SUPPORTING INFORMATION

QM/MM Nonadiabatic Dynamics: The SHARC/COBRAMM Approach

*Davide Avagliano‡ [a], Matteo Bonfanti‡ [b], Marco Garavelli[b] *, Leticia González[a,c] **

[a]Institute of Theoretical Chemistry, Faculty of Chemistry, University of Vienna, Währinger Straße 17, A-1180 Vienna, Austria.

[b] Università degli Studi di Bologna, Dipartimento di Chimica Industriale "Toso Montanari", Viale del Risorgimento, 4, Bologna, Italy

[c]Vienna Research Platform on Accelerating Photoreaction Discovery, University of Vienna, Währinger Straße 17, A-1180 Vienna, Austria.

Corresponding Authors

*marco.garavelli@unibo.it; *leticia.gonzalez@univie.ac.at

† Present address: Human Technopole, Milan, Italy

Page:

SECTION S1 ACROLEIN DYNAMICS PROTOCOL

S1.1 Generation of initial conditions

Acrolein's structure was optimized at the CASSCF(8,7)/cc-pVD[S1] level of theory, starting with an initial guess for the active space obtained from a Hartree-Fock calculation done with OpenMolcas[S2], see active orbitals in Fig. S1. A Wigner distribution at 300K was generated based on the frequency calculation at the optimized geometry. 250 geometries were extracted and solvated with Packmol[S3] with 1000 acetonitrile molecules. The systems were minimized for 500 steps with the steepest descent method and for 9.500 steps with conjugated gradient, then heated to 300K in 50 ps, and finally the volume and pressure equilibrated for additional 50 ps. The Nosè-Hoover thermostat [S4] and Berendsen barostat [S5] were used to keep constant T and P in the NVT and NPT ensembles. Periodic boundary conditions and Particle Mesh Ewald were used [S6]. The acrolein molecules were kept frozen during the whole procedure and MeCN molecules treated with GAFF force field. These calculations were run using AMBER [S7]. At the end of this solvent equilibration procedure, half of the molecules were stripped and a droplet of 500 MeCN centered at the acrolein molecule was obtained for each of the system. All these 250 systems were propagated in the ground state for 250 fs with SHARC/COBRAMM interface, calling OpenMolcas and AMBER for the computation of QM and MM energy and gradient. RATTLE [S8] algorithm was used, the external shell of 250 MeCN was kept frozen (lower layer, L), the other 250 MeCN molecule were allowed to move at MM (GAFF) level (medium layer, M) and acrolein was treated at CASSCF(8,7)/cc-pVDZ (higher layer, H).

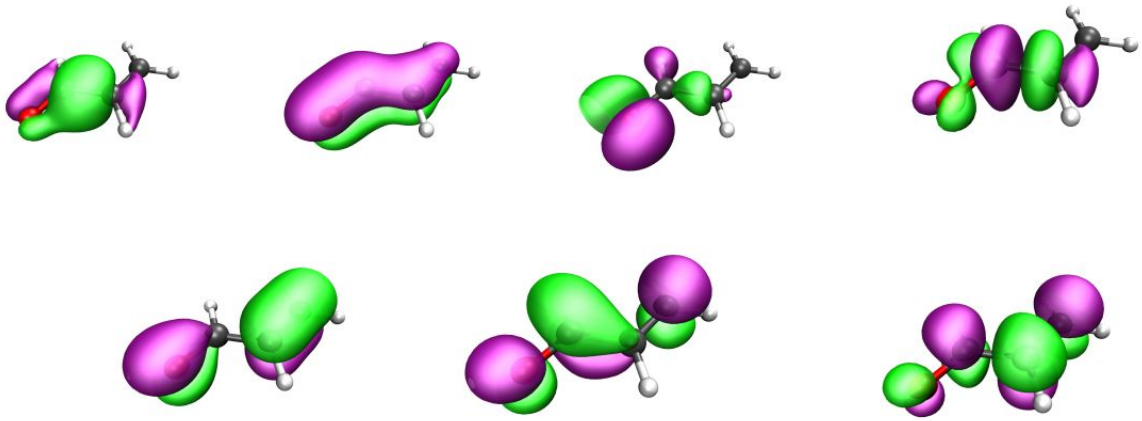

Figure S1: Orbitals included in CASSCF calculation for acrolein

S1.2 Excited states dynamics

After the ground state equilibration, 148 trajectories were selected stochastically and started in the S_1, then propagated for 500fs with a time step of 0.5fs. RATTLE and the atom mask were applied to the solvent molecules. Kinetic energy was reflected parallel to the velocity vectors, the electronic decoherence correction [S9] was applied and frustrated hop not reflected. The same H/M/L layers partition and level as the ground state SHARC/COBRAMM trajectories was kept. Two singlet and two triplet states were included.

S1.3 Total energy conservation along the TSH dynamics

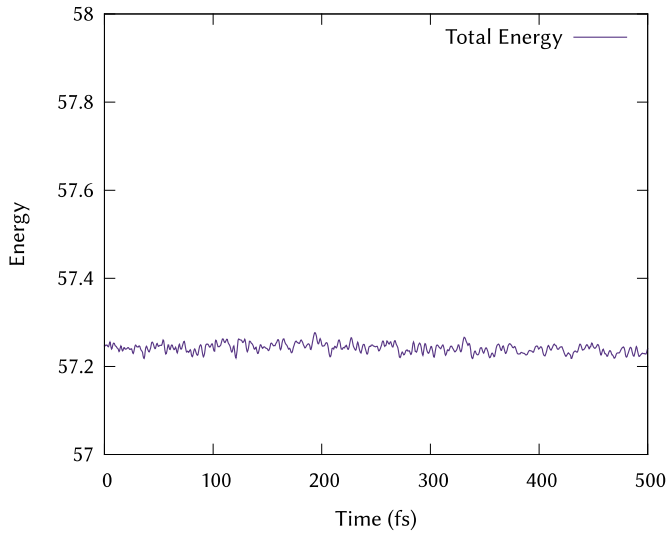

Figure S2: Total energy conservation along the 500fs dynamics of acrolein in acetonitrile for a representative trajectory

S1.4 Kinetic model

In order to estimate a kinetic model for the decay from S_1 state, we fitted the populations in a global procedure considering three possible decays. Four independent species were defined, one representing the population of each state and three reaction rates. Three possible reaction were defined: one relative to the intersystem crossing S_1 –> T_2 (reaction rate K_{ISC}), one for the internal conversion between the two triplets T_2 -> T_1 (reaction rate K_{IC_tripl}) and one between the singlets S_1 -> S_0 (reaction rate K_{IC_singl}). The initial populations were set to 1 for S_1 and null for all the other three states. Population of the states were described by:

$$S_0(\mathrm{t}) = \frac{K_{\mathrm{IC\_singl}} * \mathrm{S}_1(t_0)}{K_{ISC} + K_{\mathrm{IC\_singl}}} - \frac{K_{\mathrm{IC\_singl}} * \mathrm{S}_1(t_0) * e^{-(K_{ISC} + K_{\mathrm{IC\_singl}} * t)}}{K_{ISC} + K_{\mathrm{IC\_singl}}} \qquad \text{(eq S.1)}$$

$$S_1(\mathrm{t}) = \mathrm{S}_1(t_0) * e^{-(K_{ISC} + K_{\mathrm{IC\_singl}} * t)} \qquad \text{(eq S.2)}$$

$$T_1(\mathrm{t}) = \frac{K_{IC\_tripl} * K_{ISC} * \mathrm{S}_1(t_0) * e^{-(K_{ISC} + K_{\mathrm{IC\_singl}} * t)}}{K_{ISC}^2 + (2 * K_{IC\_singl} - K_{IC\_tripl}) * K_{ISC} - K_{IC\_singl} * K_{IC\_tripl} + K_{Ic\_singl}^2} - \frac{K_{ISC} * \mathrm{S}_1(t_0) * e^{-(K_{\mathrm{IC\_tripl}} * t)}}{K_{ISC} - K_{IC\_tripl} + K_{IC\_singl}} + \frac{K_{ISC} * S_1(t_0)}{K_{ISC} + K_{IC\_singl}}$$

(eq S.3)

$$T_2(\mathrm{t}) = \frac{K_{ISC} * \mathrm{S}_1(t_0) * e^{-(K_{\mathrm{IC\_tripl}} * t)}}{K_{ISC} - K_{IC\_tripl} + K_{IC\_singl}} - \frac{K_{ISC} * \mathrm{S}_1(t_0) * e^{-(K_{ISC} + K_{\mathrm{IC\_singl}} * t)}}{K_{ISC} - K_{IC\_tripl} + K_{IC\_singl}}$$

(eq S.4)

The time constants were optimized, as implemented in ScyPy, with the Trust Region Reflective constrain algorithm [S10]. The errors were estimated with bootstrapping methods [S11], generating 50 copies of the original ensemble of populations data and fitted as the original sample.

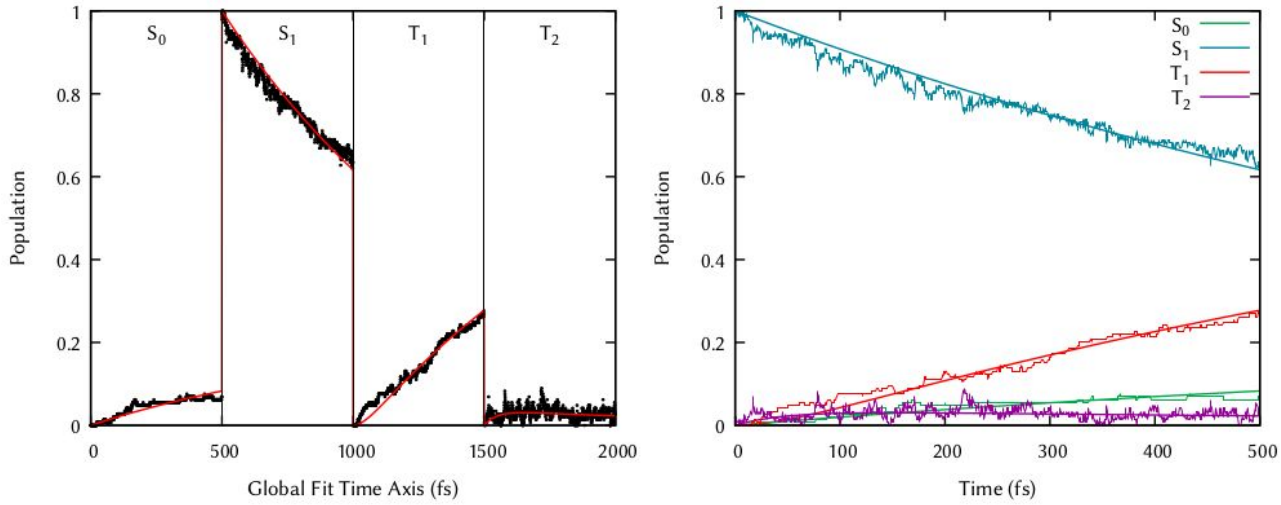

Figure S3: Fit of the four states population in the acrloein relaxation from S1 used to estimate the kinetic model

S1.5 Hopping geometries

Superimposition of 5 hopping geometries for the S1-> S0 and S1-> T2 relaxation. In pink representative structures in our ensemble of trajectories and in green the analogous structure from reference S12 (Ref. 62 of the main text). Only 5 representative geometries for each type are reported for graphical reasons and benefit of the understanding.

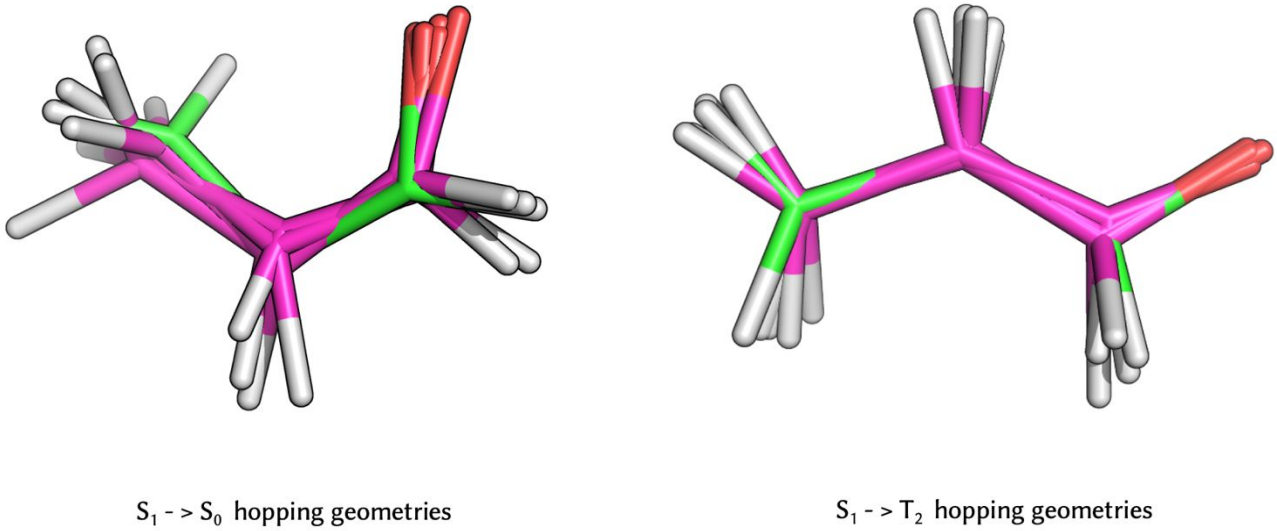

Figure S4 Hopping geometries for S1 relaxation to the ground state and ISC with T2

SECTION S.2 CODES DETAILS

S2.1 SHARC_COBRAMM.py interface

Relevant functionalities of the SHARC_COBRAMM.py interface are preparing COBRAMM calculation and return in SHARC format the information about the QM/MM calculation. The basic structure of SHARC_COBRAMM.py interface and the logic behind the code are the same as other SHARC interfaces. In particular, the interface reads the *COBRAMM.template*, the *COBRAMM.resources* and the *QM.in* files. It creates the *SCRATCH* directory and a save directory (*SAVEDIRQMMM*), where a coordinate file in AMBER format for the previous (*real.crd.old*) and current (*real.crd*) time step is written each time step. The interface writes the input file for COBRAMM, *cobram.command*, which contains the information about which *SHARC_QM.py* interface will be called, and run *cobram.py*. It writes the *QMMM.err* and *QMMM.log* files during the dynamics. Additionally, the interface copies the topology files in AMBER format for the total system (*real.top*) and

the QM part only (*model-H.top)* and the layers definition file (*real_layers.xyz)* required by COBRAMM in the correct location.

S2.2 sharcQMcalculator.py interface

sharcQMcalculator.py is called by cobram.py to run energy and gradient calculation of the QM part using a *SHARC_QM.py* interface. First of all, it writes the point charges relative to the MM part to be included in the QM calculation. Charges are written in a file called charges.dat, already correctly formatted for the specific QM software, specified in *COBRAMM.template*. *sharcQMcalculator.py* calls the *SHARC_QM.py* interface. Once the calculation is done, it reads the *QM.out*, given by the *SHARC_QM.py* interface only for the QM part, the file *gradient_charges*, including the gradient of the point charges induced by the QM region, reads the MM output and writes a complete *QM.out* for the whole system to be used by *sharc.x.*

S2.3 SHARC_QM.py modifications

Small modifications to each of the interfaces were required about reading the point charges file and returning the point charges gradient induced by the QM density. Each interface is adapted to the specific software but reads and writes the same files. All the interfaces currently available read the *charges.dat* file and print the *grad_charges* file. In case of ORCA and TURBOMOLE interfaces, the point charges gradient is explicitly written by the QM software, while in case of MOLCAS it is calculated from the electric field in the position of the point charges induced by the QM atoms, given by MOLCAS.

References

S1) Dunning, T. H. Gaussian Basis Sets for Use in Correlated Molecular Calculations. I. The Atoms Boron through Neon and Hydrogen. *J. Chem. Phys.* **1989**, *90* (2), 1007–1023.

S2) I. F. Galván, M. Vacher, A. Alavi, C. Angeli, J. Autschbach, J. J. Bao, S. I. Bokarev, N. A. Bogdanov, R. K. Carlson, L. F. Chibotaru, J. Creutzberg, N. Dattani, M. G. Delcey, S. Dong, A. Dreuw, L. Freitag, L. M. Frutos, L. Gagliardi, F. Gendron, A. Giussani, L. González, G. Grell, M. Guo, C. E. Hoyer, M. Johansson, S. Keller, S. Knecht, G. Kovačević, E. Källman, G. Li Manni, M. Lundberg, Y. Ma, S. Mai, J. P. Malhado, P. Å. Malmqvist, P. Marquetand, S. A. Mewes, J. Norell, M. Olivucci, M. Oppel, Q. M. Phung, K. Pierloot, F. Plasser, M. Reiher, A. M. Sand, I. Schapiro, P. Sharma, C. J. Stein, L. K. Sørensen, D. G. Truhlar, M. Ugandi, L. Ungur, A. Valentini, S. Vancoillie, V. Veryazov, O. Weser, T. A. Wesolowski, P.-O. Widmark, S. Wouters, A. Zech, J. P. Zobel, R. Lindh, *J. Chem. Theory Comput.* 15, 5925-5964, (2019),

S3) L. Martínez, R. Andrade, E. G. Birgin, J. M. Martínez. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of Computational Chemistry*, 30 (13) 2157-2164, 2009.

S4) Zhang, Zhijun; Liu, Xinzijian; Chen, Zifei; Zheng, Haifeng; Yan, Kangyu; Liu, Jian. A unified thermo-stat scheme for efficient configurational sampling for classical/quantum canonical ensembles via moleculardynamics.The Journal of Chemical Physics,2017,147, 034109.

S5) H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81(8):3684-90, 1984.

S6) P. P. Ewald, Ann. Phys. 1921, 369, 253–287

S7) Salomon-Ferrer, R.; Case, D. A.; Walker, R. C. An Overview of the Amber Biomolecular Simulation Package. *WIREs Comput. Mol. Sci.* **2013**, *3* (2), 198–210

S8) Andersen, H. C. Rattle: A “Velocity” Version of the Shake Algorithm for Molecular Dynamics Calculations. *J. Comput. Phys.* **1983**, *52* (1), 24–34

S9) Granucci, G.; Persico, M. Critical Appraisal of the Fewest Switches Algorithm for Surface Hopping. *J. Chem. Phys.* **2007**, *126* (13), 134114.

S10) Dormand, J.; Prince, P.J. Comput. Appl. Math.1980,6, 19 – 26.

S11) Nangia, S.; Jasper, A. W.; Miller, T. F.; Truhlar, D. G.J. Chem. Phys.2004,120, 3586–3597

S12) Cui, G.; Thiel, W. *J. Chem. Phys.* **2014**, *141* (12), 124101.