NQCDynamics.jl: A Julia Package for Nonadiabatic Quantum Classical Molecular Dynamics in the Condensed Phase

James Gardner,[1] Oscar A. Douglas-Gallardo,[1] Wojciech G. Stark,[1] Julia Westermayr,[1] Svenja M. Janke,[1,2] Scott Habershon,[1] and Reinhard J. Maurer[1]

[1)]*Department of Chemistry, University of Warwick, Gibbet Hill Road, Coventry, CV4 7AL, UK*

[2)]*Institute of Advanced Study, University of Warwick, Gibbet Hill Road, Coventry, CV4 7AL, UK*

Accurate and efficient methods to simulate nonadiabatic and quantum nuclear effects in high-dimensional and dissipative systems are crucial for the prediction of chemical dynamics in condensed phase. To facilitate effective development, code sharing and uptake of newly developed dynamics methods, it is important that software implementations can be easily accessed and built upon. Using the Julia programming language, we have developed the `NQCDynamics.jl` package which provides a framework for established and emerging methods for performing semi-classical and mixed quantum-classical dynamics in condensed phase. The code provides several interfaces to existing atomistic simulation frameworks, electronic structure codes, and machine learning representations. In addition to the existing methods, the package provides infrastructure for developing and deploying new dynamics methods which we hope will benefit reproducibility and code sharing in the field of condensed phase quantum dynamics. Herein, we present our code design choices and the specific Julia programming features from which they benefit. We further demonstrate the capabilities of the package on two examples of chemical dynamics in condensed phase: the population dynamics of the spin-boson model as described by a wide variety of semi-classical and mixed quantum-classical nonadiabatic methods and the reactive scattering of H_2 on Ag(111) using the Molecular Dynamics with Electronic Friction method. Together, they exemplify the broad scope of the package to study effective model Hamiltonians and realistic atomistic systems.

I. INTRODUCTION

Classical molecular dynamics (MD) simulations are crucial to understanding dynamical processes and chemical reactions in molecules and materials. However, the assumptions that underpin classical MD simulations are regularly violated. This is, for example, the case when nonadiabatic and quantum nuclear effects cannot be neglected, i.e. when the time scales of electronic and nuclear dynamics do not clearly separate or when the motion of atoms cannot be approximated as being classical. These effects are important for a broad range of processes in condensed phase ranging from chemical reaction dynamics at metal surfaces to photocatalysis and nonequilibrium processes in materials.

The development of accurate simulation methods that are able to capture nonadiabatic and/or quantum effects in dynamics of hundreds or thousands of atoms and electrons or in open quantum systems represents a true frontier with important emerging applications in areas such as strong light-matter coupling and quantum transport.[1–3] While significant advances in the development of full unitary quantum dynamics methods have recently been reported,[4–6] a full quantum dynamical description for high-dimensional condensed phase systems remains mostly out of reach. A variety of mixed quantum-classical and semiclassical dynamics methods have been developed over the years that retain an (approximate) description of quantum effects while providing improved computational scaling properties. Examples include: Ehrenfest dynamics,[7–9] molecular dynamics with surface hopping,[10–16] mixed quantum-classical Liouville dynamics,[17–20] the quantum-classical path integral method,[21,22] and semiclassical mapping Hamiltonian methods.[23–38] Most of these methods were conceived with a relatively small number of electronic states in mind, but some have been extended and modified to tackle the continuum of states encountered in metallic environments. In particular, these include surface hopping methods,[39–42] molecular dynamics with electronic friction,[43–47] and mapping variable techniques.[48]

Despite the plethora of proposed methods, exploring their capabilities for application cases can be challenging as software implementations are often not publicly available. Only when methods grow in popularity do efficient open-source implementations start to appear and become maintained by active user communities. Just to name some examples, this has been the case for fewest-switches surface hopping methods for molecular systems[49] as implemented in Newton-X[50,51] or SHARC[52,53] and for path-integral molecular dynamics methods as implemented in i-PI.[54] However, many recently developed nonadiabatic and quantum-classical dynamics (NQCD) methods have not yet reached this stage of maturity in their development. A possible solution to bridge the gap between early inception of new approximate NQCD methods and their realisation for applications is to develop open-source implementations during their development, as recently suggested in a Faraday Discussion.[55] While this has become common practice in many other communities (e.g. in machine learning for chemical physics applications[56]), rarely are proof-of-principle implementations of new dynamics methods released together with the publications that first report them. Doing so would allow greater insight into the inception of the method and its numerical properties and would support reproducibil-

ity and user uptake. Furthermore, few standardised benchmark model problems exist with which new NQCD methods can be assessed. The potential success of such an effort has recently been shown for a number of projects in other fields and a similar opportunity exists in the development of NQCD methods.

In this article, we present an open-source software package, `NQCDynamics.jl`, that provides a framework for performing NQCD with a diverse range of methods, and toolsets for developing new simulation methods. The package aims to provide an open-source environment that can satisfy both the needs of prototype method development and performance-sensitive method deployment for production simulations. Our aim is to support open-source availability of newly developed simulation methods and to enable the transparent comparison and benchmarking of methods against each other. We achieve this by developing the code in the *Julia* programming language and by providing a range of existing NQCD methods. The code features a range of interfaces to employ model Hamiltonians, on-the-fly ab-initio electronic structure calculations, or high-dimensional atomistic machine learning models, which we demonstrate with two example problems. In Sec. II we introduce the *Julia* programming language, and describe the features of the package in Sec. III. Secs. IV and V discuss results for two example applications together with a concise description of the relevant theory. Sec. IV presents nonequilibrium population dynamics of the spin boson model, whereas Sec. V focuses on the reactive scattering of H_2 on an Ag(111) surface. The final section, Sec. VI, discusses our vision for the software package and planned future developments.

II. THE JULIA PROGRAMMING LANGUAGE

Before introducing the package, we will briefly introduce the *Julia* language and highlight the characteristics that make it an excellent choice for a software project suitable for both method prototyping and production simulations. *Julia*[57] is a modern language designed to combine user productivity with efficient code. This is achieved by providing a user friendly interface through the dynamic type system, while achieving high performance with effective *type inference* and *just-in-time* compilation.[57] On the surface, the syntax looks much like other dynamic languages such as *Python*, but the compiler is able to produce optimised assembly code that can achieve comparable performance to static languages such as *C* and *Fortran*.[57–60]

Aside from performance, a key requirement of scientific software is its ease of transferability and reuse. *Julia*'s built-in package manager `Pkg` allows for automated installation of project dependencies which facilitates code sharing and allows for seamless integration of cutting edge developments. Through `BinaryBuilder.jl` it is even possible to include binary dependencies from other languages without requiring the user to manually compile extra libraries. This is particularly relevant when considering the vast amounts of existing scientific software written in other languages.

Compared to most languages *Julia* is relatively young, launching in only 2012, though it has grown quickly and presents itself as a strong option for scientific computing projects. *Julia* is not completely new to the realm of molecular simulation; of particular note are the `DFTK.jl` package[61] and the `Fermi.jl` package.[62] `DFTK.jl` is a plane-wave density functional theory code and `Fermi.jl` is a wave-function-based quantum chemistry code. `DFTK.jl` has already been used to investigate new developments in the self-consistent field procedure.[63,64] The success of `DFTK.jl` has shown that *Julia* is not only viable, but effective at tackling chemical problems and producing high-performance software.

III. PACKAGE OVERVIEW

The goal of the package is to provide an environment where researchers can develop new methods for NQCD simulations, compare them to existing implementations, and scale them up to full production simulations on atomistic systems. This section describes the code design choices to fulfill these requirements.

To support new users and developers it is important to provide comprehensive, yet concise documentation. This is often a challenge, particularly for research code that undergoes constant development by a small team. Using automated build procedures we provide a stable and a development version of the documentation that builds whenever new versions are published. By frequently re-building the documentation, it is easier to incorporate additions and to ensure that new features are adequately documented. Further, examples within the documentation are executed during the build procedure, acting as additional tests and ensuring the reader is able to follow along without issue.

To further reduce the complexity of the codebase, we can rely upon external packages to provide specialised functionality. This has benefits for initial development, maintainability and documentation since we are not responsible for managing external dependencies, and get immediate access to their features. Often, this also has drawbacks since it can complicate the build procedure, acting as a barrier toward new developers. Fortunately, *Julia*'s built-in package manager makes it simple to include both *Julia* packages and binary dependencies without complicating the installation process. The full set of dependencies is specified in the `Project.toml` file as is standard in *Julia* packages, and these are automatically installed along with the package.

While minimising the barrier to entry, it is also important to ensure the package has enough scope for further contributions. To achieve this, a flexible interface was created that does not unnecessarily restrict the possibilities of future work. We utilise *Julia*'s *multiple dispatch* to simplify the addition of new functionality. *Multiple dispatch* allows the developer to define a new type, then add methods specialised for that type. With this, the developer is able to take advantage of the existing framework and to modify any functions that require different behaviour. This procedure is exemplified by our central

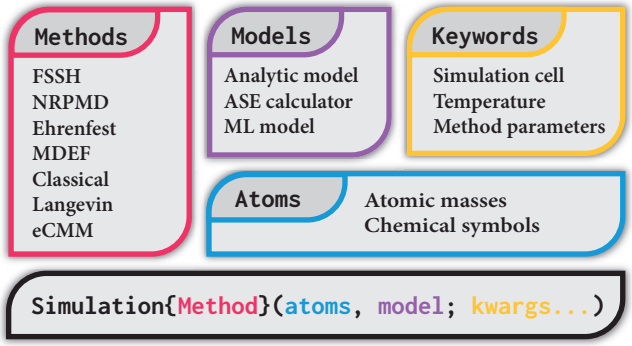

FIG. 1. The user inputs required to define the parameters for a simulation.

parameter type: `Simulation` (Fig. 1).

The `Simulation` holds the static parameters of the system such as the atom types, temperature and simulation cell. Further, its type parameter (`Method` in Fig. 1) acts as a label that determines the dynamics method. These `Methods` are regular *Julia* `structs`[65] and can be defined to contain any extra parameters. In this way, the `Simulation` type has a basic structure for shared functionality between dynamics methods, but allows for arbitrary extension through the `Method` parameter. By defining a new `Method`, *multiple dispatch* can be used to modify and implement functions to perform new dynamics methods.

Another goal of the package is to facilitate easier comparison with existing methods. Currently, this is challenging as implementations for many prototype dynamics methods are not publicly available and can be difficult to obtain. We provide implementations of many methods, along with detailed descriptions of the implementation specifics. In this way, the package can be used as a resource for obtaining benchmark data and as an educational resource that provides extra computational details, so that those details do not have to be covered in the supporting information of publications.

Finally, the prototype implementations must be easily transferable for both model Hamiltonians and realistic atomistic systems. Generally, there is a disconnect between research codes and large production applications that can lead to a duplication of effort when the developer must re-implement functionality in a more efficient or scalable format. The key difference between simple models and atomistic problems lies in the evaluation of the electronic Hamiltonian. The underlying dynamics to propagate the motion of atoms is identical. We can take advantage of this similarity by abstracting the dynamics from the electronic problem, exposing a simple interface for defining the Hamiltonian. This interface is packaged separately as `NQCModels.jl` and included as a dependency. By separating the interface, the models can be accessed individually and integrated into other codes.

NQCD simulations involve the calculation of observables over many trajectories. The initial coordinates for each trajectory are sampled from an appropriate distribution, before propagating the coordinates and momenta in time. During the propagation, at each timestep, the electronic Hamiltonian is evaluated as a function of the nuclear coordinates. The following sections discuss how each of these aspects is handled by `NQCDynamics.jl` by following the simulation workflow presented in Fig. 2.

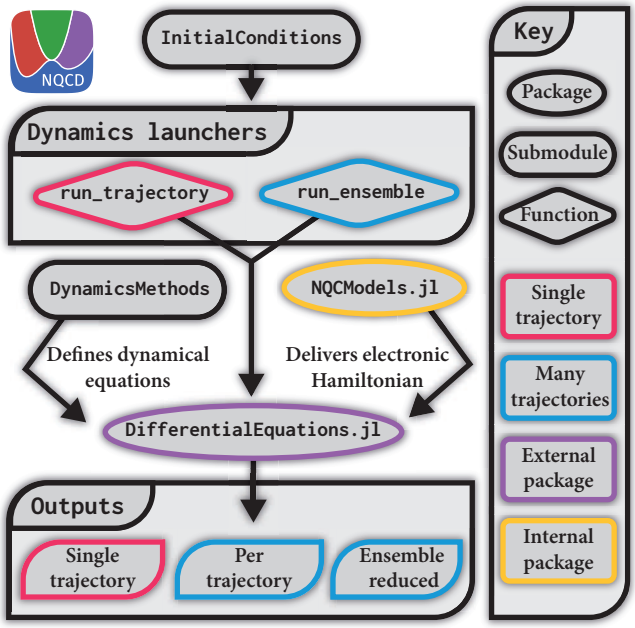

FIG. 2. Schematic diagram showing how the internal structure of the code relates to the workflow of a simulation. The ovals denote separate *Julia* packages, whilst boxes with rounded ends denote submodules within `NQCDynamics.jl`. The diamond boxes are functions exposed by `NQCDynamics.jl`. The arrows show how the outputs of each code section flow forward to produce the final output of the simulation.

A. Preparing initial conditions

Before performing dynamics simulations, it is important to ensure the initial nuclear and electronic distributions are sampled correctly, otherwise the trajectories become meaningless. Within `NQCDynamics.jl`, the submodule `InitialConditions` provides the functionality to generate these initial distributions. For simulations where the initial nuclear distribution is at thermal equilibrium we provide Monte Carlo sampling and dynamics using a Langevin thermostat.[66] Both of these methods have also been implemented in the path-integral form, which exploits the ring polymer normal mode representation to more efficiently sample the ring polymer phase space.[66–69] For non-equilibrium nuclear distributions, we have implemented Einstein-Brillouin-Keller (EBK) quantisation for diatomic molecules,[70] which generates semi-classical distributions with given vibrational and rotational quantum numbers and provides initial conditions for diatomic gas-surface scattering dynamics. In addition to these methods, we provide simple analytic distributions built on top of

`Distributions.jl`.[71] These include the Boltzmann velocity distribution, Wigner distributions for the quantum harmonic oscillator, and a ring polymer in a harmonic potential. The above methods are used to sample the nuclear degrees of freedom separately from the electronic variables. For all methods, the electronic variables are sampled analytically or set to represent a specific initial state. Currently, we provide only for initial conditions where the nuclear and electronic distributions are separable, allowing for individual sampling of each subsystem. In summary, the currently available sampling methods to create initial conditions include:

- Einstein-Brillouin-Keller quantisation[70]
- Langevin molecular dynamics (BAOAB algorithm)[66,72,73]
- Path integral Langevin dynamics (BCOCB algorithm)[68,69]
- Metropolis-Hastings Monte Carlo[66]
- Path integral Monte Carlo[66]

B. Performing dynamics

Each of the trajectory-based dynamics methods can be formulated as a set of coupled differential equations. Given the variety of differential equations that we must solve, it is easiest to use an established library for solving them, rather than implementing new algorithms and integrators for every dynamics method. This becomes especially relevant when developing new methods where, initially, the specific properties of the integration algorithm are not yet a priority. In *Julia*, the `DifferentialEquations.jl` package[74] provides a variety of algorithms for the numerical integration of differential equations. We have hence chosen to use `DifferentialEquations.jl` as the main driver for our dynamics simulations. With this choice, we need only define a function that evaluates the time-derivative of each of the dynamical variables that we can pass to any of the available solvers (defined in `DynamicsMethods`, Fig 2). Listing 1 shows the implementation of this function for the eCMM method (Sec. IV A 5).

```
function motion!(
  du, u, sim::AbstractSimulation{<:eCMM}, t
)
  # Create references to output variables
  dr = get_positions(du)
  dv = get_velocities(du)

  # Create references to input variables
  r = get_positions(u)
  v = get_velocities(u)

  # Set nuclear velocity
  velocity!(dr, v, r, sim, t)
  # Set nuclear acceleration
  acceleration!(dv, u, sim)
  # Set time-derivative of mapping variables
  set_mapping_force!(du, u, sim)
end
```

Listing 1: The function passed to the solver that governs the eCMM dynamics. The first parameter `du` is filled with the time-derivative of the dynamical variables `u`.

In some cases, such as when using ring polymer methods, there are specialised algorithms available that allow for larger timesteps and improved performance.[68,69] Although not immediately available from `DifferentialEquations.jl`, the implementation of additional integration algorithms is well documented in the online manual and, once implemented, they can be directly compared to the library of existing algorithms. We have taken this approach to implement versions of the MInt[28] and BCOCB[68,69] algorithms within `DifferentialEquations.jl` to efficiently integrate mapping variable and ring polymer dynamics, respectively.

As shown in Fig. 2, two functions are used to launch dynamics simulations: `run_trajectory` and `run_ensemble`. The former is used to perform a single trajectory at a time, whereas the latter can be used to perform multiple trajectories in parallel. The choice between the two ties directly into the output quantities from the dynamics (Fig. 2). In the single trajectory case, quantities of interest (positions, momenta, etc.) can be output at specified intervals. This functionality is also available from the ensemble interface, but there is the further option to request more complex observables, such as scattering probabilities or time-correlation functions. In doing so, it is possible to reduce the output as the trajectories finish, saving memory and reducing the burden of handling large amounts of data.

The parallelism available in the ensemble mode is provided by `DifferentialEquations.jl` and allows trajectories to be performed simultaneously using both shared memory and distributed memory parallelism. For large scale simulations on high performance computing facilities, the distributed form allows the user to leverage multi-node clusters to perform a large number of trajectories. To demonstrate the effectiveness of the parallelism we have included a scaling study (Fig. 3) carried out on a system equipped with Dell PowerEdge C6420 compute nodes with 48 cores each. These results were obtained by measuring the time taken to perform $100N$ trajectories using N compute cores. The simulations were carried out

as described in Sec. IV to obtain the eCMM result for model B. When using the simulation time span in Sec. IV, $t_{\max} = 20$, the efficiency begins to deteriorate when using more than 48 cores (1 node). However, when increasing the simulation time span to $t_{\max} = 200$, we see that the efficiency remains high across multiple nodes. This suggests that we are capable of achieving almost perfect scaling up to 768 cores (16 nodes), assuming that the time taken to simulate each trajectory is long enough to render the parallel overhead negligible.

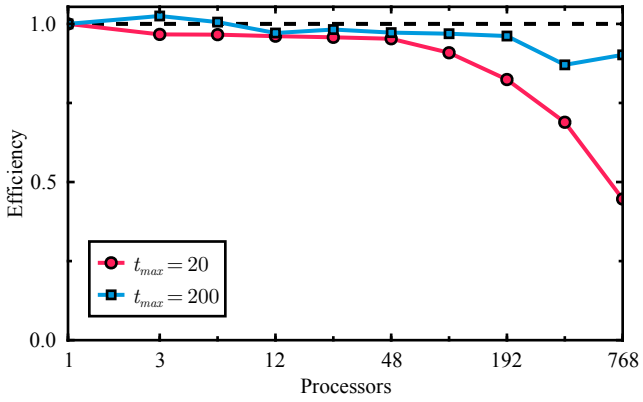

FIG. 3. The efficiency $(t(1)/t(N)$ where $t(N)$ is the time taken to perform $100N$ trajectories with N processors) obtained for eCMM trajectories using the `EnsembleDistributed` method for ensemble level parallelism. $t_{\max}$ denotes to the time span over which the trajectory was propagated (in atomic units). The dots show the mean value obtained from two samples.

Inside the `DynamicsMethods` submodule, the following dynamics methods are currently implemented:

- Classical molecular dynamics[66]
- Molecular dynamics with electronic friction (MDEF)[43,44]
- Ehrenfest molecular dynamics[7–9]
- Fewest-switches surface hopping[49]
- Ring polymer molecular dynamics (RPMD)[75,76]
- Nonadiabatic RPMD (NRPMD)[35–38]
- Centroid ring polymer surface hopping (RPSH)[14,15]
- Centroid ring polymer Ehrenfest dynamics[77]
- Extended classical mapping model (eCMM)[30,78]
- Generalized spin mapping approach[79,80]

C. Defining the Hamiltonian with NQCModels.jl

The final part of Fig. 2 that has not yet been described is the `NQCModels.jl` package. This package is responsible for providing the dynamics code with the potential energy surfaces that define the system interactions. In the case of analytic diabatic models, among others, these include Tully's two-state scattering models,[49] Coronado and Miller's three-state Morse potentials,[81] and the spin-boson model.[82] However, the package can also define or interface with high-dimensional atomistic models and *ab initio* Hamiltonians. We accomplish this by exposing a minimal set of functions that are required to take the nuclear positions and return the electronic quantities. The developer is free to wrap any code within these functions to perform the necessary computations. The flexible interface provided by `NQCModels.jl` is largely responsible for the system-size agnostic structure of the dynamics code.

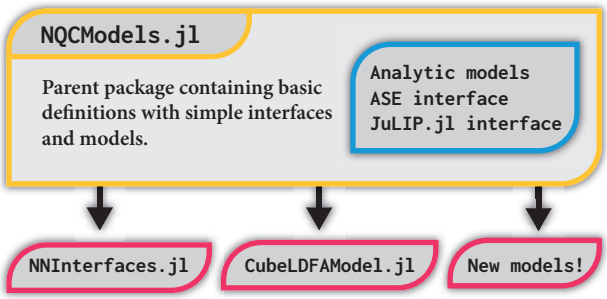

FIG. 4. Package layout diagram for `NQCModels.jl`. The blue box displays some of the models and interfaces included in the package, including ASE[83] and JuLIP.jl.[84] The bottom row shows some of the add-on packages used to obtain the results in Sec. V.

The modular design that *Julia* affords means that these models can be used separately from the dynamics code if desired, e.g. to compute energy values or to be incorporated into other codes. Fig. 4 shows the layout of the `NQCModels.jl` package. The top row shows the basic models and interfaces included in the package itself. The bottom row shows some of the child packages that implement the interface to provide add-on models. These extra models are tailored for specific applications and are packaged separately. For example, `NNInterfaces.jl` provides the H_2 on Ag(111) neural network model used in Sec. V. The advantage of this format is that `NQCModels.jl` can remain lightweight, with minimal dependencies, and add-on packages have more freedom as they operate separately from the main package.

One of the included interfaces is to the Atomic Simulation Environment (ASE)[83] written in *Python* that provides a library of calculators that give energies and forces using a variety of electronic structure programs. Using `PyCall.jl` we are able to directly access *Python* from within *Julia*. With this, we have implemented a simple wrapper for ASE calculators that allows us to access the values provided by ASE with minimal overhead. In principle, this interface can be used with any ASE calculator as easily as in native *Python*.[83] This provides capabilities to perform on-the-fly dynamics with a vast array of electronic structure and quantum chemistry codes that have existing calculator instances within ASE. It also provides access to atomistic machine learning (ML) packages via ASE, such as QUIP/GAP[85,86] and SchNetPack.[87,88] Both packages provide an ASE calculator instance that can be exposed to `NQCDynamics.jl` via our interface.

For atomistic molecular dynamics simulations, ML has become a key tool to facilitate dynamics of large systems or dynamics over long time-scales.[89,90] More recently, ML models of excited-state properties and whole Hamiltonians have become available[91–93] There has been substantial recent progress in machine learning with *Julia*.[94–96] For example the

`ACE.jl` package[97,98] provides for the parametrization of interatomic potentials based on the Atomic Cluster Expansion. Nevertheless, most existing atomistic ML models are developed and trained using other languages. Through *Julia*'s language interoperability features,[57] we connect to these models with minimal difficulty. The H_2 on Ag(111) model in `NNInterfaces.jl` (Sec. V) relies upon a *Fortran* library.[99] By directly calling the functions compiled into this library using `ccall`,[100] we are able to access the potential, forces, and electronic friction tensor from the same interface as the analytic models.

D. Example script

Now that we have introduced the fundamentals of the package, we can introduce a basic script that shows how each of the components work together. The example script (Listing 2) performs a single trajectory with classical molecular dynamics in a 1D harmonic potential. The structure of this script is typical for all dynamics simulations using `NQCDynamics.jl`.

```
1  using NQCDynamics # Import all exported symbols
2
3  atoms = Atoms([:H, :C]) # Atoms in the system
4  model = Harmonic() # External potential
5
6  # Combine simulation parameters
7  sim = Simulation{Classical}(atoms, model)
8
9  # Initialise starting position and velocity
10 velocity = zeros(1, 2) # all zero with size=(1,2)
11 position = rand(1, 2) # random with size=(1,2)
12
13 # Combine variables into single entity
14 z = DynamicsVariables(sim, velocity, position)
15
16 using Unitful # Import package for specifying units
17 tspan = (0.0, 10.0u"fs") # Simulation timespan
18
19 # Run a single trajectory
20 trajectory = run_trajectory(z, tspan, sim;
21   dt=0.1, output=(:position, :velocity))
```

Listing 2: Example script for classical molecular dynamics with two atoms in a 1D harmonic potential.

In Listing 2, line 1 imports the package. `using NQCDynamics` brings both the module name and all exported symbols into the global namespace.[101] Line 3 creates the `atoms` which tells the simulation which particles are in the system. Line 4 creates the `model`. In this example, `Harmonic` is a 1D harmonic potential, but can be replaced by any type that implements the `NQCModels.jl` interface. Line 7 shows the creation of the simulation, which is the central parameter type for all simulations (Fig. 1). This contains all of the static parameters of the system, which in this case, are the atoms and the model that defines their interactions. The `Method` type parameter, here `Classical`, is how the user chooses the dynamics method they will use, this can be any of the implemented dynamics methods. After creating the simulation, the dynamical variables are created. For classical dynamics, these are the positions and momenta but will contain electronic variables when performing nonadiabatic dynamics. The velocities and positions are provided as matrices with the number of degrees of freedom per atom along the first dimension and the number of atoms along the second dimension. The simulation time span is defined on line 17 to be 10 fs. By default, all quantities are assumed to be in atomic units, however, `Unitful.jl`[102] can be used to attach alternative units which are converted internally. Finally, the simulation is performed using `run_trajectory`. After execution, the output `trajectory` is a table containing the values for the positions and velocities at each timestep. The interface described here is similar for all dynamics methods, making it easy to switch between and compare different methods. Interfacing with the package via a *Julia* script means that the user has the ability to use any *Julia* functionality to manipulate inputs and outputs. This affords great flexibility when considering future developments.

In the preceding sections, we have provided motivation for using `NQCDynamics.jl` and described its functionality. `NQCDynamics.jl` is open-source and freely available on GitHub.[103] The package documentation and tutorials are hosted online with GitHub pages and are updated with each release.[104] The documentation provides a comprehensive introduction for new users along with implementation details and code specifications useful for developers. For each of the implemented methods, the theoretical background is introduced alongside walkthrough examples that aim to reproduce published results. In the following sections, we will present two example use cases for `NQCDynamics.jl`.

IV. EXAMPLE I: NON-EQUILIBRIUM POPULATION DYNAMICS OF THE SPIN-BOSON MODEL

In this example we will use `NQCDynamics.jl` to evaluate quantum time-correlation functions[82,105–107] of the form

$$C_{\hat{A}\hat{B}}(t) = \mathrm{Tr}\left[\hat{A}e^{i\hat{H}t/\hbar}\hat{B}e^{-i\hat{H}t/\hbar}\right], \tag{1}$$

where $\mathrm{Tr} = \mathrm{Tr}_n \mathrm{Tr}_e$ is the trace over both the nuclear and electronic subsystems and $\hat{A}$ and $\hat{B}$ are arbitrary quantum operators. $\hat{H}$ is the Hamiltonian operator of the full system. Depending on the identity of the operators $\hat{A}$ and $\hat{B}$, these correlation functions can be used to calculate reaction rates, spectra, and various transport coefficients.[105,106] Correlation functions of this form are typically challenging to evaluate using a quantum mechanically exact formalism but it is possible to approximate the quantum dynamics by using mixed quantum-classical and semiclassical dynamics.

In the following, the theory will be presented for a general Hamiltonian with F electronic states in the diabatic representation:

$$\hat{H} = \frac{1}{2}\hat{\mathbf{P}}^T\mathbf{M}^{-1}\hat{\mathbf{P}} + \sum_{n,m=1}^{F} V_{nm}(\hat{\mathbf{R}})\,|n\rangle\langle m| \tag{2}$$

Here, $\hat{\mathbf{R}}$ and $\hat{\mathbf{P}}$ are vectors of position and momentum operators and $\mathbf{M}$ is the diagonal mass matrix. Throughout we will be using bold notation for vectors and matrices. $V_{nn}(\hat{\mathbf{R}})$ are the diabatic potential energy surfaces and V_{nm} is the coupling between the two states $n \neq m$. In this section, we explore the case where $\hat{A} = \hat{\rho}(\hat{\mathbf{R}}, \hat{\mathbf{P}}) |n\rangle\langle n|$ and $\hat{B} = |m\rangle\langle m|$ such that the correlation function can be viewed as the time-dependent population of state m starting from a given initial density $\hat{\rho}(\hat{\mathbf{R}}, \hat{\mathbf{P}}) |n\rangle\langle n|$. This initial density is separable into the nuclear and electronic parts, where $\hat{\rho}(\hat{\mathbf{R}}, \hat{\mathbf{P}})$ is the thermal equilibrium nuclear distribution of the ground state, and the electronic population starts in state $|n\rangle$.

$$C_{nm}(t) = \mathrm{Tr}\left[\hat{\rho}(\hat{\mathbf{R}}, \hat{\mathbf{P}}) |n\rangle\langle n| e^{i\hat{H}t/\hbar} |m\rangle\langle m| e^{-i\hat{H}t/\hbar}\right], \quad (3)$$

In the next section we will briefly introduce a set of dynamics methods that are implemented in the package that can be used to approximate the population time-correlation function (Eq. 3). Each of these methods takes a trajectory based approach, where initial conditions are sampled from a distribution and propagated using the appropriate algorithm to evaluate the population at later time.

A. Methods

1. *Fewest switches surface hopping (FSSH)*

Tully's fewest-switches surface hopping (FSSH) method[108–110] is one of the most frequently used methods, implemented in many programs, for simulating coupled electron-nuclear dynamics in molecular systems.[50–53] In recent years, there have been several investigations focused on testing its efficacy in the condensed phase[111] and for modelling molecules on surfaces.[39,112,113] Here we will shortly summarize how it can be used to approximate time-correlation functions of the form introduced above.

In FSSH, the nuclei are described by the classical time-dependent Hamiltonian:

$$H^{\mathrm{FSSH}}(t) = \frac{1}{2}\mathbf{P}^T\mathbf{M}^{-1}\mathbf{P} + \sum_i^F \delta_{i,s(t)} E_i(\mathbf{R}) \quad (4)$$

Note that the symbols have no hats as they represent classical variables, not quantum operators. $E_i(\mathbf{R})$ is the energy of the i-th adiabatic state $|E_i\rangle$ obtained by diagonalising the electronic Hamiltonian. The time-dependent quantity $s(t)$ is the discrete state variable that takes on the integer value of the currently occupied adiabatic state. To obtain the value of $s(t)$, the time-dependent Schrödinger equation is propagated alongside the classical equations of motion for the nuclei. In the adiabatic basis, this equation can be cast in terms of the wavefuction expansion coefficients c_i as

$$i\hbar\dot{c}_i(t) = E_i(\mathbf{R})c_i(t) - i\hbar\sum_{j=1}^F \dot{\mathbf{R}}\cdot\mathbf{d}_{ij}(\mathbf{R})c_j(t). \quad (5)$$

where $\mathbf{d}_{ij}$ is the nonadiabatic coupling vector between states i and j. The basic assumption made in FSSH is that the nuclei move on one adiabatic potential energy surface at a time, as illustrated by the Kronecker delta in equation 4. After every time step, the probability to make a transition to another state is evaluated. Such a transition is called a hop and its success is determined by comparing the hopping probability to a uniform random number. If the computed hopping probability is larger than the random number, a hop takes place and the value of $s(t)$ changes. Typically, the hopping probability is evaluated based on nonadiabatic couplings,[53,109] but other approximate schemes have also been proposed.[114,115] We have implemented the hopping probability according to Subotnik *et al.*[12] which is based on the original notion of nonadiabatic couplings between adiabatic potential energy surfaces. `NQCDynamics.jl` implements the hopping procedure using callback functions from `DifferentialEquations.jl`.[74] Listing 3 shows the implementation of a general surface hopping procedure in our package. The `HoppingCallback` is given to the solver, which after every timestep performs the surface hopping. This callback approach decouples the discontinuous hopping events from the continuous dynamics and allows users to investigate alternative hopping schemes by re-implementing individual functions that appear in Listing 3. For example, different velocity rescaling procedures can be implemented by modifying the `rescale_velocity!` function.

```
"""
Return true if the proposed state differs
from the initial state.
"""
function check_hop!(u, t, integrator)::Bool
  sim = integrator.p
  dt = get_proposed_dt(integrator)
  evaluate_hopping_probability!(sim, u, dt)
  set_new_state!(sim.method, select_new_state(sim, u))
  return sim.method.new_state != sim.method.state
end

"""
If the velocity is rescaled successfully,
update the state variable.
"""
function execute_hop!(integrator)
  sim = integrator.p
  if rescale_velocity!(sim, integrator.u)
    set_state!(integrator.u, sim.method.new_state)
    set_state!(sim.method, sim.method.new_state)
  end
end

const HoppingCallback = DiscreteCallback(
  check_hop!, execute_hop!
)
```

Listing 3: Implementation of the surface hopping procedure in the code. The `HoppingCallback` will evaluate `check_hop!` at every time step. If `check_hop!` returns true, the hop is attempted using `execute_hop!`.

Since FSSH is a mixed quantum-classical method, the most appropriate approximation to Eq. 3 is the partially Wigner transformed expression:[116,117]

$$C_{nm}^{\mathrm{FSSH}}(t) = \frac{1}{(2\pi\hbar)^K}\int \mathrm{d}\mathbf{R}\,\mathrm{d}\mathbf{P} \\ \mathrm{Tr}_e\left[\rho_{\mathrm{W}}(\mathbf{R},\mathbf{P})\mathscr{P}_n(\mathbf{R},\mathbf{P},0)\mathscr{P}_m(\mathbf{R},\mathbf{P},t)\right], \quad (6)$$

K is the number of nuclear degrees of freedom, $\rho_{\mathrm{W}}(\mathbf{R},\mathbf{P})$ is the Wigner transformed nuclear density, and $\mathscr{P}_n(\mathbf{R},\mathbf{P},t)$ are the populations of state n obtained from surface hopping trajectories at time t. Recall that we are interested in the populations of the diabatic states, though we perform FSSH in the adiabatic representation. We calculate the diabatic populations using the mixed quantum classical density approach.[118,119] Numerical evaluation of Eq. 6 involves performing FSSH trajectories sampled from $\rho_{\mathrm{W}}(\mathbf{R},\mathbf{P})$ and averaging the population $\mathscr{P}_m(\mathbf{R},\mathbf{P},t)$ over all trajectories.

2. Ring polymer surface hopping (RPSH)

For FSSH, we are using the Wigner distribution to initialise the nuclear configurations. However, the Wigner distribution is difficult to sample for realistic systems,[120,121] and the classical propagation does not conserve the initial distribution, leading to zero point energy leakage.[122] A possible solution to these problems is to use ring polymer surface hopping (RPSH).[14,15] Ring polymer molecular dynamics (RPMD) uses the imaginary-time path integral formalism to map the quantum distribution of the nuclei onto the extended phase space of a classical ring polymer to approximate the simulation of real-time correlation functions.[75,76] The ring polymer is comprised of multiple replicas of the nuclei, each joined by harmonic springs with stiffness depending on the temperature and mass of the particle. At low temperatures and light particle masses, the springs become softer, leading to a swelling of the ring polymer and a particle that incorporates quantum effects such as zero point energy and, to a more limited extent, tunneling. At high temperatures, the stiff springs cause the polymer beads to coalesce, becoming equivalent to a classical particle. The key advantage of ring polymer dynamics is that the quantum Boltzmann distribution is conserved.

RPSH is an *ad hoc* combination of trajectory surface hopping and RPMD. The algorithm follows FSSH, except the classical nuclear dynamics are replaced by the ring polymer dynamics. However, the additional complexity of the ring polymer leads to some ambiguity in the implementation of the propagation of the electronic quantities and the rescaling of the momenta. Two options for treating this ambiguity exist: the bead and centroid approximations.[14] The bead approximation involves evaluating the electronic quantities for every bead and using each contribution to propagate the electronic quantities. The centroid approximation simply replaces the classical particle in the FSSH algorithm with the ring polymer centroid. When rescaling the momenta, the bead approximation conserves energy for the entire ring polymer, whereas the centroid approximation conserves energy only for the centroid. The method we use here is the centroid approximation since it is more convenient and previous results have shown little difference in results between both approaches.[15]

As with FSSH, the nuclear dynamics follow a classical Hamiltonian:

$$H^{\mathrm{RPSH}}(t) = \sum_{\alpha=1}^{N}\left[\frac{1}{2}\mathbf{P}_\alpha^T\mathbf{M}^{-1}\mathbf{P}_\alpha \right. \\ + \frac{1}{2}\omega_N^2(\mathbf{R}_\alpha-\mathbf{R}_{\alpha+1})^T\mathbf{M}(\mathbf{R}_\alpha-\mathbf{R}_{\alpha+1}) \\ \left. + \sum_i^F \delta_{i,s(t)}E_i(\mathbf{R}_\alpha)\right]. \quad (7)$$

This Hamiltonian matches equation 4, with the addition of N replicas, where each replica α is joined to the adjacent $\alpha+1$ with a harmonic spring. Since this is a ring polymer, the indices are cyclic and the final replica is connected to the first. The spring frequency is directly proportional to the temperature as $\omega_N = N/\hbar\beta$ where $\beta = (k_B T)^{-1}$. Other than the nuclear dynamics, RPSH proceeds exactly as FSSH in the approximation of equation 3, with the exception that the initial distribution is taken as the thermal ring polymer distribution $\rho_{\mathrm{RP}}(\mathbf{R},\mathbf{P})$.

$$C_{nm}^{\mathrm{RPSH}}(t) = \frac{1}{(2\pi\hbar)^{NK}}\int \mathrm{d}\mathbf{R}\,\mathrm{d}\mathbf{P} \\ \times \mathrm{Tr}_e\left[\rho_{\mathrm{RP}}(\mathbf{R},\mathbf{P})\mathscr{P}_n(\bar{\mathbf{R}},\bar{\mathbf{P}},0)\mathscr{P}_m(\bar{\mathbf{R}},\bar{\mathbf{P}},t)\right] \quad (8)$$

As with all of the methods, this integral is evaluated by simulating an ensemble of trajectories and averaging the populations. The populations $\mathscr{P}_n(\bar{\mathbf{R}},\bar{\mathbf{P}})$ are obtained as for FSSH, except that the ring polymer centroids replace the classical nuclei.

3. Ehrenfest molecular dynamics

As an alternative to surface hopping dynamics, a mean-field approach can be taken such that the force is averaged over all states, weighted by the electronic populations.[7–9] The Ehrenfest Hamiltonian can be written as

$$H^{\mathrm{E}}(t) = \frac{1}{2}\mathbf{P}^T\mathbf{M}^{-1}\mathbf{P} + \sum_i^F \left|c_i(t)\right|^2 E_i(\mathbf{R}). \quad (9)$$

Note that here, in contrast to FSSH and RPSH, the time-dependence of the classical Hamiltonian comes directly from the electronic coefficients $c_i(t)$, rather than from an auxiliary state variable. As with the surface hopping methods however, the electronic Schrödinger equation must be integrated alongside the Hamiltonian dynamics with Eq. 5. Since Ehrenfest is another mixed quantum-classical method, it approximates Eq. 3 exactly as FSSH and we can use Eq. 6. The evaluation of the populations $\mathscr{P}_n(\mathbf{R},\mathbf{P})$ is simplified here and comes directly from the electronic coefficients $c_i(t)$ after conversion to the diabatic representation.

4. Ehrenfest ring polymer molecular dynamics

As with FSSH, the same discussion surrounding the choice of initial nuclear distribution applies and the drawbacks of the Wigner distribution are still present. Similarly, we can introduce an *ad hoc* ring polymer formalism here to tackle the same problem.[77] We can directly follow the RPSH treatment for the electronic degrees of freedom (propagating the electronic equation of motion for the centroid), but obtain the nuclear forces from the Ehrenfest approach as described by the Hamiltonian:

$$H^{\mathrm{ERP}}(t) = \sum_{\alpha=1}^{N}\Bigg[\frac{1}{2}\mathbf{P}_\alpha^T\mathbf{M}^{-1}\mathbf{P}_\alpha + \frac{1}{2}\omega_N^2(\mathbf{R}_\alpha-\mathbf{R}_{\alpha+1})^T\mathbf{M}(\mathbf{R}_\alpha-\mathbf{R}_{\alpha+1}) + \sum_i^F \big|c_i(t)\big|^2 E_i(\mathbf{R}_\alpha)\Bigg]. \quad (10)$$

The dynamics of this Hamiltonian can be seen as a straightforward replacement of the classical nuclei of Ehrenfest dynamics with the classical ring polymer.

5. Extended classical mapping model (eCMM)

The classical mapping methods introduced in the following sections take a different approach to the mixed quantum-classical methods discussed previously. These methods seek to treat the nuclear and electronic degrees of freedom on equal footing by mapping the discrete electronic states onto continuous degrees of freedom, then taking the semiclassical limit. Many of the existing approaches are based upon the work of Meyer and Miller[23] and later Stock and Thoss[24] where the electronic degrees of freedom become the Meyer-Miller-Stock-Thoss mapping variables. More recently, a unified framework has been introduced[26] from which many existing mapping methods can be derived, including the Meyer-Miller Hamiltonian.

From the unified framework, the extended classical mapping model[30,78,123] uses the Meyer-Miller Hamiltonian:

$$H^{\mathrm{eCMM}} = \frac{1}{2}\mathbf{P}^T\mathbf{M}^{-1}\mathbf{P} + \sum_{n,m=1}^{F}\left[\frac{1}{2}(x_n x_m + p_n p_m) - \gamma\delta_{nm}\right]V_{nm}(\mathbf{R}) \quad (11)$$

Here, x_n and p_n are the electronic mapping variables for state n and γ is a parameter that can take any value greater than $-1/F$.

The eCMM population correlation function is

$$C_{nm}^{\mathrm{eCMM}}(t) = \frac{1}{(2\pi\hbar)^N}\int \mathrm{d}\mathbf{R}\,\mathrm{d}\mathbf{P}\int_{S(\mathbf{x},\mathbf{p})} F\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{p} \times \rho_{\mathrm{W}}(\mathbf{R},\mathbf{P})\left[\frac{1}{2}(x_n^2(0)+p_n^2(0))-\gamma\right] \times\left[\frac{1+F}{2(1+F\gamma)^2}(x_m^2(t)+p_m^2(t)) - \frac{1-\gamma}{1+F\gamma}\right] \quad (12)$$

where $\int_{S(\mathbf{x},\mathbf{p})}$ denotes integration over the constraint space

$$S(\mathbf{x},\mathbf{p}) = \sum_{n=1}^{F}\left[\frac{1}{2}(x_n^2+p_n^2)\right] = 1+F\gamma. \quad (13)$$

To evaluate this integral, the nuclear degrees of freedom are sampled from the Wigner distribution $\rho_{\mathrm{W}}(\mathbf{R},\mathbf{P})$ and the electronic degrees of freedom are sampled such that the constraint in Eq. 13 is satisfied. This is equivalent to sampling on the surface of a $2F$ dimensional hypersphere with radius $\sqrt{2+2F\gamma}$. Trajectories are then obtained using Eq. 11 to calculate the correlation at time t. Although not presented in this article, the similar spin mapping methods introduced by Runeson and Richardson[79,80] have been compared to the eCMM method[123] and are equivalent for certain choices of γ.

6. Nonadiabatic ring polymer molecular dynamics (NRPMD)

As with FSSH, an RPMD extension to classical mapping dynamics has been proposed, referred to as nonadiabatic ring polymer molecular dynamics (NRPMD).[35–37] NRPMD uses the Meyer-Miller representation for the electronic degrees of freedom and the ring polymer path-integral discretisation for the nuclear degrees of freedom. However, unlike the surface hopping alternative, NRPMD has rigorous mathematical justification through its links to the recently derived nonadiabatic Matsubara dynamics.[38] This formal theoretical footing helps to justify its implementation and removes some of the ambiguities encountered in methods such as RPSH.

The NRPMD Hamiltonian is given by

$$H^{\mathrm{NRP}}(t) = \sum_{\alpha}\Bigg[\frac{1}{2}\mathbf{P}_\alpha^T\mathbf{M}^{-1}\mathbf{P}_\alpha + \frac{1}{2}\omega_N(\mathbf{R}_\alpha-\mathbf{R}_{\alpha+1})^T\mathbf{M}(\mathbf{R}_\alpha-\mathbf{R}_{\alpha+1}) + \sum_{n,m=1}^{F}\left[\frac{1}{2}(x_{n,\alpha}x_{m,\alpha}+p_{n,\alpha}p_{m,\alpha})-\gamma\delta_{nm}\right]V_{nm}(\mathbf{R}_\alpha)\Bigg] \quad (14)$$

which bears much resemblance to the classical Meyer-Miller Hamiltonian in Equation 11. Usually this Hamiltonian is presented such that $\gamma = 1/2$, though it has been provided here in a more general form to emphasise the similarity to the Meyer-Miller Hamiltonian.

The NRPMD population time-correlation function is

$$C_{nm}^{\mathrm{NRP}}(t) = \int \mathrm{d}\mathbf{R} \int \mathrm{d}\mathbf{P} \int \mathrm{d}\mathbf{x} \int \mathrm{d}\mathbf{p} \times \rho_{\mathrm{RP}}(\mathbf{R},\mathbf{P}) \mathscr{P}_n(\mathbf{x},\mathbf{p},0) \mathscr{P}_m(\mathbf{x},\mathbf{p},t) \quad (15)$$

with the population estimator

$$\mathscr{P}_n(\mathbf{x},\mathbf{p},t) = \frac{1}{N} \sum_{\alpha=1}^{N} \left[\frac{1}{2}(x_{n,\alpha}^2(t) + p_{n,\alpha}^2(t)) - \gamma \right]. \quad (16)$$

As with the other ring polymer methods, the initial nuclear configuration is sampled from the thermal ring polymer distribution $\rho_{\mathrm{RP}}(\mathbf{R},\mathbf{P})$. The electronic variables are sampled according to the procedure detailed in ref. 37, which differs from the approach used above for eCMM. This methodology for sampling the electronic variables and evaluating the population follows the work of Chowdhury and Huo,[37] though more recently an alternative form has been presented[38] that more closely matches previous work with Meyer-Miller mapping dynamics.[32,34] When using only a single bead for NRPMD, the method becomes equivalent to the LSCI of Gao *et al.*[32] with focused initial conditions.[79]

7. Ring polymer extended classical mapping model

Before we apply all of these methods to a model system in the coming sections, it is interesting to consider how one might replace the nuclear dynamics of eCMM with ring polymer dynamics. Although not rigorously justified, the flexible structure of `NQCDynamics.jl` allows us to explore heuristic methods such as this and evaluate their effectiveness via numerical tests.

If we assume the system is described by the NRPMD Hamiltonian in Eq. 14, then we must adapt Eqs. 12 and 13 in line with the extended ring polymer phase space. Since we have N copies of each mapping variable, we can simply include the extra variables in the constraint summation such that the total population remains conserved.

$$S(\mathbf{x},\mathbf{p}) = \sum_{\alpha=1}^{N} \sum_{n=1}^{F} \left[\frac{1}{2}(x_{n,\alpha}^2 + p_{n,\alpha}^2) \right] = 1 + NF\gamma. \quad (17)$$

Then, in the spirit of the NRPMD population estimator (Eq. 16), we can rewrite the correlation function with populations accumulated over all the beads.

$$C_{nm}^{\mathrm{RPeCMM}}(t) = \frac{1}{(2\pi\hbar)^{NK}} \int \mathrm{d}\mathbf{R}\,\mathrm{d}\mathbf{P} \int_{S(\mathbf{x},\mathbf{p})} F\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{p} \times \rho_{\mathrm{RP}}(\mathbf{R},\mathbf{P}) \sum_{\alpha=1}^{N} \left[\frac{1}{2}(x_{n,\alpha}^2(0) + p_{n,\alpha}^2(0)) - \gamma \right] \times \sum_{\alpha=1}^{N} \left[\frac{1+NF}{2(1+NF\gamma)^2}(x_{m,\alpha}^2(t) + p_{m,\alpha}^2(t)) - \frac{1-\gamma}{1+NF\gamma} \right] \quad (18)$$

B. Spin-boson model

To compare each of the methods introduced in the previous section, we will use the spin-boson model. This model is comprised of a two-state system ($F = 2$) coupled to a bath of harmonic oscillators where the couplings and bath frequencies are characterised by a given spectral density $J(\omega)$. The model describes a dissipative quantum system and has been widely used as a benchmark for approximate nonadiabatic methods due to the feasibility of computing numerically exact quantum results.[30,32,124–129] The spin-boson Hamiltonian can be written in the form of Eq. 2 by setting

$$V(\hat{\mathbf{R}}) = \begin{pmatrix} \varepsilon + \mathbf{c}^T\hat{\mathbf{R}} & \Delta \\ \Delta & -\varepsilon - \mathbf{c}^T\hat{\mathbf{R}} \end{pmatrix} + \frac{1}{2}\hat{\mathbf{R}}^T\Omega^2\hat{\mathbf{R}} \quad (19)$$

and taking the mass matrix $\mathbf{M}$ to be the identity matrix. In the potential operator V, ε is the energy bias between the two states and Δ is the coupling between them. The couplings $\mathbf{c}$ to the position operators and the diagonal matrix of oscillator frequencies $\Omega = \mathrm{diag}(\omega_1,\ldots,\omega_{N_b})$ are obtained by discretisation of the spectral density.

In this work, we employ the Ohmic spectral density:

$$J(\omega) = \frac{\pi}{2}\eta\omega e^{-\omega/\omega_c}, \quad (20)$$

which can be discretised to give[30]

$$\omega_j = -\omega_c \ln[1 - j/(1+N_b)], \quad (21)$$

$$c_j = \sqrt{\frac{\eta\omega_c}{N_b+1}}\omega_j, \quad (22)$$

with $j = 1,\ldots,N_b$. For all simulations we set $N_b = 100$.

Different regimes of the model can be explored by modifying the relationship between the parameters. The model is symmetric when $\varepsilon = 0$ and asymmetric otherwise. The system-bath coupling strength is determined by the reorganisation energy $\zeta = 2\eta\omega_c$. The temperature regime is characterised by the relationship between β and Δ: the high temperature regime is encountered when $\beta\Delta < 1$ and the low temperature regime when $\beta\Delta > 1$. The balance between the adiabatic and nonadiabatic regimes is determined by ω_c and Δ. In the case of $\omega_c < \Delta$ the model represents the adiabatic regime, for $\omega_c > \Delta$ it represents the nonadiabatic regime. Throughout, $\Delta = 1$ and the parameters β and ω_c alone will determine the regime of the model. When both β and ω_c are large, this is the regime where both nuclear quantum effects and nonadiabatic effects become significant.

For this model, the initial nuclear density is given by the bath at thermal equilibrium:

$$\hat{\rho}(\mathbf{R},\mathbf{P}) = e^{-\beta\hat{H}_b(\mathbf{R},\mathbf{P})} / \mathrm{Tr}_n[e^{-\beta\hat{H}_b(\mathbf{R},\mathbf{P})}], \quad (23)$$

with the bath Hamiltonian

$$\hat{H}_b(\mathbf{R},\mathbf{P}) = \frac{1}{2}(\mathbf{P}^T\mathbf{P} + \mathbf{R}^T\Omega^2\mathbf{R}). \quad (24)$$

It is possible to sample the corresponding Wigner and ring polymer distributions analytically. The Wigner distribution is a normal distribution of the form:

$$\rho_{\mathrm{W}}(\mathbf{R},\mathbf{P}) = \prod_{j=1}^{N_b} \frac{\alpha_j}{\pi} \exp\left[-\frac{2\alpha_j}{\omega_j}(P_j^2+\omega_j^2 R_j^2)\right] \tag{25}$$

with $\alpha_j = \tanh\left(\frac{1}{2}\beta\hbar\omega_j\right)$.[79] The ring polymer distribution can be sampled in the ring polymer normal mode coordinates with the expression

$$\rho_{\mathrm{RP}}(\tilde{\mathbf{R}},\tilde{\mathbf{P}}) = \prod_{j=1}^{N_b}\prod_{\alpha=1}^{N} \exp\left[-\frac{\beta}{2N}(\tilde{P}_{j,\alpha}^2+\omega_{j,\alpha}^2\tilde{R}_{j,\alpha}^2)\right] \tag{26}$$

where $\omega_{j,\alpha}^2 = \omega_j^2+\omega_\alpha^2$ and ω_α is the frequency of the α-th normal mode. This can be converted back to the primitive coordinates using the standard ring polymer normal mode matrix.[67]

C. Simulation details

The results in the next section were obtained by performing 10^6 trajectories for each of the methods, sufficient for numerical convergence for all methods to the relevant accuracy. For NRPMD, the parameter γ was set equal to $\frac{1}{2}$ as is consistent with previous work[35,37] and for eCMM it is set to 0 as done previously.[30] A recent investigation into the value of γ suggests the effect is minimal for reasonable values.[78] For the ring polymer simulations, 50 beads were used to obtain converged results. However, for the ring polymer Ehrenfest and RPSH, using only a single bead was capable of reproducing the same population dynamics. We believe that this behaviour is specific to the fact that we are using centroid approximations and that the nuclear degrees of freedom are harmonic. The Wigner methods were integrated using the `Vern7` solver,[74,130] whilst the ring polymer methods used variants of MInt[28] and `Tsit5`[131] coupled with normal mode propagation for the ring polymer.[67–69] The fixed timestep methods used a timestep of 5×10^{-3} whereas the adaptive `Vern7` used relative and absolute tolerances of 1×10^{-10}.

D. Results and discussion

We have applied the methods described above to the five spin boson models A, B, C, D and E with parameters in Table I.[32] The first four models (A-D) have been previously investigated in the benchmark study of Gao *et al.*[32] and the final model (E) with elevated system-bath coupling appears in the work of Wang *et al.*[125]

For each method, we present two variants characterised by the representation used to model the nuclear degrees of freedom, either the Wigner or ring polymer representation. We evaluate $C_{11}(t)-C_{12}(t)$ which is the time dependent population difference between the two spin states. The numerically exact results for models A-D were calculated using the dissipation equation of motion method and are taken from the benchmark study of Gao *et al.*[32] Similarly, the exact result for model E is taken from Ref. 125 and was calculated using the multiconfiguration time-dependent Hartree approach.

TABLE I. Parameters for the five spin boson models. All quantities are given in atomic units.

Model	Description	ε	η	ω_c	β
A	Symmetric, high temperature	0	0.09	2.5	0.10
B	Symmetric, low temperature	0	0.09	2.5	5.00
C	Asymmetric, high temperature	1	0.10	1.0	0.25
D	Asymmetric, low temperature	1	0.10	2.0	5.00
E	Symmetric, high temperature	0	0.50	10.0	0.25

For the symmetric, high temperature model A (Fig. 5, left column), all methods are capable of reproducing the exact population dynamics. In fact, due to the high temperature nature of the model, the ring polymer dynamics requires only a single bead to reach convergence for all methods. Since the Wigner distribution converges to the classical Boltzmann distribution at high temperature, it is expected that the results be identical to the ring polymer dynamics.

For the asymmetric, high temperature model C (Fig. 5, second column), only eCMM is capable of capturing the exact result, with both FSSH and RPSH coming close. Both Ehrenfest variants perform worse, failing to capture the long time population. For this model, NRPMD returns the same result as (RP)Ehrenfest. Its single bead Wigner counterpart, LSCI, slightly overestimates the long-time population difference. For RPSH and ring polymer Ehrenfest, a single bead was sufficient to converge the dynamics, as expected for a high temperature model, where the methods become equivalent to their classical counterparts. However, the two ring polymer mapping methods (NRPMD, RPeCMM) differ significantly from their classical variants, this is likely due to the addition of extra electronic variables. The additional electronic variables mean that even when the ring polymer shrinks to a classical particle, the method is not equivalent to the single bead version.

With the low temperature models (Fig. 6), the difference between the ring polymer and Wigner methods is more pronounced than in the high temperature case (Fig. 5). Using the symmetric model B (Fig. 6, first column), RPSH and ring polymer Ehrenfest perform worse than their Wigner counterparts which are able to reproduce the exact dynamics. The RP variants exhibit dynamics with slower decoherence time. Similarly, eCMM also reproduces the exact dynamics, but its ring polymer version overestimates the amplitude of the Rabi oscillations. However, the decoherence time appears faster than in RPSH comparable to RPEhrenfest. In contrast, LSCI underestimates the amplitude while its ring polymer extension NRPMD matches the exact result. Across all methods, the oscillation amplitude is greater for the ring polymer variants.

Model D (Fig. 6, second column), the low temperature asymmetric model, is the most challenging for our approximate methods. As seen across all models, the exact dynamics

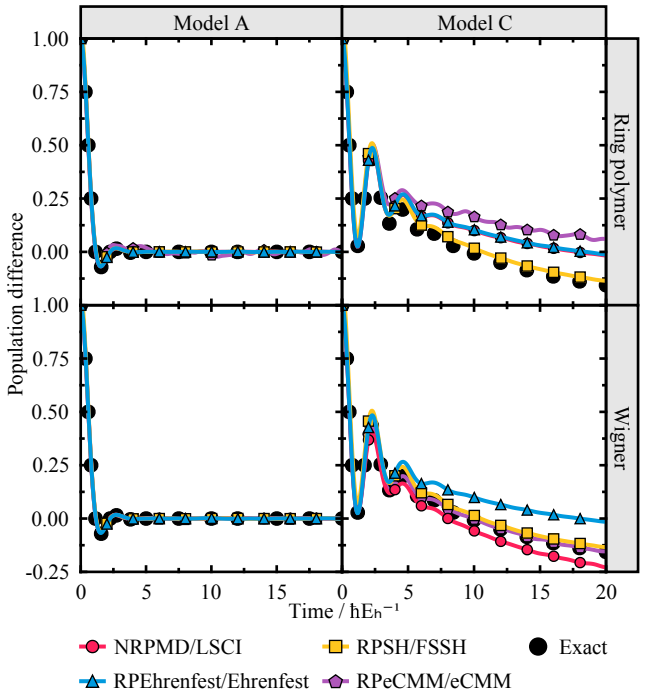

FIG. 5. Population dynamics for the high temperature models. The first column contains the results for model A, the second column for model C. The first row contains the results for the ring polymer methods (NRPMD, RPSH, RPEhrenfest, RPeCMM), and the second row contains the Wigner alternatives (LSCI, FSSH, Ehrenfest, eCMM). Refer to Table I for the model parameters.

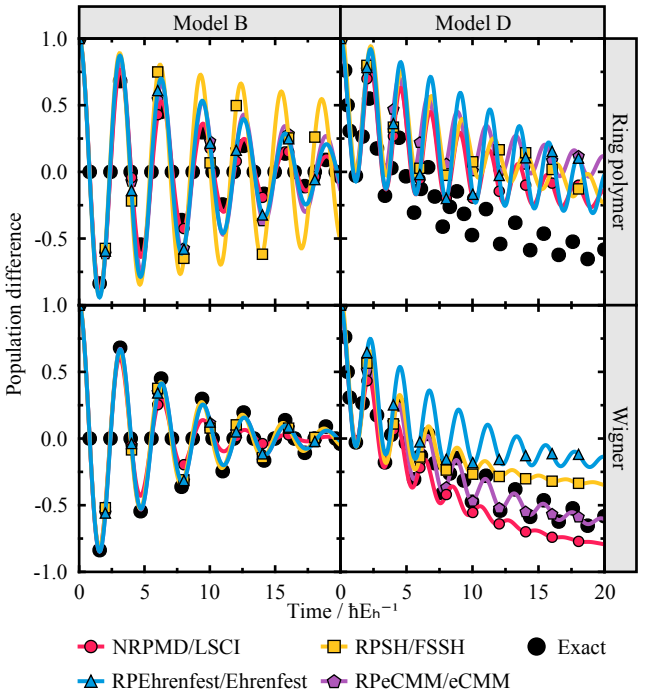

FIG. 6. Population dynamics for the low temperature models. The first column contains the results for model B, the second column for model D. Data is presented as in Fig. 5.

is captured comfortably at short times, but here, the long-time limit is out of reach for all methods except eCMM, which captures close to exact dynamics across all times. For the two low temperature models, all ring polymer method variants overestimate coherence during the dynamics.

The first four models have relatively weak system-bath coupling ($\eta \approx 0.1$), whereas model E (Fig. 7) has a larger value of $\eta = 0.5$. For this model, we see that none of the methods are capable of recovering the exact dynamics, although eCMM is the closest. Compared to the ring polymer methods, the short-time dynamics of the Wigner methods is more accurate, though a similar level of accuracy is observed at later time. The Wigner methods all underestimate the rate of population transfer, but the ring polymer methods (except for NRPMD) instead overestimate the decay.

Considering that the ring polymer modifications necessarily increase the computational expense of each method, significant improvements in the dynamics are required to justify their use. However, reviewing the results for all of the models suggests that the ring polymer dynamics cannot be reliably expected to immediately improve the population dynamics. For the low temperature models, improvement is seen only for NRPMD in model B. In all other cases, the ring polymer modification has no effect, or leads to small changes that do not directly improve the result.

Using the Wigner distribution reliably gives strong results across all five models, especially in the case of eCMM. Unfor-

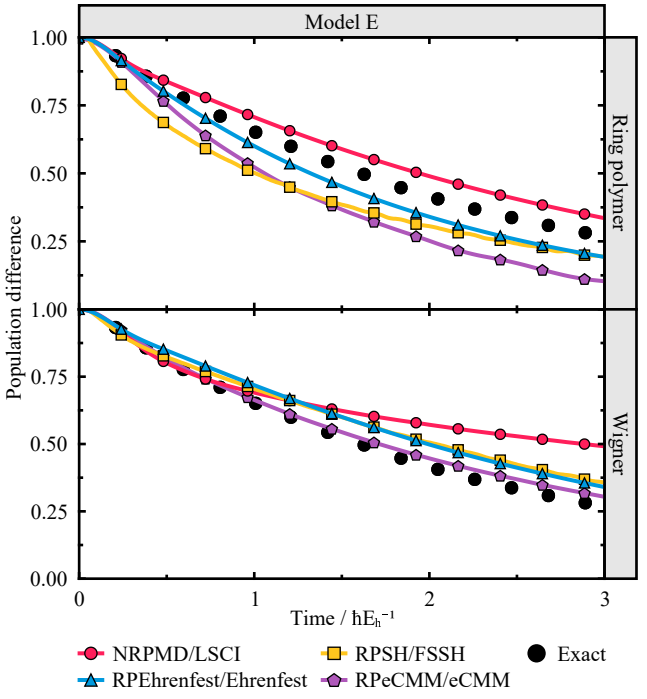

FIG. 7. Population dynamics for model E. Data is presented as in Fig. 5.

tunately, sampling the Wigner distribution for realistic atomistic systems is more challenging than the corresponding ring polymer distribution.[120,121] The high temperature results here (Fig. 5) suggest that the ring polymer methods are adequate substitutes in this regime, often requiring very few beads to obtain similar results. However, at low temperature, the ring polymer methods appear less capable of achieving the accuracy afforded by the Wigner methods.

In the benchmark study of Gao *et al.*[32] these models (A-D) were investigated using a collection of classical mapping methods using the Wigner distribution. The only methods we have used that also appear in their work are Ehrenfest and LSCI, where the results are identical. The most effective methods in their study are the LSC methods with modified population estimators that align very closely with the eCMM results presented here.

The `NQCDynamics.jl` code allows us to study all these method variants on equal footing and to systematically analyse the impact of different approximations and parameter choices. This will facilitate future method improvements to achieve accurate long-term dynamics in realistic atomistic systems.

V. EXAMPLE II: REACTIVE SCATTERING OF H_2 ON Ag(111)

In addition to the model Hamiltonian quantum dynamics of the first example, `NQCDynamics.jl` also allows us to investigate full dimensional atomistic systems. This example focuses on reproducing and augmenting the work of Zhang *et al.* [132] where they investigated the effect of hot-electrons during the reactive scattering of H_2 on a Ag(111) surface. In this system, nonadiabatic effects arise from the interaction of the molecular motion with the electronic bath of the metal substrate. Traditional adiabatic molecular dynamics evolving on a single potential energy surface ignores these effects, however, molecular dynamics with electronic friction (MDEF)[43] has been proposed as an alternative that attempts to approximately include these effects by introducing additional frictional forces due to the nonadiabatic interactions between adsorbate atoms and metal electrons. In the previous study,[132] the reactive scattering was investigated using machine learning to describe both the potential energy surface and the electronic friction.[132–135] After a brief overview of the different flavours of MDEF, using these same machine learning models[132,133] we investigate the dissociative chemisorption and state-to-state scattering of H_2 on Ag(111).

A. Molecular Dynamics with Electronic Friction

Molecular dynamics with electronic friction (MDEF) is a quasi-classical method that uses a Langevin equation to approximate weak nonadiabatic effects encountered at metal surfaces.[43,136] Within this theoretical framework, the coupling between the molecular degrees of freedom and the electron-hole pair excitations within the metal substrate is described by means of frictional and stochastic forces.[43,137] In doing so, the dynamical effects that arise due to the complex electronic structure of the metal are condensed into a single electronic friction coefficient or friction tensor in the case of multidimensional dynamics.[43,136,137]

During MDEF, the total nuclear force is given by:

$$\mathbf{M}\ddot{\mathbf{R}} = -\nabla V(\mathbf{R}) - \Lambda(\mathbf{R})\dot{\mathbf{R}} + \mathscr{R}(t). \tag{27}$$

The first term on the right hand side of Eq. 27 corresponds to the conservative force associated with the adiabatic potential energy surface. Adiabatic molecular dynamics simulations are governed solely by this unique ground-state force. The second term describes the energy losses produced by adsorbate-substrate interaction, with magnitude proportional to the friction tensor Λ and the particle velocity $\dot{\mathbf{R}}$. The final term is a temperature and friction-dependent stochastic random force that satisfies the fluctuation-dissipation relation.

Light-driven molecular dynamics processes can also be simulated using MDEF.[138–140] In this context, an external laser source is incorporated within the nuclear dynamics by modifying the temperature in the random force term as a function of time.[138,140] The temperature is chosen to model the electronic temperature which is often described by a simple 1D two-temperature model.[141] Further improvements of the dynamics can be also found by including the coupling with bulk phonons through a generalised Langevin oscillator model, but the relevance of the phonon dissipation effects on the final nuclear dynamics depends on the studied system.[138,140,142] In the example case of H_2 scattering dynamics on a Ag(111) surface, we will explicitly ignore phonon dissipation effects.

MDEF simulations are usually performed using one of two methods for calculating the electronic friction tensor. The following sections describe the two choices and discuss the details of their implementation.

1. *Local density friction approximation (LDFA)*

The simplest version of MDEF uses a *local density friction approximation* (LDFA) where a single friction coefficient is associated with each adsorbate atom based on the local electron density of the bare metal substrate.[136,138] During the dynamics, the local density is computed as a function of each adsorbate coordinate R_i, which is then converted to a friction coefficient via a fitting of pre-computed values. Analytic expressions are commonly used to fit the pre-computed values[138,143] but our implementation uses a cubic spline to fit the LDFA values reported by Gerrits and Meyer.[144]

With this, the friction tensor becomes diagonal:

$$\Lambda_{\mathrm{LDFA}}(\mathbf{R}) = \mathrm{diag}(\eta(R_1), \ldots, \eta(R_N)) \tag{28}$$

In this way, the fitting function $\eta(R_i)$ allows us to connect any point visited by the adsorbate atoms with a single electronic friction coefficient used to drive the nuclear dynamics.

LDFA friction coefficients have previously been widely used to describe surface processes such as atomic and molecular diffusion and laser-driven dynamics.[136,138,145]

2. Orbital-dependent electronic friction (ODF)

A more general formulation of the electronic friction tensor stems from time-dependent perturbation theory based on the Kohn-Sham Density Functional Theory wave functions. This has previously been coined orbital-dependent electronic friction or ODF.[137,146–149] ODF provides a coordinate-dependent tensorial representation of electronic friction that has been found to be more suitable to describe reactive dynamics of molecules at metal surfaces.[45,134,147] The ODF representation of the electronic friction tensor (EFT) captures the intrinsic mode anisotropy and internal coupling between different degrees of freedom in the molecule.[132,137,146] An efficient *ab initio* all electron electronic structure implementation of MDEF-ODF based on Kohn-Sham density functional theory was previously reported by Maurer *et al.*.[132,137,150]

For ODF, the friction tensor Λ is dense and positive semi-definite. Each element Λ_{ij} corresponds to a relaxation rate due to electron-nuclear coupling along the Cartesian coordinate i due to motion along the j direction. In general, a more convenient representation in normal mode coordinates is often used to compute the associated vibrational lifetimes or relaxation rate components.[132,137,146]

3. Machine learning neural network models

Performing MDEF simulations can be challenging due to the prohibitive computational cost associated with *ab initio* electronic structure calculations. This difficulty can be circumvented by employing machine learning techniques.[132,135,140,151] Zhang *et al*. have recently reported an efficient machine learning model based on a permutation invariant polynomial neural network, which can accurately reproduce both potential energy surfaces and electronic friction tensors at reduced computational cost.[132,151] A new family of machine learning models are currently being developed to produce accurate potential energy surfaces and other physical properties.[133,140,142,151] In the present work, we have used the machine learning model of Jiang and Guo [133] and the previously reported six-dimensional energy landscape and EFT model to compute the reactive scattering of H_2 on a frozen Ag(111) surface.[132,135] The efficiency of the models allows us to perform up to 75,000 trajectories at LDFA and ODF level for each initial condition.

B. Simulation details

As in the original paper,[132] the initial conditions are sampled from a nonequilibrium semiclassically quantised distribution in a specific ro-vibrational quantum state. This distribution was generated using EBK quantisation implemented in the `QuantisedDiatomic` submodule. The initial distribution used for all simulations contained 7.5×10^4 nuclear positions and velocities consistent with the ro-vibrational state $(v = 2, J = 0)$. The Ag metallic slab is modelled with a primitive p(2×2) unit cell with 4 atomic layers.

All the simulations were run with a 420 fs time limit with a time step of 0.1 fs, with the molecule initially located 8 Å away from the metal surface. The lateral position and orientation of the H_2 molecule were uniformly distributed within the simulation cell. The neural network models used for the potential energy surface and the EFT make a frozen surface assumption such that the metal surface is fixed with its outermost layer at $z = 0$ Å. During the simulation, if the molecule scatters to vertical distance larger than 8.1 Å from the metal surface, the outcome is considered a scattering event. If the H_2 bond length exceeds 2.5 Å, the outcome is dissociative chemisorption. When either of these conditions are satisfied, the simulation is terminated. Two specific state-to-state transitions were analysed starting from the initial ro-vibrational state $(v = 2, J = 0)$ with final states $(v = 1, J = 0)$ and $(v = 0, J = 0)$. State-to-state transition probabilities were obtained via the binning method, where the final image from the scattering trajectories was re-quantised following the initial EBK procedure.[45,132]

C. Results and discussion

The scattering probability results obtained with MD, LDFA and ODF as a function of incident translational energy E_{trans} (Fig. 8, top panel) almost perfectly reproduce the previously reported values[132] that were calculated with a modified version of the VENUS code.[152,153] The new calculations consider a wider range of translational energies up to 1.4 eV.

In addition to the dissociation probability, the two lower panels of Fig. 8 show the vibrational de-excitation probabilities for the reactive scattering of H_2 on Ag(111). The first transition shown in the middle panel experiences a peak at 0.4 eV for all methods, these results reproduce those in the original paper.[132] The second transition considered is an extension of the original data, concerning the transition to the rovibrational ground state. In this case, the highest de-excitation probability is detected when the translational energy is 0.6 eV for all the methods considered. When comparing the results across each of the methods, qualitatively similar trends are seen. However, the addition of friction appears to slightly increase the de-excitation probability and reduce the dissociation probability. This result is seen for both LDFA and ODF. For this system, while the inclusion of friction is important to capture the dissipative dynamics, the differences between ODF and LDFA in predicting inelastic vibrational state-to-state scattering are more subtle.

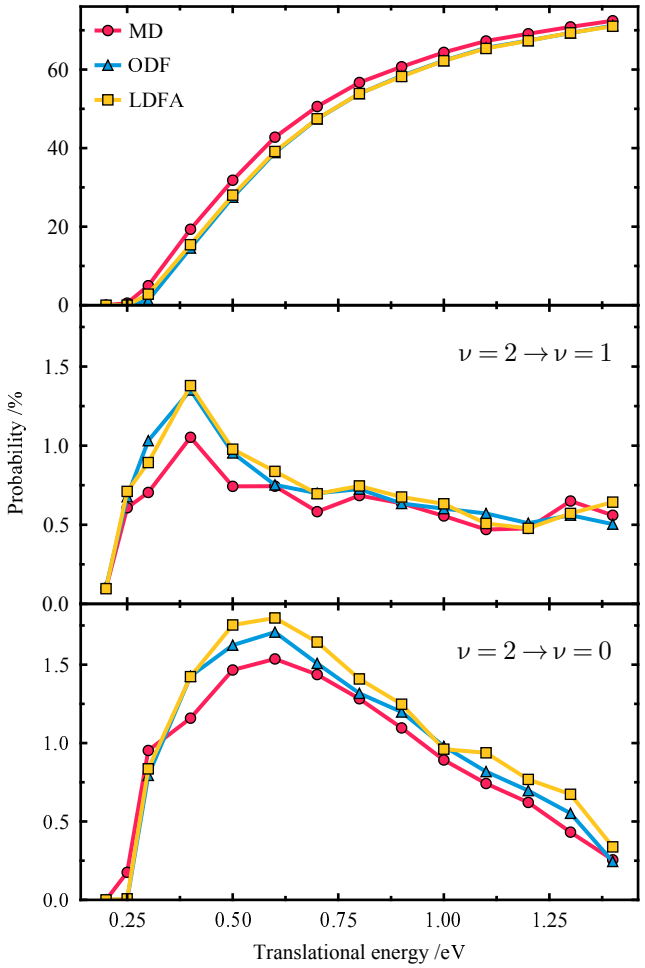

FIG. 8. Dissociative chemisorption (top panel) and vibrational de-excitation probabilities (middle and botton panels) for the scattering of H_2 on Ag(111) computed with MD, LDFA, and ODF. The middle and bottom panels show the state-to-state vibrational de-excitation probabilities computed for $(v = 2, J = 0) \rightarrow (v = 1, J = 0)$ and $(v = 2, J = 0) \rightarrow (v = 0, J = 0)$ transitions, respectively.

VI. CONCLUSIONS AND OUTLOOK

In this work, we have introduced the `NQCDynamics.jl` package for performing and developing semiclassical and mixed-quantum classical methods for nonadiabatic dynamics. It is written in *Julia*, an emerging language that promises high performance alongside an approachable development experience. The package provides a set of established and developing methods, alongside a framework for further additions. The code interfaces to a comprehensive and extendable differential equations solver, `DifferentialEquations.jl`, and thereby externalises the general integration routines. The package, via `NQCModels.jl`, interfaces to a wide variety of analytical models, *ab initio* calculators (through `ASE`) and high-dimensional machine learning models of condensed phase systems.

To demonstrate the production and prototyping capabilities of the package, we have provided two example studies: the first investigates the population dynamics of four spin-boson models with a variety of mixed quantum-classical and semiclassical methods in different temperature regimes and for different state splittings. Using the code framework, we implement several *ad hoc* extensions of existing methods, for example a ring polymer Ehrenfest method and a ring-polymer extension to the eCMM method, and we analyse their performance against other methods. In the second example, we study nonadiabatic reactive state-to-state scattering of molecular hydrogen at a Ag(111) surface as an example of realistic atomistic dynamics based on machine learning representations.

The package will be actively maintained and we will continue to expand its library of methods, models and functionality. We ourselves plan to significantly extend its capabilities to perform approximate nonadiabatic quantum dynamics in condensed phase and we invite others to contribute methods and use cases. The code is open source and presents extensive online documentation. Moving forward, we hope that the package will gain recognition within the community and become a useful resource for the development of new nonadiabatic dynamics methods. In particular, we want to encourage its use to produce reference implementations of new dynamics methods, which can be released alongside the relevant publications. This will improve code availability and method reproducibility and is an important first step to establish general benchmarks for approximate quantum dynamics methods in condensed phase.

VII. ACKNOWLEDGEMENT

This work was financially supported by The Leverhulme Trust (RPG-2019-078) and the UKRI Future Leaders Fellowship programme (MR/S016023/1) (R.J.M.). Financial support from the Austrian Science Fund (FWF) [J 4522-N] (J.W.), and the WIRL-COFUND fellowship scheme at the University of Warwick (S.M.J.), under the Marie Skłodowska Curie Actions COFUND program (grant agreement number 713548) is acknowledged. High performance computing resources were provided via the Scientific Computing Research Technology Platform of the University of Warwick, the EPSRC-funded Materials Chemistry Consortium for the ARCHER2 UK National Supercomputing Service (EP/R029431/1), and the EPSRC-funded HPC Midlands+ computing centre (EP/P020232/1). We thank Prof. Bin Jiang (USTC, Hefei) for providing us with the neural network model for H_2 scattering on Ag(111).

VIII. REFERENCE

[1] J. Flick, N. Rivera, and P. Narang, "Strong light-matter coupling in quantum chemistry and quantum photonics," Nanophotonics **7**, 1479–1501 (2018).

[2] N. S. Mueller, Y. Okamura, B. G. M. Vieira, S. Juergensen, H. Lange, E. B. Barros, F. Schulz, and S. Reich, "Deep strong light–matter coupling in plasmonic nanoparticle crystals," Nature **583**, 780–784 (2020).

[3] J. Taylor, H. Guo, and J. Wang, "Ab initio modeling of quantum transport properties of molecular electronic devices," Physical Review B **63**, 245407 (2001).

[4] H. Wang and M. Thoss, "Multilayer formulation of the multiconfiguration time-dependent Hartree theory," J. Chem. Phys. **119**, 1289–1299 (2003).

[5] H.-D. Meyer, F. Gatti, and G. A. Worth, *Multidimensional Quantum Dynamics: MCTDH Theory and Applications* (John Wiley & Sons, 2009).

[6] G. Richings, I. Polyak, K. Spinlove, G. Worth, I. Burghardt, and B. Lasorne, "Quantum dynamics simulations using Gaussian wavepackets: The vMCG method," Int. Rev. Phys. Chem. **34**, 269–308 (2015).

[7] A. McLachlan, "A variational solution of the time-dependent Schrodinger equation," Mol. Phys. **8**, 39–44 (1964).

[8] J. E. Subotnik, "Augmented Ehrenfest dynamics yields a rate for surface hopping," J. Chem. Phys. **132**, 134112 (2010).

[9] S. Choi and J. Vaníček, "High-order geometric integrators for representation-free Ehrenfest dynamics," J. Chem. Phys. **155**, 124104 (2021).

[10] S. Hammes-Schiffer and J. C. Tully, "Proton transfer in solution: Molecular dynamics with quantum transitions," J. Chem. Phys. **101**, 4657–4667 (1994).

[11] S. Hammes-Schiffer and J. C. Tully, "Nonadiabatic transition state theory and multiple potential energy surface molecular dynamics of infrequent events," J. Chem. Phys. **103**, 8528–8537 (1995).

[12] J. E. Subotnik, A. Jain, B. Landry, A. Petit, W. Ouyang, and N. Bellonzi, "Understanding the Surface Hopping View of Electronic Transitions and Decoherence," Annu. Rev. Phys. Chem. **67**, 387–417 (2016).

[13] C. C. Martens, "Surface Hopping without Momentum Jumps: A Quantum-Trajectory-Based Approach to Nonadiabatic Dynamics," J. Phys. Chem. A **123**, 1110–1128 (2019).

[14] P. Shushkov, R. Li, and J. C. Tully, "Ring polymer molecular dynamics with surface hopping," J. Chem. Phys. **137**, 22A549 (2012).

[15] F. A. Shakib and P. Huo, "Ring Polymer Surface Hopping: Incorporating Nuclear Quantum Effects into Nonadiabatic Molecular Dynamics Simulations," J. Phys. Chem. Lett. **8**, 3073–3080 (2017).

[16] S. M. Parker and C. J. Schiltz, "Surface hopping with cumulative probabilities: Even sampling and improved reproducibility," J. Chem. Phys. **153**, 174109 (2020).

[17] C. C. Martens and J.-Y. Fang, "Semiclassical-limit molecular dynamics on multiple electronic surfaces," J. Chem. Phys. **106**, 4918–4930 (1997).

[18] R. Kapral and G. Ciccotti, "Mixed quantum-classical dynamics," J. Chem. Phys. **110**, 8919–8929 (1999).

[19] S. Nielsen, R. Kapral, and G. Ciccotti, "Statistical mechanics of quantum-classical systems," J. Chem. Phys. **115**, 5805–5815 (2001).

[20] R. Kapral, "PROGRESS IN THE THEORY OF MIXED QUANTUM-CLASSICAL DYNAMICS," Annu. Rev. Phys. Chem. **57**, 129–157 (2006).

[21] R. Lambert and N. Makri, "Quantum-classical path integral. II. Numerical methodology," J. Chem. Phys. **137**, 22A553 (2012).

[22] R. Lambert and N. Makri, "Quantum-classical path integral. I. Classical memory and weak quantum nonlocality," J. Chem. Phys. **137**, 22A552 (2012).

[23] H.-D. Meyer and W. H. Miller, "A classical analog for electronic degrees of freedom in nonadiabatic collision processes," J. Chem. Phys. **70**, 3214–3223 (1979).

[24] G. Stock and M. Thoss, "Semiclassical Description of Nonadiabatic Quantum Dynamics," Phys. Rev. Lett. **78**, 578–581 (1997).

[25] M. Thoss and G. Stock, "Mapping approach to the semiclassical description of nonadiabatic quantum dynamics," Phys. Rev. A **59**, 64–79 (1999).

[26] J. Liu, "A unified theoretical framework for mapping models for the multistate Hamiltonian," J. Chem. Phys. **145**, 204105 (2016).

[27] S. J. Cotton, R. Liang, and W. H. Miller, "On the adiabatic representation of Meyer-Miller electronic-nuclear dynamics," J. Chem. Phys. **147**, 064112 (2017).

[28] M. S. Church, T. J. H. Hele, G. S. Ezra, and N. Ananth, "Nonadiabatic semiclassical dynamics in the mixed quantum-classical initial value representation," J. Chem. Phys. **148**, 102326 (2017).

[29] M. A. C. Saller, A. Kelly, and J. O. Richardson, "On the identity of the identity operator in nonadiabatic linearized semiclassical dynamics," J. Chem. Phys. **150**, 071101 (2019).

[30] X. He and J. Liu, "A new perspective for nonadiabatic dynamics with phase space mapping models," J. Chem. Phys. **151**, 024105 (2019).

[31] J. Liu, X. He, and B. Wu, "Unified Formulation of Phase Space Mapping Approaches for Nonadiabatic Quantum Dynamics," Acc. Chem. Res. **54**, 4215–4228 (2021).

[32] X. Gao, M. A. C. Saller, Y. Liu, A. Kelly, J. O. Richardson, and E. Geva, "Benchmarking Quasiclassical Mapping Hamiltonian Methods for Simulating Electronically Nonadiabatic Molecular Dynamics," J. Chem. Theory Comput. **16**, 2883–2895 (2020).

[33] M. A. C. Saller, J. E. Runeson, and J. O. Richardson, "Path-integral approaches to non-adiabatic dynamics," in *Quantum Chemistry and Dynamics of Excited States* (John Wiley & Sons, Ltd, 2020) Chap. 20, pp. 629–653.

[34] M. A. C. Saller, A. Kelly, and E. Geva, "Benchmarking Quasiclassical Mapping Hamiltonian Methods for Simulating Cavity-Modified Molecular Dynamics," J. Phys. Chem. Lett. **12**, 3163–3170 (2021).

[35] J. O. Richardson and M. Thoss, "Communication: Nonadiabatic ring-polymer molecular dynamics," J. Chem. Phys. **139**, 031102 (2013).

[36] J. O. Richardson, P. Meyer, M.-O. Pleinert, and M. Thoss, "An analysis of nonadiabatic ring-polymer molecular dynamics and its application to vibronic spectra," Chem. Phys. **482**, 124–134 (2017).

[37] S. N. Chowdhury and P. Huo, "State dependent ring polymer molecular dynamics for investigating excited nonadiabatic dynamics," J. Chem. Phys. **150**, 244102 (2019).

[38] S. N. Chowdhury and P. Huo, "Non-adiabatic Matsubara dynamics and non-adiabatic ring-polymer molecular dynamics," J. Chem. Phys. **154**, 124124 (2021).

[39] N. Shenvi, S. Roy, and J. C. Tully, "Nonadiabatic dynamics at metal surfaces: Independent-electron surface hopping," J. Chem. Phys. **130**, 174107 (2009).

[40] W. Ouyang, W. Dou, and J. E. Subotnik, "Surface hopping with a manifold of electronic states. I. Incorporating surface-leaking to capture lifetimes," J. Chem. Phys. **142**, 084109 (2015).

[41] W. Dou, A. Nitzan, and J. E. Subotnik, "Surface hopping with a manifold of electronic states. III. Transients, broadening, and the Marcus picture," J. Chem. Phys. **142**, 234106 (2015).

[42] W. Dou, A. Nitzan, and J. E. Subotnik, "Surface hopping with a manifold of electronic states. II. Application to the many-body Anderson-Holstein model," J. Chem. Phys. **142**, 084110 (2015).

[43] M. Head-Gordon and J. C. Tully, "Molecular dynamics with electronic frictions," J. Chem. Phys. **103**, 10137–10145 (1995).

[44] R. J. Maurer, M. Askerka, V. S. Batista, and J. C. Tully, "Ab initio tensorial electronic friction for molecules on metal surfaces: Nonadiabatic vibrational relaxation," Phys. Rev. B **94**, 115432 (2016).

[45] C. L. Box, Y. Zhang, R. Yin, B. Jiang, and R. J. Maurer, "Determining the Effect of Hot Electron Dissipation on Molecular Scattering Experiments at Metal Surfaces," JACS Au **1**, 164–173 (2021).

[46] W. Dou, G. Miao, and J. E. Subotnik, "Born-Oppenheimer Dynamics, Electronic Friction, and the Inclusion of Electron-Electron Interactions," Phys. Rev. Lett. **119**, 046001 (2017).

[47] R. Martinazzo and I. Burghardt, "Quantum dynamics with electronic friction," (2021), arXiv:2108.02622 [quant-ph].

[48] G. Tao, "Nonadiabatic dynamics of hydrogen diffusion on cu(001): Classical mapping model with multistate projection window in real space," ChemPhysChem **20**, 2127–2135 (2019).

[49] J. C. Tully, "Molecular dynamics with electronic transitions," J. Chem. Phys. **93**, 1061–1071 (1990).

[50] M. Barbatti, M. Ruckenbauer, F. Plasser, J. Pittner, G. Granucci, M. Persico, and H. Lischka, "Newton-X: A surface-hopping program for nonadiabatic molecular dynamics," WIREs Comput. Mol. Sci. **4**, 26–33 (2014).

[51] M. Barbatti, G. Granucci, M. Ruckenbauer, R. Crespo-Otero, J. Pittner, M. Persico, and H. Lischka, "NEWTON-X: A package for Newtonian Dynamics Close to the Crossing Seam," `www.newtonx.org` (2018).

[52] S. Mai, P. Marquetand, and L. González, "Nonadiabatic dynamics: The SHARC approach," WIREs Comput. Mol. Sci. **8**, e1370 (2018).

[53] S. Mai, M. Richter, M. Ruckenbauer, M. Oppel, P. Marquetand, and L. González, "SHARC2.0: Surface Hopping Including ARbitrary Cou-

plings – Program Package for Non-Adiabatic Dynamics," sharc-md.org (2018).
[54] V. Kapil, M. Rossi, O. Marsalek, R. Petraglia, Y. Litman, T. Spura, B. Cheng, A. Cuzzocrea, R. H. Meißner, D. M. Wilkins, B. A. Helfrecht, P. Juda, S. P. Bienvenue, W. Fang, J. Kessler, I. Poltavsky, S. Vandenbrande, J. Wieme, C. Corminboeuf, T. D. Kühne, D. E. Manolopoulos, T. E. Markland, J. O. Richardson, A. Tkatchenko, G. A. Tribello, V. Van Speybroeck, and M. Ceriotti, "I-PI 2.0: A universal force engine for advanced molecular simulations," Comput. Phys. Commun. **236**, 214–223 (2019).
[55] S. C. Althorpe, W. Barford, J. Blumberger, C. Bungey, I. Burghardt, A. Datta, S. Ghosh, S. Giannini, T. Grünbaum, S. Habershon, S. Hammes-Schiffer, S. Hay, S. Iyengar, G. Jones, A. Kelly, K. Komarova, J. Lawrence, Y. Litman, J. Mannouch, D. Manolopoulos, C. Martens, R. J. Maurer, M. Melander, M. Rossi, K. Sakaushi, M. Saller, A. Schile, S. Sturniolo, G. Trenins, and G. Worth, "Emerging opportunities and future directions: General discussion," Faraday Discuss. **221**, 564–581 (2020).
[56] J. Westermayr, M. Gastegger, K. T. Schütt, and R. J. Maurer, "Perspective on integrating machine learning into computational chemistry and materials science," J. Chem. Phys. **154**, 230903 (2021).
[57] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," SIAM Rev. **59**, 65–98 (2017).
[58] M. Lubin and I. Dunning, "Computing in Operations Research Using Julia," INFORMS J. Comput. **27**, 238–248 (2015).
[59] T. Koolen and R. Deits, "Julia for robotics: Simulation and real-time control in a high-level programming language," in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE, Montreal, QC, Canada, 2019) pp. 604–611.
[60] J. Bezanson, J. Chen, B. Chung, S. Karpinski, V. B. Shah, J. Vitek, and L. Zoubritzky, "Julia: Dynamism and performance reconciled by design," Proc. ACM Program. Lang. **2**, 1–23 (2018).
[61] M. F. Herbst, A. Levitt, and E. Cancès, "DFTK: A Julian approach for simulating electrons in solids," Proc. JuliaCon Conf. **3**, 69 (2021).
[62] G. J. R. Aroeira, M. M. Davis, J. M. Turney, and H. F. Schaefer, "Fermi.jl: A Modern Design for Quantum Chemistry," J. Chem. Theory Comput. **18**, 677–686 (2022).
[63] M. F. Herbst and A. Levitt, "Black-box inhomogeneous preconditioning for self-consistent field iterations in density functional theory," J. Phys. Condens. Matter **33**, 085503 (2021).
[64] M. F. Herbst and A. Levitt, "A robust and efficient line search for self-consistent field iterations," (2022), arXiv:2109.14018 [cond-mat.mtrl-sci].
[65] "Julia: Composite types," https://docs.julialang.org/en/v1/manual/types/#Composite-Types (), accessed: 2021-21-02.
[66] M. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation* (Oxford University Press, 2010).
[67] M. Ceriotti, M. Parrinello, T. E. Markland, and D. E. Manolopoulos, "Efficient stochastic thermostatting of path integral molecular dynamics," J. Chem. Phys. **133**, 124104 (2010).
[68] R. Korol, N. Bou-Rabee, and T. F. Miller, "Cayley modification for strongly stable path-integral and ring-polymer molecular dynamics," J. Chem. Phys. **151**, 124103 (2019).
[69] R. Korol, J. L. Rosa-Raíces, N. Bou-Rabee, and T. F. Miller, "Dimension-free path-integral molecular dynamics without preconditioning," J. Chem. Phys. **152**, 104102 (2020).
[70] A. J. Larkoski, D. G. Ellis, and L. J. Curtis, "Numerical implementation of Einstein-Brillouin-Keller quantization for arbitrary potentials," Am. J. Phys. **74**, 572–577 (2006).
[71] M. Besançon, T. Papamarkou, D. Anthoff, A. Arslan, S. Byrne, D. Lin, and J. Pearson, "Distributions.jl: Definition and modeling of probability distributions in the juliastats ecosystem," J. Stat. Softw. **98**, 1–30 (2021).
[72] B. Leimkuhler and C. Matthews, "Rational Construction of Stochastic Numerical Methods for Molecular Sampling," Applied Mathematics Research eXpress **2013**, 34–56 (2013).
[73] B. Leimkuhler and C. Matthews, "Robust and efficient configurational molecular sampling via Langevin dynamics," The Journal of Chemical Physics **138**, 174102 (2013).
[74] C. Rackauckas and Q. Nie, "DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia," J. Open Res. Softw. **5**, 15 (2017).
[75] I. R. Craig and D. E. Manolopoulos, "Quantum statistics and classical mechanics: Real time correlation functions from ring polymer molecular dynamics," J. Chem. Phys. **121**, 3368–3373 (2004).
[76] S. Habershon, D. E. Manolopoulos, T. E. Markland, and T. F. Miller, "Ring-Polymer Molecular Dynamics: Quantum Effects in Chemical Dynamics from Classical Trajectories in an Extended Phase Space," Annu. Rev. Phys. Chem. **64**, 387–413 (2013).
[77] T. Yoshikawa and T. Takayanagi, "Application of ring-polymer molecular dynamics to electronically nonadiabatic excess electron dynamics in water clusters: Importance of nuclear quantum effects," Chem. Phys. Lett. **564**, 1–5 (2013).
[78] X. He, Z. Gong, B. Wu, and J. Liu, "Negative Zero-Point-Energy Parameter in the Meyer–Miller Mapping Model for Nonadiabatic Dynamics," J. Phys. Chem. Lett. **12**, 2496–2501 (2021).
[79] J. E. Runeson and J. O. Richardson, "Spin-mapping approach for nonadiabatic molecular dynamics," J. Chem. Phys. **151**, 044119 (2019).
[80] J. E. Runeson and J. O. Richardson, "Generalized spin mapping for quantum-classical dynamics," J. Chem. Phys. **152**, 084110 (2020).
[81] E. A. Coronado, J. Xing, and W. H. Miller, "Ultrafast non-adiabatic dynamics of systems with multiple surface crossings: A test of the Meyer–Miller Hamiltonian with semiclassical initial value representation methods," Chem. Phys. Lett. **349**, 521–529 (2001).
[82] A. Nitzan, *Chemical Dynamics in Condensed Phases: Relaxation, Transfer and Reactions in Condensed Molecular Systems*, Oxford Graduate Texts (Oxford University Press, Oxford ; New York, 2006).
[83] A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment-a Python library for working with atoms," J. Phys. Condens. Matter **29**, 273002 (2017).
[84] "JuLIP.jl," https://github.com/JuliaMolSim/JuLIP.jl (), accessed: 2021-25-02.
[85] G. Csányi, S. Winfield, J. R. Kermode, A. De Vita, A. Comisso, N. Bernstein, and M. C. Payne, "Expressive programming for computational physics in fortran 95+," IoP Comput. Phys. Newsletter , Spring 2007 (2007).
[86] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**, 136403 (2010).
[87] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "SchNet – a deep learning architecture for molecules and materials," J. Chem. Phys. **148**, 241722 (2018).
[88] K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller, "Schnetpack: A deep learning toolbox for atomistic systems," J. Chem. Theory Comput. **15**, 448–455 (2019).
[89] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, "Gaussian Process Regression for Materials and Molecules," Chem. Rev. **121**, 10073–10141 (2021).
[90] J. Li, P. Reiser, B. R. Boswell, A. Eberhard, N. Z. Burns, P. Friederich, and S. A. Lopez, "Automatic discovery of photoisomerization mechanisms with nanosecond machine learning photodynamics simulations," Chem. Sci. **12**, 5302–5314 (2021).
[91] J. Westermayr, M. Gastegger, M. F. S. J. Menger, S. Mai, L. González, and P. Marquetand, "Machine learning enables long time scale molecular photodynamics simulations," Chem. Sci. **10**, 8100–8107 (2019).
[92] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer, "Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions," Nat. Commun. **10**, 5024 (2019).
[93] L. Zhang, B. Onat, G. Dusson, G. Anand, R. J. Maurer, C. Ortner, and J. R. Kermode, "Equivariant analytical mapping of first principles hamiltonians to accurate and transferable materials models," (2022), arXiv:2111.13736 [cond-mat.mtrl-sci].
[94] M. Innes, "Flux: Elegant machine learning with Julia," J. Open Source. Softw. **3**, 602 (2018).

[95] A. Blaom, F. Kiraly, T. Lienart, Y. Simillides, D. Arenas, and S. Vollmer, "MLJ: A Julia package for composable machine learning," J. Open Source Softw. **5**, 2704 (2020).

[96] K. Gao, G. Mei, F. Piccialli, S. Cuomo, J. Tu, and Z. Huo, "Julia language in machine learning: Algorithms, applications, and open issues," Comput. Sci. Rev. **37**, 100254 (2020).

[97] R. Drautz, "Atomic cluster expansion for accurate and transferable interatomic potentials," Phys. Rev. B **99**, 014104 (2019).

[98] G. Dusson, M. Bachmayr, G. Csanyi, R. Drautz, S. Etter, C. van der Oord, and C. Ortner, "Atomic cluster expansion: Completeness, efficiency and stability," (2021), arXiv:1911.03550 [math.NA].

[99] B. Jiang and H. Guo, "Six-dimensional quantum dynamics for dissociative chemisorption of H_2 and D_2 on Ag(111) on a permutation invariant potential energy surface," Phys. Chem. Chem. Phys. **16**, 24704–24715 (2014).

[100] "Julia: Calling C and Fortran code," `https://docs.julialang.org/en/v1/manual/calling-c-and-fortran-code/` (), accessed: 2021-20-01.

[101] "Julia: Standalone `using` and `import`," `https://docs.julialang.org/en/v1/manual/modules/#Standalone-using-and-import` (), accessed: 2021-20-01.

[102] "Unitful.jl," `https://github.com/PainterQubits/Unitful.jl`, accessed: 2021-18-02.

[103] "NQCDynamics.jl," `https://github.com/NQCD/NQCDynamics.jl` (), accessed: 2021-20-01.

[104] "NQCDynamics.jl documentation," `https://nqcd.github.io/NQCDynamics.jl/stable/` (), accessed: 2021-20-01.

[105] B. J. Berne and G. D. Harp, "On the Calculation of Time Correlation Functions," in *Advances in Chemical Physics* (John Wiley & Sons, Ltd, 1970) pp. 63–227.

[106] T. J. H. Hele, "Thermal quantum time-correlation functions from classical-like dynamics," Mol. Phys. **115**, 1435–1462 (2017).

[107] S. Bonella, D. Montemayor, and D. F. Coker, "Linearized path integral approach for calculating nonadiabatic time correlation functions," Proc. Natl. Acad. Sci. U. S. A. **102**, 6715 (2005).

[108] J. C. Tully, "Molecular dynamics with electronic transitions," J. Chem. Phys. **93**, 1061–1071 (1990).

[109] J. C. Tully, "Nonadiabatic molecular dynamics," Int. J. Quantum Chem. **40**, 299–309 (1991).

[110] J. C. Tully, "Mixed quantum–classical dynamics," Faraday Discuss. **110**, 407–419 (1998).

[111] H.-T. Chen and D. R. Reichman, "On the accuracy of surface hopping dynamics in condensed phase non-adiabatic problems," J. Chem. Phys. **144**, 094104 (2016).

[112] W. Dou and J. E. Subotnik, "Nonadiabatic Molecular Dynamics at Metal Surfaces," J. Phys. Chem. A **124**, 757–771 (2020).

[113] Z. Jin and J. E. Subotnik, "Nonadiabatic dynamics at metal surfaces: Fewest switches surface hopping with electronic relaxation," J. Chem. Theory Comput. **17**, 614–626 (2021).

[114] C. Zener, "Non-adiabatic crossing of energy levels," Proc. Roy. Soc. Lond. A **137**, 696–701 (1932).

[115] C. Zhu, H. Kamisaka, and H. Nakamura, "New Implementation of the Trajectory Surface Hopping Method with Use of the Zhu–Nakamura Theory. II. Application to the Charge Transfer Processes in the 3D DH_2+ System," J. Chem. Phys. **116**, 3234–3247 (2002).

[116] A. Sergi and R. Kapral, "Quantum-classical limit of quantum correlation functions," J. Chem. Phys. **121**, 7565 (2004).

[117] C.-Y. Hsieh and R. Kapral, "Correlation Functions in Open Quantum-Classical Systems," Entropy **16**, 200–220 (2013).

[118] B. R. Landry, M. J. Falk, and J. E. Subotnik, "Communication: The correct interpretation of surface hopping trajectories: How to calculate electronic properties," J. Chem. Phys. **139**, 211101 (2013).

[119] H.-T. Chen and D. R. Reichman, "On the accuracy of surface hopping dynamics in condensed phase non-adiabatic problems," J. Chem. Phys. **144**, 094104 (2016).

[120] J. Liu and W. H. Miller, "A simple model for the treatment of imaginary frequencies in chemical reaction rates and molecular liquids," J. Chem. Phys. **131**, 074113 (2009).

[121] F. X. Vázquez, S. Talapatra, and E. Geva, "Vibrational Energy Relaxation in Liquid HCl and DCl via the Linearized Semiclassical Method: Electrostriction versus Quantum Delocalization," J. Phys. Chem. A **115**, 9775–9781 (2011).

[122] S. Habershon and D. E. Manolopoulos, "Zero point energy leakage in condensed phase dynamics: An assessment of quantum simulation methods for liquid water," J. Chem. Phys. **131**, 244518 (2009).

[123] X. He, B. Wu, Z. Gong, and J. Liu, "Commutator Matrix in Phase Space Mapping Models for Nonadiabatic Quantum Dynamics," J. Phys. Chem. A **125**, 6845–6863 (2021).

[124] M. Thoss, H. Wang, and W. H. Miller, "Self-consistent hybrid approach for complex systems: Application to the spin-boson model with Debye spectral density," J. Chem. Phys. **115**, 2991–3005 (2001).

[125] H. Wang, M. Thoss, and W. H. Miller, "Systematic convergence in the dynamical hybrid approach for complex systems: A numerically exact methodology," J. Chem. Phys. **115**, 2979–2990 (2001).

[126] T. C. Berkelbach, D. R. Reichman, and T. E. Markland, "Reduced density matrix hybrid approach: An efficient and accurate method for adiabatic and non-adiabatic quantum dynamics," J. Chem. Phys. **136**, 034113 (2012).

[127] N. Rekik, C.-Y. Hsieh, H. Freedman, and G. Hanna, "A mixed quantum-classical Liouville study of the population dynamics in a model photo-induced condensed phase electron transfer reaction," J. Chem. Phys. **138**, 144106 (2013).

[128] S. Habershon, "Path integral density matrix dynamics: A method for calculating time-dependent properties in thermal adiabatic and non-adiabatic systems," J. Chem. Phys. **139**, 104107 (2013).

[129] A. Sindhu and A. Jain, "Benchmarking the Surface Hopping Method to Include Nuclear Quantum Effects," J. Chem. Theory Comput. **17**, 655–665 (2021).

[130] J. H. Verner, "Numerically optimal runge–kutta pairs with interpolants," Numer. Algorithms **53**, 383–396 (2010).

[131] C. Tsitouras, "Runge–kutta pairs of order 5 (4) satisfying only the first column simplifying assumption," Comput. Math. Appl. **62**, 770–775 (2011).

[132] Y. Zhang, R. J. Maurer, H. Guo, and B. Jiang, "Hot-electron effects during reactive scattering of H_2 from Ag(111): The interplay between mode-specific electronic friction and the potential energy landscape," Chem. Sci. **10**, 1089–1097 (2019).

[133] B. Jiang and H. Guo, "Six-dimensional quantum dynamics for dissociative chemisorption of h2 and d2 on ag(111) on a permutation invariant potential energy surface," Phys. Chem. Chem. Phys. **16**, 24704–24715 (2014).

[134] R. J. Maurer, B. Jiang, H. Guo, and J. C. Tully, "Mode specific electronic friction in dissociative chemisorption on metal surfaces: h_2 on ag(111)," Phys. Rev. Lett. **118**, 256001 (2017).

[135] R. J. Maurer, Y. Zhang, H. Guo, and B. Jiang, "Hot electron effects during reactive scattering of h2 from ag(111): assessing the sensitivity to initial conditions, coupling magnitude, and electronic temperature," Faraday Discuss. **214**, 105–121 (2019).

[136] M. Alducin, R. Díez Muiño, and J. Juaristi, "Non-adiabatic effects in elementary reaction processes at metal surfaces," Prog. Surf. Sci. **92**, 317–340 (2017).

[137] R. J. Maurer, M. Askerka, V. S. Batista, and J. C. Tully, "Ab initio tensorial electronic friction for molecules on metal surfaces: Nonadiabatic vibrational relaxation," Phys. Rev. B **94**, 115432 (2016).

[138] R. Scholz, S. Lindner, I. Lončarić, J. C. Tremblay, J. I. Juaristi, M. Alducin, and P. Saalfrank, "Vibrational response and motion of carbon monoxide on cu(100) driven by femtosecond laser pulses: Molecular dynamics with electronic friction," Phys. Rev. B **100**, 245431 (2019).

[139] M. Alducin, N. Camillone, S.-Y. Hong, and J. I. Juaristi, "Electrons and phonons cooperate in the laser-induced desorption of co from pd(111)," Phys. Rev. Lett. **123**, 246802 (2019).

[140] A. Serrano Jiménez, A. P. Sánchez Muzas, Y. Zhang, J. Ovčar, B. Jiang, I. Lončarić, J. I. Juaristi, and M. Alducin, "Photoinduced Desorption Dynamics of CO from Pd(111): A Neural Network Approach," J. Chem. Theory Comput. **17**, 4648–4659 (2021).

[141] S. I. Anisimov, B. L. Kapeliovich, and T. L. Perel'Man, "Electron emission from metal surfaces exposed to ultrashort laser pulses," J. Exp. Theor. Phys. **39**, 375–377 (1974).

[142] L. Zhu, Y. Zhang, L. Zhang, X. Zhou, and B. Jiang, "Unified and transferable description of dynamics of h2 dissociative adsorption on multiple copper surfaces via machine learning," Phys. Chem. Chem. Phys. **22**,

13958–13964 (2020).
[143] R. Scholz, G. Floß, P. Saalfrank, G. Füchsel, I. Lončarić, and J. I. Juaristi, "Femtosecond-laser induced dynamics of co on ru(0001): Deep insights from a hot-electron friction model including surface motion," Phys. Rev. B **94**, 165447 (2016).
[144] N. Gerrits, J. I. Juaristi, and J. Meyer, "Electronic friction coefficients from the atom-in-jellium model for Z = 1 – 92," Phys. Rev. B **102**, 155130 (2020).
[145] J. I. Juaristi, M. Alducin, and P. Saalfrank, "Femtosecond laser induced desorption of h_2, d_2, and hd from ru(0001): Dynamical promotion and suppression studied with ab initio molecular dynamics with electronic friction," Phys. Rev. B **95**, 125439 (2017).
[146] M. Askerka, R. J. Maurer, V. S. Batista, and J. C. Tully, "Role of tensorial electronic friction in energy transfer at metal surfaces," Phys. Rev. Lett. **116**, 217601 (2016).
[147] P. Spiering and J. Meyer, "Testing Electronic Friction Models: Vibrational De-excitation in Scattering of H2 and D2 from Cu(111)," J. Phys. Chem. Lett. **9**, 1803–1808 (2018).
[148] P. Spiering, K. Shakouri, J. Behler, G.-J. Kroes, and J. Meyer, "Orbital-Dependent Electronic Friction Significantly Affects the Description of Reactive Scattering of N2 from Ru(0001)," J. Phys. Chem. Lett. **10**, 2957–2962 (2019).
[149] A. C. Luntz and M. Persson, "How adiabatic is activated adsorption/associative desorption?" J. Chem. Phys. **123**, 074704 (2005).
[150] C. L. Box, W. G. Stark, and R. J. Maurer, "Ab initio calculation of electron-phonon linewidths and molecular dynamics with electronic friction at metal surfaces with numeric atom-centered orbitals," (2021), arXiv:2112.00121 [cond-mat.mtrl-sci].
[151] Y. Zhang, R. J. Maurer, and B. Jiang, "Symmetry-adapted high dimensional neural network representation of electronic friction tensor of adsorbates on metals," J. Phys. Chem. C **124**, 186–195 (2020).
[152] X. Hu, W. L. Hase, and T. Pirraglia, "Vectorization of the general Monte Carlo classical trajectory program VENUS," J. Comput. Chem. **12**, 1014–1024 (1991).
[153] W. L. Hase, R. J. Duchovic, X. Hu, A. Komornicki, K. F. Lim, D.-h. Lu, G. H. Peslherbe, K. N. Swamy, S. V. Linde, A. Varandas, *et al.*, "VENUS96, A general chemical dynamics computer program," QCPE **16**, 671 (1996).