with a finite-width Gaussian as in Eq. (22). The time-independent results were calculated using a grid-based method in a box of $L = 30$ with $M = 6000$ grid points. The kinetic energy operator was evaluated using a five-point stencil and exact scattering eigenstates were obtained by inverting the Hamiltonian for a given k using MATLAB's sparse-matrix inversion capabilities. The method was tested to ensure convergence by doubling the box size and simultaneously halving the grid spacing (i.e., $L = 60$ with $M = 24000$). Our results are presented in Sec. V, along with results for the STSH algorithm.

V. NUMERICAL RESULTS

To test our algorithm we applied it to four model problems, three of which were studied by Tully in his original paper on FSSH (Ref. 9) and a fourth which was used recently to elucidate the effects of decoherence.[28] All four problems are one-dimensional and involve two coupled diabatic surfaces (i.e., N = 2) with the particle initially placed on the lower adiabatic surface. Throughout this section, all the values are given in atomic units and the mass of the nucleus is assumed to be $m = 2000$.

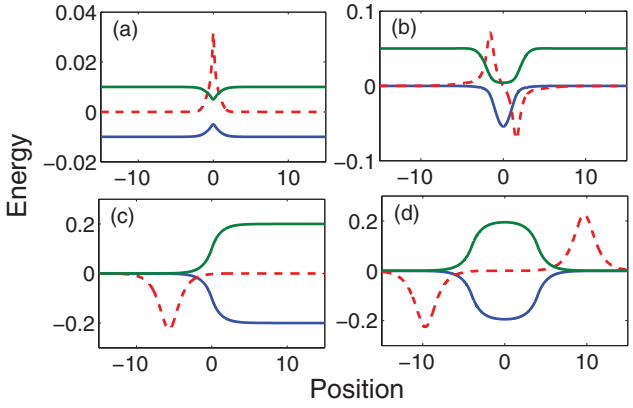

FIG. 1. Adiabatic surfaces and nonadiabatic couplings for the four model problems simulated in the paper. The solid lines are the adiabatic surfaces and the dashed line is the nonadiabatic coupling. Panel (a) shows Tully problem 1, a single avoided crossing. The nonadiabatic coupling has been divided by a factor of 50 for visualization purposes. Panel (b) shows Tully problem 2, a dual avoided crossing. The nonadiabatic coupling has been divided by a factor of 12 for visualization purposes. Panel (c) shows Tully problem 3, the extended coupling model. Panel (d) shows the double arch problem introduced in Ref. 28. All quantities shown are in atomic units.

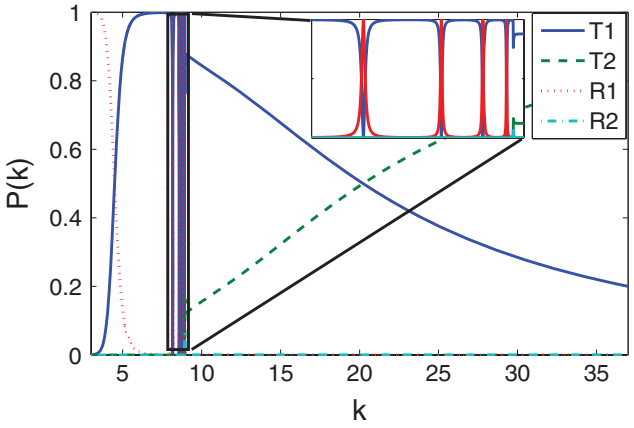

FIG. 2. The exact time-independent scattering probabilities $P(k)$ as a function of the wavevector k. T1 and T2 are the transmission probabilities on the lower and upper adiabats, respectively. R1 and R2 are the reflection probabilities on the lower and upper adiabats, respectively. The inset shows the presence of resonances between $7.7 < k < 8.9$ which lead to total reflection at certain values of k. The presence of narrow (i.e., long-lived) resonance states renders it very difficult to calculate the exact scattering probabilities from time-dependent wavepacket dynamics.

A. Tully problem 1

The diabatic Hamiltonian for Tully problem 1 is given by the following matrix elements:

$$V_{11}(x) = A[1 - \exp(-Bx)], x > 0 \tag{23}$$

$$= -A[1 - \exp(Bx)], x < 0, \tag{24}$$

$$V_{22}(x) = -V_{11}(x), \tag{25}$$

$$V_{12}(x) = V_{21}(x) = C\exp(-Dx^2) \tag{26}$$

with $A = 0.01, B = 1.6, C = 0.005,$ and $D = 1.0$. Figure 1(a) shows the adiabatic surfaces and nonadiabatic coupling for this problem. There are three relevant regimes for this problem. For $k < 7.7$, the particle has enough energy to pass over the potential barrier on the lower adiabatic surface but not enough energy to populate the upper surface, even transiently. Hence the scattering is characterized primarily by transmission on the lower adiabatic surface with some small component of above-the-barrier reflection. For $7.7 < k < 8.9$, the particle has enough kinetic energy to transiently populate the upper potential well but not enough energy to asymptotically escape. In this regime there are four resonance states with a finite width corresponding to the particle trapped in the upper adiabatic potential well. Although transmission usually dominates in this regime, if the momentum happens to align with one of these resonances, there is a complete reflection. For $k > 8.9$ the upper adiabatic state is accessible and there is a smooth transition between transmission primarily on the lower adiabat and transmission primarily on the upper adiabat.

Figure 2 shows the exact time-independent branching coefficients for Tully problem 1. We would like to call particular attention to the second regime $7.7 < k < 8.9$ and point out three things: first, the presence of resonances leads to complete reflection at certain values of k and hence cannot be ignored. These resonances do not introduce some small correction to the scattering problem, but change the result completely for certain values of k. Second, the resonances are remarkably narrow (i.e., long-lived) and hence would require an extremely wide wavepacket to effectively probe their shape. It is for this reason that their existence has been frequently missed in nonadiabatic dynamics calculations. Any mixed quantum-classical trajectory-based method is approximating a finite-width wavepacket and hence must smooth out these resonances. Even time-dependent quantum mechanical calculations will require exceptionally wide initial wavepackets that are numerically infeasible given the narrowness of the resonances. Instead, it appears that a time-independent approach is required to truly assess the location and shape of these resonance states. Once the exact time-independent scattering matrix is known, the time-dependent scattering of a finite-width wavepacket can then be calculated by convolution [see Eq. (22)]. Third, the location of these resonances depends on the energy of bound states on the upper surface and is therefore unavoidably quantum mechanical. Any method which treats the nuclei classically will be unable to calculate the energies or widths of these resonances accurately. In a sense, this fact supports the validity of the FSSH algorithm which never claims to be able to correctly capture nuclear quantum effects. Since resonances are necessarily quantum mechanical in nature, we should not expect FSSH or any other mixed quantum-classical algorithm to capture them precisely.

Having established the presence of resonances and their role in time-dependent scattering, we now apply our STSH algorithm to the problem. The three traces on Fig. 3 show the time-dependent results using the exact time-dependent scattering with a wavepacket of width $\sigma_x = \alpha^{-1/2} = 2$ (as obtained from convolution of a Gaussian envelope function with the time-independent probabilities from Fig. 2), the FSSH results, and the STSH results using an initial position of

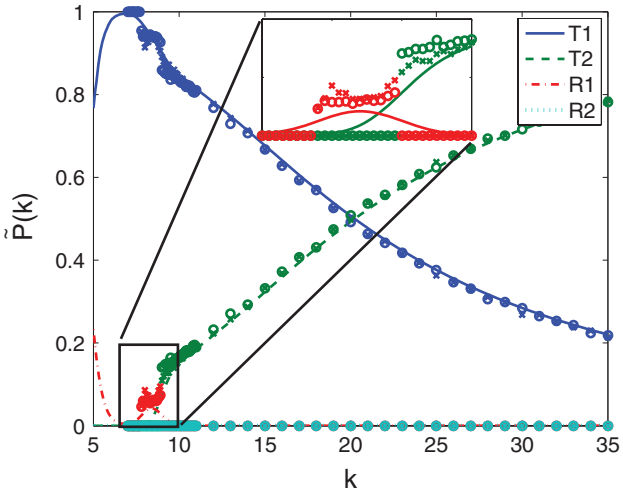

FIG. 3. A comparison of exact time-dependent scattering dynamics with $\sigma_x = 2$ (solid line) with FSSH (crosses) and STSH (circles) for Tully problem 1. FSSH and STSH values are averaged over 10 000 trajectories for each data point. The channels T1, T2, R1, and R2 are the same as in the previous plot. FSSH and STSH both give good approximations to the exact wavepacket scattering probabilities at high k. In the resonance region $7.7 < k < 8.9$, STSH gives a slightly more accurate reflection probability R1 but gives a slightly less accurate upper state transmission probability T2 just above the resonance region (see inset).

$x_0 = -10$ and an initial width parameter of $\alpha = .25$. The figure shows that both the FSSH algorithm and the STSH algorithm give good results above threshold ($k > 8.9$) as should be expected since decoherence is fairly rare in this regime. Below threshold, things are a bit more interesting. Although neither method yields results which are identical to the exact result, the STSH algorithm is slightly closer to the exact result in which it predicts a reduced resonance peak relative to FSSH.

B. Tully problem 2

The diabatic Hamiltonian for Tully problem 2 is given by the matrix elements:

$$V_{11}(x) = 0, \tag{27}$$

$$V_{22}(x) = -A\exp(-Bx^2) + E_0, \tag{28}$$

$$V_{12}(x) = V_{21}(x) = C\exp(-Dx^2) \tag{29}$$

with $A = 0.10$, $B = 0.28$, $E_0 = 0.05$, $C = 0.015$, and $D = 0.06$. This problem was originally constructed because it contains two regions of nonadiabatic coupling. As a result, the scattering probabilities exhibit Steuckelburg oscillations where the phase of the wavepackets on the upper and lower surfaces oscillates as a function of the time spent in the center region of the potential. However, what is even more interesting is the presence of a large number of resonance states below the threshold of the upper surface ($k < 14.1$).

When the total kinetic energy is sufficient to transiently populate the upper adiabat, but insufficient to populate the upper adiabat asymptotically, the particle can be trapped for a long time in the well on the upper surface. Figure 4 shows

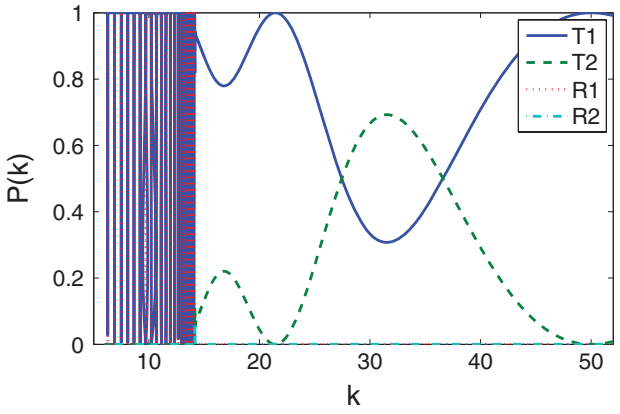

FIG. 4. The exact time-independent scattering probabilities $P(k)$ as a function of the wavevector k where T1, T2, R1, and R2 are defined as in previous plots. The plot clearly shows a large number of narrow resonances between $8.0 < k < 14.1$. These resonances make it difficult to calculate the time-independent scattering probabilities from time-dependent wavepacket simulations. Above the threshold of the upper surface ($k > 14.1$) scattering probabilities vary smoothly with k.

the time-independent scattering probabilities as a function of wavevector k. Because the well on the upper state is significantly deeper and more broad than that in Tully problem 1, there are a much larger number of quasibound resonances that produce peaks in the scattering probabilities. One important feature to notice is the narrow width of some of the resonance peaks which is due to the small spatial overlap of the resonance states with the two regions of nonadiabatic coupling. This narrow width leads to an extremely long lifetime. Computationally, the narrow width of these resonances means that even time-independent approaches may have a difficult time detecting them. Only if the value of the k-vector coincides almost precisely with the resonance energy will the resonance be detectable in time-independent scattering calculations. Again, these resonances highlight the important role of wavepacket width in the observed branching ratios. If the wavepacket is broad enough (in position space) individual resonances may be visible, but if the wavepacket is narrow enough (in position space) the individual resonances will be indistinguishable due to convolution with the initial k-space distribution.

The three traces in Fig. 5 show the exact time-dependent results using a wavepacket of width $\sigma_x = 2$, the FSSH results, and the STSH results using an initial position of $x_0 = -15$ and a width of $\sigma_x = \alpha^{-1/2} = 2$. Both the FSSH algorithm and the STSH algorithm again give good results above threshold ($k > 14.1$). Below threshold, STSH again predicts a reduced resonance peak, which is closer to the exact answer than the large resonance contribution predicted by FSSH. Still, neither algorithm is clearly more accurate for all values of k.

C. Tully problem 3

The previous examples showed that the STSH algorithm gives very similar results as the FSSH algorithm for Tully problems 1 and 2. This similarity is to be expected given that the two surfaces are roughly parallel and one-dimensional, so that decoherence events are fairly rare: wave packets on the

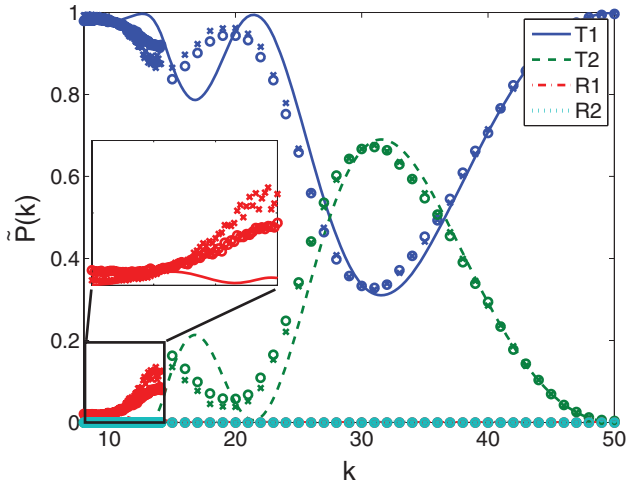

FIG. 5. A comparison of exact time-dependent scattering dynamics with $\sigma_x = 2$ (solid line) with FSSH (crosses) and STSH (circles) for Tully problem 2. FSSH and STSH values are averaged over 10 000 trajectories for each data point. The channels T1, T2, R1, and R2 are the same as in the previous plot. FSSH and STSH again both give good approximations to the exact wavepacket scattering probabilities at high k. In the resonance region, we again see that STSH provides a slightly more realistic assessment of back scattering on the lower surface (R1) than FSSH (see inset). But differences between the two algorithms are not dramatic.

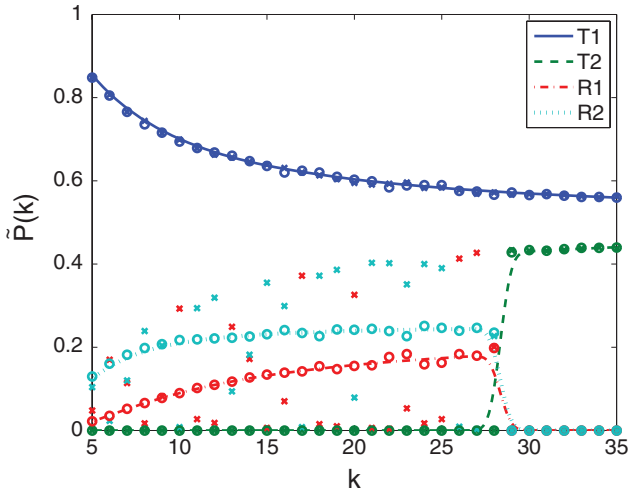

FIG. 6. A comparison of exact time-dependent scattering dynamics with $\sigma_x = 2$ (solid line) with FSSH (crosses) and STSH (circles) for Tully problem 3. FSSH and STSH values are averaged over 10 000 trajectories for each data point. The channels T1, T2, R1, and R2 are the same as in the previous plot. Due to the dramatic dependence of the nuclear forces on the electronic state, the wavepackets on the upper and lower adiabatic surfaces rapidly diverge when the particle does not have sufficient energy to overcome the barrier on the upper surface. FSSH results for R1 and R2 fluctuate rapidly and unphysically with k because the wavefunction remains fully coherent over the course of the trajectory. In contrast, STSH correctly allows the wavefunction to decohere when the upper and lower components diverge, yielding very stable and accurate scattering probabilities.

two surfaces will not readily diverge (at least at high k) and the STSH algorithm will behave almost exactly like the FSSH algorithm. There was some slight improvement in branching coefficients in certain regimes, particularly for low values of k where the wavepackets might be expected to diverge more quickly, but a clear improvement in accuracy was hard to discern. The most that could be said was that STSH did not perform any less well than FSSH, which is actually encouraging since FSSH yields fairly accurate results for the first two problems. However, the third Tully problem is a very interesting case, since it is known that the original FSSH algorithm performs quite poorly on this problem.

The diabatic Hamiltonian for Tully problem 3 is given by the matrix elements:

$$V_{11}(x) = A, \tag{30}$$

$$V_{22}(x) = -A, \tag{31}$$

$$V_{12}(x) = B\exp(Cx), x < 0, \tag{32}$$

$$V_{12}(x) = B[2 - \exp(-Cx)], x > 0, \tag{33}$$

$$V_{21}(x) = V_{12}(x) \tag{34}$$

with $A = 6 \times 10^{-4}$, $B = 0.1$, and $C = 0.9$.

The reason for the poor performance of FSSH on this problem is clear if we consider the adiabatic potentials shown in Fig. 1(c). At low k, the upper channel is closed to transmission so that all particles traveling on the upper surface will reflect while particles traveling on the lower surface ought to transmit. As a result, trajectories ought to experience strong decoherence which the original FSSH algorithm will fail to capture.

Figure 6 shows the exact results for a wavepacket of width $\sigma_x = 2$, the FSSH results and the STSH results with $x_0 = -15$ and a width of $\sigma_x = \alpha^{-1/2} = 2$. As the figure shows, STSH yields very accurate results compared with the strongly unphysical behavior of FSSH. What is more, an examination of the STSH trajectories shows that on average the particle experiences precisely one decoherence event over the course of a trajectory due to the divergence of the trajectories on the upper and lower surfaces at low k. This problem is clear evidence that STSH is adding decoherence to FSSH in a physical way which accords with our expectations and which yields results which are quantitatively and qualitatively more accurate than those of FSSH.

D. Double arch problem

Finally, we applied our algorithm to a model problem recently constructed by Subotnik *et al.*, who pieced together two instances of the third model problem in Tully's original paper, to create a potential that ought to show clear signatures of decoherence. The diabatic Hamiltonian for the double arch problem is given by the matrix elements:

$$V_{11}(x) = A, \tag{35}$$

$$V_{22}(x) = -A, \tag{36}$$

$$V_{12}(x) = -B\exp(C(x - Z)) + B\exp(C(x + Z)), x < -Z, \tag{37}$$

$$V_{12}(x) = -B\exp\left(C(x-Z)\right) - B\exp-(C(x+Z)),$$
$$-Z < x < Z, \tag{38}$$

$$V_{12}(x) = B\exp\left(-C(x-Z)\right) - B\exp-(C(x+Z)), x > Z, \tag{39}$$

$$V_{21}(x) = V_{12}(x) \tag{40}$$

with $A = 6\times10^{-4}$, $B = 0.1$, $C = 0.9$, and $Z = 4$.

Figure 1(d) shows a plot of the adiabatic surfaces and derivative coupling functions. We can expect decoherence to play a major role in this problem if we sketch out how a wavepacket should behave as it passes through the various spatial regions of this potential. For a wavepacket that begins on the lower adiabatic surface, the nonadiabatic coupling region around $x = -10$ should simply flip it onto a superposition of the upper and lower adiabatic states. If $k < 28.3$, then the portion of the wavepacket on the upper surface will not have enough kinetic energy to pass over the potential barrier. Thus, the wavepacket on the upper surface will reflect while the portion of the wavepacket on the lower surface will transmit. This reflected component will then re-enter the coupling region at $x = -10$ and will again be flipped onto both the upper and lower surface. The component of the wavepacket on the lower adiabat will pass through the valley around $x = 0$ and will eventually hit the coupling region at $x = +10$ where it will be flipped onto both the upper and lower surface. All of this reasoning leads us to expect that for $k < 28.3$ all four channels will have probabilities of $\sim$0.25 which is indeed roughly what we see.

A major change occurs for $k > 28.3$, when the wavepacket on the upper surface has sufficient energy to surmount the potential barrier. In that case, there is a forward-moving component on both the upper and lower surfaces in the intermediate region between $-5 < x < 5$. Two factors will come into play. First, a relative phase between the upper and lower wavepackets will be acquired as it passes through this intermediate region. Hence, there ought to be an oscillation between transmission on the upper and lower surfaces when the two components of the wavepacket recombine at $x = +10$. Second, and more importantly for us, the two wavepackets ought to move apart because the wavepacket on the upper surface will have a far smaller momentum than that on the lower surface. If this difference in momenta is large enough, then the two wavepackets will completely separate before they reach the recombination region at $x = +10$, removing the possibility of either constructive or destructive interference. Thus, each component will be flipped by the nonadiabatic coupling independent of the other component and we would expect to obtain transmission coefficients of roughly 0.5 for both channels. This is precisely what we mean when we talk about decoherence. As we increase k, the relative difference in momenta between the two wavepackets will become smaller. The two wavepackets will undergo less spatial separation in the middle region such that they can again recombine to produce constructive or destructive interference depending on their relative phase. Thus, we expect that as we increase k, oscillations between transmission on the upper and lower surfaces ought to become visible.

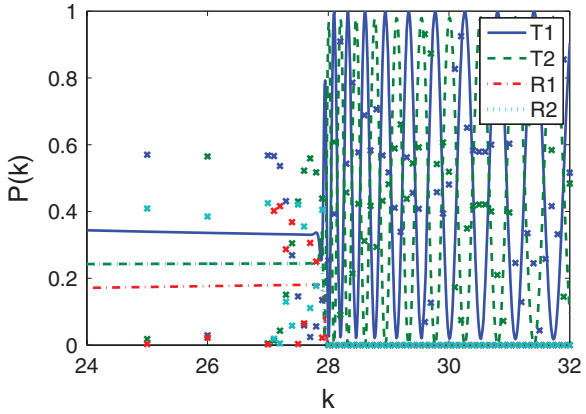

FIG. 7. The exact time-independent scattering probabilities $P(k)$ (solid lines) as a function of the wavevector k for the double arch problem where T1, T2, R1, and R2 are defined as in previous plots. When the particle does not have sufficient energy to surmount the barrier on the upper state $k < 28.3$, the scattering probabilities closely follow our classical intuition and vary smoothly with k. But when the particle can overcome the potential barrier on the upper surface, there is dramatic interference between the upper and lower components of the wavefunction and we obtain near-complete oscillation between transmission on the upper and lower adiabats due the accumulated phase difference between the upper and lower trajectories. Oscillation is most rapid just above the threshold because the difference in momenta is most pronounced. Also shown are the FSSH probabilities (crosses), which give qualitatively incorrect results at all energies. See Fig. 8 for a plot of the FSSH results averaged over stochastic initial conditions.

Figure 7 shows the exact time-independent scattering results for the double arch problem. For $k < 28.3$, Fig. 7 shows that our wavepacket-based intuition is born out for $k < 28.3$ where the scattering probabilities are all indeed roughly equal to 0.25. However, when transmission over the upper barrier is possible for $k > 28.3$, we find complete oscillation between the upper and lower adiabats, corresponding to the accumulated phase difference between the two pathways. What is more, there is no sign at all of decoherence or a suppression of this oscillation due to the separation of the wavepackets taking the upper and lower pathways. Because the time-independent picture corresponds to the infinite-width limit, the wavepacket overlap never decays and hence there is no decoherence. In fact, decoherence is a consequence of using an initial wavepacket with finite width. For a finite-width wavepacket corresponding to the time-dependent picture, we must convolute the time-independent scattering probabilities with a Gaussian in k-space. In that case, for k just above the barrier (i.e., $k = 28$), oscillation is extremely rapid due to a large difference in the accumulated phase. Hence, convolution with any finite-width Gaussian will lead to complete suppression of the oscillation such that both channels transmit with probability $\sim$0.5. As k increases, the oscillation period increases so that convolution with a finite-width Gaussian will no longer completely wash out this underlying oscillation. This is the recoherence that we see in the wavepacket picture as k increases. In other words, there is an exact correspondence—as expected—between time-dependent and time-independent scattering. However, it appears that decoherence is best understood from the perspective of time-dependent wavepacket scattering.

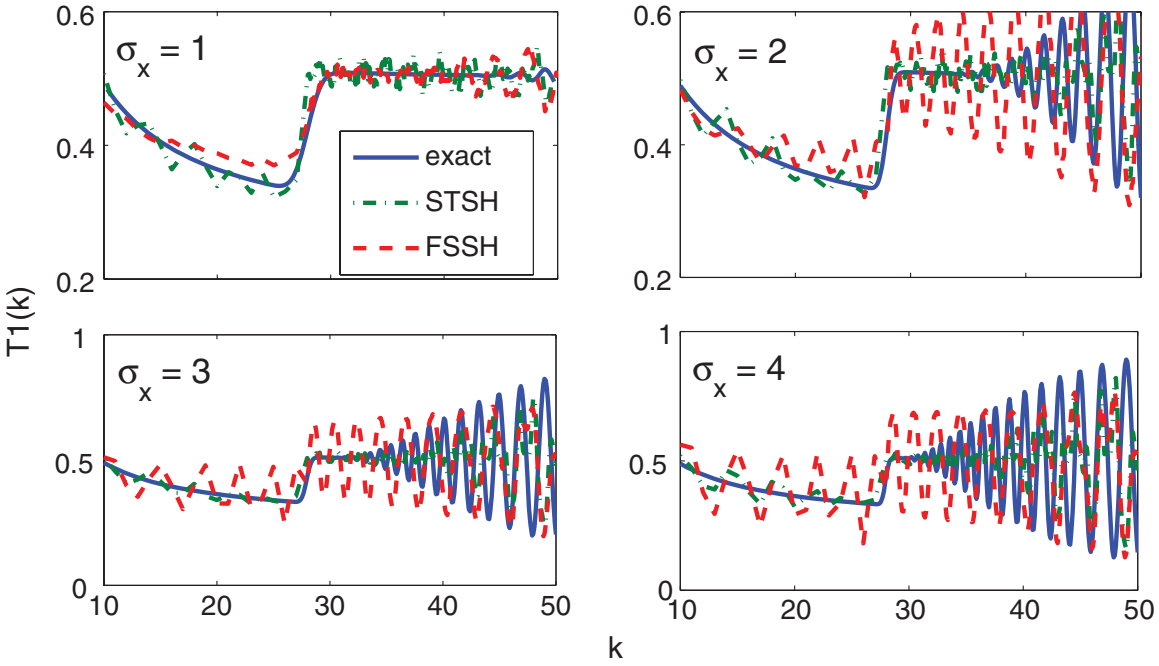

FIG. 8. The time-dependent transmission probabilities on the lower surface $T1(k)$ as a function of the wavevector k for the double arch problem. Panel (a), (b), (c), and (d) show the results with an initial wavepacket at $x_0 = -20$ with width $\sigma_x = 1, 2, 3$, and 4, respectively. Solid lines show exact time-dependent scattering probabilities, dashed–dotted lines show the corresponding STSH calculations, and dashed lines show the smoothed FSSH results. Although STSH does not provide quantitative accuracy, it qualitatively predicts the correct branching ratios below threshold ($k < 28.1$) and the onset of oscillations as a function k above threshold ($k < 28.1$).

Because the double arch problem was designed to exhibit a significant amount of decoherence, it should provide an excellent test for the STSH algorithm. However, even if STSH produces qualitatively correct results, we might wonder whether this success is merely fortuitous. How can we be sure that our heuristic involving wavepacket width and the decay of the overlap is actually capturing the correct physics? To answer this question, we calculated the transmission probability on the lower surface for four different values of the initial wavepacket width (see Fig. 8), where the exact results are compared to the STSH and FSSH results. Since the degree of decoherence will depend sensitively on the initial width, these plots act as probes of the physical plausibility of our algorithm. Following the suggestion made by Tully in Ref. 9, the FSSH results in this figure were averaged over stochastic initial conditions corresponding to the width of the initial Gaussian wavepacket. This averaging does smooth away some of the unphysical oscillations. However, the results clearly show that even when results are highly smoothed [i.e., in Fig. 8(a)], the branching probabilities exhibit quantitative deviations from the exact result. In addition, FSSH gives oscillations above threshold ($k > 28.1$) which are consistently too large in magnitude.

In contrast, in all four panels of Fig. 8, we see that STSH yields good qualitative agreement with the wavepacket results. Below threshold ($k < 28.1$), STSH does not show the systematic error in branching ratios exhibited by FSSH. Above threshold ($k > 28.1$), STSH correctly captures the onset and absolute magnitude of the oscillations a function of k, although neither STSH or FSSH correctly captures the phase of these oscillations. It should be noted that any algorithm which does not take into account the initial width of the wavepacket will not be able to differentiate between these four cases. These plots show that the extrinsic width of the wavepacket is vitally important to decoherence properties, as our theoretical arguments also support. Note also that the STSH results do not make use of the stochastic averaging employed for the FSSH results. Instead, we are using a single, deterministic initial position and momentum and introducing decoherence purely through our the STSH mechanism. These plots are evidence that the decoherence rate calculated by STSH is not an *ad hoc* heuristic, but is capturing real physical processes of decoherence that occur when wavepackets on different adiabats diverge.

VI. DISCUSSION AND CONCLUSIONS

In summary, we have constructed a new mixed quantum-classical dynamics algorithm which introduces decoherence effects into the FSSH algorithm. The STSH algorithm is able to calculate the scattering probabilities for four model problems, two of which were specifically designed to necessitate proper treatment of wavepacket decoherence. In addition to the computational accuracy of our approach, there are several theoretical features which distinguish it from previous approaches involving decoherence and surface hopping. First, our algorithm includes an explicit treatment of the initial and dynamical wavepacket width. Notably, this explicit treatment of an extrinsic width parameter is one of the main features that differentiates the STSH algorithm from the A-FSSH algorithm introduced in Ref. 28. The A-FSSH algorithm does not take into account the initial or dynamic wavepacket width, but

is limited to capturing the "intrinsic decoherence" of the problem. The effect of extrinsic width on decoherence has not—as far as we are aware—ever been previously included but was shown in Sec. V D to be vital to capturing realistic decoherence in wavepacket calculations. Second, our algorithm is completely free from parameters. There is no imposed coherence length, thermal temperature, or other measure of decoherence which needs to be specified to obtain results. Again, this is important since the real physics of wavepacket scattering does not depend on a special decoherence parameter. Third, our algorithm is computationally efficient. While multiple-spawning algorithms potentially requires the simultaneous propagation of an exponential number of wavepackets, our STSH algorithm requires the propagation of at most N simultaneous trajectories, one for each adiabatic surface on the Hilbert space. Consequently, computational resources scale linearly with the size of the Hilbert space when compared to FSSH. Finally, our algorithm makes few changes to traditional FSSH, retaining its intuitive underlying structure and ease of implementation.

One of the fundamental questions raised by our STSH algorithm and any surface-hopping algorithm which includes decoherence is whether decoherence can by understood from a time-independent perspective. For instance, based on our description of decoherence in the double arch problem in Sec. V D, it is clear that decoherence is easy to recognize in *time-dependent wavepacket scattering calculations*. But our narrative raises the important question of whether there is any signature of decoherence in the time-independent scattering matrix. For instance, can we examine the time-dependent scattering matrix $S(\mathbf{k}, \mathbf{k}')$ and determine whether the fully coherent FSSH algorithm will be successful? The answer is yes, there do seem to be certain features of the time-independent scattering matrix $S(\mathbf{k}, \mathbf{k}')$ that indicate when decoherence will play a role in wavepacket calculations such that FSSH will breakdown. In fact, there appear to be two distinguishable types of decoherence that both play a role in the breakdown of the FSSH algorithm: free-space decoherence and force-induced decoherence, both of which were identified in Ref. 28 and play a role in the double arch problem (see Sec. V D).

The first type of decoherence, free-space decoherence, is clearly visible for $k > 28.3$ in the double arch problem. In this regime, the wavepacket has sufficient energy to pass over the barrier on both surfaces. In passing over the barrier region, the two components of the wavepacket acquire a relative phase which oscillates with the time spent in the barrier region and leads to oscillations in branching ratios as a function of wavepacket momentum k (see Figs. 7 and 8). In the time-dependent picture, decoherence occurs because the two wavepackets travel with different velocities on the upper and lower surfaces such that they separate spatially. We refer to this separation as "free-space decoherence"[28] because this effect would occur in free space, whether or not there are any forces acting on the wavepackets. One important feature of free-space decoherence is that it disappears as the wavepackets become infinitely wide (i.e., as we move to the time-independent limit) because the overlap between the two wavepackets does not decay. In other words, free-space decoherence is really a consequence of the finite k-space width of the wavepacket. Therefore, we expect free-space decoherence to undermine the results of FSSH calculations whenever the exact time-dependent scattering matrix elements $S(\mathbf{k}, \mathbf{k}')_{ij}$ oscillate rapidly as a function of the *magnitude* of the outgoing wavevector $\mathbf{k}$. It may be possible to remove some of the effects of free-space decoherence simply by averaging FSSH results over some initial width in k-space as was suggested in Ref. 9, since this averaging will smooth out the oscillations that are indicative of free-space decoherence.

The second type of decoherence, force-induced decoherence, comes from the fact that scattering will often cause wavepackets moving on different surfaces to move in entirely different directions due to the different forces they experience. In 1D this effect is typified in the failure of FSSH to predict the scattering probabilities for $k < 28.3$ in the double arch problem. When $k < 28.3$ the particle will transmit on the lower surface but will be reflected on the upper surface. In other words, the direction of the outgoing wavevector k depends on the electronic state of the particle. The consequences of this effect are not as important in 1D because the particle has little freedom of motion; it can only reflect or transmit. But this effect will play a crucial role in higher dimensions where the particle can be deflected in a huge number of directions such that coherence between trajectories with different outgoing wavevectors will certainly decay. Mathematically, force-induced decoherence will undermine the results of FSSH calculations whenever the exact time-independent scattering matrix elements $S(\mathbf{k}, \mathbf{k}')_{ij}$ depend nonseparably on the *direction* of the outgoing wavevector $\hat{k}$. In other words, if $S(\mathbf{k}, \mathbf{k}')_{ij}$ cannot be written as a separable function $f(\mathbf{k}, \mathbf{k}') \times g_{ij}(\mathbf{k}')$, then there is some entanglement between the electronic state and the direction of the outgoing nuclear wavepacket. In 1D we see that this effect occurs whenever the real trajectories bifurcate according to their electronic state. If nonadiabatic transitions make a particle both reflect and transmit for a given initial wavevector $\mathbf{k}$ (as they do in the double arch problem with $k < 28.3$), then surface hopping will be unable to capture this bifurcation. It seems that there is no clear way to remove the effects of force-based decoherence within the context of the FSSH algorithm without explicitly including decoherence. Simply averaging over some distribution of initial momenta will not alter the coherent nature of each FSSH trajectory, leading to inaccuracies in the FSSH results. Thus, even though decoherence is necessarily a time-dependent phenomenon, it appears that its importance can be determined from an examination of the time-independent scattering matrix $S(\mathbf{k}, \mathbf{k}')_{ij}$.

Another fundamental question raised by our algorithm is whether a quantum wavepacket width ought to be included in all nonadiabatic dynamics algorithms. This is an important question and the answer will depend on what observables we are trying to simulate. If our desire is to approximate the results of a true time-independent scattering experiment, then the incoming state is a plane-wave. In this case, we would hope that nonadiabatic algorithm results would be width-independent. However, if we are trying to approximate the results of a time-dependent scattering experiment, then the incoming state will be a wavepacket with a real,

finite width. Physically, this situation might occur for photoinduced dynamics in which a nuclear wavepacket is formed on an excited electronic surface due to photoabsorption. In this case, the above model problems show that an algorithm which explicitly includes the wavepacket width is necessary for obtaining accurate results.

Because this paper is primarily intended to introduce a new algorithm, there are necessarily numerous avenues which are yet unexplored. Foremost among these is the question of how to rescale the momenta (or alter the nuclear configuration) during a surface hop, which has always been an open question in the surface hopping literature. The STSH algorithm offers an intriguing possibility with regards to momentum rescaling. If a hop occurs from surface i to surface j, there is no guarantee that there will be an active trajectory propagating on surface j. In this case, the traditional momentum rescaling approach must suffice. However, if there is an active trajectory propagating on surface j, we could consider hopping from the phase space point (x_i, k_i) directly to the phase-space point (x_j, k_j). Alternatively, we could consider rescaling the momenta only along the direction $k_i - k_j$. Either of these alternatives is quite distinct from traditional momentum rescaling and offers new possibilities for more accurate surface hops.

Another important avenue for research is how to replace deterministic spawning at the local maxima of $\mathbf{p} \cdot \mathbf{d}$ with stochastic spawning at some calculated spawning rate. One possibility is to run a trajectory forward until a local maximum of $\mathbf{p} \cdot \mathbf{d}$ is reached at time t_m. Based on the value of $\mathbf{p} \cdot \mathbf{d}$ at this maximum, the spawning rate could be rescaled so that the probability of spawning between time 0 and time t_m is 1. Then the algorithm could be restarted at time 0 using the new rescaling factor. The downside of this approach is that it is nonlocal in time and is not very elegant. However, it avoids the problem that spawning occurs only at particular seams in configuration space where $\mathbf{p} \cdot \mathbf{d}$ reaches a maximum and replaces it with a stochastic approach that is more reminiscent of surface hopping.

Many other minor details could also be altered. For instance, if there is insufficient energy to spawn on an excited state adiabat, should we use the current approach which does not conserve the energy of the spawned trajectory? Should we introduce forbidden spawns analogous to forbidden hops? How are we to transfer the width parameters from one surface to another during a spawning event or a surface hop? We could imagine changing one or all of these aspects of the STSH algorithm, possibly improving its accuracy significantly. Indeed, the fact that we did not apply a careful, systematic analysis of our spawning or hopping criteria implies that there is almost certainly a room for improvement.

Finally, it is unclear how our STSH algorithm will perform in higher-dimensional problems, for which efficient surface hopping schemes are particularly useful. It will be important to develop new low-dimensional model problems ($N = 2$ or $N = 3$) which include significant decoherence effects. Developing such models will enable us to differentiate between the accuracy of different algorithms and will enable us to determine where their errors arise. Obviously, there is great potential for future work and we hope that the STSH algorithm will spawn a host of new ideas.

ACKNOWLEDGMENTS

N.S. and W.Y. would like to acknowledge the support from the UNC EFRC: Solar Fuels and Next Generation Photovoltaics, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Award No. DE-SC0001011. J.E.S. thanks the University of Pennsylvania for start-up funding.

[1] J. C. Tully, Int. J. Quantum Chem. **40**, 299 (1991).
[2] A. W. Jasper, S. Nangia, C. Zhu, and D. G. Truhlar, Acc. Chem. Res. **39**, 101 (2006).
[3] J. C. Tully, Annu. Rev. Phys. Chem. **51**, 153 (2000).
[4] D. R. Yarkony, Rev. Mod. Phys. **68**, 985 (1996).
[5] S. Hammes-Schiffer and J. C. Tully, J. Chem. Phys. **101**, 4657 (1994).
[6] J.-Y. Fang and S. Hammes-Schiffer, J. Chem. Phys. **106**, 8442 (1997).
[7] S. Hammes-Schiffer, Acc. Chem. Res. **34**, 273 (2001).
[8] M. Ben-Nun, J. Quenneville, and T. J. Martinez, J. Phys. Chem. A **104**, 5161 (2000).
[9] J. C. Tully, J. Chem. Phys. **93**, 1061 (1990).
[10] J. R. Schmidt, P. V. Parandekar, and J. C. Tully, J. Chem. Phys. **129**, 044104 (2008).
[11] M. Ben-Nun and T. J. Martínez, J. Chem. Phys. **108**, 7244 (1998).
[12] R. Kapral and G. Ciccotti, J. Chem. Phys. **110**, 8919 (1999).
[13] A. Donoso and C. C. Martens, J. Chem. Phys. **112**, 3980 (2000).
[14] C.-C. Wan and J. Schofield, J. Chem. Phys. **113**, 7047 (2000).
[15] M. Thoss, W. H. Miller, and G. Stock, J. Chem. Phys. **112**, 10282 (2000).
[16] S. Yang, J. D. Coe, B. Kaduk, and T. J. Martínez, J. Chem. Phys. **130**, 134113 (2009).
[17] O. V. Prezhdo and P. J. Rossky, J. Chem. Phys. **1097**, 825 (1997).
[18] E. J. Heller and D. Beck, Chem. Phys. Lett. **202**, 350 (1993).
[19] E. J. Heller, B. Segev, and A. V. Sergeev, J. Phys. Chem. B **106**, 8471 (2002).
[20] M. 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).
[21] F. Webster, P. J. Rossky, and R. A. Friesner, Comput. Phys. Commun. **63**, 494 (1990).
[22] E. R. Bittner and P. J. Rossky, J. Chem. Phys. **103**, 8130 (1995).
[23] B. J. Schwartz, E. R. Bittner, O. V. Prezhdo, and P. J. Rossky, J. Chem. Phys. **104**, 5942 (1996).
[24] J.-Y. Fang and S. Hammes-Schiffer, J. Phys. Chem. A **103**, 9399 (1999).
[25] C. Zhu, S. Nangia, and D. G. Truhlar, J. Chem. Phys. **121**, 7658 (2004).
[26] G. Granucci and M. Persico, J. Chem. Phys. **126**, 134114 (2007).
[27] G. Granucci, M. Persico, and A. Zoccante, J. Chem. Phys. **133**, 134111 (2010).
[28] J. E. Subotnik and N. Shenvi, J. Chem. Phys. **134**, 024105 (2011).
[29] E. Neria and A. Nitzan, J. Chem. Phys. **99**, 1109 (1993).
[30] E. J. Heller, J. Chem. Phys. **62**, 1544 (1975).
[31] J. C. Tully and R. K. Preston, J. Chem. Phys. **55**, 562 (1971).