Hands-on Tutorials for the Nonadiabatic Dynamics based on Exact Factorization with Libra

Daeho Han

Department of Chemistry,
University at Buffalo, SUNY

July 10, 2024

Copy the exact calculation file and extract it to the XF tutorial directory.

cp /vscratch/grp-alexeyak/daehohan/TUT_XF/DVR_data.tar.bz2 /your_Tutorial_Libra/6_dynamics/1_trajectory_based/11_model_xf/

tar –xvf DVR_data.tar.bz2

Change the path accordingly according to your vscratch path.

Objectives

- Understand the set of parameters for running the XF methods with the Libra package
- Run the nonadiabatic dynamics calculations with the XF and conventional methods with 1D model Hamiltonians
- Visualize the dynamics results including the XF quantities
- Compare the results with the exact DVR dynamics

Setting the XF dynamics

- Parameters for the XF methods

Method	Electronic EOM	Nuclear force	Velocity rescaling for a hop	Energy conservation
SHXF	$H_{BO} + H_{XF}$	Active-state force	Yes	Yes
MQCXF		$F_{MF} + F_{XF}$	No	
MFXF		F_{MF}		No

SHXF ➡ int decoherence_algo == 5

MQCXF and MFXF ➡ int decoherence_algo == 6

$$F_\nu = F_{a,\nu}$$

$$F_\nu = -\sum_i \rho_{ii} \nabla_\nu E_i - \sum_{ij} \rho_{ij} (E_j - E_i) d_{ij,\nu} + \sum_{ij} \rho_{ii} \rho_{jj} \left[\sum_\mu \frac{2i\mathcal{P}_\mu}{\hbar M_\mu} \cdot (\phi_{j,\mu} - \phi_{i,\mu}) \right] \phi_{i,\nu}$$

int use_xf_force
- 0: only F_{MF}, MFXF
- 1: with the full force $F_{MF} + F_{XF}$, MQCXF

Setting the XF dynamics

- The coherence criteria

To determine whether to spawn an auxiliary trajectory for a wavepacket on state i, the coherence criterion is checked, $\epsilon < |C_i|^2 < 1 - \epsilon$. The threshold ϵ can be set by the coherence_threshold variable (default: 0.01).

ϵ == double coherence_threshold

- The Gaussian width

One needs to set the Gaussian width parameter used for computing the quantum momentum.

$$\mathcal{P}_\nu = \frac{i\hbar}{2\sigma_\nu^2}\left(R_{a,\nu} - \sum_i \rho_{ii} R_{i,\nu}\right)$$

The DOF-resolved Gaussian width is defined as the wp_width variable.

σ_ν == MATRIX wp_width – ndof × 1

Setting the XF dynamics

- The time-dependent width approximations

The time-dependent width can be set by the use_td_width parameter.

int use_td_width

- 0: frozen Gaussian [default]
- 1: free-particle Gaussian approximation
- 2: the Schwartz scheme
- 3: the Subotnik scheme

The Schwartz width [Bedard-Hearn, M. J.; Larsen, R. E.; Schwartz, B. J. *JCP.* **2005**, 123 (23), 234106.]

$$\sigma_\nu^{-2}(t) = \left(\frac{(w_\nu/\mathrm{Bohr})^2}{2\lambda_{D,\nu}(t)}\right)^2 = \left(\frac{(w_\nu/\mathrm{Bohr})^2 P_\nu}{4\pi\hbar}\right)^2$$

The Subotnik width [Subotnik, J. E. *JPCA.* **2011**, 115 (44), 12083–12096.]

$$\sigma_{ij,\nu}^{-2}(t) = \hbar\frac{|R_{i,\nu} - R_{j,\nu}|}{|P_{i,\nu} - P_{j,\nu}|}$$

$$\sigma_\nu^{-2}(t) = \frac{1}{N-1}\sum_{\substack{i,j=0,\\ i<j}}^{N-1} \sigma_{ij,\nu}^2(t)$$

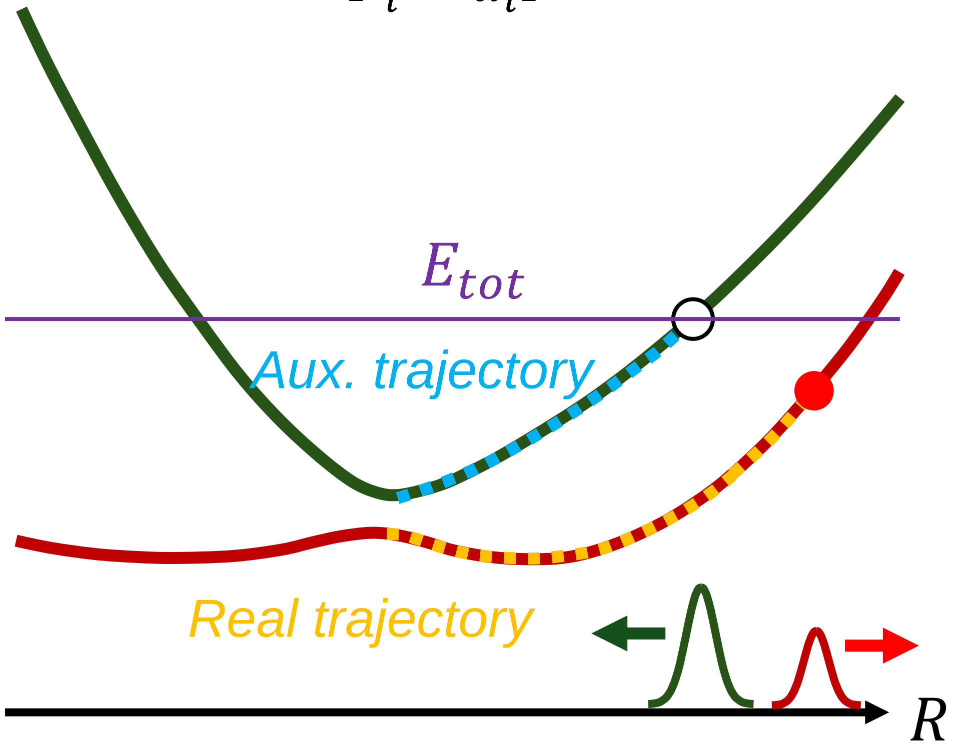
Setting the XF dynamics

- Turning-point algorithms

Case I. An auxiliary trajectory encounters the turning point.

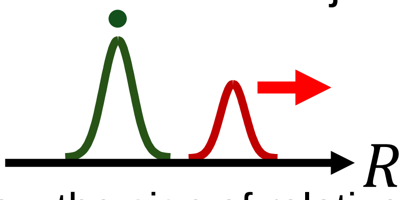
$$\frac{1}{2}\boldsymbol{P}_i^T \boldsymbol{M}^{-1}\boldsymbol{P}_i + E_i = \frac{1}{2}\boldsymbol{P}^T \boldsymbol{M}^{-1}\boldsymbol{P} + E$$

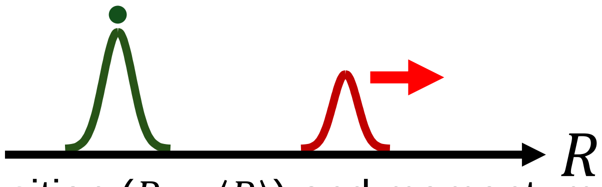
$$\boldsymbol{P}_i = \alpha_i \boldsymbol{P}$$



int project_out_aux

- 0, Fix that aux. trajectory by $\alpha_i = 0$

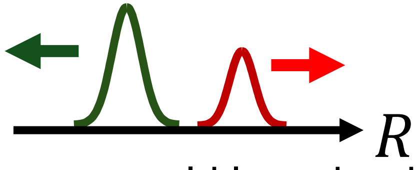




Anyway, the sign of relative position ($R - \langle R \rangle$) and momentum difference ($\Delta\phi$) is maintained, so the decoherence correction directs the dynamics as it have done.

- 1 [default], BC1: Project out the corresponding state density and initialize aux. trajectories.

$$\rho_{ii} \coloneqq 0 \Rightarrow \text{Renormalize } \sum_i \rho_{ii}.$$



Clearly, the real trajectory would lose the density on the aux. trajectory due to the reflection. Thus, clean it in the BCSH-fashion.

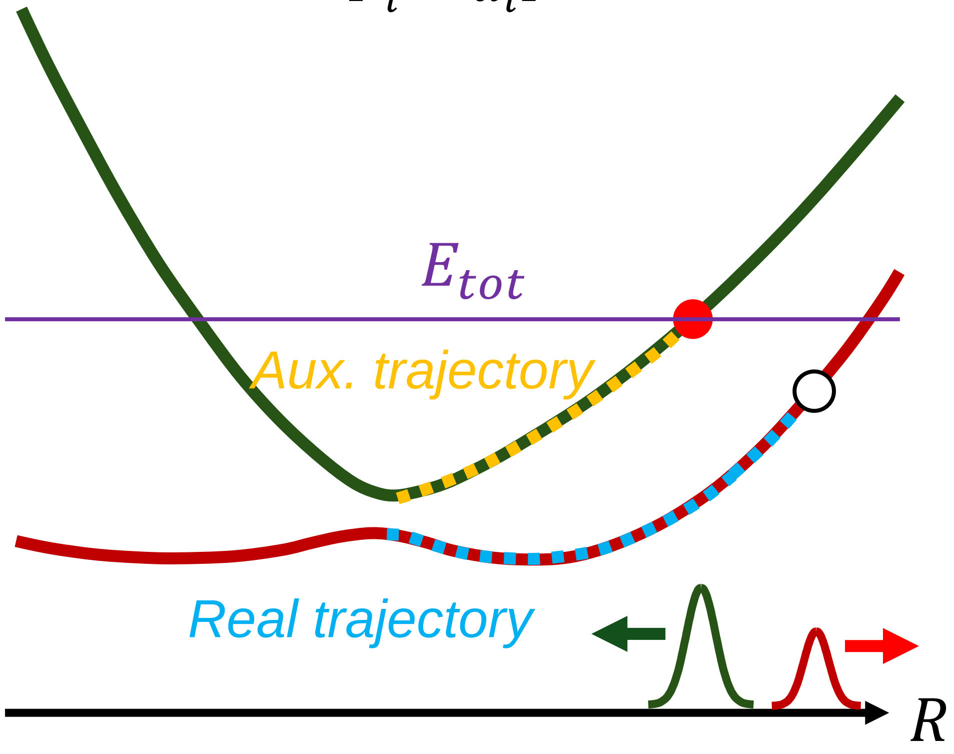
Setting the XF dynamics

- Turning-point algorithms

Case II. The real trajectory encounters the turning point.

$$\frac{1}{2}\boldsymbol{P}_i^T\boldsymbol{M}^{-1}\boldsymbol{P}_i + E_i = \frac{1}{2}\boldsymbol{P}^T\boldsymbol{M}^{-1}\boldsymbol{P} + E$$

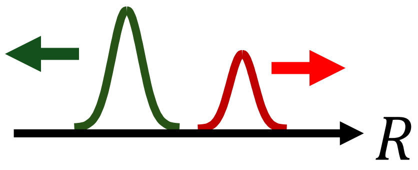
$$\boldsymbol{P}_i = \alpha_i\boldsymbol{P}$$



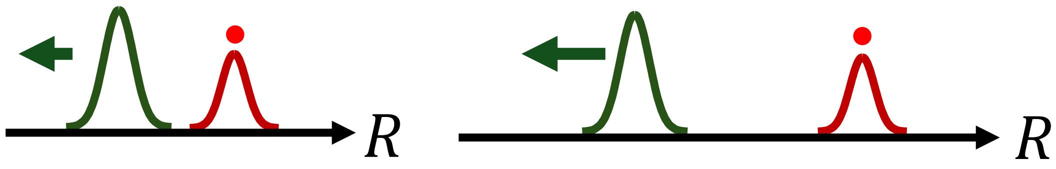
int tp_algo

- 0, don't use the turning-point algorithm
- 1 [default], BC2: Collapse the state into the active state.

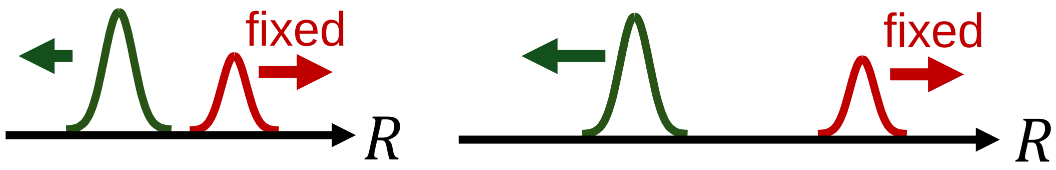
$\rho \coloneqq 0 \Rightarrow \rho_{aa} = 1 \Rightarrow$ Renormalize $\sum_i \rho_{ii}$.



- 2: Fix the aux. trajectory



- 3: Keep the last momenta in aux. trajectory



Comparison with the DVR dynamics

- Population and coherence expressed by the DVR amplitude

XF ansatz DVR ansatz

$$|\Psi(\boldsymbol{R},t)\rangle = \chi(\boldsymbol{R},t)|\Phi_{\boldsymbol{R}}(t)\rangle = \sum_i \chi_i(\boldsymbol{R},t)|i_{\boldsymbol{R}}\rangle$$

With the adiabatic basis expansion, $|\Phi_{\boldsymbol{R}}(t)\rangle = \sum_i C_i(\boldsymbol{R},t)|i_{\boldsymbol{R}}\rangle$,

$$\sum_i C_i(\boldsymbol{R},t)\chi(\boldsymbol{R},t)|i_R\rangle = \sum_i \chi_i(\boldsymbol{R},t)|i_{\boldsymbol{R}}\rangle$$

➡ $C_i(\boldsymbol{R},t)\chi(\boldsymbol{R},t) = \chi_i(\boldsymbol{R},t)$ or $C_i(\boldsymbol{R},t) = \chi_i(\boldsymbol{R},t)/\chi(\boldsymbol{R},t)$

For a local observable, $\hat{A}(R,t)$, its time-resolved average is the following.

$$\langle A(t)\rangle = \int dR\langle\Psi(R,t)|\hat{A}(R,t)|\Psi(R,t)\rangle = \int dR\,|\chi(R,t)|^2\langle\Phi_R(t)|\hat{A}(R,t)|\Phi_R(t)\rangle = \int dR|\chi(R,t)|^2 A(R,t)$$

In the MQC approach, $|\chi(R,t)|^2 \approx \frac{1}{N_{tr}}\sum_K^{N_{tr}} \delta(\boldsymbol{R}-\boldsymbol{R}^K)$ ➡ $\langle\hat{A}(t)\rangle \approx \frac{1}{N_{tr}}\sum_K^{N_{tr}} A^K(t)$

Comparison with the DVR dynamics

- Population and coherence expressed by the DVR amplitude

Population

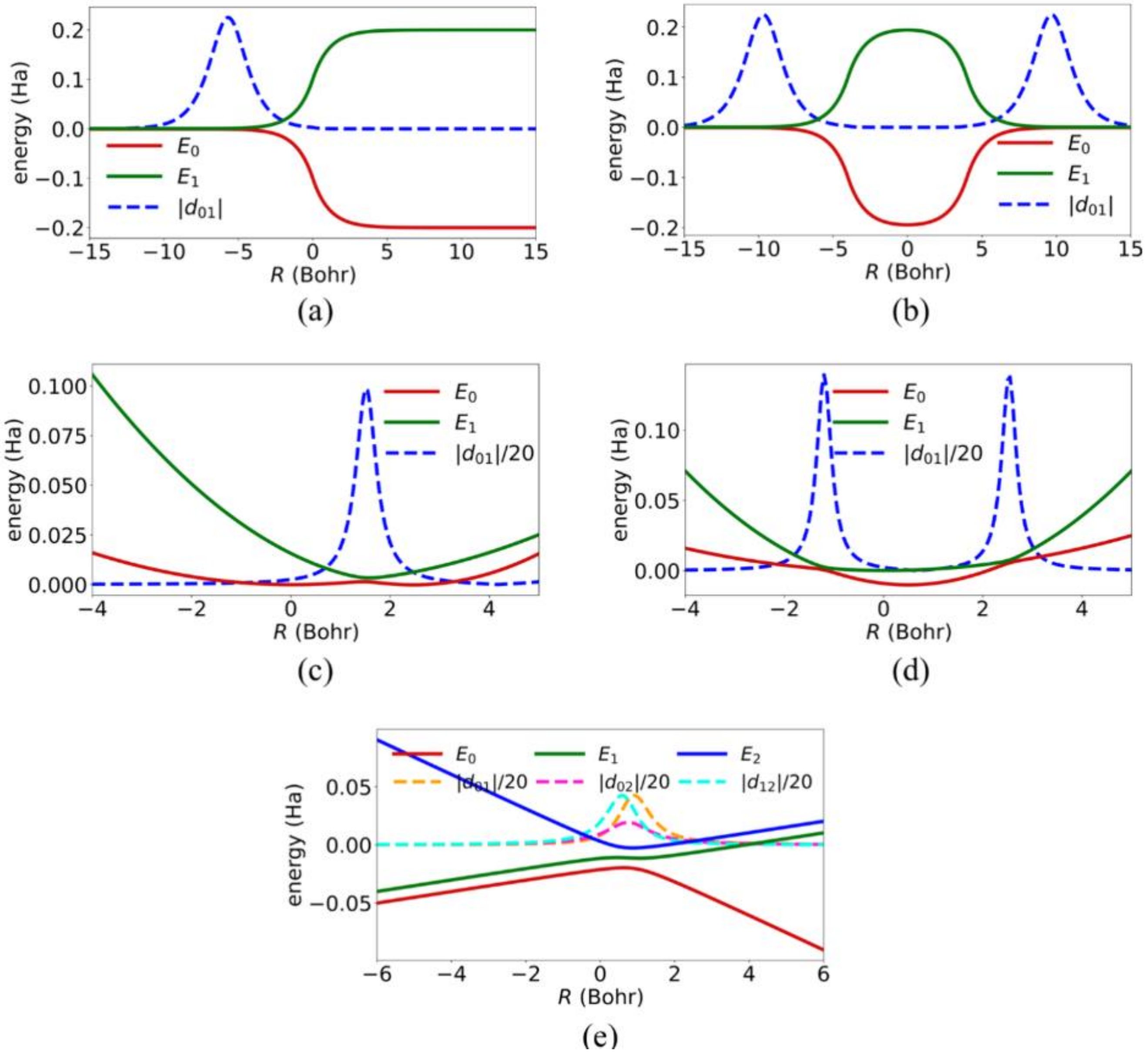
$$\langle \rho_{ii}(t) \rangle = \int dR |\chi(R,t)|^2 \cdot \frac{|\chi_i(R,t)|^2}{|\chi(R,t)|^2} = \int dR |\chi_i(R,t)|^2 \approx \frac{1}{N_{tr}} \sum_{K}^{N_{tr}} \rho_{ii}^{K}(t)$$

Coherence

$$\left\langle \left| \rho_{ij}(t) \right|^2 \right\rangle = \int dR |\chi(R,t)|^2 \cdot \frac{|\chi_i(R,t)|^2}{|\chi(R,t)|^2} \cdot \frac{\left|\chi_j(R,t)\right|^2}{|\chi(R,t)|^2} = \int dR \frac{|\chi_i(R,t)|^2 \left|\chi_j(R,t)\right|^2}{\sum_k |\chi_k(R,t)|^2}$$

$$\approx \frac{1}{N_{tr}} \sum_{K}^{N_{tr}} \left| \rho_{ij}^{K}(t) \right|^2 = \frac{1}{N_{tr}} \sum_{K}^{N_{tr}} \rho_{ii}^{K}(t)\, \rho_{jj}^{K}(t)$$

Model Hamiltonians



- Tully's extended crossing with reflection (ECWR)
- Subotnik's double arch geometry (DAG)
- Single- and double-crossing Holstein (SC, DC Holstein)
- 3-state Esch-Levine model

TUT1. Run the XF dynamics

- Section 1-7

ECWR	DAG	SC H	DC H	SHXF	MQCXF	FSSH	BCSH
0	1	2	3	0	1	3	5

Run the nonadiabatic dynamics on the ECWR, DAG, Holstein models with the SHXF, MQCXF, FSSH and BCSH methods (30 min).

In Section 3,

Choose the model to simulate here by setting `model_indx`.

```
# 0 - Single-crossing (SC) Holstein, 2 level
# 1 - Double-crossing (DC) Holstein, 2 level
# 2 - Tully, extended crossing with reflection (ECWR), 2 level
# 3 - Double arch geometry or symmetrized ECWR, 2 level
# 4 - Esch-Levine, 1 crosses 2 parallel, 3 level

#################################
# Give the model used an index
model_indx = 2
################################

model_params = all_model_params[model_indx]
```

model_indx = 0, 1, 2, 3

In Section 4,

Now, it is time to select the type of calculations we want to do. Keep in mind that some options are related to each other, so usually one would need to correlate the choices. For methods based on surface hopping, default options are used for frustrated hops and how to rescale momenta on hops.

```
#################################
# Give the recipe above an index
method_indx = 5
#################################

if method_indx == 0:
    shxf.load(dyn_general)  # SHXF
elif method_indx == 1:
    mqcxf.load(dyn_general)  # MQCXF
elif method_indx == 2:
    mfxf.load(dyn_general)  # MFXF
elif method_indx == 3:
    fssh.load(dyn_general)  # FSSH
elif method_indx == 4:
    sdm.load(dyn_general)  # SDM with default EDC parameters
elif method_indx == 5:
    bcsh.load(dyn_general)  # BCSH

map_methods = {"SHXF":0, "MQCXF":1, "MFXF":2, "FSSH":3, "SDM":4, "BCSH":5}
```

method_indx = 0, 1, 3, 5

TUT2. Run the DVR dynamics and comparison

- Section 8

ECWR	DAG	SC H	DC H	SHXF	MQCXF	FSSH	BCSH
0	1	2	3	0	1	3	5

Run the DVR dynamics on the ECWR, DAG, Holstein models and visualize the population and coherence (5 min).

In Section 3,

Choose the model to simulate here by setting `model_indx`.

```
# 0 - Single-crossing (SC) Holstein, 2 level
# 1 - Double-crossing (DC) Holstein, 2 level
# 2 - Tully, extended crossing with reflection (ECWR), 2 level
# 3 - Double arch geometry or symmetrized ECWR, 2 level
# 4 - Esch-Levine, 1 crosses 2 parallel, 3 level

##################################
# Give the model used an index
model_indx = 2
#################################

model_params = all_model_params[model_indx]
```

model_indx = 0, 1, 2, 3

In Section 8,

Run the DVR dynamics

```
wfc = dvr.init_wfc(exact_params, potential, model_params)
savers = dvr_save.init_tsh_savers(exact_params, model_params, exact_params["nsteps"], wfc)
dvr.run_dynamics(wfc, exact_params, model_params, savers)
```

Compare the results from the DVR and MQC dynamics

```
compare_pop(2, which_methods=["SHXF", "MQCXF", "FSSH", "BCSH"]);
```

```
compare_coh(2, which_methods=["SHXF", "MQCXF", "FSSH", "BCSH"])
```

TUT3. Plot phase-space diagrams

- Section 9

ECWR	DAG	SC H	DC H	SHXF	MQCXF	FSSH	BCSH
0	1	2	3	0	1	3	5

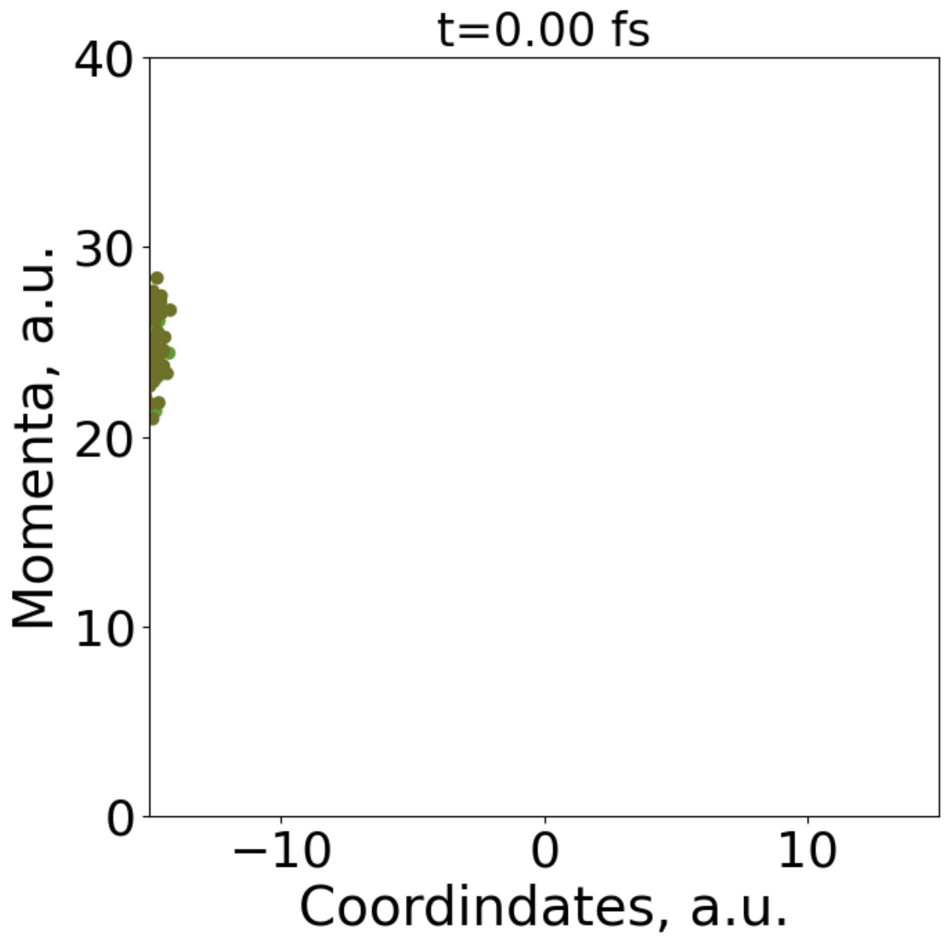
Use the make_phase_space_movie function to visualize the nuclear propagation in the phase space. This GIF file is saved to each dynamics result directory (8 min).

_model_indx = 0, 1, 2, 3

_method_indx = 0, 1, 3, 5

Set _qlim, _plim to [-15, 15] and [0,40] for ECWR and DAG

Set _qlim, _plim to [-5, 5] and [-10,10] for the Holstein models



Ex.
ECWR, SHXF

State 0

State 1

TUT4. Plot the dynamics snapshots

- Section 10

ECWR	DAG	SC H	DC H	SHXF	MQCXF	FSSH	BCSH
0	1	2	3	0	1	3	5

Compare the TDPES, nuclear density, and quantum momenta from the DVR and MQC trajectories (10 min).

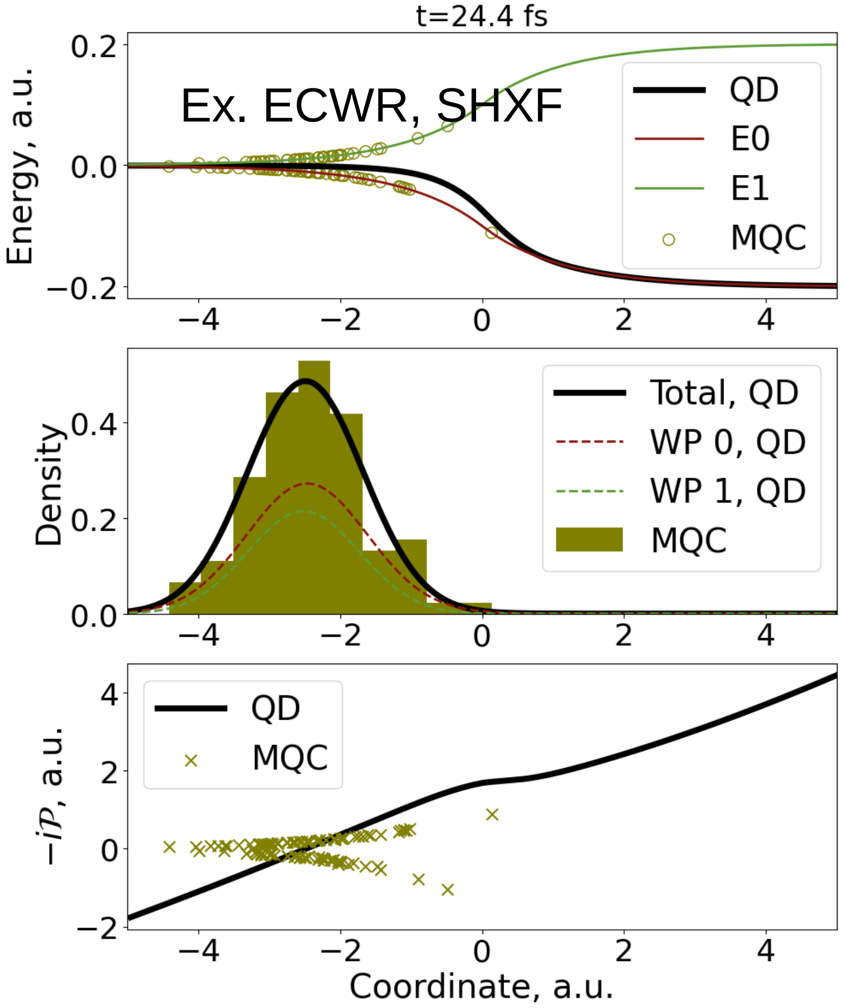
_model_indx = 0, 1, 2, 3

_method_indx = 0, 1 (for the XF methods)

R window can be between [-25, 25] for the Holstein models, and [-35, 35] for ECWR and DAG!

TDPES $$\langle \Phi_{\boldsymbol{R}} | \hat{H}_{BO} | \Phi_{\boldsymbol{R}} \rangle = \frac{\sum_i |\chi_i(\boldsymbol{R},t)|^2 E_i(\boldsymbol{R})}{\sum_i |\chi_i(\boldsymbol{R},t)|^2} \approx \sum_i \left|C_i^K\right|^2 E_i^K$$

Quantum momentum $$-i\mathcal{P}_\nu(\boldsymbol{R},t) = -\frac{\nabla_\nu |\chi(R,t)|^2}{2|\chi(R,t)|^2}$$



TUT5. Use the TD Gaussian scheme

- Section 1-7, 11

ECWR	DAG	SC H	DC H	SHXF	MQCXF	FSSH	BCSH
0	1	2	3	0	1	3	5

Run the XF dynamics with the TD Gaussians, check the behavior of the TD widths and population/coherence (25 min).

model_indx = 2, 3

method_indx = 0, 1 (for the XF methods)

param_indx = 3, 4, 5 (for the XF methods)

Schwartz width

$$\sigma_\nu^{-2}(t) = \left(\frac{(w_\nu/\mathrm{Bohr})^2}{2\lambda_{D,\nu}(t)}\right)^2 = \left(\frac{(w_\nu/\mathrm{Bohr})^2 P_\nu}{4\pi\hbar}\right)^2$$

$w_0 = 1.0\ Bohr$ param_indx = 3

$w_0 = 4.0\ Bohr$ param_indx = 4

Subotnik width

$$\sigma_{ij,\nu}^{-2}(t) = \hbar \frac{|R_{i,\nu} - R_{j,\nu}|}{|P_{i,\nu} - P_{j,\nu}|}$$

param_indx = 5

For population and coherence,

compare_pop(3, _param_indx=5, which_methods=["SHXF", "MQCXF"]);
compare_coh(3, _param_indx=5, which_methods=["SHXF", "MQCXF"])