Reference manual: Proton/hydrogen transfer dynamics via simulated ultrafast X-ray pump-probe spectroscopy

Micheline B. Soley,*[ad] Pablo E. Videla,*[ab] E. T. J. Nibbering,[c] V. S. Batista[ab]

July 27, 2021

[a]Department of Chemistry, Yale University, P.O. Box 208107, New Haven, CT, 06520-8107, USA
[b]Energy Sciences Institute, Yale University, P.O. Box 27394, West Haven, CT, 06516-7394, USA
[c]Max Born Institute for Nonlinear Optics and Short Pulse Spectroscopy, Max Born Strasse 2A, 12489 Berlin, Germany
[d]Yale Quantum Institute, Yale University, P.O. Box 208334, New Haven, CT, 06520-8263, USA.

1 Introduction

We present HBT, a code for simulation of UV pump/X-ray probe spectra and highly multidimensional chemical dynamics with fully quantum treatment of all degrees of freedom. The code is demonstrated for HBT excited state intramolecular hydrogen/proton transfer (ESIHT/ESIPT).

The program consists of three main parts: ab initio determination of the molecular system's potential energy surfaces, tensor-train Split Operator Fourier Transform (TT-SOFT) quantum dynamics, and UV pump/Xray probe spectroscopy calculation.

2 Dependencies

Implementation of the HBT code requires the following packages:

- Dynamics Code Dependencies:
  - `python` – Programming language required to run dynamics code
  - `ttpy` – Python library for low-rank tensor train manipulation
  - `matplotlib` – Python library for visualization
  - `numpy` – Python library
  - `scipy` – Python library
  - `gnuplot` – Optional software for visualization of dynamics code output with provided gnuplot scripts

3 Dynamics Code Instructions

3.1 Interpolation of the Ab Initio Potential Energy Surfaces

The **2d_interpolator.py** code interpolates the *ab initio* potential energy surfaces to a finer grid to facilitate wavepacket propagation. The code is run with the command

```
python 2d_interpolator.py
```

Note that the program produces either the ground state potential `pot_type='gs'` or the excited state potential `pot_type='es'` in a single run. The program must therefore be run twice with each potential type setting in order to create the full set of potential energy surfaces required for the quantum dynamics simulation.

The code requires the following data files, which are produced from the *ab initio* potential energy surface determination code:

- `2d_pes_*.dat` – Reactive potential energy surface
- `2d_pes_nm_ampl_*.dat` – Normal mode amplitudes
- `2d_hess_diag_*.dat` – Diagonal normal mode couplings
- `2d_hess_ampl_*.dat` – Full normal mode couplings

The code returns interpolated versions of the raw potential energy surface data files, as follows:

- `2d_pes_*_interpolated.dat` – Interpolated reactive potential energy surface
- `2d_pes_nm_ampl_*_interpolated.dat` – Interpolated normal mode amplitudes
- `2d_hess_diag_*_interpolated.dat` – Interpolated diagonal normal mode couplings
- `2d_hess_ampl_*_interpolated.dat` – Interpolated full normal mode couplings

Data files for the ground and excited state surfaces are marked with `*=gs` and `*=es`, respectively. Example input and output files are included in the `DynamicsCode` folder.

The program automatically produces plots of the raw and interpolated reactive potential energy surface, normal mode amplitudes, and nonzero normal mode couplings, which are printed to screen.

To modify the interpolation code, the interpolation/extrapolation grid can be adjusted via manipulation of the parameters for the maximum and minimum values of the x and y grid (`q1_max`, `q1_min`, `q2_max`, and `q2_min`) and the number of x and y grid divisions (`ndim1` and `ndim2`). The degree of the interpolation spines can also be toggled between `linear`, `cubic`, and `quintic` splines with the variable `interp_kind`.

3.2 Quantum Propagation in Full Dimensionality

The molecular motion associated with excited state intramolecular hydrogen/proton transfer (ESIHT/ESIPT) in HBT is simulated according to tensor-train split operator Fourier transform (TT-SOFT) quantum dynamics with the program **ttmn.py**. In the program, the interpolated potential energy surfaces is represented as a low-rank tensor train. An initial guess wavepacket is initialized in the ground state potential, and TT-SOFT imaginary-time propagation is performed to identify the ground state wavefunction. The wavepacket is then instantaneously excited to the excited state potential energy surface, and TT-SOFT real-time propagation is employed to simulate the dynamics of the molecule.

The following command is used to run the TT-SOFT code:

```
python ttmn.py
```

The code requires the following input files (the output files of the interpolator code:

- `2d_pes_*.dat` – Reactive potential energy surface
- `2d_pes_nm_ampl_*.dat` – Normal mode amplitudes
- `2d_hess_diag_*.dat` – Diagonal normal mode couplings
- `2d_hess_ampl_*.dat` – Full normal mode couplings

for both the ground state `*=gs` and excited state `*=es`. The code also requires the auxiliary function file

- `HBT_potential.py`

which converts the interpolated potential parameters into a functional form that can be evaluated at any point on the dynamics grid.

The code outputs data files that can be used to reconstruct the tensor train from the wavepacket at any time during the real time propagation.

- realpropAstateB.npy

where A is the propagation step number and B={0,1} is a technical index. The technical index has no physical meaning for HBT simulation, but can be used to model other molecular systems where inclusion of two potential energy surfaces is essential to dynamics (such as molecules that feature conical intersections and/or significant population transfer between potential energy surfaces). The code also outputs additional information on the value of observables during the dynamics simulation, including:

- populations.dat – Enol/keto populations
- norm.dat – Norm
- expectationvalues.dat – Expectation values of position
- energy.dat – Expectation values of the potential, kinetic, and total energy

Observables can be visualized with the following commands:

```
gnuplot populations.gpt
gnuplot norm.gpt
gnuplot expectationvalueOH.gpt
gnuplot expectationvalueOH.gpt
gnuplot energy.gpt
```

These commands produce the image files:

- populations.eps and populations.pdf
- norm.eps and norm.pdf
- expectationvalueOH.eps and expectationvalueOH.pdf
- expectationvalueCCC.eps and expectationvalueCCC.pdf
- energy.eps and energy.pdf

The propagation code can be changed by altering the parameter values defined in the function parameters. All parameters are given in atomic units unless stated otherwise. Adjustable parameters include:

- qmodes – Number of quantum bath modes
- nsc – Number of time steps
- tau – Time step length
- eps – Tensor train accuracy parameter
- rma – Maximum tensor train rank
- d – Argument for number of grid divisions n = 2**d (must match value in interpolation code)
- Lx – Length of x grid (must match value in interpolation code)
- Ly – Length of y grid (must match value in interpolation code)
- ro – Center of initial Gaussian wavepacket
- sig – Width of initial Gaussian wavepacket

3.3 Analysis of the Potential Energy Surface

Information about the global minimum wells of the potential energy surface and the accuracy of the low-rank approximation can be obtained with the program **hbtinformation.py**

The code requires the same input files required by the quantum propagation code:

- `2d_pes_*.dat` – Reactive potential energy surface
- `2d_pes_nm_ampl_*.dat` – Normal mode amplitudes
- `2d_hess_diag_*.dat` – Diagonal normal mode couplings
- `2d_hess_ampl_*.dat` – Full normal mode couplings

for both the ground `*=gs` and excited states `*=es`, as well as the potential function

- `HBT_potential.py`

The program automatically prints images to screen of the interpolated reactive potential energy surfaces, the tensor-train approximations to the reactive potential energy surfaces, the grid-based excited state displacements and couplings, and the tensor-train approximations to the excited state displacements and couplings. The displacement and coupling images are also saved to file as:

- `ExactParameters.png`
- `ApproximateParameters.png`

The program also prints to the terminal information about the positions, values, and curvature of the minimum wells of the ground and excited state potential energy surfaces; and the associated harmonic approximations to the frequencies, widths, and zero-point energies associated with the ground state wavefunctions in these surfaces.

The HBT information code can be modified by updating:

- `eps` – Tensor train accuracy parameter
- `rma` – Tensor train maximum rank

Other parameters should be chosen to agree with those employed in the interpolation code, including

- `xminorig`, `xmaxorig`, `yminorig`, and `ymaxorig` – Minimum and maximum positions in the x and y grids of the raw *ab initio* potential energy surfaces
- `nhbt` – Number of positions in each direction (must be the square root of the total number of xy points `totalpoints`)
- `xminhbt`, `xmaxhbt`, `yminhbt`, and `ymaxhbt` – Minimum and maximum positions in the x and y grids of the interpolated potential energy surfaces

The choice of potential parameters visualized can be modified with parameters

Displacements of different bath modes can be visualized by changing `specmode` and couplings between different bath modes can be visualized by changing `specmode0` and `specmode1`.

3.4 Analysis of Quantum Wavepacket Slices

The motion of the molecule's large-amplitude modes during the isomerization process can be visualized using the snaphots code **snapshots.py**.

The snapshots code requires as input the tensor trains for the wavefunctions at each time step:

- `realpropAstateB.npy`

for all (or selected) propagation step numbers `A=0-799` and both technical indice `B={0,1}`.

The snapshots code outputs images of the probability density as a function of the large amplitude modes:

- snapshotsCD.png

where C is the overall dimensionality of the system.

The program requires that parameters match those defined in the dynamics program, including:

- qmodes – Number of quantum bath modes
- tau – Time step length
- d – Argument for number of grid divisions n = 2**d (must match value in interpolation code)
- Lx – Length of x grid (must match value in interpolation code)
- Ly – Length of y grid (must match value in interpolation code)
- xminorig, xmaxorig, yminorig, and ymaxorig – Minimum and maximum positions in the x and y grids of the raw *ab initio* potential energy surfaces
- nhbt – Number of positions in each direction
- xminhbt, xmaxhbt, yminhbt, and ymaxhbt – Minimum and maximum positions in the x and y grids of the interpolated potential energy surfaces

The time steps at which the probability density slices can be selected by modifying the set of indices qq as marked in the code.