

The Quantum Trajectory-guided Adaptive Gaussian Methodology in the Libra Software Package

Matthew Dutra,[†] Sophya Garashchuk,^{*,†} and Alexey V. Akimov^{*,‡}

[†]Department of Chemistry and Biochemistry, University of South Carolina, Columbia, SC 29208, USA

[‡]Department of Chemistry, University at Buffalo, Buffalo, NY 14260, USA

E-mail: garashchuk@sc.edu; alexeyak@buffalo.edu

Abstract

In this paper we report an implementation of the quantum trajectory-guided adaptive Gaussian (QTAG) method in a modular open-source Libra package for quantum dynamics calculations. The QTAG method is based on a representation of wavefunctions in terms of a quantum trajectory-guided adaptable Gaussians basis and is generalized for time-propagation on multiple coupled surfaces to be applicable to model nonadiabatic dynamics. The potential matrix elements are evaluated within either the local harmonic or bra-ket-average (linear) approximations to the potential energy surfaces, the latter being a more practical option. Performance of the QTAG method is demonstrated and discussed for the Holstein and Tully models, which are the standard benchmarks for method development in the area of nonadiabatic dynamics.

Keywords — nonadiabatic dynamics, quantum trajectories, Gaussian basis, Tully models, QTAG, Libra.

1 Introduction

Nonadiabatic effects of electron-nuclear interactions are essential in many chemical processes,^{1–6} including photophysics and photochemistry involving charge and energy transfer, and reactive dynamics in the condensed phase. The systems exhibiting electronically nonadiabatic behavior range from electron and/or proton transfer in chemical and biological systems,^{4,5,7} to light-harvesting materials,^{6,8–12} to reaction control and optochemistry.¹³ Therefore, the development of time-dependent nonadiabatic dynamics methods has received a lot of attention from many theorists utilizing a variety of approaches.^{14–18} Given the expense and unfavorable scaling of exact quantum dynamics methods, an often-employed strategy is to combine a classical treatment of the nuclei with a semi-classical or quantum description of electrons.^{19–21} The trajectory surface hopping (TSH) and Ehrenfest methods – both used in numerous applications^{8,22,23} – are the most popular methods of this type. The TSH and the Ehrenfest dynamics are both based on an ensemble of independent classical trajectories: in the former, these trajectories move on one potential energy surface with a possibility of electronic transitions to different surfaces, while in the latter all trajectories move on a single effective potential determined via an averaging procedure over the electronic states. While the classical description of the nuclei makes these methods much more practical than the exactly-calculated alternatives, they fail to incorporate nuclear quantum effects (NQEs), which are at times essential for the understanding and characterization of a system. This shortcoming motivates the active development of trajectory-based methods incorporating NQEs at least approximately. An incomplete list of such methods includes those based on ring-polymers,^{24–26} quantum trajectories,^{27,28} exact factorization^{15,29–32} and various approaches employing Gaussian bases.^{33–36} We refer an interested reader to Ref.¹⁶ for a recent review of nonadiabatic dynamics methodologies.

A natural way to add NQEs to the nuclear dynamics is to represent nuclear wavefunctions in terms of basis functions that follow trajectories capturing the broad features of the system’s time-evolution. In this paper, we describe a multi-state generalization of a formally exact

quantum dynamics method based on the trajectory-guided basis representation of nuclear wavefunctions, i.e. the quantum trajectory-guided adaptable Gaussians (QTAG).^{37–39} In the QTAG method – unlike many other Gaussian methods – the basis function parameters are ‘optimized’ by the time evolution of the wavefunction. This approach offers several conceptual advantages: (1) the trajectory paths are guided by the shape and coordinate-dependent phase of the wavefunction itself, creating an efficient sampling of the configuration space; (2) the NQEs are captured intrinsically, without requiring external corrections; (3) thanks to the trajectories defining the basis functions’ centers, the QTAG formalism smoothly connects to semi-classical and classical descriptions of nuclear motion and to the hierarchical treatment of large molecular systems.

The remainder of the paper is organized in the following way: QTAG dynamics and the propagation algorithm are reviewed, and the nonadiabatic generalization is presented in Section 2. The multi-surface, multidimensional QTAG implementation within the Libra software suite⁴⁰ – including organization of the package, its key components and the main algorithms used in their implementation – is described in Section 3. The utility of the QTAG dynamics is demonstrated and discussed in the context of several standard nonadiabatic models in Section 4. Summary and outlook is presented in Section 5.

2 Theoretical Background

2.1 Notation, Wavefunction, and Basis Definitions

The theoretical underpinnings of the QTAG method are described in full elsewhere.³⁸ A synopsis of the fundamental equations is presented here employing atomic units ($\hbar = 1$), Cartesian coordinates and the diagonal form of the kinetic energy for clarity. At its core, this method seeks to solve the time-dependent Schrödinger equation (TDSE):

$$i\frac{\partial}{\partial t}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle = [\hat{T} + \hat{V}]|\Psi(t)\rangle. \quad (1)$$

Here, \hat{T} is the nuclear kinetic energy operator and \hat{V} is the electronic Hamiltonian operator. The electron-nuclear wavefunction, $|\Psi(t)\rangle$ is represented in the basis of electronic states ($|i\rangle$ such that $\langle i|j\rangle = \delta_{ij}$) and nuclear Gaussian basis functions (GBFs, $|G_k\rangle$):

$$|\Psi(t)\rangle = \sum_{i=0}^{M-1} \sum_{k=0}^{N_b-1} C_{(i,k)}(t) |i, G_{(i,k)}(\lambda_{(i,k)}(t))\rangle = |\mathbf{G}\rangle \mathbf{C} = |\tilde{\mathbf{G}}\rangle \mathbf{B}, \quad (2a)$$

$$|i, G_{(i,k)}(\lambda_{(i,k)}(t))\rangle = |i\rangle |G_{(i,k)}(\lambda_{(i,k)}(t))\rangle. \quad (2b)$$

In our notation, bold-faced letters denote matrices and vectors. Here, $C_{(i,k)}(t)$ are the time-dependent coefficients corresponding to the non-orthogonal GBF $|G_{(i,k)}(\lambda_{(i,k)}(t))\rangle$ associated with electronic state i and trajectory k on that state, M is the number of electronic states considered, and N_b is the number of GBFs per electronic state. The adaptable GBF parameters $\lambda_{(i,k)}(t)$ are explained below. In the current implementation, we populate each state with the same number of GBFs, although they may carry zero amplitudes. In principle, each surface may need a different number of GBFs and one may allow GBF transitions (hops) across states, although such an adaptive approach is not included in the current implementation. One may argue that incorporating the trajectory (or GBF) “hopping” is redundant, since such events are naturally included in the time-dependence of the GBF coefficients on different surfaces. Therefore, a better alternative to GBF transitions (not pursued here) would be to adjust the sizes of the GBF sets on each surface as the quantum system evolves according to a given Hamiltonian.

The notation $|\mathbf{G}\rangle$ is used to denote a vector of state vectors: $|\mathbf{G}\rangle = (|G_0\rangle, |G_1\rangle, \dots, |G_{N_{traj}-1}\rangle)$, whereas the matrix \mathbf{C} represents a vector-column $\mathbf{C} = (C_0, C_1, \dots, C_{N_{traj}-1})^T$, such that the short-hand notation $|\mathbf{G}\rangle \mathbf{C}$ corresponds to the scalar product, $|\mathbf{G}\rangle \mathbf{C} = \sum_{I=0}^{N_{traj}-1} |G_I\rangle C_I$. Here, $I \leftrightarrow (i, k)$ is a mapped index of the GBFs in the whole set of functions for all electronic states, and $N_{traj} = M \times N_b$ is the total number of such functions on all surfaces. Since both \mathbf{C} and $|\mathbf{G}\rangle$ are essentially one-dimensional (1D) vectors, the indices corresponding to the state and GBF are combined within parenthesis in Eq. 2b, to not confuse them with the

indices of a column and row of a regular matrix. Furthermore, we assume the state-block ordering of the coefficients and GBFs, that is:

$$\mathbf{C} = (C_{(0,0)}, \dots, C_{(0,N_b-1)}, C_{(1,0)}, \dots, C_{(1,N_b-1)}, \dots, C_{(M-1,0)}, \dots, C_{(M-1,N_b-1)})^T, \quad (3a)$$

$$|\mathbf{G}\rangle = (|G_{(0,0)}\rangle, \dots, |G_{(0,N_b-1)}\rangle, |G_{(1,0)}\rangle, \dots, |G_{(1,N_b-1)}\rangle, \dots, |G_{(M-1,0)}\rangle, \dots, |G_{(M-1,N_b-1)}\rangle), \quad (3b)$$

$$\langle \mathbf{G}| = (\langle G_{(0,0)}|, \dots, \langle G_{(0,N_b-1)}|, \langle G_{(1,0)}|, \dots, \langle G_{(1,N_b-1)}|, \dots, \langle G_{(M-1,0)}|, \dots, \langle G_{(M-1,N_b-1)}|)^T. \quad (3c)$$

The coefficients \mathbf{B} in Eq. 2a correspond to the wavefunction expansion in an auxiliary orthogonalized basis $|\tilde{\mathbf{G}}\rangle$, related to the non-orthogonal GBF basis as:

$$|\tilde{\mathbf{G}}\rangle = |\mathbf{G}\rangle \mathbf{U}. \quad (4)$$

Here, \mathbf{U} is a unitary transformation matrix that orthogonalizes the GBF basis, that is, the one that diagonalizes the overlap matrix \mathbf{S} :

$$\mathbf{S} = \langle \mathbf{G} | \mathbf{G} \rangle. \quad (5)$$

Indeed, the basis $|\tilde{\mathbf{G}}\rangle$ is defined as $\langle \tilde{\mathbf{G}} | \tilde{\mathbf{G}} \rangle = I$, so $\langle \tilde{\mathbf{G}} | \tilde{\mathbf{G}} \rangle = \mathbf{U}^\dagger \langle \mathbf{G} | \mathbf{G} \rangle \mathbf{U} = \mathbf{U}^\dagger \mathbf{S} \mathbf{U} = \mathbf{I}$. One can also note a useful relationship $\mathbf{S}^{-1} = \mathbf{U} \mathbf{U}^\dagger$.

The GBFs are constructed as a product of 1D Gaussians for each degree of freedom. In the position representation, this takes the form:

$$\langle \mathbf{q} | G_I \rangle = G_I(\mathbf{q}; \lambda_I(t)) = \prod_{n=0}^{N_d-1} g(q_n; \lambda_I(t)), \quad (6)$$

where \mathbf{q} is the coordinate vector of N_d nuclear degrees of freedom, $\mathbf{q} = (q_0, q_1, \dots, q_{N_d-1})^T$.

The 1D Gaussians are defined as:

$$g(q_n; \lambda_I(t)) = \left(\frac{a_n^{(I)}(t)}{\pi} \right)^{1/4} \exp \left(- \frac{a_n^{(I)}(t)}{2} (q_n - Q_n^{(I)}(t))^2 + i P_n^{(I)}(t) (q_n - Q_n^{(I)}(t)) + i s_n^{(I)}(t) \right) \quad (7)$$

The QTAG GBFs – Eqs 6 and 7 – are parametrized by a set of real, adaptable, time-dependent parameters $\lambda_I(t) = \{\mathbf{Q}^{(I)}, \mathbf{P}^{(I)}, \mathbf{a}^{(I)}, \mathbf{s}^{(I)}\}$ whose tuneability offers an efficient means to represent the system wavefunction. Here, $Q_n^{(I)}(t)$ and $P_n^{(I)}(t)$ can be regarded as the n -th component of the coordinate and momentum of the I -th GBF ($I \leftrightarrow (i, k)$), whereas $a_n^{(I)}(t)$ and $s_n^{(I)}(t)$ are the n -th components of its width and phase vectors.

Although all the parameters are formally time-dependent quantities, our implementation allows for the opportunity of constraining the values of any of these parameters to remain constant. In the special case of all constant values, the approach reduces to the regular TDSE solution in a static basis. Such a limiting case has an important flaw which the QTAG approach is designed to overcome: the efficiency of the basis. In the fixed approach, an unnecessarily large basis is required to cover all the regions of configuration space where the system may evolve. However, under evolution in bounded potentials, the system’s wavefunction is likely to be reasonably localized, and the actual basis needed for an accurate description may be smaller. In the QTAG procedure, the basis functions instead evolve along quantum trajectories to achieve a compact and accurate representation of the wavefunctions.

2.2 Computing Hamiltonian Matrix Elements

The QTAG computational workflow described in the following sections requires evaluating matrices of three kinds – the GBF overlap matrix $\mathbf{S} = \langle \mathbf{G} | \mathbf{G} \rangle$, the kinetic energy matrix, $\mathbf{T} = \langle \mathbf{G} | \hat{T} | \mathbf{G} \rangle$, and the potential energy operator, $\mathbf{V} = \langle \mathbf{G} | \hat{V} | \mathbf{G} \rangle$. Considering the ordering of the electron-nuclear basis functions (Eq. 3b) and the orthogonality of the electronic components, one can note that the matrix \mathbf{S} – as well as any other matrices of the form $\langle \mathbf{G} | \hat{A} | \mathbf{G} \rangle$ (e.g. with $\hat{A} = \hat{T}$ or \hat{V}) – have a block structure with each block being an $N_b \times N_b$ sub-matrix whose elements correspond to different trajectories for a given set of electronic state indices (Fig. 1). The blocks of the \mathbf{S} and \mathbf{T} matrices corresponding to different electronic states are zero because of the orthogonality of the electronic components of the GBF, $\langle i | j \rangle = \delta_{ij}$, whereas the matrix \mathbf{V} may have non-zero elements in blocks connecting

different electronic states.

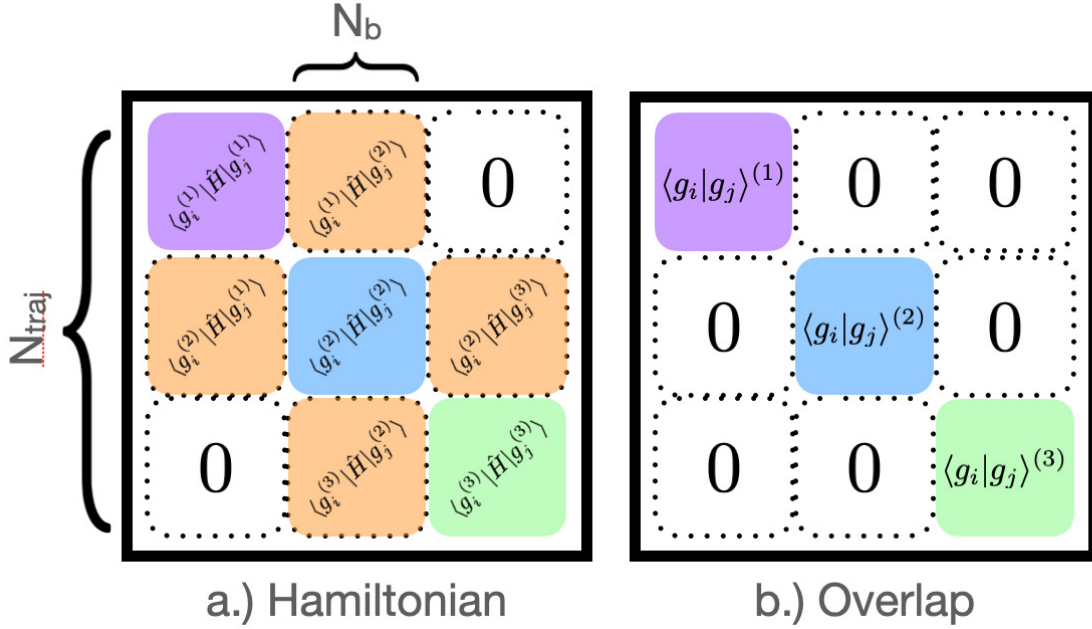


Figure 1: Hamiltonian (a.) and overlap (b.) super-matrices, each of dimension $N_{traj} \times N_{traj}$. The sub-matrices (dimension $N_b \times N_b$) correspond to single-surface Hamiltonians on the diagonal, or coupling Hamiltonians on the off-diagonals. In (a.), the **0**-blocks coupling states 1 and 3 are merely given as an example of uncoupled states 1 and 3; in (b.) all **0**-blocks are enforced, as basis functions of different surfaces do not overlap. The text within each sub-block indicates the calculation required for its entries, while the colors purple, blue and green are used for clarity to distinguish the surfaces 1, 2 and 3, respectively. The non-zero coupling blocks in the Hamiltonian super-matrix are marked as orange.

In general, the Hamiltonian and overlap matrices for an arbitrary number of quantum states, M , are referred to as the super-Hamiltonian and super-overlap, and are constructed from their single-surface counterparts (Fig. 1). The numeric superscripts on the right-hand side of sub-blocks index the electronic states; for example, $\mathbf{S}^{0,0}$ refers to an overlap matrix containing single-surface elements corresponding to state 0, and $\mathbf{H}^{1,M-1}$ corresponds to the Hamiltonian with elements which couple states 1 and $M-1$. The use of **0** in the definition of the super-overlap matrix indicates the zero-matrix, a matrix of the same rank as the block-diagonal entries and whose elements are all zeroes. This property of the super-overlap means that QTAG GBFs on different surfaces have no overlap with one another by construction; if this were not enforced, the resulting matrix would be singular and Eq. 21 would be rendered

un-invertible in the basis orthogonalization and transformation (BOT) algorithm described in Section 2.3.

The computation of the \mathbf{S} and \mathbf{T} matrices is a straightforward invocation of the functions implemented in Libra's Gaussian wavepacket (GWP) library, `libgwp`, for the GWP parameter values at time t (see Appendix A).

As for the matrix elements of the potential energy operator, the resulting integrals cannot be evaluated analytically (except for special cases of \hat{V}), or the potential energy is not available in an analytic form altogether – only numeric values and local derivatives are known, for example, from electronic structure calculations. In these instances, the potential can be approximated via local Taylor expansion at the GBF centers. Denoting the gradient vector and the Hessian matrix of $V(\mathbf{Q})$ as \mathbf{V}' and \mathbf{V}'' , respectively, the approximated potential at the location of the k -th GBF is

$$V(\mathbf{Q}; \mathbf{Q}^{(k)}) \approx V[k] + (\mathbf{Q} - \mathbf{Q}^{(k)})^T \mathbf{V}'[k] + \frac{1}{2}(\mathbf{Q} - \mathbf{Q}^{(k)})^T \mathbf{V}''[k](\mathbf{Q} - \mathbf{Q}^{(k)}), \quad (8)$$

where the GBF center at which the functions are evaluated is labeled as $[k]$, $\mathbf{V}'[k] = \mathbf{V}'(\mathbf{Q}^{(k)})$, etc. Within the QTAG dynamics, the potential matrix elements \mathbf{V}_{kl} are evaluated employing the overlap matrix \mathbf{S}_{kl} from the `libgwp` library and the following integrals for each pair of GBFs, $\langle \mathbf{G}_k |$ and $|\mathbf{G}_l \rangle$. For *each dimension* q_n the required integrals of the k -th and l -th GBFs, $\langle g(q_n; \lambda_k) |$ and $|g(q_n; \lambda_l) \rangle$, are given in terms of the GBF overlap and an auxiliary variable $\zeta_n^{(kl)}$ (n labels dimension; k and l label bra and ket basis functions):

$$\zeta_n^{(kl)} = \frac{a_n^{(k)} Q_n^{(k)} + a_n^{(l)} Q_n^{(l)}}{a_n^{(k)} + a_n^{(l)}} + i \frac{P_n^{(l)} - P_n^{(k)}}{a_n^{(k)} + a_n^{(l)}} \quad (9)$$

Then, an N_d -dimensional vector $\tilde{\mathbf{Q}}^{(kl)}$ is constructed from the first moments,

$$\tilde{Q}_n^{(kl)} = \frac{\langle G_k | q_n | G_l \rangle}{\langle G_k | G_l \rangle} = \zeta_n^{(kl)} \quad (10)$$

and an $N_d \times N_d$ matrix $\widetilde{\mathbf{Q}}^2^{(kl)} = \langle G_k | q_n q'_n | G_l \rangle / \langle G_k | G_l \rangle$ is constructed from the second moments,

$$\widetilde{\mathbf{Q}}^2_{nn}^{(kl)} = (\zeta_n^{(kl)})^2 + \frac{1}{a_n^{(k)} + a_n^{(l)}}, \quad (11a)$$

$$\widetilde{\mathbf{Q}}^2_{nn'}^{(kl)} = \zeta_n^{(kl)} \zeta_{n'}^{(kl)}, \quad n \neq n'. \quad (11b)$$

With that, the potential matrix elements are computed using the average of the Taylor expansions (Eq. 8) at the GBF centers $\mathbf{Q}^{(k)}$ and $\mathbf{Q}^{(l)}$. The first-order expansion yields the bra-ket-average (BAT) approximation,⁴¹

$$V_{kl}^{BAT} = \widetilde{V}_{kl}^{BAT} S_{kl}, \quad (12a)$$

$$\widetilde{V}_{kl}^{BAT} = \frac{1}{2} \left(V[k] + V[l] + (\widetilde{\mathbf{Q}}^{(kl)} - \mathbf{Q}^{(k)})^T \mathbf{V}'[k] + (\widetilde{\mathbf{Q}}^{(kl)} - \mathbf{Q}^{(l)})^T \mathbf{V}'[l] \right) \quad (12b)$$

The half-sum of the second-order expansions gives the local harmonic approximation (LHA),

$$V_{kl}^{LHA} = \widetilde{V}_{kl}^{LHA} S_{kl}, \quad (13a)$$

$$\begin{aligned} \widetilde{V}_{kl}^{LHA} = \frac{1}{4} \left(\text{Tr}\{(\mathbf{V}''[k] + \mathbf{V}''[l]) \widetilde{\mathbf{Q}}^2^{(kl)}\} - 2(\widetilde{\mathbf{Q}}^{kl})^T \left(\mathbf{V}''[k] \mathbf{Q}^{(k)} + \mathbf{V}''[l] \mathbf{Q}^{(l)} \right) \right. \\ \left. + (\mathbf{Q}^{(k)})^T \mathbf{V}''[k] \mathbf{Q}^{(k)} + (\mathbf{Q}^{(l)})^T \mathbf{V}''[l] \mathbf{Q}^{(l)} \right) + \widetilde{V}_{kl}^{BAT}. \end{aligned} \quad (13b)$$

The LHA approximation is exact for parabolic potentials, but also obviously more expensive than BAT in general because of the Hessian evaluation. To estimate the BAT error, we may expand V at the midpoint of the GBF centers, considering only one dimension for

simplicity. Using the following notations,

$$\bar{Q} = \frac{Q^{(k)} + Q^{(l)}}{2}, \quad \tilde{a} = a^{(k)} + a^{(l)} \quad (14a)$$

$$\Delta_Q = Q^{(l)} - Q^{(k)}, \quad \Delta_P = P^{(l)} - P^{(k)} \quad (14b)$$

and taking the potential as:

$$V = V_0 + V'|_{q=\bar{Q}}(q - \bar{Q}) + V''|_{q=\bar{Q}} \frac{(q - \bar{Q})^2}{2}, \quad (15)$$

the integral approximation is equal to:

$$V_{kl}^{BAT} - V_{kl} = \frac{V''|_{q=\bar{Q}}}{2\tilde{a}^2} \left(\Delta_P^2 - \tilde{a} - \imath(a^{(l)} - a^{(k)})\Delta_P\Delta_Q - \frac{(a^{(k)})^2 + (a^{(l)})^2}{2}\Delta_Q^2 \right) S_{kl}. \quad (16)$$

The error in the BAT and LHA approximations depends on the separation between GBFs as $\sim \Delta_Q^2 \exp(-a\Delta_Q^2)$. In other words, the long-time dynamics that leads to notable wavefunction dispersion may incur additional inaccuracies due to the potential matrix element approximations.

2.3 Propagation of the Wavefunctions

To integrate the TDSE, Eq. 1 is first projected onto the basis of the GBFs (Eq. 3b) to yield:

$$\imath \mathbf{S} \frac{d}{dt} \mathbf{C} = \mathbf{H}_{vib} \mathbf{C}, \quad (17)$$

$$\mathbf{H}_{vib} = \left\langle \mathbf{G} \left| \left(\hat{H} - \imath \sum_{jj'} |j\rangle \frac{\partial}{\partial t} \langle j'| \right) \right| \mathbf{G} \right\rangle = \mathbf{T} + \mathbf{V} - \imath \left\langle \mathbf{G} \left| \left(\sum_{jj'} |j\rangle \frac{\partial}{\partial t} \langle j'| \right) \right| \mathbf{G} \right\rangle. \quad (18)$$

Here, \mathbf{H}_{vib} is the vibronic Hamiltonian matrix, which includes the effects of nonadiabatic couplings connecting the different electronic states labeled $|j\rangle$ (and $\langle j'|$). In the present approach, however, it is assumed that the solution is obtained in the basis of locally static

(on the interval $[t, t + \Delta t]$) GBFs. In other words, the basis of the GBFs is considered to be the crude diabatic basis at every time interval $[t, t + \Delta t]$ that does not depend on time explicitly. Hence, the time-derivative coupling term, $-\imath \sum_{jj'} |j\rangle \frac{\partial}{\partial t} \langle j'|$, can be dropped, leading to:

$$\mathbf{H}_{vib} = \langle \mathbf{G} | \hat{H} | \mathbf{G} \rangle = \mathbf{T} + \mathbf{V} \quad (19)$$

Note that the crude diabatic basis Hamiltonian^{35,42} \mathbf{H}_{vib} may still contain diabatic couplings connecting distinct electronic states.

To solve Eq. 17 using the Hamiltonian in Eq. 19, we use the following propagation scheme for the non-orthogonal basis:

$$\mathbf{C}(t + \Delta t) = \exp(-\imath \mathbf{S}^{-1} \mathbf{H}_{vib} \Delta t) \mathbf{C}(t) = \mathbf{Z} \exp(-\imath \mathbf{E} \Delta t) \mathbf{Z}^\dagger \mathbf{S} \mathbf{C}(t). \quad (20)$$

Here, \mathbf{S} is the overlap matrix defined in Eq. 5, while \mathbf{Z} and \mathbf{E} are matrices that contain the eigenvectors and eigenvalues of the following generalized eigenvalue problem, respectively:

$$\mathbf{H}_{vib} \mathbf{Z} = \mathbf{S} \mathbf{Z} \mathbf{E}. \quad (21)$$

The second identity in Eq. 20 can be proven by applying the Taylor expansion to the second form of the propagator:

$$\mathbf{Z} \exp(-\imath \mathbf{E} \Delta t) \mathbf{Z}^\dagger \mathbf{S} = \mathbf{Z} \mathbf{Z}^\dagger \mathbf{S} - \imath \mathbf{Z} \mathbf{E} \mathbf{Z}^\dagger \mathbf{S} \Delta t - \mathbf{Z} \mathbf{E} \mathbf{E} \mathbf{Z}^\dagger \mathbf{S} \frac{\Delta t^2}{2} + \dots \quad (22)$$

One can insert the orthogonality property $\mathbf{I} = \mathbf{Z}^\dagger \mathbf{S} \mathbf{Z}$, which also suggests that $\mathbf{S}^{-1} = \mathbf{Z} \mathbf{Z}^\dagger$,

and use Eq. 21 in the form $\mathbf{Z}\mathbf{E} = \mathbf{S}^{-1}\mathbf{H}\mathbf{Z}$ to obtain the following chain of derivations:

$$\begin{aligned}
\mathbf{Z}\exp(-i\mathbf{E}\Delta t)\mathbf{Z}^\dagger\mathbf{S} &= \mathbf{Z}\mathbf{Z}^\dagger\mathbf{S} - i\mathbf{Z}\mathbf{E}\mathbf{Z}^\dagger\mathbf{S}\Delta t - \mathbf{Z}\mathbf{E}(\mathbf{Z}^\dagger\mathbf{S}\mathbf{Z})\mathbf{E}\mathbf{Z}^\dagger\mathbf{S}\frac{\Delta t^2}{2} + \dots \\
&= \mathbf{S}^{-1}\mathbf{S} - i\mathbf{S}^{-1}\mathbf{H}\mathbf{Z}\mathbf{Z}^\dagger\mathbf{S}\Delta t - (\mathbf{S}^{-1}\mathbf{H}\mathbf{Z})\mathbf{Z}^\dagger\mathbf{S}(\mathbf{S}^{-1}\mathbf{H}\mathbf{Z})\mathbf{Z}^\dagger\mathbf{S}\frac{\Delta t^2}{2} + \dots \\
&= \mathbf{I} - i\mathbf{S}^{-1}\mathbf{H}\mathbf{S}^{-1}\mathbf{S}\Delta t - \mathbf{S}^{-1}\mathbf{H}\mathbf{S}^{-1}\mathbf{S}\mathbf{S}^{-1}\mathbf{H}\mathbf{S}^{-1}\mathbf{S}\frac{\Delta t^2}{2} + \dots \\
&= \mathbf{I} - i\mathbf{S}^{-1}\mathbf{H}\Delta t - \mathbf{S}^{-1}\mathbf{H}\mathbf{S}^{-1}\mathbf{H}\frac{\Delta t^2}{2} + \dots
\end{aligned} \tag{23}$$

The resulting expression is equivalent to the Taylor expansion of the propagator in another form:

$$\exp(-i\mathbf{S}^{-1}\mathbf{H}\Delta t) = \mathbf{I} - i\mathbf{S}^{-1}\mathbf{H}\Delta t - \mathbf{S}^{-1}\mathbf{H}\mathbf{S}^{-1}\mathbf{H}\frac{\Delta t^2}{2} + \dots \tag{24}$$

The quantum amplitudes also change when the GBFs are moved in tandem with the trajectories. Thus, we incorporate the re-projection technique, taking advantage of the fact that the wavefunction represented in the bases associated with times t and $t + \Delta t$ is the same.³⁸ The expansion coefficients $\mathbf{C}'(t + \Delta t)$ of the evolved wavefunction in the evolved basis are found by minimizing the error between the two representations:

$$|\Psi(t + \Delta t)\rangle = |\mathbf{G}(t)\rangle\mathbf{C}(t + \Delta t) \approx |\mathbf{G}(t + \Delta t)\rangle\mathbf{C}'(t + \Delta t), \tag{25}$$

$$\begin{aligned}
\mathbf{C}'(t + \Delta t) &= [\langle\mathbf{G}(t + \Delta t)|\mathbf{G}(t + \Delta t)\rangle]^{-1}\langle\mathbf{G}(t + \Delta t)|\mathbf{G}(t)\rangle\mathbf{C}(t) \\
&= \mathbf{S}^{-1}(t + \Delta t)\mathbf{S}(t + \Delta t, t)\mathbf{C}(t),
\end{aligned} \tag{26}$$

where $\mathbf{S}(t + \Delta t)$ and $\mathbf{S}(t + \Delta t, t)$ are the new/new and new/old GBF overlap matrices, respectively. In principle, the time-increment Δt in Eqs 25 and 26 does not need to be small, and the basis parameters, including the number of the GBFs, can be changed in any manner, as long as the new basis provides a sufficiently complete representation of $|\Psi(t + \Delta t)\rangle$. The re-projection technique allows us to update the expansion coefficients analytically over finite time-increments for which the basis functions are taken as time-independent, thereby bypassing explicit computation of the GBF time-derivatives.

2.4 Propagation of the GBFs

In the QTAG method, the GBFs are adaptive, meaning that the time-evolution of their parameters is related to (but not the same as) that of Bohmian dynamics.⁴³ A basis function can then be described as “trajectory-guided” by associating its position and momentum to that of an underlying trajectory, where the equations of motion (EOMs) of the latter determine the parameters of the former. The defining characteristic of the QTAG method, then, lies within how these basis-guiding trajectories are computed.

The definitions of the Bohmian equations of motion can be recognized by inserting a polar wavefunction $\Psi(\mathbf{q}, t) = |\Psi(\mathbf{q}, t)|\exp(iS(\mathbf{q}, t))$ with real amplitude ($|\Psi|$) and phase (S) into the TDSE (Eq. 1) and separating real and imaginary components. The former yields the well-known quantum Hamilton-Jacobi equation,

$$\frac{\partial S}{\partial t} = -\frac{1}{2}(\nabla S)^T \mathbf{M}^{-1} \nabla S - (V + U), \quad U = -\sum_{n=0}^{N_d-1} \frac{\nabla_n^2 |\Psi|}{2M_n |\Psi|}, \quad (27)$$

while the latter gives the continuity equation for probability density ρ ,

$$\frac{\partial \rho}{\partial t} = -(\nabla \rho)^T \mathbf{M}^{-1} (\nabla S) - \sum_{n=0}^{N_d-1} \frac{\nabla_n^2 S \cdot \nabla_n \rho}{M_n}, \quad \rho = |\Psi|^2 \quad (28)$$

Here \mathbf{q} is the position vector in N_d -dimensional space, and \mathbf{M} is the diagonal matrix with the particle mass M_n for each dimension $n = [0, N_d-1]$. The potential-like term U —termed the quantum potential—incorporates the quantum effects into the otherwise classical-looking time-evolution equations, once the trajectory momentum is identified with the gradient of the phase:

$$\mathbf{P} = \nabla S(\mathbf{q}, \mathbf{t}). \quad (29)$$

Strictly speaking, the above formalism is valid only for a single-surface wavefunction; in the nonadiabatic case, a more complicated form of the wavefunctions should be used.⁴⁴ In this work, we extend the QTAG approach initially formulated for adiabatic dynamics³⁷ to

the nonadiabatic domain by assuming that the GBF centers on each surface move according to the momentum defined as in Eq. 29. The nonadiabatic effects are fully accounted for in the evolution of the GBF coefficients, as explained in Section 2.3, but ignored in the EOMs for the GBF parameters.

The functions $V(\mathbf{q}, t)$ and $U(\mathbf{q}, t)$ seen in Eq. 27 are the classical and quantum potentials found in the TDSE, respectively. With the definition of Eq. 29, transformation of the gradient of Eq. 27 into the Lagrangian frame of reference yields an expression for its time-dependence, and hence the equations of motion for a quantum trajectory ($\dot{\mathbf{P}} \equiv d\mathbf{P}/dt$ and so on):

$$\dot{\mathbf{Q}} = \mathbf{M}^{-1}\mathbf{P} \quad (30a)$$

$$\dot{\mathbf{P}} = -\nabla_{\mathbf{q}}(V + U)\Big|_{\mathbf{q}=\mathbf{Q}} \quad (30b)$$

As demonstrated in Ref.,³⁸ the EOMs of the GBF width parameters, $\{a_n\}$, and phase s (Eq. 7) aggregated over the dimensions take the form:

$$\dot{a}_n = -\frac{2a_n}{M_n}\nabla_{q_n}P_n\Big|_{\mathbf{q}=\mathbf{Q}}, \quad (31a)$$

$$\dot{s} = \sum_n s_n = \sum_n \frac{P_n^2}{2M_n} - (V + U)\Big|_{\mathbf{q}=\mathbf{Q}} \quad (31b)$$

The GBF parameters $\{s_n\}$ in Eq. 7 (or their N_d -dimensional counterpart in Eq. 31b) can be omitted entirely, as their role in the basis definition (i.e. via the $\exp(\imath s)$ term) is that of an overall phase factor. Such a factor is already accounted for in the expansion coefficients $\{C_i\}$; thus, the Libra implementation of QTAG utilizes this omission for efficiency. Note, however, that inclusion of s within the GBF reduces oscillations of the expansion coefficients, which may help to stabilize their time-evolution. We do not use Eq. 30b to evolve the GBF positions, since evaluation of its right-hand side involves third order derivatives of $|\Psi|$.

Instead \mathbf{P} is computed from the wavefunction according to

$$\mathbf{P} = \text{Im}\left(\frac{\nabla\Psi(\mathbf{q}, t)}{\Psi(\mathbf{q}, t)}\right)\Big|_{\mathbf{q}=\mathbf{Q}}, \quad (32)$$

which allows us to bypass integration of stiff EOMs arising from the quantum potential. There are still two drawbacks associated with Eq. 32, however. First, the momentum is ill-defined in regions of negligible wavefunction amplitude or at the wavefunction nodes, leading to potentially unstable dynamics. Second, the wavefunction must be well-defined (i.e. have appreciable amplitude) for a given quantum state. Thus, we introduce the following modifications to the definition of \mathbf{P} (Eq. 32).

First, we smooth out the momenta defined by Eq. 32. This reduces micro-features in the trajectory dynamics without introducing formal errors into the wavefunction propagation, since within an adequately full basis the guiding trajectories do not need to follow Bohmian equations of motion exactly; they just need to cover the same configuration space. In principle, this can be achieved in a variety of ways. The approach in Libra is to use a first-order polynomial to fit $\mathbf{P}(\mathbf{Q})$. Such a fitting introduces an efficient means of calculating the gradient of the momentum $\nabla^T\mathbf{P}$, generally a matrix, used in Eq. 31a, although this approximation comes at the expense of having the GBF width parameters $\{a^{(i)}\}$ change in the same manner. Some alternatives to this approach (not included in the Libra implementation) are to use a Gaussian convolution of either \mathbf{P} or Ψ .³⁷

Second, the following protocol is introduced to handle situations where trajectories evolve on multiple surfaces. The motion of GBFs on surfaces with little to no probability density is a concern, since the momentum of Eq. 32 is ill-defined in these cases.²⁷ We take the most straightforward approach of synchronizing the GBFs on surfaces with no appreciable population density to the ones on the most populated surface, thus allowing the latter to dictate the position of the former until probability density is built up on the initially unpopulated states. This approach requires that both states' trajectories are governed by

the wavefunction of the initially occupied state, which is reasonable for systems in which the relevant surfaces are bound over similar spatial domains, or over short time scales before the wavefunction decoheres. Neither modification leads to additional approximations of the wavefunction, but they do make the GBF basis non-ideal unless the evolving-in-time wavefunction itself is a Gaussian wavepacket.

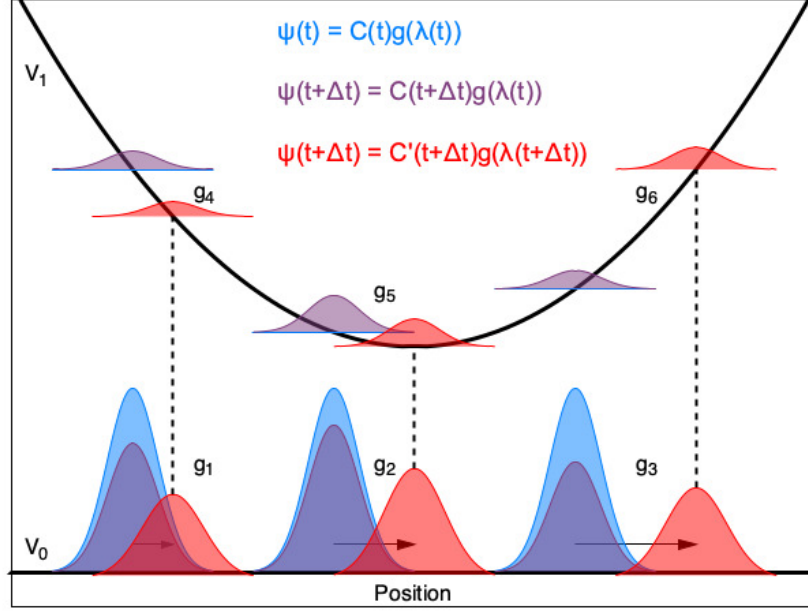
2.5 Overall Computational Workflow

In summary, the QTAG wavefunction evolution proceeds according to the BOT algorithm, which is composed of three parts (Fig. 2). The first part is responsible for evolving the coefficients $\{C_i\}$ given a locally time-independent set of GBFs, while the second part evolves the coefficients $\{C_i\}$ in response to the time-evolution of the GBFs along the trajectories via re-expansion of the wavefunction. These two parts are detailed in Section 2.3. The third component of the algorithm is to evolve the parameters of the GBFs, which can be interpreted classically according to the Bohmian EOMs with the momentum computed from the single-surface wavefunctions, as explained in Section 2.4.

3 Software Description

3.1 Software Architecture and Functionality

The package consists of both Python and C++ layers. The former is represented by the `libra.py.dynamics.qtag` module and implements the high-level functions of immediate importance to the user. The latter is represented by the `dyn/qtag` library (`libqtag` when compiled) as well as several other C++ sub-libraries in the core of the Libra software, notably `libgwp` (Fig. 3). The C++ level functions and classes are leveraged to construct the algorithms at the higher level, including in the `libra.py.dynamics.qtag` module. These functions are implemented in C++, although most of them are exposed to Python via the `Boost.Python` library.⁴⁵ They may be of higher interest to the methodology developers.



-
1. Propagate $C(t + \Delta t)$ in orthonormal basis stored in \mathbf{Z} .
 2. Update $P(t + \Delta t)$ according to Ψ defined by $C(t + \Delta t)$ and $\lambda(t)$.
 3. Update $Q(t + \Delta t)$ and $a(t + \Delta t)$ via $P(t + \Delta t)$ and $\nabla P(t + \Delta t)$.
 4. Reexpand $C(t + \Delta t) \rightarrow C'(t + \Delta t)$ with bases defined by $\lambda(t + \Delta t)$.
-

$$\mathbf{S}(t + \Delta t)\mathbf{C}'(t + \Delta t) = \mathbf{S}(t + \Delta t, t)\mathbf{C}(t + \Delta t)$$

Figure 2: The BOT algorithm for a generic nonadiabatic system involving two surfaces, V_0 and V_1 , for an initially unpopulated V_1 state. A three-function basis (blue) is initialized on the ground surface and propagated for one iteration (i.e. $t \rightarrow t + \Delta t$). Propagation of the coefficients (step 1) changes the overall GBF contributions, shown as the corresponding purple functions which represent the new wavefunction $\Psi(t + \Delta t)$ in the old basis defined by $\lambda(t)$. Upon updating the basis parameters (steps 2 and 3), the new functions (red) have moved according to the Bohmian EOMs and have new widths. Note that the amplitudes have also changed, as they must describe the same overall wavefunction as the purple basis, but with ‘new’ functions described by $\lambda(t + \Delta t)$ (4). The dashed vertical lines indicate the basis synchronization described in-text, whereby the excited state functions are moved according to their ground-state counterparts due to low V_1 population.

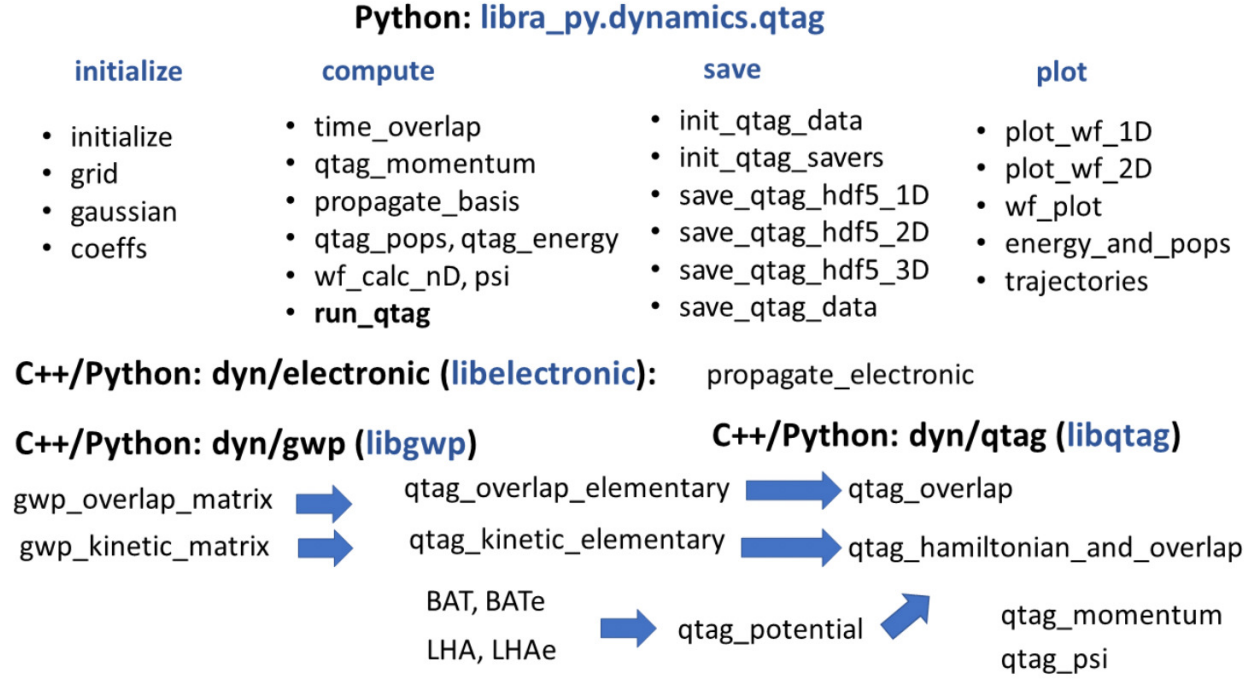


Figure 3: Structure of the QTAG components of Libra in the Python (top) and C++ (bottom) layers. Arrows in the C++ layer indicate dependencies relating certain functions.

At the C++ layer (Fig. 3, bottom), multiple time-consuming functions are implemented. The functions “BAT” and “LHA” implement the calculation of the $\tilde{\mathbf{V}}_{IJ}^{BAT}$ and $\tilde{\mathbf{V}}_{IJ}^{LHA}$ potentials of Eqs 12 and 13, respectively. These elementary functions compute the pair potentials between two arbitrary GBFs, and are called via the `qtag_potential` module for all pairs of GBFs to compute the corresponding elements of the appropriate $N \times N$ matrix $\tilde{\mathbf{V}}$. The `qtag_overlap` function is used to construct the $N \times N$ super-overlap matrix \mathbf{S}_{IJ} (Eq. 5), which is assembled from the smaller $N_b \times N_b$ overlaps $\mathbf{S}^{(i,j)}$ for all $i, j \in 0, \dots, M-1$, as illustrated in Fig. 1. The smaller overlap matrices are computed by the `qtag_overlap_elementary` function, which is essentially a wrapper of the `libgwp gwp_overlap_matrix` function customized for the QTAG definitions of the GBFs – Eqs 6 and 7 – which is itself the main driver for computing the overlaps of multidimensional Gaussians (not accounting for the electronic state they belong to). The direct (element-wise) product of the super-overlap \mathbf{S} and potential $\tilde{\mathbf{V}}$ matrices yields the potential energy matrix

of the Hamiltonian:

$$\mathbf{V} = \tilde{\mathbf{V}} \otimes S \leftrightarrow V_{IJ} = \tilde{V}_{IJ} S_{IJ}. \quad (33)$$

These calculations are conducted in the `qtag_hamiltonian_and_overlap` function as a part of the super-Hamiltonian construction. The remaining part – the kinetic energy matrix in the basis of GBFs – is computed using the `qtag_kinetic_elementary` function, which is isomorphic to the `qtag_overlap_elementary` function and, like the latter, an adaptation of the `libgwp` function `gwp_kinetic_matrix` to the QTAG definition of the GBFs. Unlike the output of `qtag_overlap`, however, a super-matrix of the kinetic energy elements alone is unnecessary: only the super-Hamiltonian and super-overlap are needed for propagation. The super-Hamiltonian matrix is constructed by the `qtag_hamiltonian_and_overlap` function directly.

Another C++ function – `propagate_electronic` from the `libelectronic` library – is used to implement the non-orthogonal basis propagator, Eq. 20. This function is called within the `run_qtag` Python function, which is the main driver of the present QTAG algorithm implementation. Finally, the `qtag_momentum` function on the C++ side implements the quantum momentum calculations according to Eq. 32, and the `qtag_psi` function implements the evaluation of the multidimensional GBFs at points according to Eqs 6 and 7.

The Python layer introduces a number of workflow-forming and convenience functions, all organized into four self-explanatory modules: `initialize`, `compute`, `save`, and `plot`. The functions `initialize.grid` and `initialize.gaussian` create the GBFs by placing them either on a regular grid (`grid`) or by sampling them from a normal distribution (`gaussian`). In both cases, the user provides the number of GBFs to be placed. For the grid placement, the boundaries of the box where the functions are added are defined either manually or by specifying a wavefunction amplitude threshold; in the latter case, the GBFs will be placed in those regions where the initial Gaussian wavepacket has amplitude larger than the specified value. Placing the GBFs near the initial wavepacket leads to a very economic representa-

tion of the basis, considering that the GBFs also evolve in the simulations. The approach based on defining manual boundaries (which usually requires more GBFs) is reminiscent of wavepacket dynamics where a sufficiently large, fixed basis is used during the entire simulation. This approach may also lead to better energy conservation in some cases, e.g. when the QTAG wavefunction and trajectories do not stray significantly outside of the manually-defined region. Both types of grids are produced by the `initialize.initialize` function, which returns the matrices of the GBF parameters shown in Fig. 4.

[illegible]

Figure 4: Sample output of the basis initialization procedure computed via the `libra_py.dynamics.qtag.initialize.initialize` function.

The `initialize.coeffs` function computes the initial expansion coefficients – also referred to as ‘amplitudes’ – of a starting Gaussian wavepacket $|\Psi\rangle = |\tilde{G}\rangle$ in terms of the GBFs used in the simulations:

$$|\Psi\rangle = |\mathbf{G}(0)\rangle \mathbf{C}(0) \leftrightarrow \mathbf{C}(0) = \mathbf{S}^{-1}(0) \langle \mathbf{G} | \tilde{G} \rangle. \quad (34)$$

The module `compute` implements the main computational elements of the QTAG workflow. The main entry point to the QTAG calculations is the `compute.run_qtag` function, which is the main wrapper of the entire workflow (excluding the initialization and plotting). This function takes the dynamical variables (coefficients and GBF parameters), the function object for computing the system’s Hamiltonian, and the dynamical and model Hamiltonian parameter dictionaries as inputs. The initial dynamical variables are generated by the

```

Q0 = MATRIX(1,1); Q0.set(0,0, -2.0)
P0 = MATRIX(1,1); P0.set(0,0, 0.0)
A0 = MATRIX(1,1); A0.set(0,0, 1.0)
S0 = MATRIX(1,1); S0.set(0,0, 0.0)
istate = [0]

C = initialize.coeffs(Q0, P0, A0, S0, istate, Q, P, A, S, active_states)

data_outs.print_matrix(C)

(0.0035165773448428642+0j)
(0.0008074729237409066+0j)
(0.019570498821938256+0j)
(0.023370831298085926+0j)
(0.07169063086741813+0j)
(0.09972220147760633+0j)
(0.1715524473692758+0j)
(0.19994712431913392+0j)
(0.23851894505989182+0j)

```

Figure 5: Sample output of the coefficients initialization procedure computed via the `libra_py.dynamics.qtag.initialize.coeffs` function.

`initialize.initialize` and `initialize.coeffs` functions, as shown above (Figs. 4 and 5, respectively). The `compute.run_qtag` function also takes care of calculating various observables and saving the computed results, as implemented in the `save` module. A typical call of this function is shown in Fig. 6.

Internally, the `compute.run_qtag` function cycles through calls of the following functions: `propagate_electronic`, `compute.propagate_basis`, `qtag_hamiltonian_and_overlap`, `compute.qtag_energy`, `compute.qtag_pops`, and `save.save_qtag_data`. In order, these functions propagate the basis coefficients and parameters, calculate the Hamiltonian matrix elements, compute the system energy and surface populations, and save the output data. The `compute.propagate_basis` function implements the basis re-expansion procedure, Eq. 26, together with the evolution of the GBF parameters according to Eqs 30a and 31a, unless any constraints are specified. The function `compute.qtag_momentum` implements Eq. 32 together with the single-surface linear fitting scheme to update the basis momenta and mitigate numerical instabilities. The time-overlaps of the GBFs, $\langle \mathbf{G}(t') | \mathbf{G}(t) \rangle$, are needed for the basis re-expansions and computed by the function `compute.time_overlap`. The function `compute.propagate_basis` also takes care of the GBF synchronization across surfaces, as necessary. As described previously, when there is no wavefunction amplitude on any state, the direct evaluation of the quantum momentum, Eq. 32, may be problematic. In these

```

# 1D mass
iM = MATRIX(1, 1); iM.set(0,0,1.0);

#Parameters for general simulation
dyn_params = {
    "target_states":[0,1], "states":[0,1],
    "dt":0.01, "nsteps":500, "iM":iM,
    "hdf5_output_level":0, "txt2_output_level":3,
    "properties_to_save":["time","Etot","pops","q","p","a","coeffs"],
    "q_update_method":1, "p_update_method":1, "a_update_method":1, "s_update_method":0,
    "q_sync_method":1, "p_sync_method":0, "a_sync_method":0, "s_sync_method":0,
    "mom_calc_type":1, "linfit_beta":1e-3, "decpl_den":0.1,
    "qtag_pot_approx_method":1,
    "prefix":"out"
}

compute.run_qtag(Q, P, A, S, active_states, C, iM, dyn_params, compute_model, model_params)

21.749998548252773 [0.9999999849956125, 3.5917587897516737e-13]
21.74999854040308 [0.9999999849015216, 3.1990148045349754e-12]
21.749998511548725 [0.999999984371664, 8.694703285637612e-12]
21.749998502474185 [0.9999999839138096, 1.650388062602081e-11]
21.749998508284843 [0.99999998344329, 2.616568577000186e-11]
21.749998526017475 [0.999999983077347, 3.713139576979547e-11]
21.74999854966568 [0.9999999827429888, 4.8785917426209756e-11]
21.749998577389764 [0.9999999824895912, 6.047088739881983e-11]
21.749998605301485 [0.9999999821382933, 7.151780132011366e-11]
21.749998633216293 [0.9999999817231867, 8.129363852624073e-11]
21.749998663316678 [0.9999999813219979, 8.925494864698767e-11]

```

Figure 6: Sample output of the main code computed via the `libra.py.dynamics.qtag.compute.run_qtag` function. The first column shows total energy, while the second and third columns (which are elements of a Python list) display surface populations.

situations, certain properties of the GBFs evolving on the surfaces with little or no probability density may be set to be equal to those GBF parameters for the basis functions on the most-populated surface (synchronized). The user may control which properties – coordinates, momenta, width parameters, or any combinations of these – should be synchronized. The synchronization is also controlled based on the individual population(s) of the wavefunction-deficient state(s) – for populations larger than a user-controlled threshold value, the GBFs on these state(s) would evolve independently, and be synchronized to the GBFs on the most populated surface only when their population is lower than the specified threshold.

The role of the `propagate_electronic` and `qtag_hamiltonian_and_overlap` functions has been explained above. The function `compute.qtag_energy` computes the total energy of the system as

$$E = \langle \Psi | \hat{T} + \hat{V} | \Psi \rangle = \mathbf{C}^\dagger \mathbf{H} \mathbf{C}. \quad (35)$$

The function `compute.qtag_pops` computes the populations on all states:

$$p_i = (\mathbf{C}^{(i)})^\dagger \mathbf{S}^{(i,i)} \mathbf{C}^{(i)} = \sum_{k,k'=0}^{N_b-1} C_{(i,k)}^* \langle i, G_{(i,k)} | i, G_{(i,k')} \rangle C_{(i,k')}. \quad (36)$$

The modules `save` and `plot` implement the functionality for storing dynamical data in specialized formats and retrieving them for plotting, respectively. The functions of the `save` module replicate the structure of the `save` modules of other Libra workflows, such as HEOM, TSH, and exact quantum simulations (DVR). Most likely, the user is not exposed to them directly; instead they are called in the main workflow function `compute.run_qtag`. The `save.init_qtag_savers` function creates the objects of the saver classes defined in the `libra_py.data_savers` module, which can work in the 'mem', 'hdf5', and 'txt' formats. The first two have been briefly described in another account.⁴⁶ The third format, 'txt', is designed to save data in a human-readable ACSII (text) format. In our experience, this approach has been quite convenient compared to the other two, since it has fast write operations (unlike the HDF5 saver) and is not as memory-intensive as the 'mem' mode. The function also takes care of preparing the filesystem for the output of the files. The `save.init_qtag_data` function allocates memory (as multidimensional numpy arrays) for variables of interest. For instance, to store information about the nuclear coordinates ('q') of N_{traj} trajectories with N_{dof} degrees of freedom for N_{steps} timesteps, one allocates a rank-3 real ('R') array of size $N_{steps} \times N_{traj} \times N_{dof}$ via the command `saver.add_dataset('q', (nsteps, ntraj, ndof), 'R')`.

Once such arrays are allocated, they are used to collect the desired data during the simulation. The `save.init_qtag_data` function is needed for the 'mem' and 'hdf5' modes, but not for the 'txt' mode. The user can control what variables are to be saved by either providing the list of variables (`properties_to_save`) to the constructors of the savers class (e.g. to `save.init_qtag_savers`) or by providing it as an input to the `save.init_qtag_data` function. In addition, the user can limit the dimensionality of stored/output data by using

the `*_output_level` keywords, where the asterisk stands for 'txt2', 'hdf5', or 'mem'. For the output level of 1, only one-dimensional data would be stored and saved, such as the timestep index ('time') or the total energy ('Etot') of the system, (unless they are excluded from the `properties_to_save` list); for the output level of 2, one can store populations ('pops') and quantum coefficients ('coeffs'); for the output level of 3, one can store the GBF parameters ('q', 'p', 's', 'a'). Note that this is the level of output required to reconstruct a wavefunction snapshot, if desired.

Finally, the process of saving is conducted via the `save.save_qtag_data` function. Saving data with the `txt2` saver occurs with a given frequency (controlled by the `progress_frequency` keyword), whereas for the 'hdf5' format the common practice is to write the data only at the end of simulations. This function is called within the `compute.run_qtag` function. For the 'hdf5' format savers, the function calls any of the `save.save_qtag_hdf5_nD` ($n = 1, 2$, or 3) functions, depending on the dimensionality of the output data.

The plotting of the results is conducted by the functions of the 'plot' module. With the help of `plot.wf_plt`, one can plot the snapshots of the wavefunctions, provided the necessary parameters were designated to be saved via the `properties_to_save` list. The files necessary for such visualization are computed using the `compute.wf_calc_nD` function (Fig. 7a). Analogously, with the help of the `plot.energy_and_pops` function, one can plot the total energy and state populations as a function of time (Fig. 7b). Finally, the `plot.trajectories` function plots the evolution of the trajectories, which can be regarded as the visualization of the time-dependent grid of GBF centers (Fig. 7c).

4 Application to Model Problems

To illustrate our dynamics method and to understand the effect of the approximations described in Section 2, we apply the nonadiabatic QTAG to one-dimensional benchmark systems, i.e. a coupled harmonic oscillator model ('Holstein') and to single ('Tully 1') and

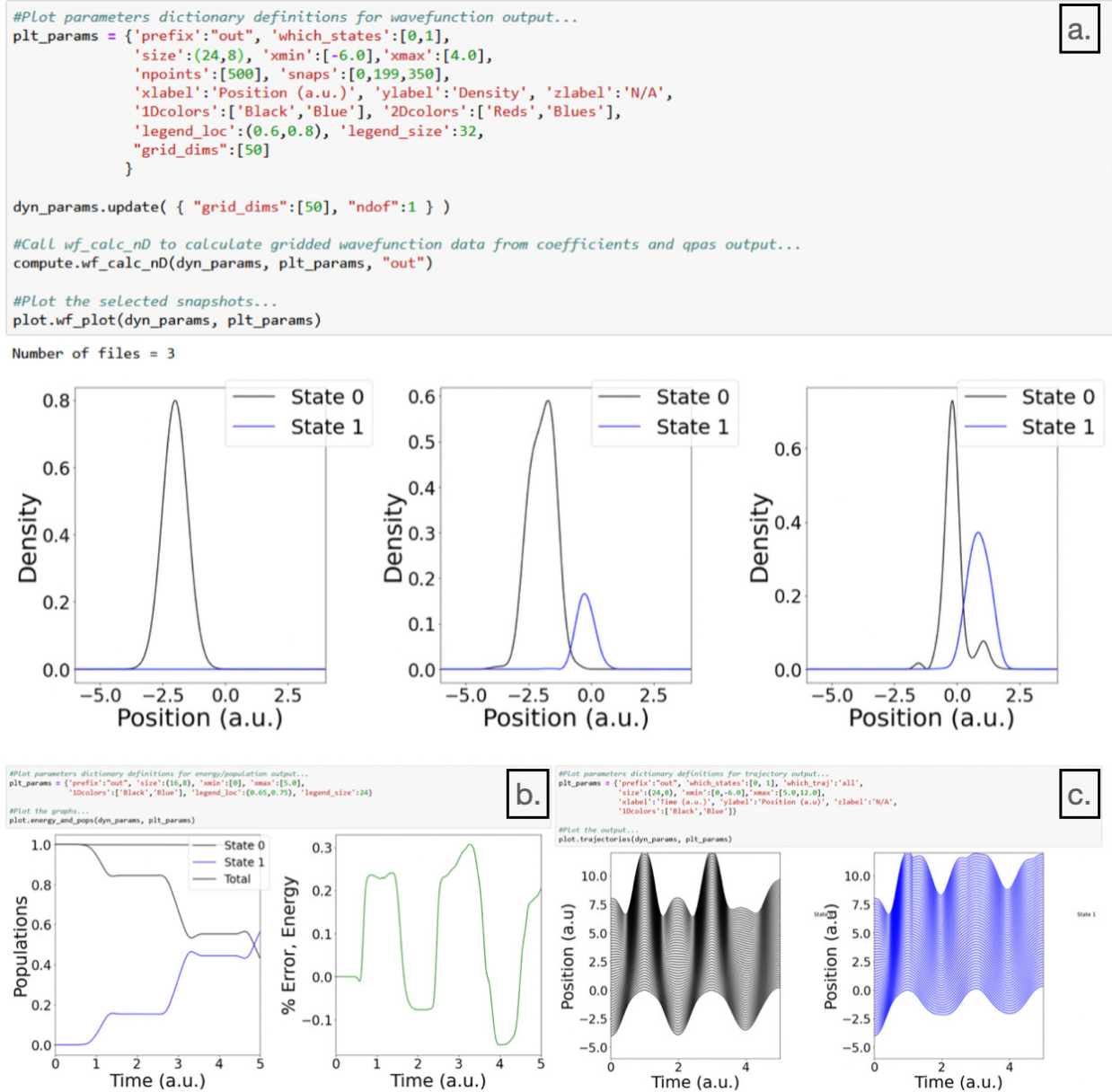


Figure 7: Sample outputs of the various plotting functions computed via the `libra.py.dynamics.qtag.plot` module. (a.) Wavefunction snapshots. (b.) Surface populations and energy. (c.) Trajectory positions as functions of time.

dual ('Tully 2') avoided crossing models in the diabatic representation. This choice is made primarily for convenience, although the recently developed quasi-diabatic approximation^{47,48} provides a link to electronic structure calculations in an adiabatic scheme, if desired. Indeed, the QTAG method is insensitive to the chosen representation (diabatic vs. adiabatic) from an algorithmic perspective, and the Libra software library is capable of performing diabatic-to-adiabatic transformations on-the-fly as necessary. For simplicity, the spatial coordinate and particle mass are labeled as x and m , respectively.

In each case, an initial wavepacket ψ_0 is prepared according to Eq. A.1 with parameters specified by the appropriate entry in the upper half of Table 1,

$$\psi_0(x) = \left(\frac{2\alpha_0}{\pi}\right)^{1/4} \exp(-\alpha_0(x - \bar{x}_0)^2 + i\bar{p}_0(x - \bar{x}_0) + i\gamma_0), \quad (37)$$

where \bar{x}_0 and \bar{p}_0 denote the initial average position and momentum of the wavepacket, α_0 is used for the initial width and γ_0 for the overall phase. Basis functions are then placed with equal spacing between points defined by a wavepacket density threshold ρ_c on the populated state and subsequently mirrored on all remaining states, so that each state is represented by N_b GBFs. Projection of the basis onto the wavepacket yields an initial projection vector with components $b_i = \langle g_i(x) | \psi_0(x) \rangle$ for the populated state; basis projections onto the remaining states are set to zero. As mentioned in Section 2, the unpopulated state bases' positions are synchronized to the populated state until sufficient population accumulates, as defined by ρ_d – once a state exceeds this threshold, its trajectories evolve independently of the other states. These basis parameters are summarized in the lower half of Table 1; the only remaining quantity to define is κ , which is a scaling factor for the basis widths, i.e. $a_0 = \alpha\kappa$. For exact reference calculations, we employ the split-operator Fourier transform (SOFT) method^{49,50} as already implemented in the Libra software suite.

Before progressing to model-specific results, it is important to comment on the initial choice of parameters in the `dyn_params` dictionary. The three parameters N_b , ρ_c , and a_{scl}

Table 1: Initial wavepacket and basis parameters for the model systems. Atomic units are implied throughout, where appropriate. S(D)AC: = Single (Dual) Avoided Crossing models.

ψ_0	\bar{x}_0	\bar{p}_0	α_0	s_0
Holstein	-2.0	0.0	1.0	0.0
SAC/DAC	-5.0	10.0	2.0	0.0
Basis	N_b	κ	ρ_c	ρ_d
Holstein	25	6.0 (LHA)	10^{-12}	0.3
	50	32.0 (BAT)		
SAC	25	10.0 (LHA)	10^{-12}	0.1
	35	15.0 (BAT)		
DAC	35	25.0 (LHA/BAT)	10^{-12}	0.01

primarily control the quality of the basis representation: in general, increasing the size of the basis improves the description of the wavefunction; however, the QTAG algorithm will fail if the overlap of any two basis functions strays too close to unity due to the matrix inversion step performed when solving Eq. 19. Furthermore, there are competing effects influencing the degree of error during a simulation originating with the basis width parameter. On one hand, the degree of error incurred in approximating the potential surfaces with the LHA/BAT techniques grows with the basis width (i.e. as a_{scl} shrinks); this is due to non-negligible basis overlap at large distances, where the approximation of the potential as a quadratic/linear function breaks down. On the other hand, a wider basis is more robust, as this leads to more functions being able to describe the wavefunction at any given point. The adaptability of the QTAG basis may somewhat mitigate the error incurred through these width effects, although we emphasize that consideration should still be given to both basis size and model potential when establishing initial conditions, and that an 'optimal' set of basis parameters will be system-dependent.

There are also certain scenarios in which expressions for one or more of the potential integrals in a Gaussian basis can be found analytically; in these cases, the error associated with approximating the potential elements via the LHA/BAT methods is removed and more accurate results can be achieved. We take advantage of this for the 1D model systems presented below, evaluating the coupling potential elements exactly (the coupling in each

case is a Gaussian function). In principle, these values could be approximated with the LHA/BAT methods, although we note that the propagation is less stable and results are less accurate at longer times.

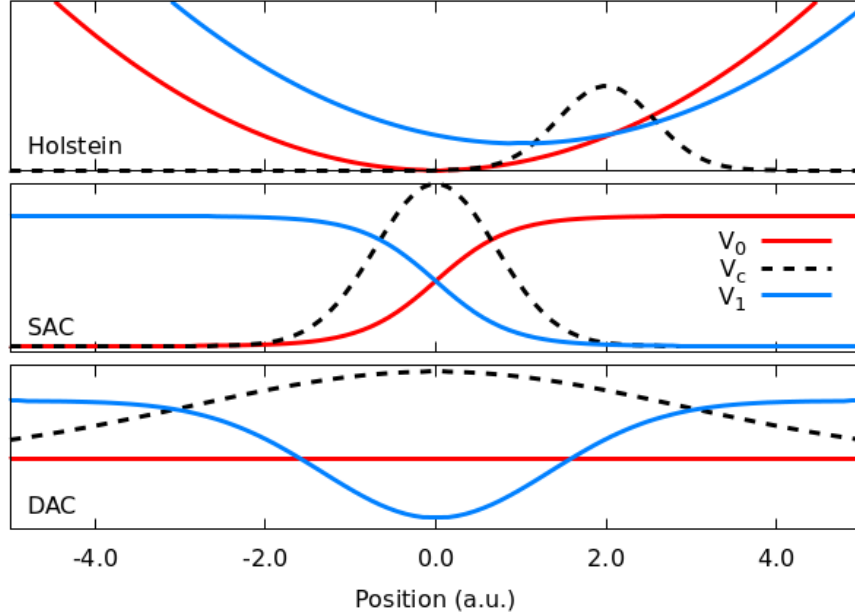


Figure 8: Diabatic potential surfaces (V_0 and V_1) with coupling (V_c) for 1D model systems; SAC = Single Avoided Crossing, DAC = Dual Avoided Crossing. The horizontal axis applies to all panels, while V_c is scaled by a factor of 50 in the Holstein model and by a factor of 5 in the avoided crossing models for visual clarity.

Overall numerical results for each of the three systems have been tabulated in Table 2 for both BAT- and LHA- approximated potentials. These quantities, which can be taken as indicators of simulation quality, are in good agreement with exact split-operator values for the times tested. The maximum and average errors in energy are computed as $\max(|E - E_{init}|)/E_{init}$ and $(E_{avg} - E_{init})/E_{init}$, respectively; further details and graphical results are given in the appropriate section.

Table 2: Indicators of simulation accuracy for all systems in the LHA/BAT-approximated cases with exact coupling elements. The normalization and population quantities are taken at the end of each run (further detail is discussed in the appropriate section in-text).

	Max. % Error, E	Avg. % Error, E	Norm.	S_0/S_1 Pops.
Holstein				
SOFT	0.024	0.011	1.000	0.587/0.413
LHA	0.193	0.005	0.999	0.585/0.414
BAT	1.649	0.051	0.999	0.578/0.421
Single Avoided Crossing				
SOFT	2.09×10^{-4}	4.22×10^{-6}	1.000	0.114/0.886
LHA	0.090	-0.005	1.000	0.121/0.879
BAT	0.428	-0.170	0.999	0.133/0.866
Dual Avoided Crossing				
SOFT	5.64×10^{-4}	5.43×10^{-5}	1.000	0.990/0.0098
LHA	2.856	0.437	0.994	0.967/0.027
BAT	2.111	0.277	0.992	0.966/0.026

4.1 Holstein Model

The first model considered is reminiscent of the phonon component of the Holstein model, consisting of two offset parabolic potentials V_{11} and V_{22} locally coupled by a Gaussian function centered at their crossing (Eqs 38a–38c).

$$V_{11}^{Hol} = \frac{k_1}{2}x^2 \quad (38a)$$

$$V_{22}^{Hol} = \frac{k_2}{2}(x - x_0)^2 + y_0 \quad (38b)$$

$$V_{12}^{Hol} = V_{21} = c_1 e^{-c_2(x-c_3)^2} \quad (38c)$$

The spring constants k_1 and k_2 are taken to be $10 E_h/a_0$ each, while the displacement coordinates of the first excited surface are $(x_0, y_0) = (1.0, 5\sqrt{k_1}/m)$. The parameters $c_1 = 1.0$, $c_2 = \frac{1}{2}\sqrt{k_1/m}$, and $c_3 = 2.0$ define the coupling potential V_{12} , and a mass of $m = 1$ a.u. is used. A useful feature of this model, as seen in the top panel of Fig. 8, is the nature of its quadratic surfaces – since the LHA for the potential integrals (Eq. 13) is exact in these instances, the only approximations entering the algorithm originate in the basis propagation step through the momentum fitting and GBF synchronization procedures, thus allowing the

impact of the parameters relating to this step to be explored.

A wavepacket is prepared according to Eq. 37 on the ground state surface with initial parameters specified in Table 1 and allowed to propagate for $t = 800$ a.u., which is long enough for multiple oscillations into the coupling region (see Fig. 9, for example). Satisfactory results are obtained when the ground and excited state trajectories are decoupled from one another after ~ 3 a.u., as evidenced by the wavefunction snapshots ($|\psi|^2$) displayed in Fig. 10. Discrepancies between QTAG and SOFT results are seen primarily in the final panel of the BAT-approximated simulations. The overall energy conservation is still quite good – within 0.5% and 1.5% of the system total for the LHA and BAT approximations, respectively – as is displayed by the dashed green curve (right axis) of Fig. 11. The regions of poorest conservation coincide with the times when the trajectories are predominantly in the coupling region. State populations are also in good agreement with the SOFT values, as can be also be seen in Fig. 11. Note that the BAT-approximated results require twice as many basis functions to be in good agreement with those computed with the LHA method; we interpret this not necessarily as a failing of the former, but as an instance of a system engineered to demonstrate particularly good results for the latter due to its exact nature.

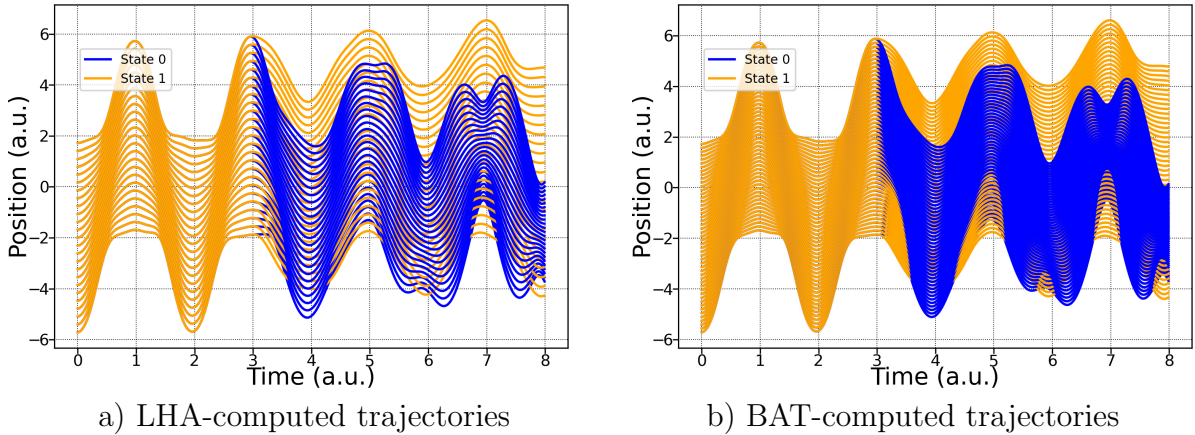
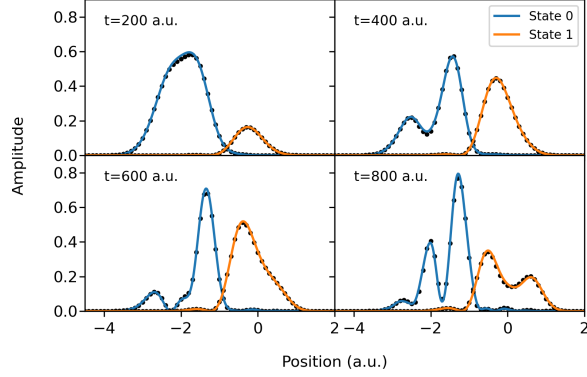
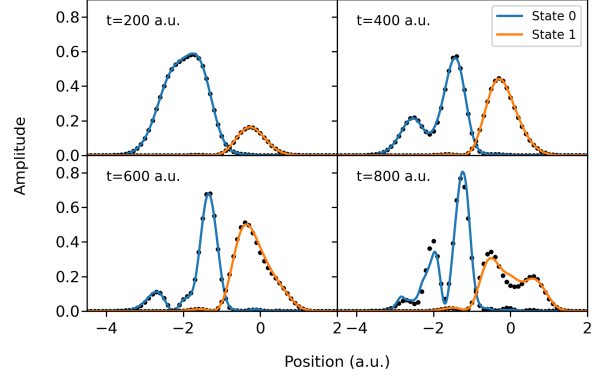


Figure 9: Trajectories for the Holstein model, computed within the LHA (a) and BAT (b) approximations. The desynchronization between states is evident in both instances at $t \approx 3$ a.u.; prior to that, the excited state trajectories (State 1, orange) move in tandem with their ground state partners (State 0, blue).

As mentioned previously, the nature of the Holstein model evaluated via the LHA also

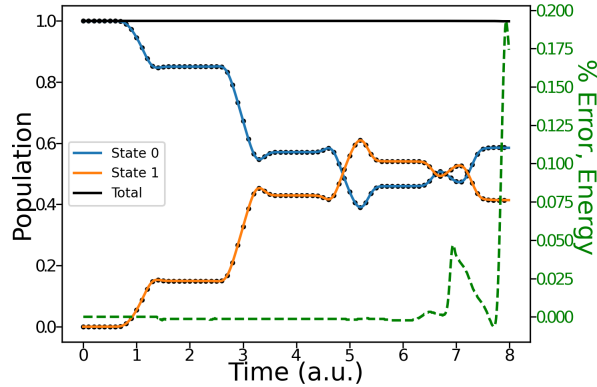


a) LHA-computed wavefunctions

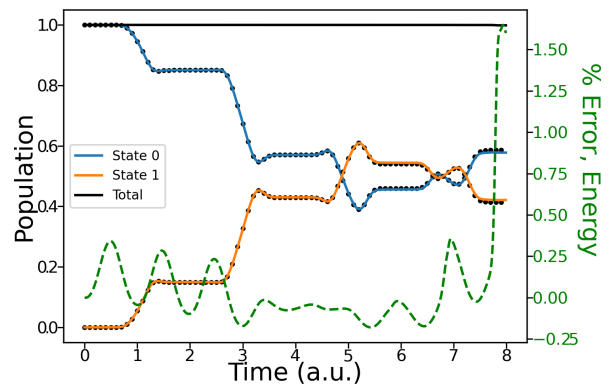


b) BAT-computed wavefunctions

Figure 10: Wavefunction snapshots taken every 200 iterations on the ground (State 0, blue) and excited (State 1, orange) states, computed within the LHA (a) and BAT (b) approximations. Black circles indicate exact quantum results, as calculated using the SOFT method with 2048 grid points. The snapshot frequency is chosen to coincide with oscillations out of the coupling region.



a) LHA-computed observables



b) BAT-computed observables

Figure 11: State populations and energy conservation for the LHA- and BAT-approximated Holstein model potentials. Black circles represent exact (SOFT) results.

allows for some direct testing of the basis initialization parameters N_b , a_{scl} , and ρ_c , free from error introduced by potential approximations. Taken together, these values (along with the initialization scheme) define the size and quality of the initial basis, which impacts the stability and accuracy of the calculations at later times. Using a fixed value of $\rho_c = 1 \times 10^{-12}$, we test combinations of N_b and a_{scl} by monitoring the maximum absolute error in total energy for simulations up to $t = 8$ a.u. The results, displayed as a heatmap in Fig. 12, suggest three regimes of basis quality. In the first regime, the basis is large and each function wide – a pairing that would conceptually lead to accurate results, although the near-unity of the overlap matrix elements causes numerical issues with the inversion of \mathbf{S} and ultimately leads to premature crashing of the calculation. The second regime corresponds to a compromise between basis size and overlap, and yields stable results that improve with an increase in basis width (decreasing a_{scl}). Finally, the third regime seen in Fig. 12 describes a basis – albeit stable – that is too small and leads to large errors in energy conservation regardless of width. Note that the specific values and domains of this figure will be system-dependent; the goal here is to emphasize the qualitative aspects of the initial basis parameter selection procedure and its impact on possible simulation outcomes.

4.2 Avoided Crossing Models

We also demonstrate the performance of the QTAG algorithm using the well known two-state models of Tully,⁵¹ which have served as a testing ground for numerous algorithms to date. In the diabatic representation, Model 1 (single avoided crossing model) is a pair of crossing surfaces with Gaussian coupling localized at their intersection, as displayed in the middle panel of Fig. 8. Their form is also seen in Eqs 39a – 39c; a hyperbolic tangent representation is used here instead of the original piecewise definition, as this yields smooth and continuous

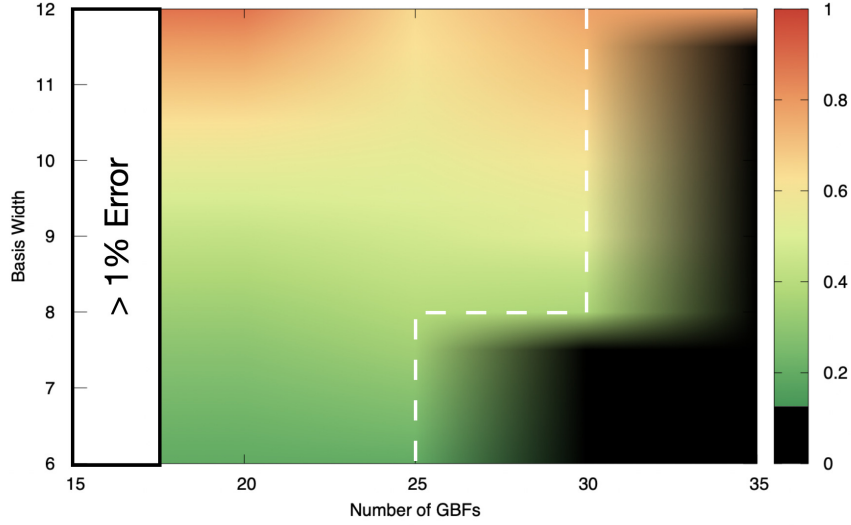


Figure 12: Heatmap of maximum percent error in energy for the LHA-approximated Holstein model; the black region in the lower right corresponds to simulations which failed due to singularities in the overlap matrix, meaning the interpolation in this region (demarcated by the dashed line) is hypothetical. The white region indicates simulations with percent errors larger than the color scale (i.e. $>1\%$).

derivative expressions.

$$V_{11}^{SAC} = A(1 + \tanh(Bx)) \quad (39a)$$

$$V_{22}^{SAC} = A(1 - \tanh(Bx)) \quad (39b)$$

$$V_{12}^{SAC} = V_{21} = Ce^{-Dx^2} \quad (39c)$$

The surface parameters are $A = 0.01$ and $B = 1.147$, while the coupling parameters are $C = 0.005$ and $D = 1.0$.

The wavepacket, represented by 25(35) basis functions in the LHA(BAT) approximation, is initialized on the lower S_0 surface and propagated for $t = 2000$ a.u. – i.e. for a sufficiently long time to have crossed the coupling region. Snapshots of $|\Psi|^2$ calculated in the BAT approximation scheme are shown in Fig. 13a in increments of 500 a.u.; agreement with SOFT results computed from 2048 points is quite good. The corresponding surface populations and energy conservation (again represented as a percent error) are displayed in Fig. 13b. The

quality of the BAT-based results is improved in this model relative to the Holstein one, both in terms of basis compactness and energy fluctuations; this can be attributed both to the favorable contour of the potential surfaces – which are nearly linear here than in the parabolic shapes seen before – and to the relative decrease in quality of the LHA since it is no longer exact in this case. Moreover, both approximation schemes benefit from the design of the first Tully system, which prevents multiple visits to the coupling region by the traveling wavepacket and thereby removes some of the more complex interference effects seen in the longer times of the Holstein model.

The second Tully model is a dual avoided crossing, as seen in the bottom panel of Fig. 8 and defined by Eqs 40a – 40c in the diabatic representation:

$$V_{11}^{DAC} = 0 \tag{40a}$$

$$V_{22}^{DAC} = -Ae^{-Bx^2} + E \tag{40b}$$

$$V_{12}^{DAC} = V_{21} = Ce^{-Dx^2} \tag{40c}$$

The constants here are the same as in the original formulation ($A=0.1$, $B=0.28$, $C=0.015$, $D=0.06$, $E=0.05$); the coupling surface defined in this way is quite wide, spanning both crossing regions. One point of note unique to this system is the use of a small decoupling parameter (see Table 1), a choice that allows the surfaces’ trajectories evolve nearly independently of one another for the duration of the simulation. We choose such a low value to prevent spurious resynchronization of the low-population trajectories back to their high-population counterparts: this artificial ‘snapping’ of trajectories away from their current positions is a possible concern in systems where state populations return to zero, as in the S_1 case here.

The wavepacket is composed of 35 GBFs in both the LHA- and BAT-approximated cases, and initialized on the S_0 surface. Propagation for $t = 2000$ a.u. is sufficiently long for passage through both crossing regions, as seen in the wavefunction snapshots of Fig. 14a.

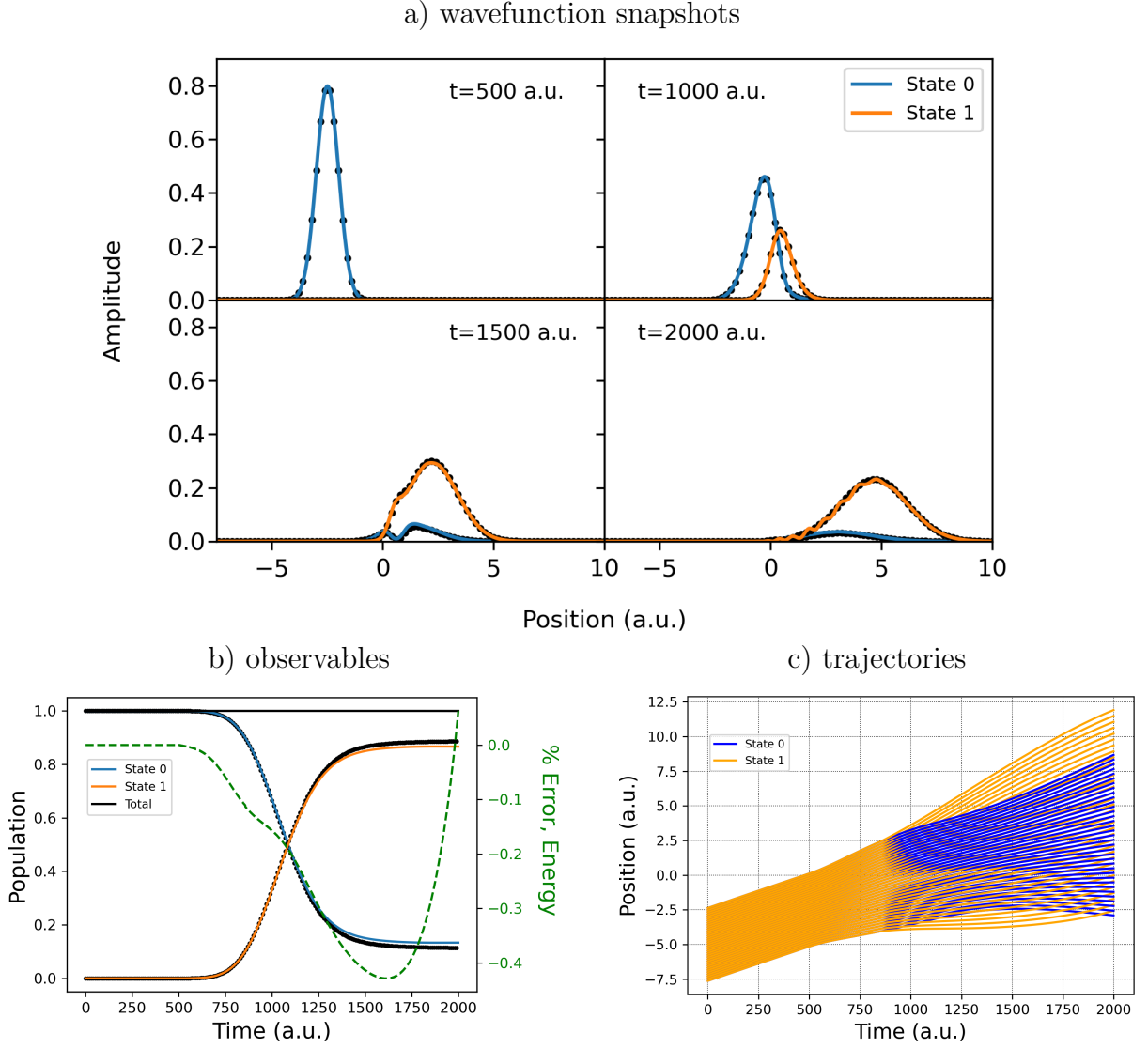


Figure 13: Wavefunction snapshots (a), energies and populations (b), and trajectories (c) for the single avoided crossing model described in-text, calculated using BAT-approximated potentials. In each case, blue and orange lines correspond to the S_0 and S_1 states, respectively, while black dots indicate exact quantum results.

The corresponding populations and trajectories (again, visualized for the BAT results) are displayed in Fig. 14b and 14c, respectively. Both the LHA and BAT results show good agreement with the SOFT data for the majority of the simulation, deviating at longer times due to a small but persistent population on the S_1 state. This is reflected in the dispersion of S_1 trajectories seen near the end of panel 14c, and serves as a visual aid for the possibility of trajectory ‘snapping’, as discussed above.

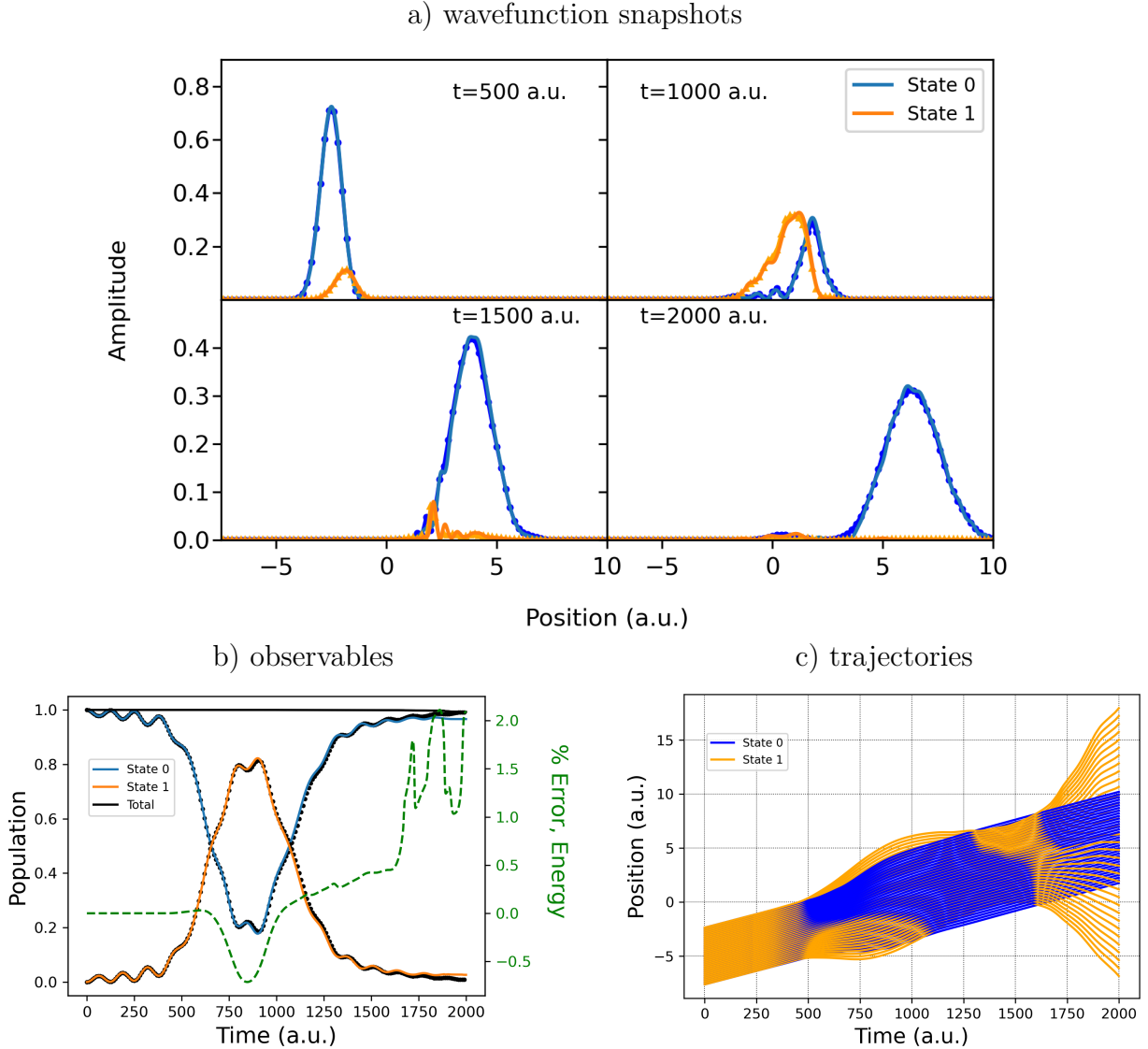


Figure 14: Wavefunction snapshots (a), energies and populations (b), and trajectories (c) for the dual avoided crossing model described in-text, calculated using BAT-approximated potentials. In each case, blue and orange lines correspond to the S_0 and S_1 states, respectively, while the corresponding points indicate exact quantum results.

5 Conclusions

In this work, we present the nonadiabatic extension of the single-surface QTAG method and its implementation in the Libra software suite. The QTAG functionality is available in Libra package starting since version v5.3 reported here. This version is available at the Zenodo server, as well as from the Quantum Dynamics Hub GitHub Libra software repository <https://github.com/Quantum-Dynamics-Hub/libra-code>. The current work provides a comprehensive account of the underlying theoretical foundations and terminology of the method, the key algorithms used, and the important implementation details and use guidance. The detailed examples to run the calculations presented in this work are available at the Zenodo server⁵² (<https://zenodo.org/record/7106561#.Yy2rhLTMK3A>), as well as from the GitHub data repository https://github.com/AkimovLab/Project_QTAG.

The flexibility of the Libra interface allows us to construct model potentials, commonly used in nonadiabatic dynamics methods benchmarking, with which we demonstrate various facets of our algorithm. Inspired by the Bohmian formulation of quantum mechanics, the biggest obstacle in our trajectory propagation scheme is the inherent instability of the trajectories themselves – to mitigate these effects and enforce numerical smoothness, we incorporate both single- and multi-surface adjustments to the basis. Furthermore, as integrable potentials in the GBF basis cannot be guaranteed, we explore the impact of the use of first- and second-order Taylor series approximations as a way to estimate the required matrix elements on the computed dynamics. We argue that the BAT expansion is expected to be especially useful in practical applications of the method, where second derivatives are not generally available.

Using the one-dimensional, two-state Holstein model, we demonstrate a “best-case” simulation, in which all potential elements can be evaluated exactly (via the LHA, which is exact for quadratic surfaces) and the single-surface momentum smoothing is most appropriate. The quality of results are relatively insensitive to the choice of basis initialization parameters, although extreme cases of basis over- or under-population lead to numerical errors and

poor wavefunction representation, respectively. We demonstrate that results obtained using the BAT approximation for this system can be comparable in quality to those of the exact case, although in general a larger number of basis functions are required.

We demonstrate how the developed QTAG approach applies to famous avoided crossing models (SAC/DAC) of Tully’s - the problems designed to highlight the nonadiabatic effects in electron-nuclear dynamics. In our test, both the BAT and LHA approaches are approximate in nature and perform comparably well. Further comparison of the SAC and DAC systems reveals an apparent insensitivity of the simulations to the basis desynchronization parameter ρ_d ; similar quality results are obtained in each case, despite ρ_d being an order of magnitude smaller in the latter. Although not explicitly tested in the present work, it seems at least plausible that this insensitivity could allow for nearly fully independent evolution of basis functions across electronic states.

Finally, we note that the suite of numerical basis modifications made in the current QTAG implementation could be extended to include numerous other approaches, which in turn may further improve numerical accuracy and computational efficiency.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant CHE-1955768 (SG). The research used an XSEDE⁵³ allocation TG-DMR110037 and the UofSC HPC cluster, funded by the National Science Foundation under Grant No. CHE-1048629. AVA acknowledges support of the National Science Foundation, Grant OAC-1931366.

Author contributions

Matthew Dutra: conceptualization, methodology, software, validation, data curation, writing – original draft, writing – review & editing. **Sophya Garashchuk:** conceptual-

ization, methodology, software, validation, data curation, writing – original draft, writing – review & editing, supervision, project administration, funding acquisition. **Alexey Aki-mov**: conceptualization, methodology, software, validation, data curation, writing – original draft, writing – review & editing, supervision, project administration, funding acquisition.

Appendix A. Gaussian integrals in the libgwp library

The `libgwp` library in `Libra` contains a multitude of useful modules for calculating various Gaussian integrals, which are used extensively in the QTAG algorithm. A Gaussian wavepacket G_i (GWP) in one dimension is defined as:

$$G_i = G(x; x_i, p_i, \alpha_i, \gamma_i) = \left(\frac{2\alpha_i}{\pi} \right)^{1/4} \exp \left(-\alpha_i(x - x_i)^2 + \frac{ip_i(x - x_i)}{\hbar} + \frac{i\gamma_i}{\hbar} \right), \quad (\text{A.1})$$

with center, momentum, width, and phase parameters defined by x_i , p_i , α_i , and γ_i , respectively. Notice that this definition differs from that of the i -th QTAG Gaussian basis function (GBF) of Eq. 7 in the following ways:

$$\underbrace{\alpha_i^{(GWP)}}_{\text{Eq. A.1}} = \underbrace{\frac{a_i^{(GBF)}}{2}}_{\text{Eq. 7}}, \quad \underbrace{\gamma_i^{(GWP)}}_{\text{Eq. A.1}} = \underbrace{s_i^{(GBF)}}_{\text{Eq. 7}}. \quad (\text{A.2})$$

The QTAG basis can thus be mapped onto the form of a GWP expected by `libgwp`, where the following formulas are implemented. The overlap matrix element S_{12} between Gaussian functions G_1 and G_2 is equal to:

$$S_{12} = \langle G_1 | G_2 \rangle = \left(\frac{2\bar{\alpha}}{\alpha_1 + \alpha_2} \right)^{1/4} \exp \left(-\frac{\bar{\alpha}}{2}(x_2 - x_1)^2 - \frac{(p_2 - p_1)^2}{4\hbar^2(\alpha_1 + \alpha_2)} + \frac{i(\gamma_2 - \gamma_1)}{\hbar} \right) \quad (\text{A.3})$$

where the following notations have been used:

$$\bar{\alpha} := \frac{2\alpha_1\alpha_2}{\alpha_1 + \alpha_2} \quad (\text{A.4a})$$

$$\bar{p} := \frac{\alpha_1 p_2 + \alpha_2 p_1}{\alpha_1 + \alpha_2} \quad (\text{A.4b})$$

The matrix element of the transition dipole moment is given by:

$$\mu_{12} = \langle G_1 | x | G_2 \rangle = \left(\frac{\alpha_1 x_1 + \alpha_2 x_2}{\alpha_1 + \alpha_2} + \frac{i(p_2 - p_1)}{2\hbar(\alpha_1 + \alpha_2)} \right) S_{12}. \quad (\text{A.5})$$

The first-order nonadiabatic derivative coupling matrix element d_{12} is equal to:

$$d_{12} = \langle G_1 | \nabla | G_2 \rangle = \left(\bar{\alpha}(x_2 - x_1) + \frac{i\bar{p}}{\hbar} \right) S_{12}. \quad (\text{A.6})$$

The kinetic energy matrix element for Gaussians G_1 and G_2 with particle mass M is:

$$T_{12} = -\frac{\hbar^2}{2M} \langle G_1 | \nabla^2 | G_2 \rangle = \frac{\hbar^2}{2M} \left(\bar{\alpha} - \left(\bar{\alpha}(x_2 - x_1) + \frac{i\bar{p}}{\hbar} \right)^2 \right) S_{12}. \quad (\text{A.7})$$

The product of two GWPs (note complex conjugation) gives a new GWP,

$$G_1^* G_2 = \left(\frac{2\bar{\alpha}}{\pi} \right)^{1/4} \exp \left(-\frac{\bar{\alpha}}{2} (x_2 - x_1)^2 \right) G(x; x_0, p_0, \alpha_0, \gamma_0), \quad (\text{A.8})$$

with the following parameters:

$$x_0 = \frac{\alpha_1 x_1 + \alpha_2 x_2}{\alpha_1 + \alpha_2} \quad (\text{A.9a})$$

$$p_0 = p_1 + p_2 \quad (\text{A.9b})$$

$$\alpha_0 = \alpha_1 + \alpha_2 \quad (\text{A.9c})$$

$$\gamma_0 = \gamma_2 - \gamma_1 + \bar{p}(x_2 - x_1). \quad (\text{A.9d})$$

References

- (1) Butler, L. J. Chemical reaction dynamics beyond the Born-Oppenheimer approximation. *Annu. Rev. Phys. Chem.* **1998**, *49*, 125–171.
- (2) Sun, L.; Song, K.; Hase, W. L. A S(N)2 reaction that avoids its deep potential energy minimum. *Science* **2002**, *296*, 875–878.
- (3) Ammal, S. C.; Yamataka, H.; Aida, M.; Dupuis, M. Dynamics-driven pathway in an intramolecular rearrangement. *Science* **2003**, *299*, 1555–1557.
- (4) Barbatti, M.; Ruckebauer, M.; J.Â Szymczak, J.; A.Â Aquino, A. J.; Lischka, H. Nonadiabatic excited-state dynamics of polar π -systems and related model compounds of biological relevance. *Physical Chemistry Chemical Physics* **2008**, *10*, 482–494, Publisher: Royal Society of Chemistry.
- (5) Blumberger, J. Recent advances in the theory and molecular simulation of biological electron transfer reactions. *Chemical reviews* **2015**, *115*, 11191–11238, Publisher: ACS Publications.
- (6) Zheng, Q.; Chu, W.; Zhao, C.; Zhang, L.; Guo, H.; Wang, Y.; Jiang, X.; Zhao, J. Ab initio nonadiabatic molecular dynamics investigations on the excited carriers in condensed matter systems. *WIREs Computational Molecular Science* **2019**, *9*, e1411, eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1411>.
- (7) Bolton, J. R., Mataga, N., McLendon, G., Eds. *Electron transfer in inorganic, organic and biological systems*; American Chemical Society, Washington, DC, 1991.
- (8) Nelson, T. R.; White, A. J.; Bjorgaard, J. A.; Sifain, A. E.; Zhang, Y.; Neugebauer, B.; Fernandez-Alberti, S.; Mozyrsky, D.; Roitberg, A. E.; Tretiak, S. Non-adiabatic Excited-State Molecular Dynamics: Theory and Applications for Modeling Photo-physics in Extended Molecular Materials. *Chemical Reviews* **2020**, *120*, 2215–2287.

- (9) Kilina, S. V.; Kilin, D. S.; Prezhd, O. V. Breaking the Phonon Bottleneck in PbSe and CdSe Quantum Dots: Time-Domain Density Functional Theory of Charge Carrier Relaxation. *ACS Nano* **2009**, *3*, 93–99.
- (10) Madjet, M. E.; Berdiyrov, G. R.; El-Mellouhi, F.; Alharbi, F. H.; Akimov, A. V.; Kais, S. Cation Effect on Hot Carrier Cooling in Halide Perovskite Materials. *The Journal of Physical Chemistry Letters* **2017**, *8*, 4439–4445, Publisher: American Chemical Society.
- (11) Senanayake, R. D.; Akimov, A. V.; Aikens, C. M. Theoretical Investigation of Electron and Nuclear Dynamics in the [Au₂₅(SH)₁₈]¹⁻ Thiolate-Protected Gold Nanocluster. *The Journal of Physical Chemistry C* **2017**, *121*, 10653–10662, Publisher: American Chemical Society.
- (12) Long, R.; Fang, W.; Akimov, A. V. Nonradiative Electron–Hole Recombination Rate Is Greatly Reduced by Defects in Monolayer Black Phosphorus: Ab Initio Time Domain Study. *The Journal of Physical Chemistry Letters* **2016**, *7*, 653–659, Publisher: American Chemical Society.
- (13) Li, H.; Gong, X.; Ni, H.; Lu, P.; Luo, X.; Wen, J.; Yang, Y.; Qian, X.; Sun, Z.; Wu, J. Light-Induced Ultrafast Molecular Dynamics: From Photochemistry to Optochemistry. *The Journal of Physical Chemistry Letters* **2022**, *13*, 5881–5893.
- (14) Tully, J. C. Perspective: Nonadiabatic dynamics theory. *The Journal of Chemical Physics* **2012**, *137*, 22A301.
- (15) Agostini, F.; Curchod, B. F. E. Different flavors of nonadiabatic molecular dynamics. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2019**, *9*:e1417, 1–21.
- (16) Crespo-Otero, R.; Barbatti, M. Recent Advances and Perspectives on Nonadiabatic Mixed Quantum–Classical Dynamics. *Chemical Reviews* **2018**, *118*, 7026–7068.

- (17) Wang, L.; Akimov, A.; Prezhd, O. V. Recent Progress in Surface Hopping: 2011–2015. *The Journal of Physical Chemistry Letters* **2016**, *7*, 2100–2112.
- (18) Smith, B.; Akimov, A. V. Modeling nonadiabatic dynamics in condensed matter materials: some recent advances and applications. *Journal of Physics: Condensed Matter* **2020**, *32*, 073001.
- (19) Donoso, A.; Kohen, D.; Martens, C. C. Simulation of nonadiabatic wave packet interferometry using classical trajectories. *J. Chem. Phys.* **2000**, *112*, 7345–7354.
- (20) Lopreore, C. L.; Wyatt, R. E. Electronic transitions with quantum trajectories. II. *J. Chem. Phys.* **2002**, *116*, 1228–1238.
- (21) Tully, J. C. Nonadiabatic dynamics. Modern methods for multidimensional dynamics computations in chemistry. 1998; pp 34–72.
- (22) Mai, S.; Marquetand, P.; Gonzalez, L. Nonadiabatic dynamics: The SHARC approach. *WIREs Computational Molecular Science* **2018**, *8*, e1370.
- (23) Smith, B.; Shakiba, M.; Akimov, A. V. Nonadiabatic Dynamics in Si and CdSe Nanoclusters: Many-Body vs Single-Particle Treatment of Excited States. *Journal of Chemical Theory and Computation* **2021**, *17*, 678–693.
- (24) Richardson, J. O.; Thoss, M. Communication: Nonadiabatic ring-polymer molecular dynamics. *The Journal of Chemical Physics* **2013**, *139*, 031102.
- (25) Ananth, N. Mapping variable ring polymer molecular dynamics: A path-integral based method for nonadiabatic processes. *The Journal of Chemical Physics* **2013**, *139*, 124102.
- (26) Shakib, F. A.; Huo, P. Ring Polymer Surface Hopping: Incorporating Nuclear Quantum Effects into Nonadiabatic Molecular Dynamics Simulations. *The Journal of Physical Chemistry Letters* **2017**, *8*, 3073–3080.

- (27) Wyatt, R. E.; Lopreore, C. L.; Parlant, G. Electronic transitions with quantum trajectories. *J. Chem. Phys.* **2001**, *114*, 5113–5116.
- (28) Garashchuk, S.; Rassolov, V. A.; Schatz, G. C. Semiclassical nonadiabatic dynamics using a mixed wave-function representation. *J. Chem. Phys.* **2005**, *123*, 174108.
- (29) Min, S. K.; Agostini, F.; Gross, E. K. U. Coupled-Trajectory Quantum-Classical Approach to Electronic Decoherence in Nonadiabatic Processes. *Phys. Rev. Lett.* **2015**, *115*, 073001.
- (30) Talotta, F.; Morisset, S.; Rougeau, N.; Lauvergnat, D.; Agostini, F. Internal Conversion and Intersystem Crossing with the Exact Factorization. *Journal of Chemical Theory and Computation* **2020**, *16*, 4833–4848.
- (31) Agostini, F.; Min, S. K.; Abedi, A.; Gross, E. K. U. Quantum-Classical Nonadiabatic Dynamics: Coupled- vs Independent-Trajectory Methods. *J. Chem. Theory Comput.* **2016**, *12*, 2127–2143.
- (32) Ha, J.-K.; Lee, I. S.; Min, S. K. Surface Hopping Dynamics beyond Nonadiabatic Couplings for Quantum Coherence. *J. Phys. Chem. Lett.* **2018**, *9*, 1097–1104.
- (33) Makhov, D. V.; Glover, W. J.; Martinez, T. J.; Shalashilin, D. V. Ab initio multiple cloning algorithm for quantum nonadiabatic molecular dynamics. *J. Chem. Phys.* **2014**, *141*, 054110.
- (34) Saller, M. A. C.; Habershon, S. Quantum Dynamics with Short-Time Trajectories and Minimal Adaptive Basis Sets. *J. Chem. Theory Comput.* **2017**, *13*, 3085–3096.
- (35) Joubert-Doriol, L.; Izmaylov, A. F. Nonadiabatic Quantum Dynamics with Frozen-Width Gaussians. *The Journal of Physical Chemistry A* **2018**, *122*, 6031–6042.
- (36) Curchod, B. F. E.; Martínez, T. J. Ab Initio Nonadiabatic Quantum Molecular Dynamics. *Chemical Reviews* **2018**, *118*, 3305–3336.

- (37) Gu, B.; Garashchuk, S. Quantum Dynamics with Gaussian Bases Defined by the Quantum Trajectories. *J. Phys. Chem. A* **2016**, *120*, 3023–3031.
- (38) Dutra, M.; Wickramasinghe, S.; Garashchuk, S. Quantum Dynamics with the Quantum Trajectory-Guided Adaptable Gaussian Bases. *Journal of Chemical Theory and Computation* **2020**, *16*, 18–34.
- (39) Dutra, M.; Wickramasinghe, S.; Garashchuk, S. Multidimensional Tunneling Dynamics Employing Quantum-Trajectory Guided Adaptable Gaussian Bases. *Journal of Physical Chemistry A* **2020**, *124*, 9314–9325.
- (40) Akimov, A. V. Libra: An open-Source “methodology discovery” library for quantum and classical dynamics simulations. *Journal of Computational Chemistry* **2016**, *37*, 1626–1649.
- (41) Makhov, D. V.; Glover, W. J.; Martinez, T. J.; Shalashilin, D. V. Ab initio multiple cloning algorithm for quantum nonadiabatic molecular dynamics. *The Journal of Chemical Physics* **2014**, *141*, 054110.
- (42) Mandal, A.; Yamijala, S. S.; Huo, P. Quasi-Diabatic Representation for Nonadiabatic Dynamics Propagation. *Journal of Chemical Theory and Computation* **2018**, *14*, 1828–1840.
- (43) Bohm, D. A suggested interpretation of the quantum theory in terms of “hidden” variables, I and II. *Phys. Rev.* **1952**, *85*, 166–193.
- (44) Wyatt, R. E. *Quantum Dynamics with Trajectories: Introduction to Quantum Hydrodynamics*; Springer-Verlag, 2005.
- (45) Abrahams, D.; Grosse-Kunstleve, R., W. Building Hybrid Systems with Boost.Python - 1.78.0. **2003**,

- (46) Temen, S.; Jain, A.; Akimov, A. V. Hierarchical equations of motion in the Libra software package. *International Journal of Quantum Chemistry* **2020**, *120*, e26373, [eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.26373](https://onlinelibrary.wiley.com/doi/pdf/10.1002/qua.26373).
- (47) Zhou, W.; Mandal, A.; Huo, P. Quasi-Diabatic Scheme for Nonadiabatic On-the-Fly Simulations. *Journal of Physical Chemistry Letters* **2019**, *10*, 7062–7070.
- (48) Maskri, R.; Joubert-Doriol, L. The moving crude adiabatic alternative to the adiabatic representation in excited state dynamics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2022**, *380*, 20200379.
- (49) Kosloff, R. Time-dependent quantum-mechanical methods for molecular dynamics. *J. Phys. Chem.* **1988**, *92*, 2087–2100.
- (50) Feit, M. D.; Fleck, J. A.; Steiger, A. *J. Comp. Phys.* **1982**, *47*, 412.
- (51) Tully, J. C. Molecular dynamics with electronic transitions. *J. Chem. Phys.* **1990**, *93*, 1061–1071.
- (52) Akimov, A. V.; Shakiba, M.; Smith, B.; Dutra, M.; Sato, K.; Temen, S.; Li, W.; Sun, X.; Stippell, L.; Chan, M. Quantum-Dynamics-Hub/libra-code: Libra with the QTAG method (v5.3.0). Zenodo, 2022.
- (53) Towns, J.; Cockerill, T.; Dahan, M.; Foster, I.; Gaither, K.; Grimshaw, A.; Hazelwood, V.; Lathrop, S.; Lifka, D.; Peterson, G. D.; Roskies, R.; Scott, J.; Wilkins-Diehr, N. XSEDE: Accelerating Scientific Discovery. *Computing in Science and Engineering* **2014**, *16*, 62–74.