

From Fewest-Switches Surface Hopping to Surface Hopping including Arbitrary Couplings and a Brief Overview over Practical Aspects

Sebastian Mai

Institute of Theoretical Chemistry, Faculty of Chemistry, University of Vienna, Austria

July 7th, 2022

“Excited States and Nonadiabatic Dynamics CyberTraining Workshop 2022” in Buffalo, NY



universität
wien

What we already have heard during the last days:

- ▶ Classical nuclear dynamics
- ▶ Ab initio dynamics
- ▶ Born-Oppenheimer approximation
- ▶ Excited states

Recap: Surface hopping

What we already have heard during the last days:

- ▶ Classical nuclear dynamics
- ▶ Ab initio dynamics
- ▶ Born-Oppenheimer approximation
- ▶ Excited states

Goal: Perform excited-state dynamics with classical nuclei.

Recap: Surface hopping

What we already have heard during the last days:

- ▶ Classical nuclear dynamics
- ▶ Ab initio dynamics
- ▶ Born-Oppenheimer approximation
- ▶ Excited states

Goal: Perform excited-state dynamics with classical nuclei.

We need to consider the Born-Oppenheimer approximation:

- ▶ **The electronic wavefunction** can change during dynamics.
- ▶ **The electronic wavefunction** has an effect on the nuclear motion.

Electronic wavefunction evolution

Classical nuclear dynamics \Rightarrow No nuclear wavefunction!

Electronic wavefunction evolution

Classical nuclear dynamics \Rightarrow No nuclear wavefunction!

Electronic wavefunction as linear combination of basis functions:

$$|\Psi(t)\rangle = \sum_{\alpha} c_{\alpha}(t) |\phi_{\alpha}\rangle \quad (1)$$

Electronic wavefunction evolution

Classical nuclear dynamics \Rightarrow No nuclear wavefunction!

Electronic wavefunction as linear combination of basis functions:

$$|\Psi(t)\rangle = \sum_{\alpha} c_{\alpha}(t) |\phi_{\alpha}\rangle \quad (1)$$

Inserting this into the TDSE:

$$i\hbar \frac{\partial}{\partial t} |\Psi(t)\rangle = \hat{H}^{\text{el}} |\Psi(t)\rangle \quad (2)$$

and premultiplying with $\langle \phi_{\beta} |$ gives:

Electronic wavefunction evolution

Classical nuclear dynamics \Rightarrow No nuclear wavefunction!

Electronic wavefunction as linear combination of basis functions:

$$|\Psi(t)\rangle = \sum_{\alpha} c_{\alpha}(t) |\phi_{\alpha}\rangle \quad (1)$$

Inserting this into the TDSE:

$$i\hbar \frac{\partial}{\partial t} |\Psi(t)\rangle = \hat{\mathcal{H}}^{\text{el}} |\Psi(t)\rangle \quad (2)$$

and premultiplying with $\langle \phi_{\beta} |$ gives:

$$\frac{\partial}{\partial t} \vec{c}(t) = - [i\mathbf{H} + i\mathbf{T}] \vec{c}(t) \quad (3)$$

where:

$$(\mathbf{H})_{\beta\alpha} = \langle \phi_{\beta} | \hat{\mathcal{H}}^{\text{el}} | \phi_{\alpha} \rangle \quad (\mathbf{T})_{\beta\alpha} = \langle \phi_{\beta} | \partial / \partial t | \phi_{\alpha} \rangle \quad (\vec{c})_{\alpha} = \langle \phi_{\alpha} | \Psi(t) \rangle \quad (4)$$

Simultaneous propagation of electrons and nuclei

We combine the equation of motion for electrons with Newton's equation:

$$\frac{\partial}{\partial t} \vec{c}(t) = - \left[i\mathbf{H}(\vec{R}(t)) + i\mathbf{T}(\vec{R}(t), \partial\vec{R}(t)/\partial t) \right] \vec{c}(t) \quad \text{and} \quad \frac{\partial^2 \vec{R}(t)}{\partial t^2} = \mathbf{M}^{-1} \frac{\partial E_{\alpha}(\vec{R}(t))}{\partial \vec{R}} \quad (5)$$

Simultaneous propagation of electrons and nuclei

We combine the equation of motion for electrons with Newton's equation:

$$\frac{\partial}{\partial t} \vec{c}(t) = - \left[i\mathbf{H}(\vec{R}(t)) + i\mathbf{T}(\vec{R}(t), \partial\vec{R}(t)/\partial t) \right] \vec{c}(t) \quad \text{and} \quad \frac{\partial^2 \vec{R}(t)}{\partial t^2} = \mathbf{M}^{-1} \frac{\partial E_{\alpha}(\vec{R}(t))}{\partial \vec{R}} \quad (5)$$

- ▶ Electrons and nuclei affect each other
- ▶ Evolution of \vec{c} depends on energies and couplings, which depend on \vec{R}
- ▶ Evolution of \vec{R} depends on gradients, which depend on electronic energy of active state α

Simultaneous propagation of electrons and nuclei

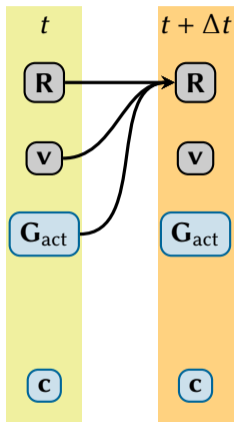
We combine the equation of motion for electrons with Newton's equation:

$$\frac{\partial}{\partial t} \vec{c}(t) = - \left[i\mathbf{H}(\vec{R}(t)) + i\mathbf{T}(\vec{R}(t), \partial\vec{R}(t)/\partial t) \right] \vec{c}(t) \quad \text{and} \quad \frac{\partial^2 \vec{R}(t)}{\partial t^2} = \mathbf{M}^{-1} \frac{\partial E_{\alpha}(\vec{R}(t))}{\partial \vec{R}} \quad (5)$$

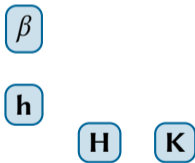
- ▶ Electrons and nuclei affect each other
- ▶ Evolution of \vec{c} depends on energies and couplings, which depend on \vec{R}
- ▶ Evolution of \vec{R} depends on gradients, which depend on electronic energy of active state α

How does this work in detail?

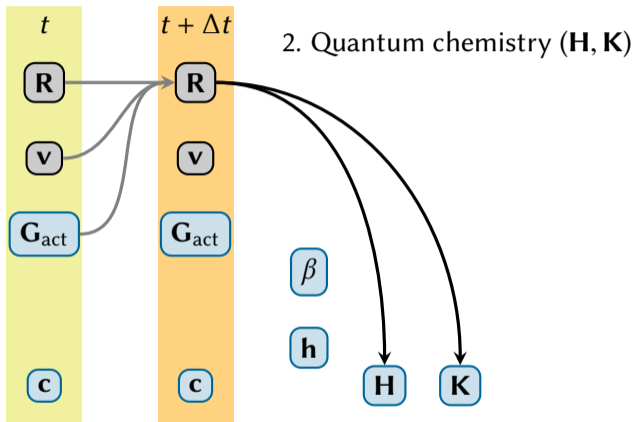
General surface hopping algorithm



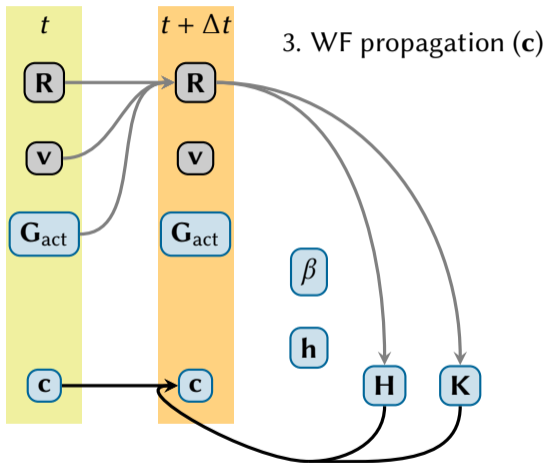
1. Velocity-Verlet (\mathbf{R})



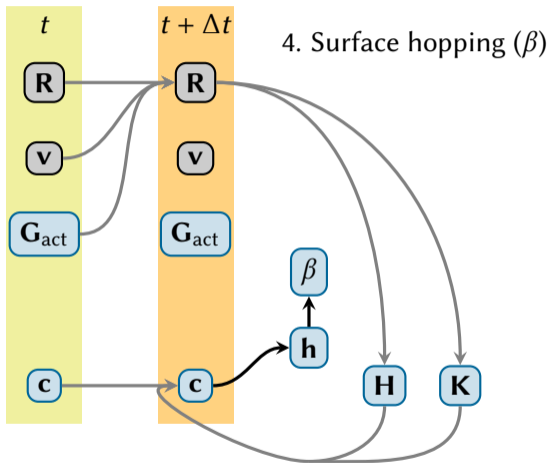
General surface hopping algorithm



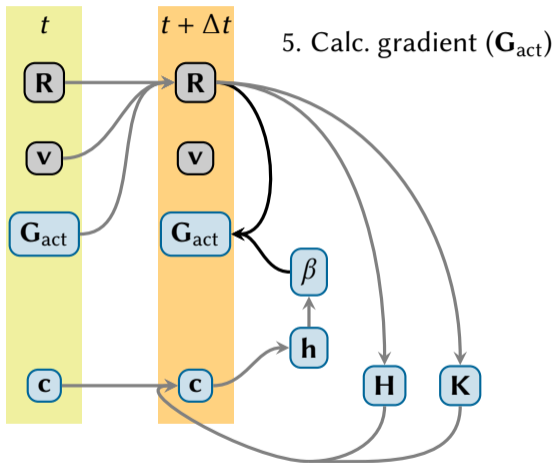
General surface hopping algorithm



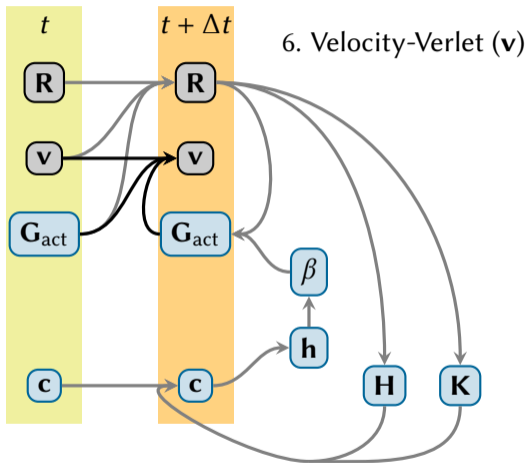
General surface hopping algorithm



General surface hopping algorithm



General surface hopping algorithm



Complicated electronic wavefunction **represented** through vector $\vec{c}(t)$:

$$|\Psi(t)\rangle = \sum_{\alpha} c_{\alpha}(t) |\phi_{\alpha}\rangle \quad (6)$$

Complicated electronic wavefunction **represented** through vector $\vec{c}(t)$:

$$|\Psi(t)\rangle = \sum_{\alpha} c_{\alpha}(t) |\phi_{\alpha}\rangle \quad (6)$$

Examples:

- ▶ $\vec{c}(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$: wavefunction identical to the first basis function $|\phi_1\rangle$
- ▶ $\vec{c}(t) = \begin{pmatrix} \sqrt{0.5} \\ \sqrt{0.5} \end{pmatrix}$: wavefunction superposition of $|\phi_1\rangle$ and $|\phi_2\rangle$

Electronic wavefunction interpretation

Complicated electronic wavefunction **represented** through vector $\vec{c}(t)$:

$$|\Psi(t)\rangle = \sum_{\alpha} c_{\alpha}(t) |\phi_{\alpha}\rangle \quad (6)$$

Examples:

- ▶ $\vec{c}(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$: wavefunction identical to the first basis function $|\phi_1\rangle$
- ▶ $\vec{c}(t) = \begin{pmatrix} \sqrt{0.5} \\ \sqrt{0.5} \end{pmatrix}$: wavefunction superposition of $|\phi_1\rangle$ and $|\phi_2\rangle$

The description of the wavefunction depends on the choice of the basis functions, the **representation!**

Arbitrary couplings

Would like to describe different processes: IC, ISC, laser excitation, ...

Arbitrary couplings

Would like to describe different processes: IC, ISC, laser excitation, ...
These are mediated by specific coupling terms in \hat{H}^{el} .

Arbitrary couplings

Would like to describe different processes: IC, ISC, laser excitation, ...
These are mediated by specific coupling terms in \hat{H}^{el} .

$$\hat{H}^{\text{full}} = \hat{H}^{\text{MCH}} + \hat{H}^{\text{additional}} \quad (7)$$

- ▶ MCH: Molecular Coulomb Hamiltonian
(only E_{kin} and Coulomb interaction, no relativistics/external fields)
⇒ standard quantum chemistry
- ▶ additional: relativistic effects (spin-orbit couplings),
field-dipole interactions, ...

Arbitrary couplings

Would like to describe different processes: IC, ISC, laser excitation, ...
These are mediated by specific coupling terms in \hat{H}^{el} .

$$\hat{H}^{\text{full}} = \hat{H}^{\text{MCH}} + \hat{H}^{\text{additional}} \quad (7)$$

- ▶ MCH: Molecular Coulomb Hamiltonian
(only E_{kin} and Coulomb interaction, no relativistics/external fields)
⇒ standard quantum chemistry
- ▶ additional: relativistic effects (spin-orbit couplings),
field-dipole interactions, ...

Spin-orbit couplings (SOC):

- ▶ Relativistic effect: goes beyond \hat{H}^{MCH}
- ▶ Coupling of intrinsic electron spin momentum with orbital angular momentum
- ▶ Couples states of different multiplicity ⇒ ISC

Arbitrary couplings – Matrix representation of \mathcal{H}^{MCH}

Matrix representation of $\hat{\mathcal{H}}^{\text{MCH}}$:

$$H_{\beta\alpha} = \langle \phi_\beta | \hat{\mathcal{H}}^{\text{MCH}} | \phi_\alpha \rangle \quad (8)$$

Arbitrary couplings – Matrix representation of \mathcal{H}^{MCH}

Matrix representation of $\hat{\mathcal{H}}^{\text{MCH}}$:

$$H_{\beta\alpha} = \langle \phi_\beta | \hat{\mathcal{H}}^{\text{MCH}} | \phi_\alpha \rangle \quad (8)$$

With basis = diabatic states:

$$\vec{H}^{\text{MCH}} = \left(\begin{array}{c} \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \right) \quad \vec{T}^{\text{MCH}} = \left(\begin{array}{c} \\ \\ \end{array} \right)$$

Arbitrary couplings – Matrix representation of $\hat{\mathcal{H}}^{\text{MCH}}$

Matrix representation of $\hat{\mathcal{H}}^{\text{MCH}}$:

$$H_{\beta\alpha} = \langle \phi_\beta | \hat{\mathcal{H}}^{\text{MCH}} | \phi_\alpha \rangle \quad (8)$$

With basis = diabatic states:

$$\vec{H}^{\text{MCH}} = \left(\begin{array}{c} \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \right) \quad \vec{T}^{\text{MCH}} = \left(\begin{array}{c} \\ \\ \\ \end{array} \right)$$

With basis = eigenstates of $\hat{\mathcal{H}}^{\text{MCH}}$:

$$\vec{H}^{\text{MCH}} = \left(\begin{array}{c} \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \right) \quad \vec{T}^{\text{MCH}} = \left(\begin{array}{c} \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \right)$$

Arbitrary couplings – Matrix representation of $\mathcal{H}^{\text{full}}$

Matrix representation of $\hat{\mathcal{H}}^{\text{full}}$:

$$H_{\beta\alpha} = \langle \phi_\beta | \hat{\mathcal{H}}^{\text{full}} | \phi_\alpha \rangle \quad (9)$$

With basis = eigenstates of $\hat{\mathcal{H}}^{\text{MCH}}$:

$$\vec{H}^{\text{MCH}} = \begin{pmatrix} \text{blue} & & \text{green} \\ & \text{blue} & \\ \text{green} & & \text{red} \\ & & & \text{red} \end{pmatrix} \quad \vec{T}^{\text{MCH}} = \begin{pmatrix} \text{blue} & \\ & \text{red} \end{pmatrix}$$

Arbitrary couplings – Matrix representation of $\hat{\mathcal{H}}^{\text{full}}$

Matrix representation of $\hat{\mathcal{H}}^{\text{full}}$:

$$H_{\beta\alpha} = \langle \phi_\beta | \hat{\mathcal{H}}^{\text{full}} | \phi_\alpha \rangle \quad (9)$$

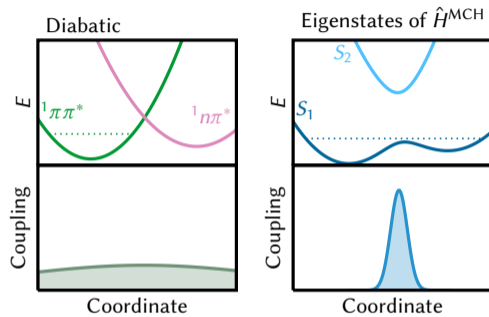
With basis = eigenstates of $\hat{\mathcal{H}}^{\text{MCH}}$:

$$\vec{H}^{\text{MCH}} = \begin{pmatrix} \text{blue} & & \text{green} \\ & \text{blue} & \\ \text{green} & & \text{red} \\ & & & \text{red} \end{pmatrix} \quad \vec{T}^{\text{MCH}} = \begin{pmatrix} \text{blue} & \\ & \text{red} \end{pmatrix}$$

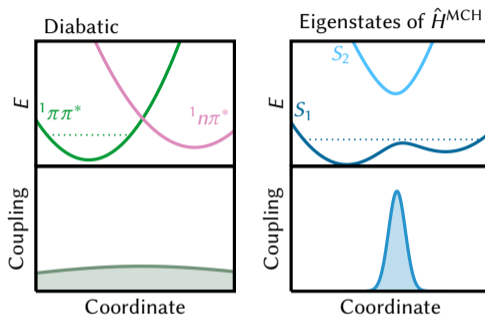
With basis = eigenstates of $\hat{\mathcal{H}}^{\text{full}}$:

$$\vec{H}^{\text{diag}} = \begin{pmatrix} \text{purple} & & & \\ & \text{purple} & & \\ & & \text{purple} & \\ & & & \text{purple} \end{pmatrix} \quad \vec{T}^{\text{MCH}} = \begin{pmatrix} \text{purple} & & & \\ & \text{purple} & & \\ & & \text{purple} & \\ & & & \text{purple} \end{pmatrix}$$

Representations of \mathcal{H}^{MCH} – Potentials



Representations of \mathcal{H}^{MCH} – Potentials

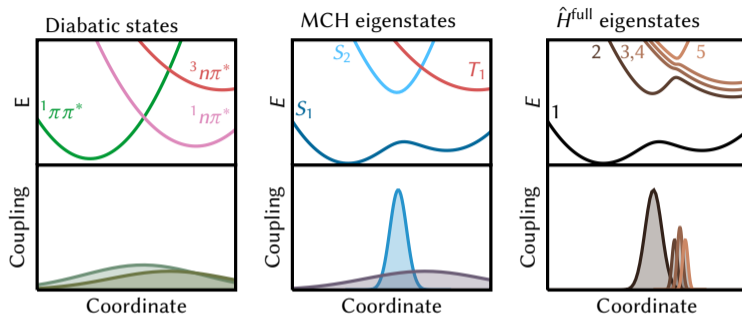


Choice of representation affects surface hopping dynamics:

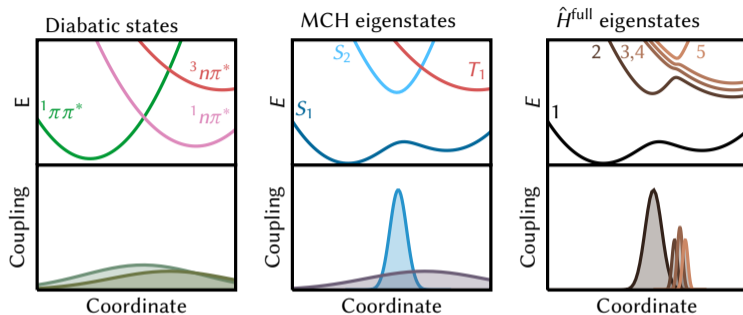
- ▶ Energetics
- ▶ Localization of couplings: where/how often to hop
- ▶ Number of states necessary

Representations of $\mathcal{H}^{\text{full}}$ – Potentials

Eigenstate representations:



Representations of $\hat{\mathcal{H}}^{\text{full}}$ – Potentials

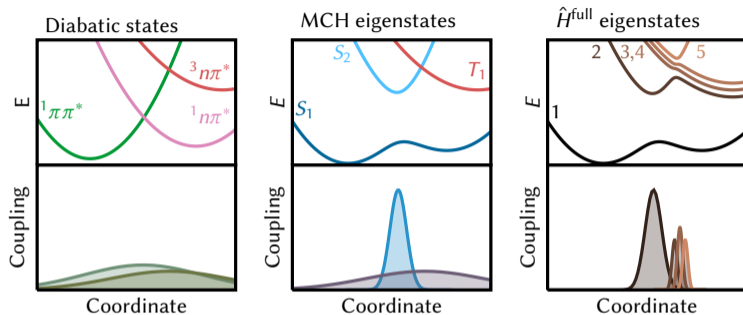


Eigenstate representations:

Surface hopping done optimally in basis of eigenstates of $\hat{\mathcal{H}}^{\text{full}}$:

- + All couplings localized \Rightarrow less hops
- + Multiplets treated correctly
- + Energetics most accurate

Representations of $\hat{\mathcal{H}}^{\text{full}}$ – Potentials



Eigenstate representations:

Surface hopping done optimally in basis of eigenstates of $\hat{\mathcal{H}}^{\text{full}}$:

- + All couplings localized \Rightarrow less hops
- + Multiplets treated correctly
- + Energetics most accurate

Problem: Quantum chemistry programs calculate only eigenstates of $\hat{\mathcal{H}}^{\text{MCH}}$ but not of $\hat{\mathcal{H}}^{\text{full}}$!

The basic idea of SHARC is to perform surface hopping in the diagonal representation, using only information from the MCH representation.

The basic idea of SHARC is to perform surface hopping in the diagonal representation, using only information from the MCH representation.

Transformation from MCH to diagonal representation:

$$\vec{H}^{\text{diag}} = \mathbf{U}^\dagger \vec{H}^{\text{MCH}} \mathbf{U} \quad (10)$$

Can transform wavefunction:

$$\vec{c}^{\text{diag}} = \mathbf{U}^\dagger \vec{c}^{\text{MCH}} \quad (11)$$

The basic idea of SHARC is to perform surface hopping in the diagonal representation, using only information from the MCH representation.

Transformation from MCH to diagonal representation:

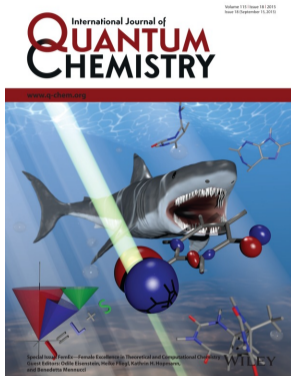
$$\vec{H}^{\text{diag}} = \mathbf{U}^\dagger \vec{H}^{\text{MCH}} \mathbf{U} \quad (10)$$

Can transform wavefunction:

$$\vec{c}^{\text{diag}} = \mathbf{U}^\dagger \vec{c}^{\text{MCH}} \quad (11)$$

Needs modifications to algorithms:

- ▶ Propagating the electronic wavefunction using the MCH data
- ▶ Calculating the hopping probabilities for the diagonal states
- ▶ Getting the gradients of the diagonal states

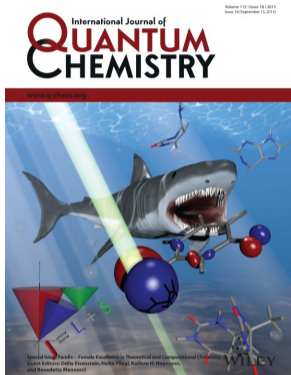


Surface Hopping including Arbitrary Couplings

Program package for nonadiabatic dynamics

SM, Marquetand, González: *IJQC* **115**, 1215 (2015).

SM, P. Marquetand, L. González, *WIREs Comput. Mol. Sci.* **8**, e1370, (2018).



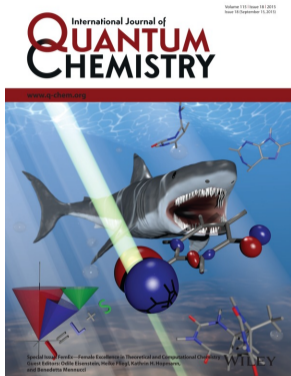
Surface Hopping including Arbitrary Couplings

Program package for nonadiabatic dynamics

- ▶ 2011: Begin of development
- ▶ 2014: 1.0 release
- ▶ 2018: 2.0 release
- ▶ 2019: 2.1 release

SM, Marquetand, González: *IJQC* **115**, 1215 (2015).

SM, P. Marquetand, L. González, *WIREs Comput. Mol. Sci.* **8**, e1370, (2018).



Surface Hopping including Arbitrary Couplings

Program package for nonadiabatic dynamics

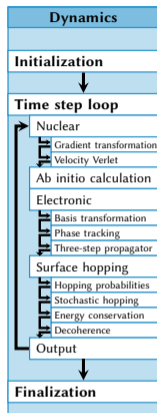
- ▶ 2011: Begin of development
- ▶ 2014: 1.0 release
- ▶ 2018: 2.0 release
- ▶ 2019: 2.1 release

Publicly available, see: sharc-md.org and github.com/sharc-md/sharc

SM, Marquetand, González: *IJQC* **115**, 1215 (2015).

SM, P. Marquetand, L. González, *WIREs Comput. Mol. Sci.* **8**, e1370, (2018).

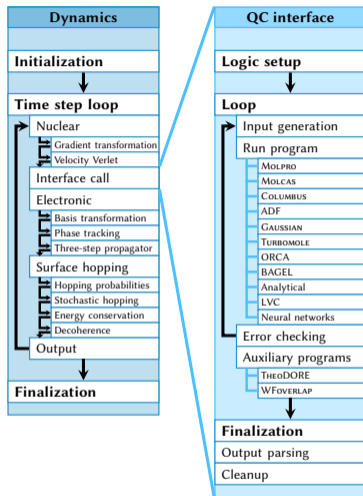
Why need an entire software package?



Important steps in a SHARC project:

- ▶ Running SHARC trajectories

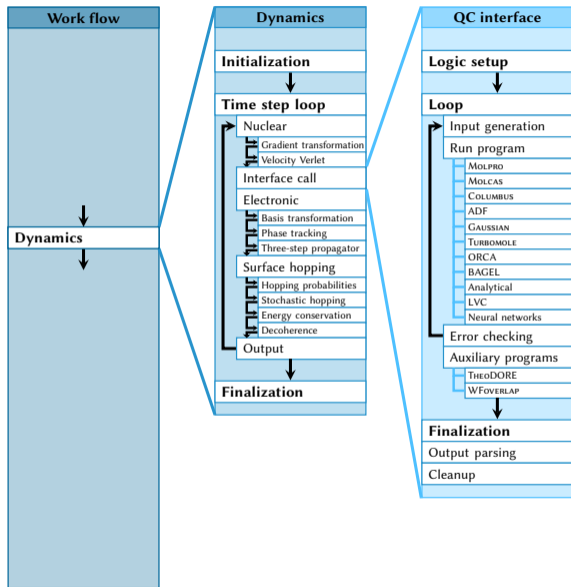
Why need an entire software package?



Important steps in a SHARC project:

- ▶ Running SHARC trajectories
- ▶ Communication with electronic structure codes

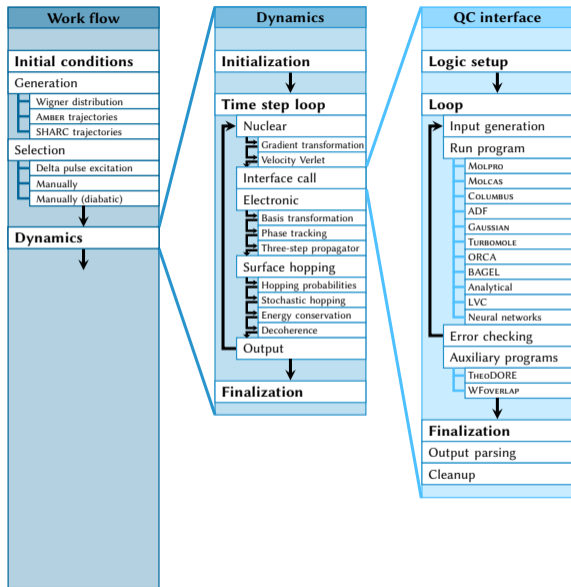
Why need an entire software package?



Important steps in a SHARC project:

- ▶ Running SHARC trajectories
- ▶ Communication with electronic structure codes
- ▶ Trajectory management

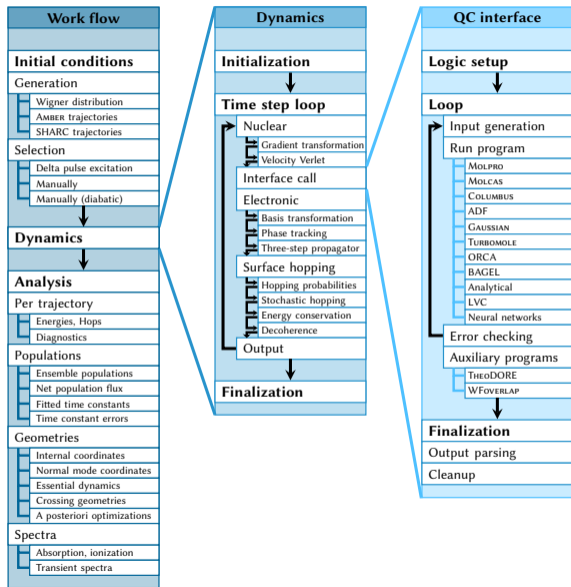
Why need an entire software package?



Important steps in a SHARC project:

- ▶ Running SHARC trajectories
- ▶ Communication with electronic structure codes
- ▶ Trajectory management
- ▶ Initial condition generation

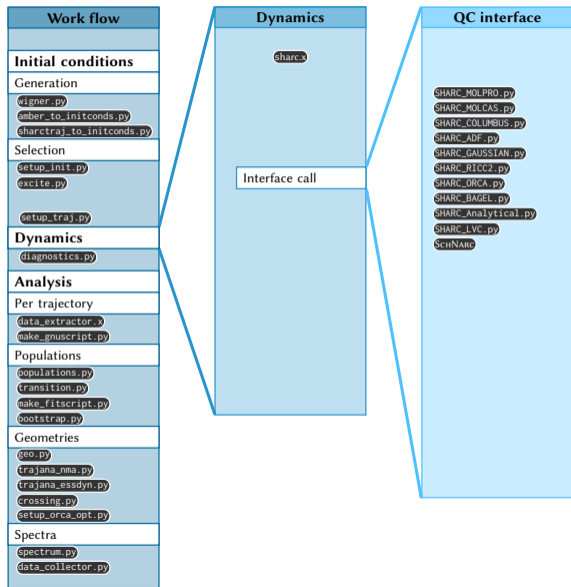
Why need an entire software package?



Important steps in a SHARC project:

- ▶ Running SHARC trajectories
- ▶ Communication with electronic structure codes
- ▶ Trajectory management
- ▶ Initial condition generation
- ▶ Analysis

Why need an entire software package?



Important steps in a SHARC project:

- ▶ Running SHARC trajectories
- ▶ Communication with electronic structure codes
- ▶ Trajectory management
- ▶ Initial condition generation
- ▶ Analysis

Requires a large degree of **automatization** and **modularity**.



So how do you use the SHARC package to perform actual simulations?
What is needed to setup nonadiabatic dynamics?

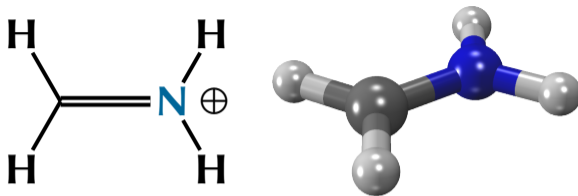
- ① Defining the chemical problem
- ② Choosing the level of theory
- ③ Preparing the initial conditions
- ④ Setting up the trajectories
- ⑤ Running the trajectories

- 1 Defining the chemical problem
- 2 Choosing the level of theory
- 3 Preparing the initial conditions
- 4 Setting up the trajectories
- 5 Running the trajectories



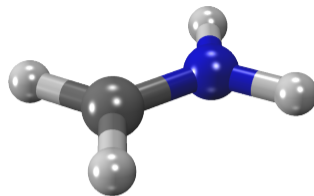
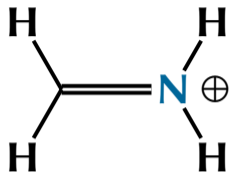
I'm interested in this molecule, but what is it doing?

The chemical problem



- ▶ Ultrafast dynamics after photo-excitation of **methylene immonium cation** CH_2NH_2^+
- ▶ Dynamics similar to ethylene? ($\pi\pi^*$ state, torsion around double bond)

The chemical problem



- ▶ Ultrafast dynamics after photo-excitation of **methylene immonium cation** CH_2NH_2^+
- ▶ Dynamics similar to ethylene? ($\pi\pi^*$ state, torsion around double bond)

Questions:

- ▶ Involved electronic states?
- ▶ Deactivation processes?
- ▶ Time scales?
- ▶ Photochemical products and yields?

- ① Defining the chemical problem
- ② **Choosing the level of theory**
- ③ Preparing the initial conditions
- ④ Setting up the trajectories
- ⑤ Running the trajectories



What numerical simulations can we use?

What do we need from the dynamics method?

- ▶ Nonadiabatic dynamics (interactions between electronic states)
- ▶ Feasibility (computational cost, user-friendliness through on-the-fly)

What do we need from the dynamics method?

- ▶ Nonadiabatic dynamics (interactions between electronic states)
- ▶ Feasibility (computational cost, user-friendliness through on-the-fly)

Possible dynamics methods:

- ▶ ~~Standard quantum dynamics~~
- ▶ ~~MCTDH~~
- ▶ Direct dynamics vMCG
- ▶ Ab initio multiple spawning/cloning
- ▶ **Surface hopping**
- ▶ ~~Born-Oppenheimer MD~~
- ▶ ~~Classical MD~~

What do we need from the electronic structure method?

- ▶ Accurate PESs
- ▶ Usable implementation
- ▶ Computationally feasible

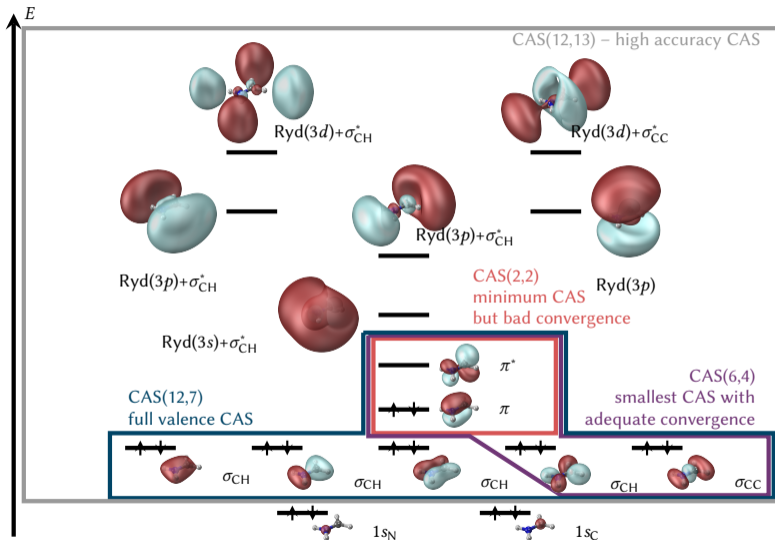
What do we need from the electronic structure method?

- ▶ Accurate PESs
- ▶ Usable implementation
- ▶ Computationally feasible

Possible electronic structure methods:

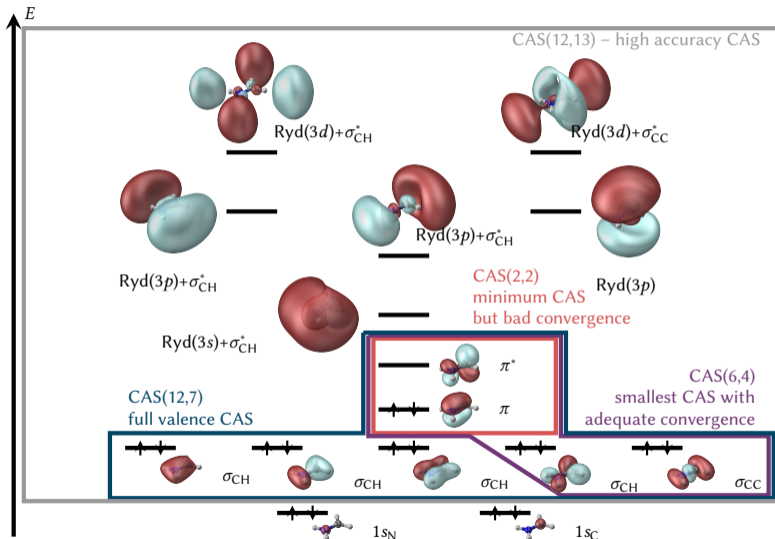
- ▶ CIS
- ▶ TDDFT
- ▶ ADC
- ▶ EOM-CC
- ▶ CASSCF
- ▶ MS-CASPT2
- ▶ **MRCI**

Active space



- ▶ **Minimal:** π and π^*
CAS(2,2)
- ▶ **Stable convergence:**
 π , π^* , σ_{CH} , σ_{CC}
CAS(6,4)
- ▶ **Full valence:** π , π^* , all σ
CAS(12,7)
- ▶ **Everything:** π , π^* , all σ , all $\sigma^*/$ Rydberg
CAS(12,13)

Active space



- ▶ **Minimal:** π and π^*
CAS(2,2)
- ▶ **Stable convergence:**
 π , π^* , σ_{CH} , σ_{CC}
CAS(6,4)
- ▶ **Full valence:** π , π^* , all σ
CAS(12,7)
- ▶ **Everything:** π , π^* , all σ ,
all $\sigma^*/Rydberg$
CAS(12,13)
- ▶ **Final level of theory:**
MRCISD(6,4)/aug-cc-
pVDZ

Vertical excitation energies

State	Sym. C_{2v}	Energy/eV				Oscillator strength
		MCSCF	MR-CISD	MR-CISD + Q	[14]	
Ground	1^1A_1	0.00 ^a	0.00 ^b	0.00 ^c	0.00	–
$\sigma_2-\pi^*$	1^1A_2	8.99	8.50	8.35	8.59	0.00
$\pi-\pi^*$	2^1A_1	10.21	9.43	9.17	9.37	0.35
$\pi-3s$	1^1B_1	10.93	11.45	11.50	11.45	0.05
$\pi-3p_y$	2^1A_2	11.97	12.50	12.55	13.01	0.00
$\sigma_1-\pi^*$	3^1A_2	12.56	12.70	12.63	–	0.00
$\pi-3p_z$	2^1B_1	12.78	13.10	13.10	–	0.00
σ_2-3p_y	3^1A_1	13.47	14.07	13.96	–	0.14
$\pi-3p_x$	4^1A_1	14.17	14.33	14.43	14.44	0.00

- ▶ Literature shows that there are only two important excited states for our problem

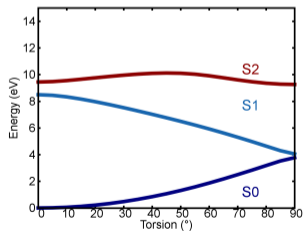
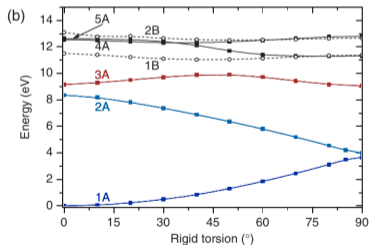
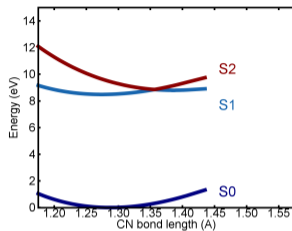
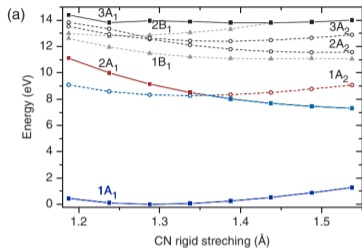
Vertical excitation energies

State	Sym. C_{2v}	Energy/eV				Oscillator strength
		MCSCF	MR-CISD	MR-CISD + Q	[14]	
Ground	1^1A_1	0.00 ^a	0.00 ^b	0.00 ^c	0.00	–
$\sigma_2-\pi^*$	1^1A_2	8.99	8.50	8.35	8.59	0.00
$\pi-\pi^*$	2^1A_1	10.21	9.43	9.17	9.37	0.35
$\pi-3s$	1^1B_1	10.93	11.45	11.50	11.45	0.05
$\pi-3p_y$	2^1A_2	11.97	12.50	12.55	13.01	0.00
$\sigma_1-\pi^*$	3^1A_2	12.56	12.70	12.63	–	0.00
$\pi-3p_z$	2^1B_1	12.78	13.10	13.10	–	0.00
σ_2-3p_y	3^1A_1	13.47	14.07	13.96	–	0.14
$\pi-3p_x$	4^1A_1	14.17	14.33	14.43	14.44	0.00

- ▶ Literature shows that there are only two important excited states for our problem
- ▶ **Three states are sufficient**

Barbatti, Aquino, Lischka: Mol. Phys., 104, 1053 (2006).

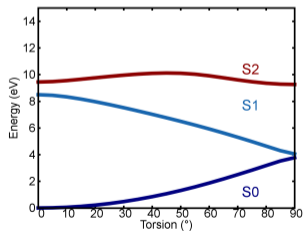
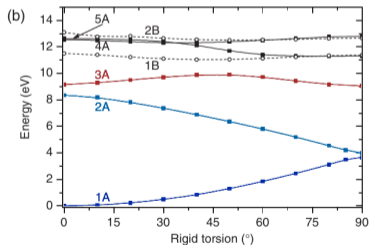
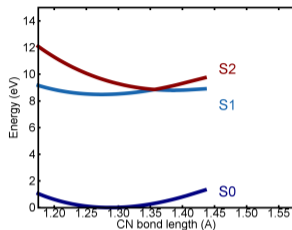
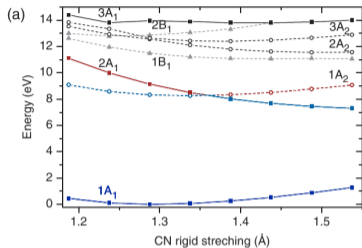
PES validation



- ▶ PES scans agree qualitatively with literature (slightly better level of theory)

Barbatti, Aquino, Lischka: Mol. Phys., 104, 1053 (2006).

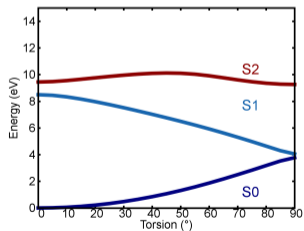
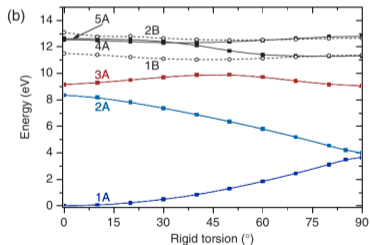
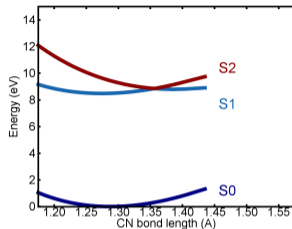
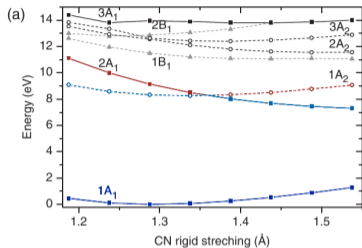
PES validation



- ▶ PES scans agree qualitatively with literature (slightly better level of theory)
- ▶ Should also compare to different spectra, structural data, ...

Barbatti, Aquino, Lischka: Mol. Phys., 104, 1053 (2006).

PES validation



- ▶ PES scans agree qualitatively with literature (slightly better level of theory)
- ▶ Should also compare to different spectra, structural data, ...
- ▶ Should explore PES as much as possible before dynamics simulations

Barbatti, Aquino, Lischka: Mol. Phys., 104, 1053 (2006).

- ① Defining the chemical problem
- ② Choosing the level of theory
- ③ **Preparing the initial conditions**
- ④ Setting up the trajectories
- ⑤ Running the trajectories



What process is it exactly that we want to simulate?

Sampling from Wigner distribution

Initial conditions:

- ▶ Define the “computational experiment”
- ▶ Molecule? Orientation? Energy? Excitation?

Sampling from Wigner distribution

Initial conditions:

- ▶ Define the “computational experiment”
- ▶ Molecule? Orientation? Energy? Excitation?

For pump-probe setups:

- 1 Prepare molecule in equilibrated ground state (distribution)
- 2 Excite molecule in defined way

Sampling from Wigner distribution

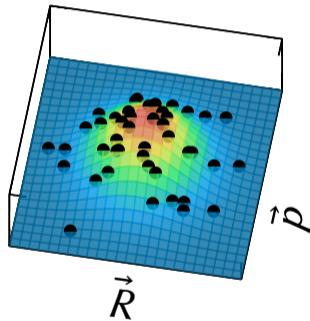
Initial conditions:

- ▶ Define the “computational experiment”
- ▶ Molecule? Orientation? Energy? Excitation?

For pump-probe setups:

- 1 Prepare molecule in equilibrated ground state (distribution)
- 2 Excite molecule in defined way

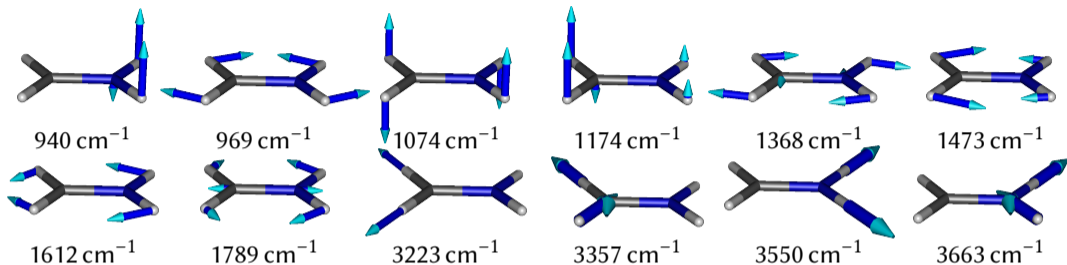
(1) Sample randomly from Wigner distribution:



Ground state vibrational wave function

- 1 Compute Hessian matrix at minimum geometry

Ground state vibrational wave function

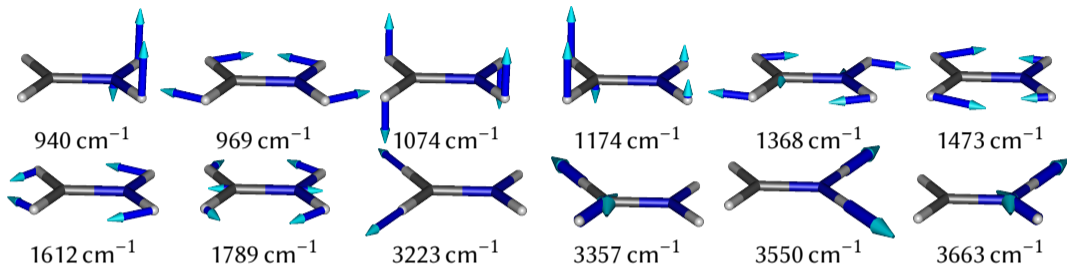


1 Compute Hessian matrix at minimum geometry

2 Find uncoupled normal mode coordinates \vec{Q} :

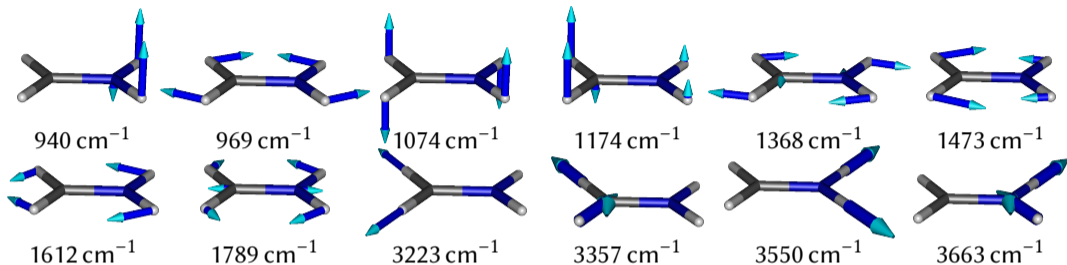
$$V(\vec{R}) = \sum_j \frac{\omega_j}{2} Q_j^2$$

Ground state vibrational wave function



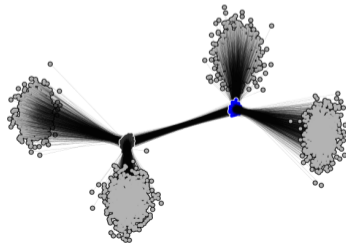
- 1 Compute Hessian matrix at minimum geometry
- 2 Find uncoupled normal mode coordinates \vec{Q} :
$$V(\vec{R}) = \sum_j \frac{\omega_j}{2} Q_j^2$$
- 3 Solve harmonic oscillator to get Wigner distribution

Ground state vibrational wave function



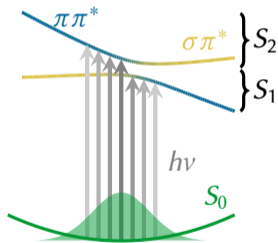
- 1 Compute Hessian matrix at minimum geometry
- 2 Find uncoupled normal mode coordinates \vec{Q} :
$$V(\vec{R}) = \sum_j \frac{\omega_j}{2} Q_j^2$$
- 3 Solve harmonic oscillator to get Wigner distribution

Result: **1000 geometries (with momenta)**



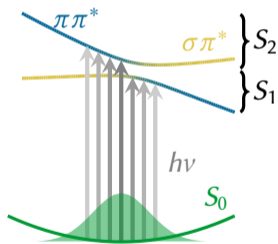
Vertical excitation spectrum

Compute excitations for
each of 1000 geometries:

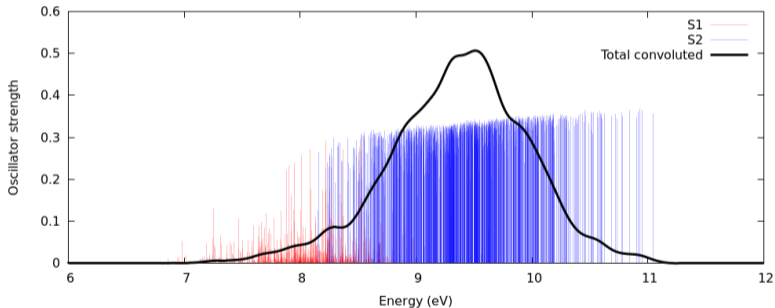


Vertical excitation spectrum

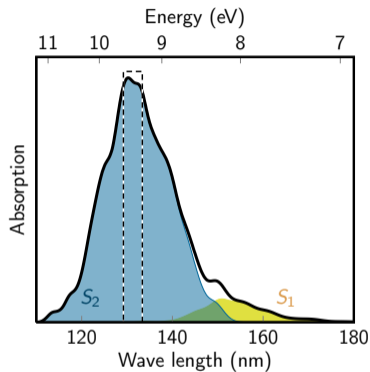
Compute excitations for each of 1000 geometries:



- ▶ Produces stick spectrum
- ▶ Convolute to obtain approximate band spectrum



Initial state selection

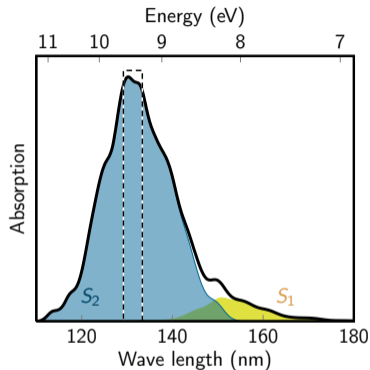


Excitation model:

- ▶ Implicit delta pulse

	Selected	In range
S_1	0	0
S_2	210	233

Initial state selection



	Selected	In range
S_1	0	0
S_2	210	233

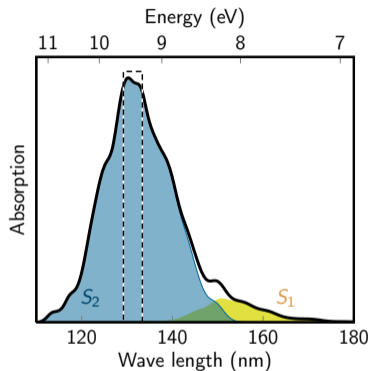
Excitation model:

- ▶ Implicit delta pulse

Initial state selection:

- ▶ Window 9.29–9.59 eV (129–133 nm)

Initial state selection



	Selected	In range
S_1	0	0
S_2	210	233

Excitation model:

- ▶ Implicit delta pulse

Initial state selection:

- ▶ Window 9.29–9.59 eV (129–133 nm)

Result: **210 initial conditions starting in S_2**

- ① Defining the chemical problem
- ② Choosing the level of theory
- ③ Preparing the initial conditions
- ④ **Setting up the trajectories**
- ⑤ Running the trajectories



Do we need to create all 1470 input files manually?

What needs to be considered?

Initial conditions

- ▶ One data set for each trajectory
- ▶ Add random number seeds

What needs to be considered?

Initial conditions

- ▶ One data set for each trajectory
- ▶ Add random number seeds

Dynamics settings

- ▶ Time step, length
 $\Delta t = 0.5 \text{ fs}$,
 $T_{\text{max}} = 100 \text{ fs}$
- ▶ Choice of numerical algorithms
- ▶ **Requires careful choices**

Some surface hopping reviews:

- ▶ Wang, Akimov, Prezhdo: J. Phys. Chem. Lett., 7, 2100(2016).
- ▶ Subotnik, Jain, Landry, Petit, Ouyang, Bellonzi: Annu. Rev. Phys. Chem., 67, 387 (2016).
- ▶ Crespo-Otero, Barbatti: Chem. Rev., 118, 7026 (2018).
- ▶ Mai, Marquetand, González: Chapter 16 in “Quantum Chemistry and Dynamics of Excited States: Methods and Applications”, Eds: González, Lindh (2020)

What needs to be considered?

Initial conditions

- ▶ One data set for each trajectory
- ▶ Add random number seeds

Dynamics settings

- ▶ Time step, length
 $\Delta t = 0.5 \text{ fs}$,
 $T_{\text{max}} = 100 \text{ fs}$
- ▶ Choice of numerical algorithms
- ▶ **Requires careful choices**

Electronic structure

- ▶ SA-CASSCF(6,4)/aug-cc-pVDZ with MRCISD
- ▶ Computer resources

Some surface hopping reviews:

- ▶ Wang, Akimov, Prezhdo: J. Phys. Chem. Lett., 7, 2100(2016).
- ▶ Subotnik, Jain, Landry, Petit, Ouyang, Bellonzi: Annu. Rev. Phys. Chem., 67, 387 (2016).
- ▶ Crespo-Otero, Barbatti: Chem. Rev., 118, 7026 (2018).
- ▶ Mai, Marquetand, González: Chapter 16 in “Quantum Chemistry and Dynamics of Excited States: Methods and Applications”, Eds: González, Lindh (2020)

What needs to be considered?

Initial conditions

- ▶ One data set for each trajectory
- ▶ Add random number seeds

Dynamics settings

- ▶ Time step, length
 $\Delta t = 0.5 \text{ fs}$,
 $T_{\text{max}} = 100 \text{ fs}$
- ▶ Choice of numerical algorithms
- ▶ **Requires careful choices**

Electronic structure

- ▶ SA-CASSCF(6,4)/
aug-cc-pVDZ with
MRCISD
- ▶ Computer resources

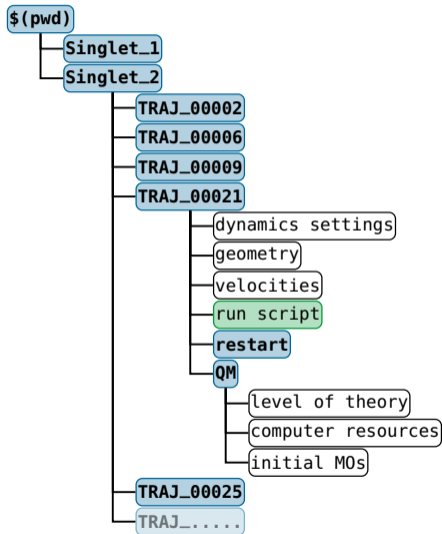
Output

- ▶ Amount of data
- ▶ Memory needs

Some surface hopping reviews:

- ▶ Wang, Akimov, Prezhdo: J. Phys. Chem. Lett., 7, 2100(2016).
- ▶ Subotnik, Jain, Landry, Petit, Ouyang, Bellonzi: Annu. Rev. Phys. Chem., 67, 387 (2016).
- ▶ Crespo-Otero, Barbatti: Chem. Rev., 118, 7026 (2018).
- ▶ Mai, Marquetand, González: Chapter 16 in “Quantum Chemistry and Dynamics of Excited States: Methods and Applications”, Eds: González, Lindh (2020)

Automatic trajectory management



Automatic trajectory management in SHARC:

- ▶ Tools combine all information
- ▶ Create 210 directories with several input files
- ▶ Each directory is completely independent

- ① Defining the chemical problem
- ② Choosing the level of theory
- ③ Preparing the initial conditions
- ④ Setting up the trajectories
- ⑤ Running the trajectories



How long does it compute and what does it cost?

$$T_{\text{complete}} = T_{\text{step}} \cdot \frac{T_{\text{max}}}{\Delta t}. \quad (12)$$

$$T_{\text{complete}} = T_{\text{step}} \cdot \frac{T_{\text{max}}}{\Delta t}. \quad (12)$$

In the present case:

$$T_{\text{complete}} = 342 \text{ sec} \cdot \frac{100 \text{ sec}}{0.5 \text{ sec}} = 19 \text{ h}. \quad (13)$$

Time to completion

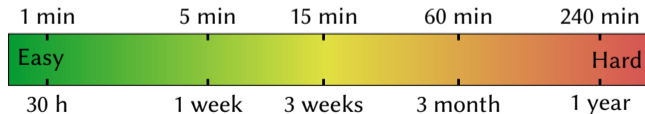
$$T_{\text{complete}} = T_{\text{step}} \cdot \frac{T_{\text{max}}}{\Delta t}. \quad (12)$$

In the present case:

$$T_{\text{complete}} = 342 \text{ sec} \cdot \frac{100 \text{ sec}}{0.5 \text{ sec}} = 19 \text{ h}. \quad (13)$$

- ▶ Takes longer if not enough CPU cores available ($N_{\text{cores}} < N_{\text{traj}}$)
- ▶ Very small project!

Time per timestep



Time for 1 ps (2000 steps)

$$T_{\text{cost}} = T_{\text{step}} \cdot \frac{T_{\text{max}}}{\Delta t} \cdot N_{\text{traj}} \quad (14)$$

$$T_{\text{cost}} = T_{\text{step}} \cdot \frac{T_{\text{max}}}{\Delta t} \cdot N_{\text{traj}} \quad (14)$$

In the present case:

$$T_{\text{cost}} = 342 \text{ sec} \cdot \frac{100 \text{ sec}}{0.5 \text{ sec}} \cdot 210 = 3990 \text{ hours.} \quad (15)$$

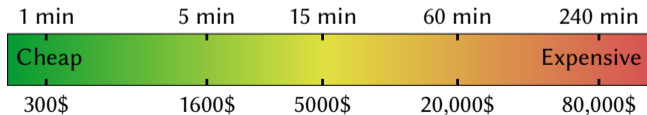
$$T_{\text{cost}} = T_{\text{step}} \cdot \frac{T_{\text{max}}}{\Delta t} \cdot N_{\text{traj}} \quad (14)$$

In the present case:

$$T_{\text{cost}} = 342 \text{ sec} \cdot \frac{100 \text{ sec}}{0.5 \text{ sec}} \cdot 210 = 3990 \text{ hours}. \quad (15)$$

- ▶ Very small project!
- ▶ Cost on Amazon AWS HPC: ≈ 0.10 \$/CPU hour, in total 400\$ (cheaper with own cluster).

Time per timestep



Cost for 100 trajectories and 2000 steps (0.10\$/CPU hour)

If T_{complete} is too large, project cannot be done.

If T_{complete} is too large, project cannot be done.

With N CPU cores in parallel, completion time decreases (diminishing returns):

$$T_{\text{complete}}(N) = T_{\text{complete}}(1) \left(1 - r + \frac{r}{N} \right) \quad (16)$$

The CPU core cost grows linearly:

$$T_{\text{cost}}(N) = T_{\text{cost}}(1) \left(1 - r + \frac{r}{N} \right) \cdot N \quad (17)$$

If T_{complete} is too large, project cannot be done.

With N CPU cores in parallel, completion time decreases (diminishing returns):

$$T_{\text{complete}}(N) = T_{\text{complete}}(1) \left(1 - r + \frac{r}{N} \right) \quad (16)$$

The CPU core cost grows linearly:

$$T_{\text{cost}}(N) = T_{\text{cost}}(1) \left(1 - r + \frac{r}{N} \right) \cdot N \quad (17)$$

Rules:

- 1 Only parallelize if r is close to 1. Use $N \approx \frac{r}{1-r}$.
- 2 Only parallelize if $T_{\text{complete}}(1)$ is very large.
- 3 Only parallelize if $N_{\text{cores}} \geq N_{\text{traj}}$.

Motivation to use SHARC (surface hopping including arbitrary couplings)

- ▶ SH: describes nonadiabatic transitions and branching, computationally/conceptually simple, parallelizable, easy to interpret
- ▶ ARC: generalize SH to ISC and eventually other processes

Summary

Motivation to use SHARC (surface hopping including arbitrary couplings)

- ▶ SH: describes nonadiabatic transitions and branching, computationally/conceptually simple, parallelizable, easy to interpret
- ▶ ARC: generalize SH to ISC and eventually other processes

The SHARC package

- ▶ SH requires initial condition generation, trajectory management, and statistical analyses
- ▶ All these steps need to be automatized

Motivation to use SHARC (surface hopping including arbitrary couplings)

- ▶ SH: describes nonadiabatic transitions and branching, computationally/conceptually simple, parallelizable, easy to interpret
- ▶ ARC: generalize SH to ISC and eventually other processes

The SHARC package

- ▶ SH requires initial condition generation, trajectory management, and statistical analyses
- ▶ All these steps need to be automatized

An example SHARC project: CH_2NH_2^+

- ▶ Defined chemical problem: Relaxation after photoexcitation to $\pi\pi^*$
- ▶ Surface hopping (no ISC, so regular) combined with MRCISD(6,4)/aug-cc-pVDZ
- ▶ Initial conditions from Wigner distribution and vertical excitation
- ▶ Setup involves several methodological choices
- ▶ Running the trajectories depends strongly on computational costs

Thank you for your attention!

Thank you for your attention!

My further thanks goes to:



universität
wien



FWF