MANUAL

SHARC-MN: A Computer Program for Electronically Nonadiabatic Trajectories with Arbitrary Couplings

***SHARC-MN* is based on *SHARC* 2.1**

Yinan Shu
University of Minnesota

Linyao Zhang
Harbin Institute of Technology and University of Minnesota

Donald G. Truhlar
University of Minnesota

The current version is *SHARC-MN* 2.0
SHARC-MN 2.0 was finalized on April 6, 2023
SHARC-MN 2.0 is available at Zenodo [DOI:10.5281/zenodo.7818894]
This manual was last updated on April 11, 2023
SHARC-MN 2.0 is built on *SHARC* 2.1.
Most features of *SHARC-MN 2.0* will be merged into *SHARC 3.0*,
which is in preparation.
For recent versions of *SHARC*, see https://sharc-md.org

LICENSING 3

ACKNOWLEDGMENTS 3

CITATIONS 3

1 INTRODUCTION 4

1.1. INTRODUCTION TO SEMICLASSICAL EHRENFEST AND CSDM 7
1.3. HISTORY OF UPDATES 9
1.3.1 NEW FEATURES IN *SHARC-MN*–V1.0 (ADDITIONS TO *SHARC*–V2.1) 9
1.3.2 NEW FEATURES IN *SHARC-MN*–V1.1 9
1.3.3 NEW FEATURES IN *SHARC-MN*–V1.2 10
1.3.4 NEW FEATURES IN *SHARC-MN*–V2.0 10
1.4. REFERENCES: BASIC METHODS 11
1.5. REFERENCES: SPECIFIC METHODS 11
1.6. INSTALLATION 11

2 INPUT KEYWORDS 12

2.1 LIST OF KEYWORDS 12
2.2 DETAILED DESCRIPTIONS OF THE KEYWORDS 17
2.3 COMBINATIONS OF KEYWORDS 29

3 OUTPUT 30

4 AUXILIARY SCRIPTS 32

4.1 STATE SELECTED INITIAL CONDITION SAMPLING: STATE_SELECT.PY 32
4.1.1 BASIC USAGE 32
4.1.2 DESCRIPTION OF MAJOR OPTIONS 32
4.1.3 NORMAL MODE TYPES 36
4.1.4 NON-DEFAULT MASSES 36
4.1.5 OUTPUT 36

APPENDIX A. GNU GENERAL PUBLIC LICENSE V3 38

APPENDIX B. CC-BY-4.0 LICENSE 47

Licensing

SHARC 2.1 is licensed under the GNU general public license–v3. This license is reproduced in Appendix A of this manual. The license requires that if one distributes copies of *SHARC* 2.1, one must pass on to recipients the same freedoms that were received under the GNU published license, and we hereby do so.

SHARC-MN 2.0 is licensed under the GNU general public license–v3.

The *SHARC-MN 2.0* manual is licensed under CC-BY-4.0. This license is reproduced in Appendix B of this manual.

Acknowledgments

We requested permission from the authors of *SHARC* 2.1 to distribute a modified version of the code, and we were given permission. We are grateful to the authors of *SHARC* 2.1 for making their code available and for their cooperation every step of the way.

Some of the modifications in *SHARC-MN 2.0* are scheduled to be included in a future version of *SHARC* itself, in particular in *SHARC 3.0*.

Our work on *SHARC-MN 2.0* and *SHARC 3.0* was supported in part by the U. S. Department of Energy, Office of Basic Energy Sciences.

Citations

Users of *SHARC-MN 2.0* are requested to cite both *SHARC* and *SHARC-MN*. Examples of appropriate referencing are as follows:

1. Y. Shu, L. Zhang, and D. G. Truhlar, *SHARC-MN 2.0*: A Computer Program for Electronically Nonadiabatic Trajectories with Arbitrary Couplings (University of Minnesota, Minneapolis, 2023), DOI:10.5281/zenodo.7818894
2. S. Mai, M. Richter, M. Heindl, M. F. S. J. Menger, A. Atkins, M. Ruckenbauer, F. Plasser, L.M. Ibele, S. Kropf, M. Oppel, P. Marquetand, and L. González, *SHARC* 2.1: Surface Hopping Including Arbitrary Couplings – Program Package for Non-Adiabatic Dynamics, 2019, sharc-md.org

See also the references for methods in Sections 1.4 and 1.5.

1 INTRODUCTION

SHARC-MN is an extended version of *SHARC*. It includes all features of *SHARC 2.1* plus enhancements. Most of these enhancements will also be in *SHARC 3.0*, which is in preparation. Both *SHARC-MN* and *SHARC* are designed for direct dynamics calculations of electronically nonadiabatic processes. A direct dynamics calculation is one in which all needed energies, gradients, and nonadiabatic couplings (NACs) are calculated by performing electronic structure calculations as they are needed in the dynamics calculations.

When molecules and materials are electronically excited, the nuclear motion is affected by more than one potential energy surface. Processes involving more than one electronic state are called electronically nonadiabatic dynamics. The *SHARC* and *SHARC-MN* codes are used for semiclassical simulation of nonadiabatic dynamical processes, i.e., for computing processes where nuclear motion, electronic-state populations, and electronic coherences are affected by more than one potential energy surface as they evolve in time.

Within the semiclassical approach, one typically treats the nuclei as undergoing classical motion on an effective potential energy surface or as classical motion that switches between two or more potential energy surfaces, while in either case the electrons are treated quantum mechanically with the nuclear and electronic propagations being coupled. There are two main categories of methods:

- trajectory surface hopping (TSH): Nuclei are propagated at any one time on a single potential energy surface (PES) that is created by electronic structure, and the PES is switched stochastically from time-to-time as the probability of being in different electronic states changes; this switch is called a "hop". We refer the readers to the *SHARC* manual (https://sharc-md.org/?page_id=15) for detailed information about using *SHARC* or *SHARC-MN* for TSH calculations that employ the original fewest-switches algorithm or fewest switches with energy-based decoherence (TSH-EDC). The *SHARC-MN* program also includes additional TSH algorithms and options described in this manual.
- methods based on a self-consistent potential (SCP) in which nuclei are propagated on a mean-field PES. These methods are in *SHARC-MN* but not in *SHARC 2.1,* although most of them will be in *SHARC 3.0.* These methods are described in the present manual. *SHARC-MN 2.0* includes three classes of methods of this type:
  - semiclassical Ehrenfest (SE),
  - self-consistent decay of mixing (SCDM)
  - coherent switching with decay of mixing (CSDM)

Note that SE is not recommended for most practical applications because it does not produce physical final states. SCDM is an older version that CSDM and has been found to be less accurate than CSDM in comparison to converged quantum dynamics. Therefore, although *SHARC-MN* contains all three methods, the one that is recommended is CSDM.

There are two equations of motion (EOMs) for semiclassical nonadiabatic dynamics algorithms, namely the electronic EOM and the nuclear EOM.

Multi-state dynamics calculations may be carried out using in either the adiabatic or diabatic representation. In the adiabatic representation, the electronic Hamiltonian is diagonal, with the adiabatic potential energy surfaces (V_I) on the diagonal, and the coupling between states is semiclassical due to the nuclear momentum operator acting on the electronic wave function. The resulting coupling matrix elements are tensors having the form

$$\mathbf{d}_{IJ}(\mathbf{R}) = \left\langle \phi_I \left| \frac{\partial}{\partial \mathbf{R}} \right| \phi_J \right\rangle$$

where ϕ_I and ϕ_J are electronic wave functions, **R** denotes the nuclear coordinates, $\partial/\partial\mathbf{R}$ is the gradient operator, and $\mathbf{d}_{IJ}$ is called a nonadiabatic coupling vector or a NAC. In a diabatic representation, which is not unique, the electronic Hamiltonian is not diagonal, and the effect of the NACs is assumed to be negligible compared to the effect of the off-diagonal elements of the electronic Hamiltonian; this can be a good approximation if the electronic wave functions are smooth functions of **R**. The electronic Hamiltonian in a diabatic representation is called the diabatic potential energy matrix (DPEM, **U**); the diagonal elements are the diabatic potential energy surfaces (U_{II}), and the off-diagonal elements are called the diabatic couplings (U_{IJ}).

Based on the ingredients (i.e., inputs) required by the electronic and nuclear EOMs, we distinguish between three types of algorithms for semiclassical nonadiabatic dynamics implemented in *SHARC-MN*. This classification applies to both TSH and SCP methods.

- Algorithms based on NACs. In these algorithms, NACs are computed and used whenever required in the nuclear and/or electronic EOMs.
- Algorithm based time derivatives. In some methods (e.g., the fewest-switches TSH method), only the NAC component along the nuclear velocity vector ($\dot{\mathbf{R}}$) is required. This component can be re-written as a time derivative coupling (TDC) , which is defined as

  $$\mathbf{T}_{IJ}(\mathbf{R}) = \left\langle \phi_I \left| \frac{d}{dt} \right| \phi_J \right\rangle = \mathbf{d}_{IJ}(\mathbf{R}) \cdot \dot{\mathbf{R}}$$

  where *t* is the time. This can be approximated by overlaps of electronic adiabatic wave functions at successive time steps. Alternatively, one can employ the overlap integrals to construct a locally diabatic Hamiltonian. In both cases, the computation of NACs (i.e., of all the components of the NAC vectors) can be avoided.
- Algorithms based on the second derivatives (curvatures) of the adiabatic potential energy surfaces. In curvature-driven methods, one approximates the TDC in terms of the curvatures of the adiabatic potential energy surfaces. Employing such an approximation avoids computation of both NACs and overlaps. The curvature-driven algorithms are more efficient than overlap-based algorithms or NAC-based algorithm, but they have been found to have similar accuracy. An advantage of curvature-driven algorithms, in addition to efficiency, is that they can be interfaced with electronic structure methods for which the analytical implementation of NACs is not available or the wave function is not defined. Curvature-driven algorithms are available in *SHARC-MN 2.0* but not in *SHARC* 2.1.

Although nuclear EOM of SCP methods involve the whole NAC vector, not just the component in eq 2, we have defined an effective NAC that can be used in place of the real NAC when one uses time-derivative or curvature-driven methods.

The NACs that are usually used for direct dynamics are computed in the laboratory frame, and they do not conserve angular momentum or the position of the center of mass. *SHARC-MN* corrects this problem by using a projection operator to remove the translational and rotational components of the originally computed NACs, in effective NACs, and in the momentum adjustment vectors and frustrated hop reflection vectors of TSH methods.

The **first set of major extensions** of *SHARC* are included in all versions of *SHARC-MN* and consist of

- the inclusion of the SE, SCDM, and CSDM methods

- NAC projection
- the incorporation of adaptive time steps
- addition of the time-derivative versions of the SE, SCDM, and CSDM methods; these methods are called are tSE, tSCDM, and tCSDM

Among the methods available in all versions of SHARC-MN, we recommend tCSDM. Not only is it more convenient and more efficient to use time derivatives rather than NACs, but also the time-derivative modification allows more accurate integrations in regions where the NACs show sharp spikes.

The **second set of major extensions** (included in versions 1.1 and later) consists of adding the curvature-driven versions of the SE, CSDM, and TSH methods. These are called κSE, κCSDM, and κTSH. Among these, κCSDM is recommended. The curvature-driven modification allows one to use curvatures of the adiabatic potential energy surfaces along the nuclear-motion path instead of NACs or time derivatives. This is most convenient of all. Note that there was a bug in the numerical-second-derivative-of-energy option of the first implementation of κTSH in SHARC-MN v1.2, but this is corrected in *SHARC-MN 2.0*.

The **third set of major extensions** (included in *SHARC-MN 2.0* and later)) consists of adding TSH with fewest switches time uncertainty algorithm (TSH-FSTU), a new gradient correction scheme for intersystem crossing processes, a new option for the momentum adjustment vector at hops in TSH methods, a new frustrated-hop velocity-reflection criterion, and new initial-condition sampling methods.

The following paragraphs provide more details of the new options in *SHARC-MN 2.0*:

- The TSH-FSTU method is a variant of TSH that allows one to perform a nonlocal hop when a frustrated hop is encountered in fewest-switches TSH. One cannot completely remove the frustrated hops in TSH algorithms, but the FSTU option significantly reduces the number of frustrated hops.
- The new gradient correction scheme is implemented for simulations of intersystem crossing processes. In all available electronic structure codes, when spin-orbit coupling is present, analytic gradients are not available for calculations in the fully adiabatic representation (which SHARC calls the diagonal representation); the only gradients that are directly available in electronic structure software are gradients in the spin-diabatic representation (which SHARC calls the molecular Coulomb Hamiltonian (MCH) representation). And therefore, in *SHARC* before *SHARC 3.0* and in *SHARC-MN* before *SHARC-MN 2.0*, the gradients in the fully adiabatic (diagonal) representation are approximated by neglecting the nuclear derivative of the spin-orbit couplings and correcting gradients of the spin-diabatic (MCH) representation with NACs. This scheme is denoted as nuclear gradient tensor (NGT) scheme. The new gradient correction scheme, named the time-derivative-matrix (TDM) scheme, is especially useful for overlap-based algorithms and curvature-driven algorithms. As compared to the NGT scheme, the TDM scheme approximates the time derivative of potential energies in the fully adiabatic (diagonal) representation by neglecting the time-derivative of the spin-orbit couplings and corrects the time-derivative of potential energies of the spin-diabatic (MCH) representation by TDCs. One can further approximate the gradients in in the fully adiabatic (diagonal) representation by knowing the time-derivative of potential energies and gradients in the spin-diabatic (MCH) representation with the chain rule. A big advantage of overlap-based algorithms and curvature-driven algorithms is that

these algorithms do not need NACs, and the TDM scheme is able to retain that advantage by obtaining the gradients in the fully adiabatic (diagonal) representation without computing NACs.

- The decoherence correction for SCDM and CSDM methods can now be performed on diagonal states, i.e. – the pointer state basis can be set to the diagonal basis.
- Additional momentum adjustment vector options have been implemented for a TSH trajectory that undergoes a hop. In total, a user has seven options available: velocity vector, projected velocity vector (vibrational velocity vector), NAC, projected NAC, gradient difference vector, effective NAC, and projected effective NAC. Notice that if one does not have enough momentum to adjust the velocity vector along the chosen direction, the hop will be frustrated. Thus the momentum adjustment vector affects not just the momentum after a hop but also helps to determine whether a hop is allowed.
- When a TSH trajectory encounters a frustrated hop, one can chose between ignoring the frustrated hop or reflecting the velocity along a certain direction based on input criteria. There are two criteria, the SHARC criterion which is in both SHARC v2.1 and SHARC-MN, and the ∇V criteria, which is in SHARC-MN starting with version 2.0 and which will be n SHARC staring with version 3.0. If the chosen criterion is satisfied, one performs the reflection of the velocity along a certain direction. There are now seven choices of such directions: velocity vector, projected velocity vector (vibrational velocity vector), NAC, projected NAC, gradient difference vector, effective NAC, and projected effective NAC.
- State-selected initial conditions for unimolecular process are now implemented with auxiliary python scripts named state_selected.py. With state-selected initial conditions, one can mandate specific amounts of initial energies or specific initial vibrational quantum numbers in selected normal modes. As for a previously available initial-condition auxiliary python script wigner.py, which is used to perform Wigner distribution sampling, the script state_selected.py can be used with a *Molden* file from a frequency calculation.

1.1. Introduction to semiclassical Ehrenfest and CSDM

Semiclassical Ehrenfest (SE), like surface hopping, is a semiclassical method in which the nuclear wave packet motion is approximated by set of classical trajectories. However, in contrast to surface hopping, the nuclei in semiclassical Ehrenfest dynamics are propagated on a self-consistent potential that is unique to each trajectory. The program obtains the self-consistent potential by averaging the PESs or the PESs and NACs or the PESs and effective NACs over the included electronic states with weights given by the time-dependent electronic density matrix, which is propagated as part of the algorithm.

The advantages of semiclassical Ehrenfest dynamics are:

- The method is derived from the time-dependent Schrödinger equation by a self-consistent field approximation. Employing the self-consistent potential is a natural way to simulate the unitary (fully coherent) motion in which the electronic wave function evolves with decoherence as a linear combination involving multiple electronic states.[1]

- Employing the self-consistent potential makes semiclassical Ehrenfest dynamics invariant with respect to the choice of electronic-state representation, which may be either adiabatic states or diabatic states.
- As in the original surface hopping method, the propagation of the classical trajectories only requires local information about the PESs and couplings, and the nuclear equations of motion follow Newtonian mechanics on an effective potential (although the motion is coupled to electronic motion in a way that has no classical analog).

However, semiclassical Ehrenfest dynamics is notoriously bad for long-time propagation as the trajectory may propagate on unphysical averaged PESs after leaving a strong interaction region, e.g., a region with locally avoided crossing of adiabatic PESs. This poor behavior is due to the lack of decoherence.[2,3,4] In molecules and materials, the electronic density matrix is a reduced density matrix for the electronic subsystem embedded in a "bath" of nuclei, and the nuclear motion causes decoherence of the electronic density matrix. Decoherence causes the density matrix to tend to a diagonal form in an environmentally selected basis called the pointer basis. The robustness of semiclassical Ehrenfest dynamics with respect to the choice of representation is important because in general one does not know which representation (adiabatic, diabatic, or something else) is closest to the pointer basis,[5,6] although one does know that the adiabatic basis is the pointer basis when the adiabatic approximation is a good approximation.

Coherent switching with decay of mixing (CSDM) adds non-Markovian decoherence to the semiclassical Ehrenfest method so that the electronic coherences (i.e., the off-diagonal elements of the electronic density matrix) decay to zero in the assumed pointer basis after one leaves a region of strongly coupled potential energy surfaces.[4,7] *SHARC-MN* assumes that the pointer basis is the adiabatic one for internal conversion processes and the fully adiabatic (diagonal) one for intersystem crossing processes; this is a reasonable assumption for many cases. As a consequence, a CSDM trajectory propagates on a single PES in a pure adiabatic state in asymptotic regions or other regions far from strong interaction regions.

Self-consistent decay of mixing (SCDM), like SE and CSDM, is a self-consistent potential method, and it is an older version [3] of CSDM with a less accurate (but simpler) treatment of decoherence. We recommend CSDM over SCDM because it is more accurate, and therefore we de-emphasize SCDM in the manual. The user who is interested in SCDM (for historical reasons or because it is simpler) should note, however, that any option that is available for CSDM is also available for SCDM.

!!

1.2. Important note

To understand *SHARC-MN,* users must read the original *SHARC* manual. This manual only has descriptions of methods and keywords that are added to the S*HARC* code.

!!

1.3. History of Updates

1.3.1 New features in *SHARC-MN*–v1.0 (additions to *SHARC*–v2.1)

These features are new in *SHARC-MN*–v1.0 (2020):

- In the dynamics program **sharc.x**:
  - New methods:
    - SE (semiclassical Ehrenfest) dynamics
    - CSDM (coherent switching with decay of mixing) dynamics, which adds non-Markovian decoherence to SE)
    - SCDM (Semiclassical decay of mixing) dynamics, which is an older, older, less accurate, but simpler version of CSDM
  - New versions of SE, CSDM, and SCDM dynamics based on overlap (time-derivative couplings):
    - tSE (time-derivative SE)
    - tCSDM (time-derivative CSDM)
    - tSCDM (time-derivative SCDM)
  - Adaptive time-step propagation:
    - adaptive Velocity Verlet
    - Bulirsch-Stoer
  - Projected NAC for conserving of nuclear angular momentum and the center of mass in direct dynamics with TSH, SE, CSDM, or SCDM
- Data extraction:
  - New options in **data_extractor.x**
- Auxiliary scripts:
  - **setup_traj.py** works for setting up both surface hopping and self-consistent potential methods, where the self-consistent potential method may be SE, CSDM, or SCDM.

1.3.2 New features in *SHARC-MN*–v1.1

These features are new in *SHARC-MN*–v1.1 (2021):

- In the dynamics program **sharc.x**:
  - New methods: curvature-based approximation of time derivative coupling. This involves a series of new methods, including
    - κCSDM (curvature-driven CSDM)
    - κSE (curvature-driven SE)
    - κTSH (curvature-driven TSH)
    - κTSH-EDC (curvature-driven TSH-EDC)

    These nonadiabatic methods do not require explicit computation of nonadiabatic coupling or wave function overlap. Therefore, they only require evaluation of adiabatic energies (potential energy surfaces) and their gradients at each time step. Note that a bug in κTSH and κTSH-EDC is corrected in *SHARC-MN 2.0*.
- Auxiliary scripts:
  - **setup_traj.py** added the curvature-based approximation option for coupling, the keyword is “etdc”.

1.3.3 New features in *SHARC-MN*–v1.2

These features are new in *SHARC-MN*–v1.2 (2022):

- In the dynamics program **sharc.x**:
  - There are now two choices for evaluation of the curvature-driven coupling in the curvature-driven methods:
    - first-order difference of dot product of gradients and velocity vector
    - second-order finite difference of energy [There was a bug in this option in v1.2, but it is corrected in version 2.0.]

    This option can be controlled with new keyword, **ktdc_method**. The default setup for κTSH is that κTDC is computed by second order difference of energy; the default for κCSDM is that κTDC is computed by first order difference of dot product of gradients and velocity vector.
  - For the equations of motion of the electronic coefficients, the program now uses the decoherence time at time step $t + \Delta t$. This is found to conserve energy better than the original implementation. See the treatment of τ in eqs (3a) ad (3b) of section IV.A; we now use $\tau\left(t+\Delta t\right)$ instead of $\tau\left(t\right)$.
  - For TSH, κTSH, TSH-EDC, and κTSH-EDC, one can now adjust the kinetic energy and reflected momentum after a frustrated hop in directions of effective nonadiabatic coupling vector and projected effective nonadiabatic coupling vector (as explained in Section II.B).
- Auxiliary scripts:
  - **setup_traj.py** changed the curvature-based approximation option for coupling, the keyword is now changed to “ktdc”.

1.3.4 New features in *SHARC-MN*–v2.0

These features are new in *SHARC-MN*–v2.0 (2023):

- In the dynamics program **sharc.x**:
  - One can perform TSH-FSTU algorithm for all types of TSH algorithms with an additional keyword, **time_uncertainty**. In the default setup, time uncertainty is turned off.
  - One can perform the new gradient correction scheme for intersystem crossing. One can employ the TDM scheme by using the keyword **gradcorrect tdm** or **gradcorrect kmatrix**. The original correction scheme – the NGT scheme – can be selected by using keyword **gradcorrect** or **gradcorrect ngt** or **gradcorrect nac**.
  - The pointer basis can be set to either MCH or diagonal representation by using keywords **pointer_basis MCH** and **pointer_basis diag**. The default is **pointer_basis diag**.
  - The spin-orbit coupling can be scaled by calling **soc_scaling X**, where **X** is a user defined scaling factor.
- Auxiliary scripts:
  - **setup_traj.py** has new options that can incorporate new capabilities in interface python scripts. For example, one can interface *SHARC-MN* with

compressed multi-state pair density functional theory (CMS-PDFT) implemented in *OpenMolcas*.

- A new initial condition sampling auxiliary script called **state_selected.py** is added.

1.4. References: basic methods

The following references may be cited for the basic methods in *SHARC-MN*:

- S. Mai, P. Marquetand, L. González: "Nonadiabatic dynamics: The SHARC approach". WIREs Comput. Mol. Sci., 8, e1370 (2018).
- Y. Shu, L. Zhang, S. Mai, S. Sun, L. González, D. G. Truhlar: "Implementation of Coherent Switching with Decay of Mixing into the SHARC program". J. Chem. Theory Comput., 16, 3464-3475 (2020).

In addition, users may cite the references in Section I.5 for specific dynamical methods.

1.5. References: specific methods

- semiclassical Ehrenfest (SE): [8,9]
- coherent switching with decay of mixing (CSDM): [4,9]
- overlap-based methods, tSE and tCSDM: [10]
- projected NAC: [11]
- curvature-driven methods, κCSDM, κSE, κTSH, and κTSH-EDC: [12]
- time uncertainty algorithm, TSH-FSTU, TSH-FSTU-EDC: [13,14]
- time derivative Hamiltonian gradient correction scheme: [15]

1.6. Installation

Installation of *SHARC-MN* requires system-specific modification of **Makefile**, which is located in the **source** folder. Installation is done by two commands from the **source** folder:

make
make install

The *SHARC-MN* executable is **sharc.x**; the output data extractor is called **data_extractor.x**.

2 INPUT KEYWORDS

In *SHARC-MN*, we tried to maintain consistency with *SHARC*. Therefore, except for the exception in the next bullet point, all inputs that work for *SHARC* should execute the identical calculation in *SHARC-MN*. Note that maintaining consistency with SHARC means that the default value of some keywords is not the recommended value.

There is only one exception to compatibility with *SHARC*, namely that we have set the default to perform NAC projection, which ensures the conservation of nuclear orbital angular momentum and center of mass motion. One can turn NAC projection off by using keyword **no_nacprojection**.

2.1 List of keywords

Table 2.1 Input keywords for sharc.x. The first column gives the name of the keyword, the second lists possible arguments, and the third column provides an explanation. Defaults are marked in green like **this**. $n denotes the n-th argument to the keyword.

The table gives keywords that are new or have new options in *SHARC-MN* as compared to *SHARC*. Users should refer to Table 4.1 in the original *SHARC* manual for other keywords.

When two keywords are given (e.g., $1=ddr, nacdr), it means that both keywords have the same effect, and one may use either one.

Note that there is no difference between the MCH basis and the diagonal basis when spin-orbit coupling is neglected.

Keyword	Arguments	Explanation
method	**string** $1=**tsh** $1=scp, ehrenfest	Set the simulation method. Use trajectory surface hoping (TSH). Use a self-consistent potential scheme (SE, CSDM, or SCDM). Either $1=scp or $1=ehrenfest will use a self-consistent potential; these two choices of keyword have the same meaning. Note that CSDM and SCDM are both SCP method; they combine semiclassical Ehrenfest with decay of mixing and with pointer-state switching.
nsubsteps	**integer** **25**	Number of substeps for the integration of the electronic equation of motion. This is equal to the ratio of the step size for integrating the nuclear equations of motion to the step size for integrating the electronic equations. It is recommended to use a large number of substeps (e.g., 200) when performing SE and CSDM.

coupling	**string** $1=ddr, nacdr $1=ddt, nacdt $1=**overlap** $1=ktdc	*Ab initio* quantities describing the inter electronic states couplings. Uses vectorial nonadiabatic couplings $\langle\phi_I\vert\,\partial/\partial\mathbf{R}\,\vert\phi_J\rangle$. Uses time-derivative couplings $\langle\phi_I\vert\,\partial/\partial t\,\vert\phi_J\rangle$ (Notice that since no electronic structure software is available to compute the TDC directly, this option is basically not used). Uses the overlaps of electronic wave functions $\langle\phi_I(t-\Delta t)\vert\phi_J(t)\rangle$ to approximate TDC. Uses the curvature-driven approximation for the time-derivative coupling $\langle\phi_I\vert\,\partial/\partial t\,\vert\phi_J\rangle$. This option is called κTDC.
ktdc_method (applies only to curvature-driven methods)	**string** $1=**energy** **(default for TSH)** $1=**gradient** **(default for SCP)**	Method used to compute κTDC Second-order finite difference of energy gap(default for TSH) First-order finite difference of dot product of gradients and velocity vector (default for SCP)
decoherence_scheme	**string** $1=**none** $1=edc $1=afssh $1=dom	Method for decoherence correction. No decoherence correction. Energy-difference-based decoherence correction for TSH. The TSH-EDC scheme is explained in Refs. [16,17] and is based on an earlier approximation [4] to the decoherence time in terms of energy gaps and nuclear kinetic energies. Augmented FSSH for TSH (refer to original SHARC manual for details of augmented FSSH) Add decay-of-mixing decoherence terms to SE to perform CSDM or SCDM.
switching_procedure (applies only to CSDM and SCDM methods)	**string** $1=**csdm** $1=scdm	Method for scheme used to switch the pointer state in decay-of-mixing calculations Coherent switching with decay of mixing.[4,8] Self-consistent decay of mixing.[3,8]
neom (applies only to SCP methods)	**string** $1=**ddr, nacdr** $1=**gdiff**	Method for propagating nuclear equations of motion for self-consistent-potential methods (SE, CSDM, or SCDM) Use full NAC (this is the default when coupling is set to ddr or nacdr) Use effective NAC, which is a combination of difference gradient vector and velocity (this is the default when coupling is set to overlap)

eeom	**string** $1=ci $1=**li** **(default for coupling ddr or coupling ktdc)** $1=**ld** **(default for TSH methods and coupling overlap)** $1=**npi** **(default for SCP methods and coupling overlap)**	Method for propagating electronic equations of motion. We suggest that only advanced users with specialized objectives should change this option from the default. constant interpolation Linear interpolation. This is the default when using coupling ddr or coupling ktdc. Local diabatization. This is the default when using set coupling keyword to overlap Norm preserving interpolation.[10]
time_uncertainty		Employ TSH-FSTU algorithm. Default is off.
pointer_basis (applies only to SCP methods)	**string** $1=**diag** $1=MCH	Method to control the pointer basis employed in decay-of-mixing algorithms. Use diagonal basis as the pointer basis. Use MCH basis as the pointer basis.
ekincorrect (applies only to TSH methods)	**string** $1=none $1=**parallel_vel** $1=parallel_pvel $1=parallel_nac $1=parallel_diff $1=parallel_pnac $1=parallel_enac $1=parallel_penac	Adjustment of the kinetic energy after a surface hop. Kinetic energy is not adjusted. Jumps are never frustrated. Not recommended. Velocity is rescaled to adjust kinetic energy. Only the velocity component in the direction of vibrational motion is rescaled. Only the velocity component in the direction of $\langle\phi_I\vert\,\partial/\partial\mathbf{R}\,\vert\phi_J\rangle$ is rescaled. Only the velocity component in the direction of $\Delta\nabla E$ is rescaled. Only the velocity component in the direction of projected NAC is rescaled. Only the velocity component in the direction of effective NAC is rescaled. Only the velocity component in the direction of projected effective NAC is rescaled.

reflect_frustrated (applies only to TSH methods)	**string** $1=**none** $1=parallel_vel $1=parallel_pvel $1=parallel_nac $1=parallel_diff $1=parallel_pnac $1=parallel_enac $1=parallel_penac $1=delV_vel $1=delV_pvel $1=delV_nac $1=delV_diff $1=delV_pnac $1=delV_enac $1=delV_penac	Adjustment of the kinetic energy after a frustrated hop. No reflection. Full velocity vector is reflected. Only the velocity component in the direction of vibrational motion is reflected. Only the velocity component in the direction of $\langle\phi_I\vert\,\partial/\partial\mathbf{R}\,\vert\phi_J\rangle$ is reflected. Only the velocity component in the direction of $\Delta\nabla$E is reflected. Only the velocity component in the direction of projected NAC is reflected. Only the velocity component in the direction of effective NAC is reflected. Only the velocity component in the direction of projected effective NAC is reflected. Full velocity vector is reflected according to ∇V criteria. Only the velocity component in the direction of vibrational motion is reflected according to ∇V criteria. Only the velocity component in the direction of $\langle\phi_I\vert\,\partial/\partial\mathbf{R}\,\vert\phi_J\rangle$ is reflected according to ∇V criteria. Only the velocity component in the direction of $\Delta\nabla$E is reflected according to ∇V criteria. Only the velocity component in the direction of projected NAC is reflected according to ∇V criteria. Only the velocity component in the direction of effective NAC is reflected according to ∇V criteria. Only the velocity component in the direction of projected effective NAC is reflected according to ∇V criteria.
integrator	**string** $1=**fvv** $1=avv $1=bsh	Method to control the integrator. fixed time-step Velocity Verlet integrator Adaptive time-step Velocity Verlet integrator (Note that adaptive integrator does not work with **coupling overlap**) Bulirsch-Stoer integrator (Note that adaptive integrator does not work with **coupling overlap**)
soc_scaling (applies only to intersystem crossing processes)	**float** $1=**1.0**	Spin-orbit coupling scaling factor.

gradcorrect **nogradcorrect**		Include $(E_I - E_J)\langle\phi_I\vert\,\partial/\partial\mathbf{R}\,\vert\phi_J\rangle$ in gradient transformation. Transform only the gradient matrix.
gradcorrect (applies only to intersystem crossing processes)	**string** $1=ngt, nac $1=tdm, kmatrix	keyword gradcorrect can also be followed by a string with additional options to control the gradient correction scheme. Include $(E_I - E_J)\langle\phi_I\vert\,\partial/\partial\mathbf{R}\,\vert\phi_J\rangle$ in gradient transformation. Include $(E_I - E_J)\langle\phi_I\vert\,\partial/\partial t\,\vert\phi_J\rangle$ in gradient transformation and perform a chain rule to approximate the diagonal gradient.
tdm_method (applies only to TDM gradient correction scheme)	**string** $1=**gradient (default for coupling ddr and coupling overlap)** $2=**energy (default for coupling ktdc)**	Method to control the computations of the time derivatives of potential energies in the diagonal basis in the TDM gradient correction scheme. Time derivatives of potential energies in diagonal basis are computed from transformation of that in MCH basis. And the time derivatives of potential energies in MCH basis are computed by a dot product between nuclear gradient and velocity vector Time derivatives of potential energies in diagonal basis are computed from finite differences. This is more accurate because in curvature-driven algorithms we approximate TDCs instead of computing TDCs accurately form electronic structure software.
convthre (applies only adaptive time step integrators)	**float** $1=**1e-04** $1=**1e-04**	Convergence threshold for successive steps Use with adaptive Velocity-Verlet integrator; the unit is eV. Use with Bulirsch-Stoer integrator.
stepsize_min (applies only to adaptive time step integrators)	**float** **stepsize/16**	Minimum time step allowed in adaptive Velocity Verlet in fs.
stepsize_max (applies only to adaptive time step integrators)	**float** **stepsize*2**	Maximum time step allowed in adaptive Velocity Verlet in fs.
nac_projection nonac_projection		Applies projected NAC (default) Applies original NAC

2.2 Detailed descriptions of the keywords

This section provides a detailed description of keywords with theoretical background.

method

The **method** keyword controls the method to be used for dynamics

- **tsh** for trajectory surface hopping
- **scp** for methods with self-consistent potentials, which are SE, CSDM, and SCDM).

To run CSDM, which is recommended, set **method** to **scp** and **decoherence_scheme** to **dom**.

coupling

The **coupling** keyword controls the coupling of electronic states in propagation of electronic coefficients

- **ddr**, **nacdr**: nonadiabatic coupling vector

$$\mathbf{d}_{IJ}^{\mathrm{MCH}}\big(\mathbf{R}(t)\big) = \left\langle \phi_I^{\mathrm{MCH}}\big(\mathbf{r};\mathbf{R}(t)\big)\middle|\frac{\partial}{\partial \mathbf{R}}\middle|\phi_J^{\mathrm{MCH}}\big(\mathbf{r};\mathbf{R}(t)\big)\right\rangle$$

  where $\phi_I^{\mathrm{MCH}}\big(\mathbf{r};\mathbf{R}(t)\big)$ is the MCH electronic wave function of state *I* at nuclear configuration $\mathbf{R}(t)$ at time *t*.

- **overlap**: overlap integrals of electronic adiabatic wave functions at successive time steps.

$$S_{IJ}^{\mathrm{MCH}}(t, t+\Delta t) = \left\langle \phi_I^{\mathrm{MCH}}(t)\middle|\phi_J^{\mathrm{MCH}}(t+\Delta t)\right\rangle$$

- **kdtc**: curvature-driven time derivative coupling κTDC,

$$T_{IJ}^{\mathrm{MCH}} = \left\langle \phi_I^{\mathrm{MCH}}\middle|\frac{\partial}{\partial t}\middle|\phi_J^{\mathrm{MCH}}\right\rangle \approx \frac{1}{2}\left[\frac{d^2\Delta V_{IJ}(t)}{dt^2}\frac{1}{\Delta V_{IJ}(t)}\right]^2 \quad \text{for } J > I$$

$$\Delta V_{IJ}(t) = H_{II}^{\mathrm{elec[MCH]}}\big(\mathbf{R}(t)\big) - H_{JJ}^{\mathrm{elec[MCH]}}\big(\mathbf{R}(t)\big)$$

  where $H_{II}^{\mathrm{elec[MCH]}}\big(\mathbf{R}(t)\big)$ is the MCH potential energy of state *I* at nuclear configuration $\mathbf{R}(t)$ at time *t*.

This keyword therefore selects between three types of electronic state couplings that can be evaluated from *ab initio* electronic structure calculations:

- NAC (**coupling ddr**, **nacdr**) – requires analytical implementation of NAC for the electronic structure method. This is often only available for state-averaged complete active space self-consistent potential (SA-CASSCF) method.
  We note some of the software for which NAC is available for other methods: COULUMBUS has analytical implementation of NAC for multi-reference configuration interaction (MRCI) method; BAGEL has analytical implementation of NAC for extended multi-state complete active space second order perturbation theory (XMS-CASPT2).
- Overlap (**coupling overlap**) – requires computation of electronic wave functions for the electronic structure method.
- Curvature approximated TDC (**coupling ktdc**) – requires computation of energies, or computation of energies and analytical implementation of gradients for the electronic structure method.

ktdc_method

The **ktdc_method** keyword controls how the κTDC is computed. And therefore it is only applied to curvature-driven methods.

- **energy** for second-order difference of energy. For the third step, it is computed as,

$$\frac{d^2\Delta V_{IJ}(t)}{dt^2} \approx \frac{1}{\Delta t^2}\left[\Delta V_{IJ}(t) - 2\Delta V_{IJ}(t-\Delta t) + \Delta V_{IJ}(t-2\Delta t)\right]$$

  Starting from fourth step, it is computed as,

$$\frac{d^2\Delta V_{IJ}(t)}{dt^2} \approx \frac{1}{\Delta t^2}\left[2\Delta V_{IJ}(t) - 5\Delta V_{IJ}(t-\Delta t) + 4\Delta V_{IJ}(t-2\Delta t) - \Delta V_{IJ}(t-3\Delta t)\right]$$

$$\Delta V_{IJ}(t) = H_{II}^{\text{elec[MCH]}}\left(\mathbf{R}(t)\right) - H_{JJ}^{\text{elec[MCH]}}\left(\mathbf{R}(t)\right)$$

- **gradient** for first-order difference of dot product of gradients and velocity vector

$$\frac{d^2\Delta V_{IJ}(t)}{dt^2} \approx \frac{\Delta\dot{V}_{IJ}(t-\Delta t) - \Delta\dot{V}_{IJ}(t)}{\Delta t}$$

$$\Delta\dot{V}_{IJ}(t) = \frac{\partial H_{II}^{\text{elec[MCH]}}(t)}{\partial \mathbf{R}} \cdot \dot{\mathbf{R}} - \frac{\partial H_{JJ}^{\text{elec[MCH]}}(t)}{\partial \mathbf{R}} \cdot \dot{\mathbf{R}}$$

decoherence_scheme

When performing methods based on self-consistent-potentials, the decoherence is introduced with the decay-of-mixing algorithm. Use of the decay-of-mixing decoherence scheme is turned on by setting **decoherence_scheme dom**. The other two options, i.e., **deocherence_sceheme edc** and **decoherence_scheme afssh**, are used for calculations with the trajectory surface hopping method.

switching_procedure

There are two options for the computation of the probabilities of switching pointer states in the decay-of-mixing algorithms, and they can be set with the **switching_procedure** keyword.

- **switching_procedure csdm** turns on coherent switching with decay of mixing,[4] which is the default and is recommended.
- **switching_procedure scdm** turns on the older self-consistent decay of mixing.[3]

neom

The **neom** keyword only applies to SCP methods. It is used to control the form of the nuclear equations of motion in coherent nuclear EOM for SCP methods; note that this is a separate choice from the form of the coupling used in the electronic equations of motion. One can directly use the nonadiabatic coupling vector by using either **neom nacdr** or **neom ddr**; this will require using an electronic structure theory code to compute the nonadiabatic coupling vector. Therefore, the coherent nuclear equation of motion has the original form for SE and CSDM, i.e.,

$$\left[\dot{\mathbf{P}}\right]_{\text{SE}} = -\sum_{I} \rho_{II} \frac{\partial H_{II}^{\text{elec[diag]}}}{\partial \boldsymbol{R}} + \sum_{IJ}^{N_{\text{states}}} \text{Re}\left(\rho_{JI}^{\text{diag}}\left(H_{II}^{\text{elec[diag]}} - H_{JJ}^{\text{elec[diag]}}\right)\mathbf{d}_{IJ}^{\text{diag}}\right)$$

Alternatively, one can use an effective nonadiabatic coupling vector by setting **neom gdiff**. The effective nonadiabatic coupling vector is a vector defined as a combination of the difference gradient vector and the velocity vector. The dot product between the effective nonadiabatic

coupling vector and velocity vector equals TDC. See reference [10] for detailed description of effective nonadiabatic coupling vector.

Using the effective nonadiabatic coupling vector is the default option if one uses overlap-based methods, which are chosen by using **coupling overlap**, or if one uses curvature-driven methods, which are chosen by using **coupling ktdc**. When using an effective NAC, the nuclear equation of motion is

$$\left[\dot{\mathbf{P}}\right]_{\mathrm{SE}} = -\sum_{I} \rho_{II} \frac{\partial H_{II}^{\mathrm{elec[diag]}}}{\partial \boldsymbol{R}} + \sum_{IJ}^{N_{\mathrm{states}}} \mathrm{Re}\left(\frac{\rho_{JI}^{\mathrm{diag}} \left(H_{II}^{\mathrm{elec[diag]}} - H_{JJ}^{\mathrm{elec[diag]}} \right) T_{IJ}^{\mathrm{diag}}}{\mathbf{G}_{IJ}^{\mathrm{diag}} \cdot \dot{\mathbf{R}}} \mathbf{G}_{IJ}^{\mathrm{diag}} \right)$$

where $\mathbf{G}_{IJ}^{\mathrm{diag}}$ is effective NAC in the diagonal basis, and T_{IJ}^{diag} is time derivative coupling between states *I* and *J*. This generalization of coherent equation of motion for SE and CSDM is called generalized SE.

One does not need to set the **neom** keyword. If it is not set, the code will employ the default choice of nuclear EOM based on the **coupling** keyword. The following are default maps between **coupling** and **neom**:

coupling ddr, **nacdr**	→	**neom ddr**, **nacdr**
coupling overlap	→	**neom gdiff**
coupling ktdc	→	**neom gdiff**

eeom

The **eeom** keyword controls the coherent propagation of the electronic coefficients. We suggest that only advanced users change this option (for example, for benchmark purposes). In general, we recommend that one accepts the defaults for this option.

There are four algorithms to propagate electronic coefficients, namely, constant interpolation (CI), linear interpolation (LI), local diabatization (LD), and norm-preserving interpolation (NPI). In *SHARC* and *SHARC-MN*, the diagonal electronic coefficients are propagated according to,

$$\mathbf{c}^{\mathrm{diag}}(t+\Delta t) = \mathbf{U}^{\dagger}(t+\Delta t)\boldsymbol{\mathcal{P}}^{\mathrm{MCH}}(t+\Delta t, t)\mathbf{U}(t)\mathbf{c}^{\mathrm{diag}}(t)$$

where $\boldsymbol{\mathcal{P}}^{\mathrm{MCH}}(t+\Delta t, t)$ is the MCH propagator, and $\mathbf{U}(t)$ is the MCH-to-diagonal representation transformation matrix defined as,

$$\mathbf{H}^{\mathrm{elec[diag]}}(t) = \mathbf{U}^{\dagger}(t)\mathbf{H}^{\mathrm{elec[MCH]}}(t)\mathbf{U}(t)$$

where $H_{II}^{\mathrm{elec[GB]}}\left(\mathbf{R}(t)\right)$ is the potential energy of state *I* at time *t* with GB representation. There are several ways to define the propagator $\boldsymbol{\mathcal{P}}^{\mathrm{MCH}}(t+\Delta t, t)$, and the definition of $\boldsymbol{\mathcal{P}}^{\mathrm{MCH}}(t+\Delta t, t)$ distinguishes different algorithms. Notice that here we only consider coherent propagation, i.e., $\boldsymbol{\mathcal{P}}^{\mathrm{MCH}} = \boldsymbol{\mathcal{P}}_{\mathrm{C}}^{\mathrm{MCH}}$.

For CI, LI, and NPI algorithms, propagation of coherent coefficients is divided into substeps,

$$\mathbf{c}^{\mathrm{diag}}(t+\Delta t) = \mathbf{U}^{\dagger}(t+\Delta t)\left(\prod_{l=1}^{n} \boldsymbol{\mathcal{P}}_{\mathrm{C},l}^{\mathrm{MCH}}\right)\mathbf{U}(t)\mathbf{c}^{\mathrm{diag}}(t)$$

where *n*, which is set by keyword **nsubsteps**, is the number of substeps employed, and *l* is the index of a substep. The coherent propagator $\boldsymbol{\mathcal{P}}_{\mathrm{C},l}$ **for substeps** is defined as

$$\boldsymbol{\mathcal{P}}_{\mathrm{C},l}^{\mathrm{MCH}} = \exp\left(-\left(\frac{i}{\hbar}\mathbf{H}_l + \mathbf{T}_l\right)\frac{\Delta t}{n}\right)$$

where,

$$\mathbf{H}_l = \mathbf{H}^{\mathrm{elec[MCH]}}(t) + \frac{l}{n}\left(\mathbf{H}^{\mathrm{elec[MCH]}}(t+\Delta t) - \mathbf{H}^{\mathrm{elec[MCH]}}(t)\right)$$

- In the constant-interpolation (**eeom ci**) algorithm,
$$\mathbf{T}_l = \mathbf{T}^{\mathrm{MCH}}(t)$$
where $\mathbf{T}^{\mathrm{MCH}}(t)$ is the time derivative coupling at time *t*.
- In the linear-interpolation (**eeom li**) algorithm,
$$\mathbf{T}_l = \mathbf{T}^{\mathrm{MCH}}(t) + \frac{l}{n}\left(\mathbf{T}^{\mathrm{MCH}}(t+\Delta t) - \mathbf{T}^{\mathrm{MCH}}(t)\right)$$
- In the norm-preserving interpolation (**eeom npi**) algorithm,
$$\mathbf{T}_l = \mathbf{T}^{\mathrm{MCH}}\left(t + \frac{l\Delta t}{n}\right)$$
and $\mathbf{T}^{\mathrm{MCH}}\left(t + \frac{l\Delta t}{n}\right)$ is evaluated according to the norm-preserving interpolation approximation of evaluation TDC from overlap integrals.[10,18] Between time step t and $t+\Delta t$, the TDC at time $\mathcal{T}$ is
$$\mathbf{T}_{IJ}^{\mathrm{MCH}}(\mathcal{T}) = \left\langle \phi_I^{\mathrm{MCH}}\left(\mathbf{r};\mathbf{R}(t)\right)\middle|\boldsymbol{\Theta}^{\dagger}(\mathcal{T})\frac{\partial}{\partial\mathcal{T}}\boldsymbol{\Theta}(\mathcal{T})\middle|\phi_J^{\mathrm{MCH}}\left(\mathbf{r};\mathbf{R}(t)\right)\right\rangle$$
where $\boldsymbol{\Theta}(\mathcal{T})$ is a time-dependent transformation matrix that interpolates the MCH (adiabatic) electronic wave functions between time t and $t+\Delta t$,
$$\phi_J^{\mathrm{MCH}}\left(\mathbf{r};\mathbf{R}(\mathcal{T})\right) = \boldsymbol{\Theta}(\mathcal{T})\phi_J^{\mathrm{MCH}}\left(\mathbf{r};\mathbf{R}(t)\right)$$
$$\boldsymbol{\Theta}_{II}(\mathcal{T}) = \cos\left(\cos^{-1}\left(S_{IJ}^{\mathrm{MCH}}(t,t+\Delta t)\right)\frac{\mathcal{T}-t}{\Delta t}\right)$$
$$\boldsymbol{\Theta}_{IJ}(\mathcal{T}) = \sin\left(\sin^{-1}\left(S_{IJ}^{\mathrm{MCH}}(t,t+\Delta t)\right)\frac{\mathcal{T}-t}{\Delta t}\right)$$
The NPI algorithm is probably the most accurate way to evaluate the TDC for intermediate steps. The LI algorithm can be interpreted as the first-order Taylor's expansion of the NPI algorithm.
- For the LD algorithm, propagation of coherent coefficients is divided into substeps. The coefficients are first propagate in a locally diabatic basis and then transformed back to he MCH basis,
$$\mathbf{c}^{\mathrm{diag}}(t+\Delta t) = \mathbf{U}^{\dagger}(t+\Delta t)\left(\mathbf{S}^{\mathrm{MCH}}(t,t+\Delta t)^{\dagger}\prod_{l=1}^{n}\boldsymbol{\mathcal{P}}_{\mathrm{C},l}^{\mathrm{MCH}}\right)\mathbf{U}(t)\mathbf{c}^{\mathrm{diag}}(t)$$
The substep coherent propagator $\boldsymbol{\mathcal{P}}_{\mathrm{C},l}$ is defined as,
$$\boldsymbol{\mathcal{P}}_{\mathrm{C},l}^{\mathrm{MCH}} = \exp\left(-\frac{i}{\hbar}\mathbf{H}_l^{\mathrm{LD}}\frac{\Delta t}{n}\right)$$
$$\mathbf{H}_l^{\mathrm{LD}} = \mathbf{H}^{\mathrm{MCH}}(t) + \frac{l}{n}\left(\mathbf{H}_{\mathrm{trans}}^{\mathrm{elec[MCH]}}(t+\Delta t) - \mathbf{H}^{\mathrm{MCH}}(t)\right)$$
$$\mathbf{H}_{\mathrm{trans}}^{\mathrm{elec[MCH]}}(t+\Delta t) = \mathbf{S}^{\mathrm{MCH}}(t,t+\Delta t)\mathbf{H}^{\mathrm{elec[MCH]}}(t+\Delta t)\mathbf{S}^{\mathrm{MCH}}(t,t+\Delta t)^{\dagger}$$

Therefore, *in principle*, both CI and LI algorithms can be employed for any type of electronic state couplings: **coupling ddr**, **coupling nacdr**, **coupling overlap**, and **coupling ktdc**; while NPI is only available for **coupling overlap**.

However, we have implemented methods by which evaluation of one type of coupling allows us to approximate the other couplings. Specifically,

- When using **coupling ddr**, or **coupling nacdr** with other couplings not computed:
  The TDC is computed as:
  $$\mathrm{T}_{\mathrm{IJ}}^{\mathrm{MCH}}(\mathrm{t}) = \mathbf{d}_{\mathrm{IJ}}^{\mathrm{MCH}}\big(\mathbf{R}(\mathrm{t})\big) \cdot \dot{\mathbf{R}}$$
  The overlap integral is approximated from $T_{IJ}^{\mathrm{MCH}}(t)$:
  $$S_{IJ}^{\mathrm{MCH}}(t, t+\Delta t) \approx \begin{cases} \mathbf{T}_{IJ}^{\mathrm{MCH}}(t)\Delta t & \text{for } J \neq I \text{ and } \sum_{J\neq I}^{N_{\mathrm{state}}} \big(\mathbf{T}_{IJ}^{\mathrm{MCH}}(t)\Delta t\big)^2 \leq 1.0 \\ \dfrac{\mathbf{T}_{IJ}^{\mathrm{MCH}}(t)\Delta t}{\sum_{J\neq I}^{N_{\mathrm{state}}}\big(\mathbf{T}_{IJ}^{\mathrm{MCH}}(t)\Delta t\big)^2} & \text{for } J \neq I \text{ and } \sum_{J\neq I}^{N_{\mathrm{state}}} \big(\mathbf{T}_{IJ}^{\mathrm{MCH}}(t)\Delta t\big)^2 > 1.0 \end{cases}$$
  $$S_{II}^{\mathrm{MCH}}(t, t+\Delta t) \approx 1 - \sum_{J\neq I}^{N_{\mathrm{state}}} \big(\mathbf{T}_{IJ}^{\mathrm{MCH}}(t)\Delta t\big)^2$$
- When using **coupling overlap**, with other couplings not computed:
  The TDC is computed from norm-preserving interpolation method (notice the norm-preserving interpolation is a method to compute TDC from overlap integrals, this is a different definition when we say NPI algorithm which is discussed previously. But these two are closely related) because
  $$T_{IJ}^{\mathrm{MCH}}(t + \Delta t/2) \approx \frac{\int_t^{t+\Delta t} T_{IJ}^{\mathrm{MCH}}(\mathcal{T})d\mathcal{T}}{\Delta t}$$
  where $T_{IJ}^{\mathrm{MCH}}(\mathcal{T})$ is defined above where norm preserving interpolation algorithm is introduced.
  The NAC is approximated with effective NAC when needed.
- When using **coupling ktdc**, with other couplings not computed:
  The overlap integral is approximated from curvature-driven $T_{IJ}^{\mathrm{MCH}}(t)$ similar to the way described when using **coupling ddr** or **coupling nacdr**.
  The NAC is approximated by the effective NAC when needed.

These electronic state coupling approximation schemes allow one to use any of the electronic coefficients propagation algorithms for any of the *ab initio* electronic state couplings.

One does not need to set the **eeom** keyword. In it is not set, the code will employ the default choice of nuclear EOM based on **coupling** and **method** keywords. The following are default maps between **coupling** and **neom**,

method tsh	and	**coupling ddr**, **nacdr**	→	**eeom li**
method tsh	and	**coupling overlap**	→	**eeom ld**
method tsh	and	**coupling ktdc**	→	**eeom li**
method scp	and	**coupling ddr**, **nacdr**	→	**eeom li**
method scp	and	**coupling overlap**	→	**eeom npi**
method scp	and	**coupling ktdc**	→	**eeom li**

time_uncertainty

The **time_uncertainty** keyword controls whether the fewest switches with time uncertainty algorithm is used or the original Tully's fewest switches algorithm is used.

The time uncertainty algorithm is the same as the TSH-FS algorithm except when a frustrated hop is encountered. When a frustrated hop is encountered, the TSH-FSTU method

looks for nearby regions where a hop can be successful. Then, if a such a region is found, the TSH-FSTU method allows a nonlocal hop. One can interpret this as the FSTU algorithm borrowing some energy along the timeline of the trajectory according to time-energy uncertainty principle. The FSTU algorithm can greatly reduce the number of frustrated hops.[13,14]

pointer_basis

The **pointer_basis** keyword controls how decoherence is introduced in the electronic EOM. This keyword only applies to SCP methods. The diagonal-representation electronic coefficients are propagated according to

$$c^{\mathrm{diag}}(t+\Delta t) = \boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t)c^{\mathrm{diag}}(t)$$

When the propagation is fully coherent, we have

$$\boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t) = \mathbf{U}^{\dagger}(t+\Delta t)\boldsymbol{\mathcal{P}}_{\mathrm{C}}^{\mathrm{MCH}}(t+\Delta t, t)\mathbf{U}(t)$$

When decoherence is introduced, there exist two ways to introduce a decoherent propagator,

$$\boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t)\big|_{\mathrm{MCH}} = \mathbf{U}^{\dagger}(t+\Delta t)\boldsymbol{\mathcal{P}}_{\mathrm{D}}^{\mathrm{MCH}}(t+\Delta t)\boldsymbol{\mathcal{P}}_{\mathrm{C}}^{\mathrm{MCH}}(t+\Delta t, t)\mathbf{U}(t)$$

and

$$\boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t)\big|_{\mathrm{diag}} = \boldsymbol{\mathcal{P}}_{\mathrm{D}}^{\mathrm{diag}}(t+\Delta t)\mathbf{U}^{\dagger}(t+\Delta t)\boldsymbol{\mathcal{P}}_{\mathrm{C}}^{\mathrm{MCH}}(t+\Delta t, t)\mathbf{U}(t)$$

where $\boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t)\big|_{\mathrm{GB}}$ denotes the diagonal propagator in the GB pointer basis, where GB is either diag or MCH. Phenomenologically, $\boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t)\big|_{\mathrm{MCH}}$ will increase the population of the pointer state in the MCH basis, and decay the population of the rest of the states; and $\boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t)\big|_{\mathrm{diag}}$ will increase the population of the pointer state in the diagonal basis, and decay the population of the rest of the states. $\boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t)\big|_{\mathrm{MCH}}$ was implemented in *SHARC-MN* as described in ref. [9]. And $\boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t)\big|_{\mathrm{diag}}$ is also implemented in *SHARC-MN*. *SHARC-MN* allows both ways to introduce the decay of mixing propagator: either in the MCH basis as $\boldsymbol{\mathcal{P}}_{\mathrm{D}}^{\mathrm{MCH}}$ or in the diagonal basis as $\boldsymbol{\mathcal{P}}_{\mathrm{D}}^{\mathrm{diag}}$.

We recommend the use of $\boldsymbol{\mathcal{P}}^{\mathrm{diag}}(t+\Delta t, t)\big|_{\mathrm{diag}}$, and this is set as the default.

ekincorrect

The **ekincorrect** keyword controls how the momentum is adjusted after a surface hop; for all the options except the first, this conserves the total energy. The recommended methods are **ekincorrect parallel_pnac** for TSH with **coupling nacdr**, and **ekincorrect parallel_penac** for TSH with **coupling overlap** and **coupling ktdc**. Note that these are not the default (because we try to keep the defaults consistent with *SHARC*). Note that the choice for this keyword not only controls how the momentum is adjusted after a hop, but also it changes which hops are allowed.

In TSH, the trajectories propagate on the active potential energy surface, which corresponds to active state *K*. During the propagation, this active state can stochastically switch to a different state. This switching is achieved according to a probability called the hopping probability, which is given by

$$P_{K\to I}^{\mathrm{GB}}(t, t+\Delta t) = \max\left(\frac{b_{KI}^{\mathrm{GB}}(t)\Delta t}{\rho_{KK}^{\mathrm{GB}}(t)}, 0\right)$$

where b_{KI}^{GB} is the rate of population change from state *K* to state *I* during time t to $t+\Delta t$ according to the electronic EOM (the equation of motion provided in the TSH method by the time-dependent Schrödinger equation) in the GB representation. This is called the fewest-

switches (FS) criterion. Because it minimizes the number of state switches and – in the limit of degenerate surfaces – it maintains the correct statistical distribution of electronic state populations. The fewest switches criterion is the most widely used criterion for hopping in TSH calculations in the literature. Note the "hopping probability" is the probability that a hop is attempted; as discussed elsewhere, some attempted hops are frustrated such that no hop occurs.

In the MCH representation, the TSH-FS hopping probability is

$$P_{K\to I}^{\mathrm{MCH}}(t,t+\Delta t)=\max\left(\frac{b_{KI}^{\mathrm{MCH}}(t)\Delta t}{\rho_{KK}^{\mathrm{MCH}}(t)},0\right)$$

$$b_{KI}^{\mathrm{MCH}}(t)=2\left(\frac{1}{\hbar}\mathrm{Im}\left(H_{IL}^{\mathrm{elec[MCH]}}\rho_{LI}^{\mathrm{MCH}}\right)-\mathrm{Re}\left(T_{IL}^{\mathrm{MCH}}\rho_{LI}^{\mathrm{MCH}}\right)\right)$$

In the diagonal representation, the TSH-FS hopping probability is

$$P_{K\to I}^{\mathrm{diag}}(t,t+\Delta t)=\max\left(\frac{b_{KI}^{\mathrm{diag}}(t)\Delta t}{\rho_{KK}^{\mathrm{diag}}(t)},0\right)$$

$$b_{KI}^{\mathrm{diag}}(t)=-2\mathrm{Re}\left(T_{IL}^{\mathrm{diag}}\rho_{LI}^{\mathrm{diag}}\right)$$

As a result of these equations, a hop is most likely to be attempted in the vicinity of an intersection between state *K* and state *I*, i.e., at a locally avoided crossing. A hop from one state to another (i.e., from one surface to another) causes the governing potential energy surface to be discontinuous. To compensate this discontinuous change in potential energy, one has to adjust the momentum to conserve the total energy (make it be the same before and after the hop). To enforce the conservation of the trajectory's energy, the momentum is adjusted along a prescribed direction called the momentum adjustment vector or the hopping direction. The velocity $\dot{\mathbf{R}}$ is changed after a $K\to I$ hop according to

$$\dot{\mathbf{R}}_{\mathrm{A}}(t)\big|_{\mathrm{post}K\to I}=\dot{\mathbf{R}}_{\mathrm{A}}(t)\big|_{\mathrm{pre}K\to I}-f\frac{\mathbf{h}_{KI,\mathrm{A}}(t)}{M_{\mathrm{A}}}$$

where A is an atom index, $\mathbf{h}_{KI,\mathrm{A}}(t)$ and M_{A} are the momentum adjustment vector and the mass of atom A, and f is a factor to be determined. The choice of $\mathbf{h}_{KI}$ in *SHARC* and *SHARC-MN* is controlled by the **ekincorrect parallel_*XX*** keyword, where ***XX*** denotes one of the available choices of vectors, for example, **ekincorrect parallel_nac**. The factor *f* is computed by requiring energy conservation after a hop:

$$\sum_{\mathrm{A}}\frac{1}{2}M_{\mathrm{A}}\left(\dot{\mathbf{R}}_{\mathrm{A}}(t)\big|_{\mathrm{post}K\to I}\right)^2+H_{II}^{\mathrm{elec[GB]}}=\sum_{\mathrm{A}}\frac{1}{2}M_{\mathrm{A}}\left(\dot{\mathbf{R}}_{\mathrm{A}}(t)\big|_{\mathrm{pre}K\to I}\right)^2+H_{KK}^{\mathrm{elec[GB]}}$$

Therefore,

$$f=\frac{E_{\mathrm{h}}^{KI}(t)\pm\sqrt{\left(E_{\mathrm{h}}^{KI}(t)\right)^2+4E_{\mathrm{kin,h}}^{KI}(t)\Delta H_{KI}^{\mathrm{GB}}(t)}}{2E_{\mathrm{kin,h}}^{KI}(t)}$$

where

$$\Delta H_{KI}^{\mathrm{GB}}(t)=H_{KK}^{\mathrm{elec[GB]}}-H_{II}^{\mathrm{elec[GB]}}$$

$$E_{\mathrm{h}}^{KI}(t)=\sum_{\mathrm{A}}\left(\dot{\mathbf{R}}_{\mathrm{A}}(t)\big|_{\mathrm{pre}K\to I}\cdot\mathbf{h}_{KI,\mathrm{A}}(t)\right)$$

$$E_{\mathrm{kin,h}}^{KI}=\sum_{\mathrm{A}}\frac{1}{2}\frac{\left|\mathbf{h}_{KI,\mathrm{A}}(t)\right|^2}{M_{\mathrm{A}}}$$

To have a real solution of f, following condition must be fulfilled,

$$\Delta E^{KI}(t) \equiv \frac{E_{\mathrm{h}}^{KI}(t)^2}{4E_{\mathrm{kin,h}}^{KI}(t)} + \Delta H_{KI}^{\mathrm{GB}}(t) \geq 0$$

When the above condition is not fulfilled, the hop is frustrated, which is a notorious drawback of all surface hopping methods.

In *SHARC-MN*, seven choices are available for the momentum adjustment vector, in addition to the option of no adjustment after a hop, which is not recommended:

- **none** deactivates the adjustment, so that the total energy is *not* preserved after a hop, although nuclear angular momentum is conserved. Using this option, jumps can never be frustrated and are always performed according to the hopping probabilities.
- **parallel_vel** (the default): The momentum is adjusted by simply scaling all components of the nuclear velocities by the same factor so that the new kinetic energy is $E_{\mathrm{tot}} - E_{\mathrm{pot}}$. Note that this is the same as scaling all components of the nuclear momenta by the same factor so that the new kinetic energy is $E_{\mathrm{tot}} - E_{\mathrm{pot}}$. Jumps are frustrated if the new potential energy would exceed the total energy. When using this option, energy is conserved, but angular momentum is not conserved.
- **parallel_pvel**: The momentum is adjusted by scaling the component of the vibrational velocity. The hop is frustrated if there is not enough momentum in this direction to conserve total energy. The vibrational velocity is computed by projecting out the translational and rotational components out of the velocity vector.
- **parallel_nac**: The momentum is adjusted by scaling the component of the nuclear velocities parallel to the nonadiabatic coupling vector between the old and new state. The hop is frustrated if there is not enough momentum in this direction to conserve total energy. Note that **ekincorrect parallel_nac** requires the calculation of the nonadiabatic coupling vectors, even if they are not used for the wave function propagation. When using this option, angular momentum will *not* be conserved.
- **parallel_diff**: The momentum is adjusted by scaling the component of the nuclear velocities parallel to the difference gradient vector between the old and new state. The hop is frustrated if there is not enough kinetic energy in this direction to conserve total energy. When using this option, angular momentum is conserved.
- **parallel_pnac**: The momentum is adjusted by scaling the component of the nuclear velocities parallel to the projected nonadiabatic coupling vector between the old and new state. The hop is frustrated if there is not enough momentum in this direction to conserve total energy. Note that **ekincorrect parallel_pnac** requires the calculation of the nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum is conserved.
- **parallel_enac**: The momentum is adjusted by scaling the component of the nuclear velocities parallel to the effective nonadiabatic coupling vector between the old and new state. The hop is frustrated if there is not enough momentum in this direction to conserve total energy. Note that **ekincorrect parallel_enac** requires the calculation of the effective nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum will NOT be conserved.
- **parallel_penac**: The momentum is adjusted by rescaling the component of the nuclear velocities parallel to the effective nonadiabatic coupling vector between the

old and new state. The hop is frustrated if there is not enough momentum in this direction to conserve total energy. Note that **ekincorrect parallel_penac** implies the calculation of the projected effective nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum is conserved.

reflect_frustrated

The **reflect_frustrated** keyword controls whether the velocity is reversed after a frustrated hop. The recommended methods are **reflect_frustrated parallel_pnac** for TSH with **coupling nacdr**, and **reflect_frustrated parallel_penac** for TSH with **coupling overlap** and **coupling ktdc**. Note that these are not the default.

- **none** (the default) after a frustrated hop, the velocity vector is not modified. This option is sometimes described as ignoring frustrated hops.
- **parallel_vel** the full velocity vector is inverted when a frustrated hop is encountered.
- **parallel_pvel** only the velocity component parallel to the vibrational velocity vector is inverted. The vibrational velocity is computed by projecting the translational and rotational components out of the velocity vector.
- **parallel_nac** only the velocity component parallel to the nonadiabatic coupling vector between the active and frustrated states is inverted. This requires the calculation of the nonadiabatic coupling vectors, even if they are not used for the wave function propagation. When using this option, angular momentum will *not* be conserved.
- **parallel_diff** only the velocity component parallel to the difference gradient vector between the active and frustrated states is inverted. When using this option, angular momentum is conserved.
- **parallel_pnac** only the velocity component parallel to the projected nonadiabatic coupling vector between the active and frustrated states is inverted. This implies the calculation of the nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum is conserved.
- **parallel_enac** only the velocity component parallel to the effective nonadiabatic coupling vector between the active and frustrated states is inverted. This implies the calculation of the effective nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum will *not* be conserved.
- **parallel_penac** only the velocity component parallel to the projected effective nonadiabatic coupling vector between the active and frustrated states is inverted. This implies the calculation of the projected effective nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum is conserved.
- **delV_vel** the full velocity vector is inverted when a frustrated hop is encountered according to ∇V criteria.[19]
- **delV_pvel** only the velocity component parallel to the vibrational velocity vector is inverted according to ∇V criteria. The vibrational velocity is computed by projecting out the translational and rotational components out of the velocity vector.

- **delV_nac** only the velocity component parallel to the nonadiabatic coupling vector between the active and frustrated states is inverted according to ∇V criteria. This implies the calculation of the nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum will NOT be conserved.
- **delV_diff** only the velocity component parallel to the difference gradient vector between the active and frustrated states is inverted according to ∇V criteria. When using this option, angular momentum is conserved.
- **delV_pnac** only the velocity component parallel to the projected nonadiabatic coupling vector between the active and frustrated states is inverted according to ∇V criteria. This implies the calculation of the nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum is conserved.
- **delV_enac** only the velocity component parallel to the effective nonadiabatic coupling vector between the active and frustrated states is inverted according to ∇V criteria. This implies the calculation of the effective nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum will NOT be conserved.
- **delV_penac** only the velocity component parallel to the projected effective nonadiabatic coupling vector between the active and frustrated states is inverted according to ∇V criteria. This implies the calculation of the projected effective nonadiabatic coupling vector, even if they are not used for the wave function propagation. When using this option, angular momentum is conserved.

integrator

The **integrator** keyword controls the integrator to be used in dynamics

- **fvv** for Velocity Verlet with a fixed time step
- **avv** for Velocity Verlet with an adaptive time step
- **bsh** for Bulirsch-Stoer integrator

The **fvv** integrator is the default integrator in the *SHARC* and *SHARC-MN* programs. We recommend using **fvv** or **avv**. Notice that the adaptive timestep integrators **integrator avv** and **integrator bsh** are not compatible with **coupling overlap** in the current versions of the *SHARC* and *SHARC-MN* programs.

soc_scaling

The **soc_scaling** keyword allows one to scale the spin-orbit coupling in the Hamiltonian involve electronic states with multiple spins. The default is 1.0 (no scaling).

gradcorrect and **nogradcorrect**

These two keywords control the use of the gradient correction scheme in he diagonal basis when spin-orbit coupling is included in the calculation. Using the keyword **gradcorrect** (without arguments) will turn on the NGT gradient correction scheme, and using the keyword **nogradcorrect** (without arguments) will turn off gradient correction scheme.

Alternatively, one can specify an additional keyword after **gradcorrect**, and this specifies which gradient correction scheme is used: **gradcorrect ngt** or **gradcorrect nac** turns on the NGT scheme; **gradcorrect tdm** or **gradcorrect kmatrix** turns on the TDM scheme.

The default is **nogradcorrect** to be consistent with *SHARC*. However, we strongly recommend that users turn on **gradcorrect**. Our recommendations are:

- NAC-based algorithm : **gradcorrect** or **gradcorrect ngt** or **gradcorrect nac**
- Overlap-based algorithm : **gradcorrect tdm** or **gradcorrect kmatrix**
- Curvature-drive algorithm : **gradcorrect tdm** or **gradcorrect kmatrix**

The gradient correction scheme is very useful to obtain accurate gradients in the diagonal basis. When spin-orbit coupling is present, the electronic Hamiltonian can be divided into two parts,

$$H^{\mathrm{elec}} = H^{\mathrm{SF}} + H^{\mathrm{SOC}}$$

where H^{SF} is the spin-free Hamiltonian (electronic kinetic energy plus all Coulomb interactions, including nuclear repulsion), and H^{SOC} is the spin-orbit coupling. Currently available electronic structure software only provides potential energies and gradients in the MCH basis, rather than the diagonal basis; and the gradient of the spin-orbit coupling is not available. For that reason, the gradient of the potential energy in the diagonal basis $\left(\frac{\partial H_{II}^{\mathrm{elec[diag]}}}{\partial \mathbf{R}}\right)$ is computed by making a correction to the gradient in MCH basis $\left(\frac{\partial H_{II}^{\mathrm{elec[diag]}}}{\partial \mathbf{R}}\right)$.

- The NGT scheme achieves this goal by correcting $\frac{\partial H_{II}^{\mathrm{elec[MCH]}}}{\partial \mathbf{R}}$ with $\mathbf{d}^{\mathrm{MCH}}$.
- The TDM scheme achieves this goal by correcting $\frac{dH_{II}^{\mathrm{elec[MCH]}}}{dt}$ with $\mathbf{T}^{\mathrm{MCH}}$, and $\frac{\partial H_{II}^{\mathrm{elec[diag]}}}{\partial \mathbf{R}}$ is approximated based on

$$\frac{\partial H_{II}^{\mathrm{elec[diag]}}}{\partial \mathbf{R}} = \frac{dH_{II}^{\mathrm{elec[diag]}}}{dt} \cdot \dot{\mathbf{R}}$$

tdm_method

The keyword **tdm_method** controls how $\frac{d\mathbf{H}^{\mathrm{elec[diag]}}}{dt}$ is computed. Because both $\mathbf{H}^{\mathrm{elec[diag]}}$ and $\mathbf{H}^{\mathrm{elec[MCH]}}$ are available, one can compute $\frac{d\mathbf{H}^{\mathrm{elec[diag]}}}{dt}$ either by finite differences or by a correction from $\frac{d\mathbf{H}^{\mathrm{elec[MCH]}}}{dt}$

- **Energy** selects the backward finite-difference method. For the second time step, it is computed as this first-order backward difference:

$$\frac{dH_{II}^{\mathrm{elec[diag]}}(t+\Delta t)}{dt} \approx \frac{1}{\Delta t}\left[H_{II}^{\mathrm{elec[diag]}}(t+\Delta t) - H_{II}^{\mathrm{elec[diag]}}(t)\right]$$

Starting from the third time step, it is computed as the second-order backward difference:

$$\frac{dH_{II}^{\mathrm{elec[diag]}}(t+\Delta t)}{dt} \approx \frac{1}{2\Delta t}\left[3H_{II}^{\mathrm{elec[diag]}}(t+\Delta t) - 4H_{II}^{\mathrm{elec[diag]}}(t) + H_{II}^{\mathrm{elec[diag]}}(t-\Delta t)\right]$$

- **gradient** selects the method of computing $\frac{d\mathbf{H}^{\mathrm{elec[diag]}}}{dt}$ from the transformation of the time-derivative Hamiltonian matrix. For a general basis (GB), the matrix elements of the TDM matrix are defined as,

$$\mathcal{K}_{IJ}^{\mathrm{GB}} \equiv \left\langle \phi_I^{\mathrm{GB}} \middle| \frac{d}{dt} H^{\mathrm{elec}} \middle| \phi_J^{\mathrm{GB}} \right\rangle = k_{IJ}^{\mathrm{GB}} + \left\langle \phi_I^{\mathrm{GB}} \middle| \frac{dH^{\mathrm{SOC}}}{dt} \middle| \phi_J^{\mathrm{GB}} \right\rangle$$

$$k_{IJ}^{\mathrm{GB}} \equiv \left\langle \phi_I^{\mathrm{GB}} \middle| \frac{dH^{\mathrm{SF}}}{dt} \middle| \phi_J^{\mathrm{GB}} \right\rangle$$

  From electronic structure information, one can compute k_{IJ}^{MCH}. And when one sets the keyword **tdm_method gradient**, the code computes k_{IJ}^{MCH} by,

$$k_{IJ}^{\mathrm{MCH}} = \frac{dH_{II}^{\mathrm{SF[MCH]}}}{d\mathbf{R}} \cdot \dot{\mathbf{R}}\delta_{IJ} - \left(H_{II}^{\mathrm{SF[MCH]}} - H_{JJ}^{\mathrm{SF[MCH]}}\right) T_{IJ}^{\mathrm{MCH}}$$

  With k_{IJ}^{MCH}, we approximate $\boldsymbol{\mathcal{K}}^{\mathrm{diag}}$ by,

$$\boldsymbol{\mathcal{K}}^{\mathrm{diag}} \approx \mathbf{U}^{\dagger}\mathbf{k}^{\mathrm{MCH}}\mathbf{U}$$

  And this gives

$$\frac{dH_{II}^{\mathrm{elec[diag]}}}{dt} = \mathcal{K}_{II}^{\mathrm{diag}}$$

The default for **tdm_method** depends on the **coupling** keyword,

coupling ddr, **nacdr**	→	**tdm_method gradient**
coupling overlap	→	**tdm_method gradient**
coupling ktdc	→	**tdm_method energy**

convthre

The **convthre** keyword controls the maximum allowed differences between successive time steps in adaptive integrators. For **avv**, it is the maximum allowed energy difference in the current and previous time steps.

dtstep

The **dtstep** keyword controls the time step for the nuclear EOM. This keyword is not in Table 2.1 because it is the same in *SHARC* and *SHARC-MN*. We refer readers to *SHARC* manual for keyword **dtstep**.

stepsize_min

The **stepsize_min** keyword controls the minimum allowed time step for adaptive integrators. The recommended value is the default, which is **dtstep**/16.

stepsize_max

The **stepsize_max** keyword controls the maximum allowed time step for adaptive integrators. The recommended value is the default, which is **dtstep***2.

nac_projection and **nonac_projection**

These two keywords control the use of projected (effective) nonadiabatic coupling vector or original (effective) nonadiabatic coupling vector. The projected (effective) nonadiabatic coupling vector conserves nuclear angular momentum and the center of mass, and is the recommended

choice; it is also the default. We refer readers to reference [11] for detailed description of projection operator.

2.3 Combinations of keywords

Possible combinations of the keywords are shown in Table 2.2.

Table 2.2 Possible combinations of **coupling**, **neom,** and **nac_projectio**n keywords for running NAC-based, overlap-based, and curvature-driven SE or CSDM calculations[a]

keywords			nuclear EOM	Notes
coupling	**neom**	**nac_projection**		
nardr(ddr)	nardr(ddr)	nac_projection	eq	The method used in refs. [8,11]
nardr(ddr)	nardr(ddr)	nonac_projection	eq	
overlap	nardr(ddr)	nac_projection	eq	recommended to be most accurate
overlap	nacdr(ddr)	nonac_projection	eq	
overlap	gdiff	nac_projection	eq	The method used in ref. [10]
overlap	gdiff	nonac_projection	eq	

[a]All three keywords are used in all of the self-consistent-potential methods.

3 OUTPUT

SHARC provides an auxiliary program **data_extractor.x**, which can operate on the **output.dat** file to extract properties as a function of time along the trajectory. We refer the users to **Table 7.5** and **Section 7.10 Data Extractor: data_extractor.x** in the original *SHARC* manual for a detailed description of the **data_extractor.x** program. Here we describe additional options for this program.

Table 3.1 Additional command-line options for **data_extractor.x**

Option	Description	Default
-h	Display help message and quit.	Help not displayed
-f	File name	None – file name must be given.
-sk	skip parsing of geom., vel., grad., NAC.	False
-ccd	Write **ccoeff_diag.out**	False
-dend	Write **density_diag.out**	False
-cdend	Write **cdensity_diag.out**	False
-denm	Write **density_mch.out**	False
-rdd	Write **den_state_diag.out**	True
-crdd	Write **cden_state_diag.out**	False
-rdm	Write **den_state_mch.out**	True
-xs	**-e**, **, -rdm**, **, -rdd, , -d**, **, -cd**, **, -cm**, **, -p**, **, -x**	
-s	**-ccd**, **, -sp**, **, -xm**, **, -cb**, **, -da** plus **-xs**	Default
-l	**-crdd**, **, -id**, **, -im**, **,** plus **-s**	
-xl	**-ccd**, **, -dend**, **, -cdend**, **, -denm, , plus -l**	

Table 3.2 Content of the files written by **data_extractor.x** corresponding to the additional options shown in Table 3.1

n is the total number of states.
j is a state index ($j \in \{1, ..., n\}$).

File	#Columns	Columns	
ccoeff_diag.out	$2 + 2n$	1	Time: t (fs)
		2	Norm of coherent wave function: $\sum_j \left\| \left[c_j^{diag} \right]_{\text{coherent}} \right\|^2$
		$1 + 2j$	$\text{Re}\left(\left[c_j^{diag} \right]_{\text{coherent}} \right)$
		$2 + 2j$	$\text{Im}\left(\left[c_j^{diag} \right]_{\text{coherent}} \right)$
density_diag.out	$1+(n+1)*n/2$	1	Time: t (fs)
		$1+n*(j-1)+k-j+1$	Density matrix element $\mathbf{D}_{ij}$ in the diagonal representation
cdensity_diag.out	$1+(n+1)*n/2$	1	Time: t (fs)
		$1+n*(j-1)+k-j+1$	Coherent density matrix element $\left[\mathbf{D}_{ij} \right]_{\text{coherent}}$ in the diagonal representation.
density_mch.out	$1+(n+1)*n/2$	1	Time: t (fs)
		$1+n*(j-1)+k-j+1$	Density matrix element $\mathbf{D}_{ij}$ in MCH representation
den_state_diag.out	$1 + n$	1	Time t (fs)
		n	Real diagonal density matrix element $\text{Re}\left[\mathbf{D}_{ii} \right]$ in the diagonal representation
cden_state_diag.out	$1 + n$	1	Time t (fs)
		n	Real diagonal coherent density matrix element $\text{Re}\left[\left[\mathbf{D}_{ii} \right]_{\text{coherent}} \right]$ in the diagonal representation.
den_state_mch.out	$1 + n$	1	Time t (fs)
		n	Real diagonal density matrix element $\text{Re}\left[\mathbf{D}_{ii} \right]$ in the MCH representation

4 AUXILIARY SCRIPTS

In this section, we describe additions to auxiliary scripts. We refer the readers to chapter 7 of *SHARC* manual for descriptions of other auxiliary scripts.

4.1 State selected initial condition sampling: state_select.py

4.1.1 Basic usage

The **state_selected.py** script implements the vibrational state selection for initial conditions. This script performs similar function as the unimolecular initial condition as implemented in *ANT* software.[20] It also adopts many options from **wigner.py** (which is the Wigner distribution sampling script in *SHARC*). The vibrational state selected initial condition sampling involves five major inputs:

- Method to control the selection process, by option **--vibselec**.
- Vibrational energy distribution, by option **--vibdist**.
- Vibrational energy of each mode if required, by option **--vibstate** or **--vibene**.
- The number of initial condition to be sampled, by option **-n**.
- A file with ⇨*Molden* format from a frequency calculation. For usage with state_selected.py, only the following blocks have to be present in the *Molden* file:
  - [FREQ]
  - [FR-COORD]
  - [FR-NORM-COORD]

The general usage of the script is:

```
user@host> $SHARC/state_selected.py [options] filename.Molden
```

4.1.2 Description of major options

--vibstate

The **--vibstate** option controls the vibrational quantum number assigned to each mode. It can be followed by a series integers separated by "," or one single integer. The example of usage of this option will be described when introducing the usage of **--vibselect**.

--vibene

The **--vibene** option controls the vibrational energy assigned to each mode. It can be followed by a series float numbers separated by "," or one single float number. The example of usage of this option will be described when introducing the usage of **--vibene**.

--vibselect

The **--vibselect** option controls which vibrational state selection method is used. It involves following 6 options,

- **--vibselect 1** The user provides vibrational quantum numbers by option **–vibstate**. Examples:
  --vibselect 1 –vibstate $n_1, n_2, n_3, \ldots, n_{3N-6}$ for a local minimum; and,
  --vibselect 1 –vibstate $n_1, n_2, n_3, \ldots, n_{3N-7}$ for a saddle point.

The amount of energy in each mode *i* equals $E_i = (0.5 + n_i)h\upsilon_i$, where n_i is the vibrational quantum number of mode *i*, and *N* is the number of atoms, and υ_i is the vibrational frequency of mode *i*.

- **--vibselect 2** Only applies to minimum-energy structures (not saddle points). The script will assign a vibrational quantum number at a randomly selected from Boltzmann distribution at a user-specified temperature. The temperature is specified by option **--t**.
  Example:
  --vibselect 2 --t 273.15
  where temperature is set as 273.15 with unit of kelvin.
- **--vibselect 4** The amount of energy in each mode is the same. The mode energy is specified by option **--vibene**.
  Example:
  --vibselect 4 --vibene E_1
  The amount of energy in each mode equals E_1, and notice E_1 is in unit of eV.
- **--vibselect 5** The amount of energy in mode *i* is E_i, which can be different for each mode. The mode energies are specified by option **--vibene**.
  Examples:
  --vibselect 5 --vibene E_1, E_2, E_3, …, E_{3N-6}
  --vibselect 5 --vibene E_1, E_2, E_3, …, E_{3N-7}
  The amount of energy in each mode *i* equals E_i in unit of eV.
- **--vibselec 6** Similar as **--vibselec 4** except that the energy in each mode is calculated as $\min\,[0.5h\upsilon_i, E_1]$. Therefore each mode will have at least zero-point energy. The usage is similar to **--vibselec 4**.
- **--vibselec 7** Similar as **--vibselec 5** except that the energy in each mode is calculated as $\min\,[0.5h\upsilon_i, E_i]$. Therefore each mode will have at least zero-point energy. The usage is similar to **--vibselec 5**.

--vibdist

The **--vibdist** option controls the type of phase space distribution to be used for initial conditions. We suggest that users simply use the default for this option.

- **--vibdist 0** This is the default – a classical or quasiclassical distribution. It is a uniform distribution. This distribution is quasiclassical if using **--vibselect 1** or **--vibselect 2**; it is classical if using **--vibselect 3**, **--vibselect 4**, **--vibselect 5**, and **--vibselect 6**.
  With this option, the initial displacements are distributed between inner turning points and outer turning points. The inner and outer turning point are computed according to the formula for a classical harmonic oscillator with energy specified by **--vibene** or computed from **--vibstate**. The inner and outer turning point $\pm q_{\mathrm{turn},i}$ are computed by:
  $$\frac{1}{2}k_i q_{\mathrm{turn},i}^2 = E_i$$
  where k_i and E_i are the force constant and energy of mode *i* respectively. When this option is used, the following steps are taken:
  (1). Normal mode ordering is shuffled.
  (2). Loop over shuffled modes, and for each mode $i = 1,2,\ldots,3N-6$,

(2.1). A random number λ between 0 and 1 is generated, and the initial displacement is

$$q_i = q_{\mathrm{turn},i}\cos(2\pi\lambda)$$

where q_i is the normal mode displacement coordinate.

(2.2). Potential energy at geometry $\mathbf{R}_i = \mathbf{R}_0 + \sum_{\alpha=1}^{i} q_\alpha$ is evaluated, where α is a normal mode index, and $\mathbf{R}_0$ is the ground state minimum geometry. If the potential energy at geometry $\mathbf{R}_i$ – denoted as $V(\mathbf{R}_i)$ has too high energy that $V(\mathbf{R}_i) - V(\mathbf{R}_0) > E_i$, then $|q_i|$ is decreased by 10%. And this is repeated if necessary until $(\mathbf{R}_i) - V(\mathbf{R}_0) \leq E_i$. Notice that E_i is the assigned vibrational energy to mode *i*.

The potential energy can be evaluated in two methods: using classical harmonic approximation or directly from *ab initio* calculation. These two evaluations can be set by **--method** option: **--method 0** uses classical harmonic approximation; **--method 1** requires *ab initio* calculations. And default is **--method 0**. Harmonic approximation is suggested because *ab initio* calculation requires an additional complicated interface, and we only suggest advanced users to use **--method 1**. We will discuss **--method** option in **--method** option section.

(2.3). Another random number is chosen to determine the sign of momentum p_i of mode *i*.

(2.4). The momentum of mode i is then,

$$p_i = \mathrm{sign}(\lambda')\sqrt{2\mu\left(E_i - \left[V(\mathbf{R}_i) - V(\mathbf{R}_0)\right]\right)}$$

where μ is the normal-mode reduced mass.

- **--vibdist 1** This option only works for the ground vibrational state $n_i = 0$. The coordinate q_i is selected from the quantum mechanical harmonic oscillator coordinate distribution, which is the square of the ground-state wave function and is a Gaussian.

$$q_i = \sigma_x\sqrt{-2\ln(\lambda_1)}\cos(2\pi\lambda_2)$$
$$\sigma_x = \sqrt{E_i/k_i}$$

where λ_1 and λ_2 are two random numbers between 0 and 1. Then steps (1) and (2) in **--vibdist 0** are performed.

- **--vibdist 2** This option only works for the ground vibrational state $n_i = 0$. This corresponds to a Wigner distribution. The distribution is generated using the Box-Muller algorithm for the normal mode coordinate displacement and momentum.

$$q_i = \sigma_x\sqrt{-2\ln(\lambda_1)}\cos(2\pi\lambda_2)$$
$$\sigma_x = \sqrt{E_i/k_i}$$
$$p_i = \sigma_p\sqrt{-2\ln(\lambda_1)}\sin(2\pi\lambda_2)$$
$$\sigma_p = 1/(2\sigma_x)$$

where λ_1 and λ_2 are two random numbers between 0 and 1. Then steps (1) and (2) in **--vibdist 0** are performed.

--method

The **--method** option controls the approximation used in the evaluation of potential energies. The default is **--method 0**, which uses the harmonic approximation. Option **--method 1** uses the *ab initio* potential, which means it requires an additional step to compute potential energies from electronic structure software.

- **--method 0** This option approximates the potential energy of geometries that deviates from the minimum-energy structure $\mathbf{R}_0$ by using the harmonic approximation, and, therefore, the potential energy is only approximately correct. This means hat the sampled vibrational state selected initial conditions do not exactly correspond to a microcanonical ensemble because the energies of each of the initial conditions may be slightly different because of the harmonic approximation.
- **--method 1** This option uses the ab initio potential, and thus it requires an additional option **--template** to specify an additional user-written script. When using **--method 1**, the **state_selected.py** will generate a template geometry with units of Å called **tmp_state_selected.xyz**. Therefore, the function of the user-written script is to convert **tmp_state_selected.xyz** to an energy that is saved in file called **energy_state_selected**. The detailed function of this user-written script should include the following steps:
  - Generate electronic structure calculation input with **tmp_state_selected.xyz**
  - Call the electronic structure software to execute the generated input
  - Extract the energy form electronic structure output and save it to **energy_state_selected**

  Example of using --method 1:
  --method 1 --template MN15.script
  where MN15.script is an example of user-written script and is shown in Figure 4.1

```
#!/bin/bash

echo "%Chk=e.chk" > e1.tmp
echo "%Mem=11800mb" >> e1.tmp
echo "#P Def2TZVP SCF=(Tight,novaracc,maxcycle=500) guess=read Integral(Grid=UltraFine) MN15" >> e1.tmp
echo "  " >> e1.tmp
echo "krhf" >> e1.tmp
echo "  " >> e1.tmp
echo "0  1" >> e1.tmp

echo "  " > e2.tmp
echo "  " >> e2.tmp

cat e1.tmp tmp_state_selected.xyz e2.tmp > e.com

rm e1.tmp
rm e2.tmp
rm tmp_state_selected.xyz

export GAUSS_SCRDIT=/scratch.global/userID/wigner
export GAUSS_EXEDIR=/software/g09.c01/g09

/software/g09.c01/g09/g09 < e.com > e.out

grep 'SCF Done' e.out|awk '{print $5}' > energy_state_selected

rm e.com
rm e.out
```

Generate a Gaussian input file called e.com with geometry tmp_state_selected.xyz
It is a density functional calculation with MN15 functional

Execute a Gaussian calculation with e.com

Extract MN15 energy and save it to energy_state_selected

Figure 4.1 An example of user-written script MN15.script used for **--template**.

4.1.3 Normal mode types

This usage of **state_selected.py** is adapted from **wigner.py**. The normal mode vectors contained in a *Molden* file can follow different conventions. By default, state_selected.py attempts to identify the convention automatically, which is achieved by the **-f 0** option. Otherwise, there are four possible options: **-f 1** to assume normal modes in the *Gaussian* convention (used by *Gaussian, Turbomole, Q-Chem, ADF*, and *Orca*); **-f 2** to assume Cartesian normal modes (used by *Molcas*, *OpenMolcas*, and *Molpro*); **-f 3** to assume the Columbus convention; or **-f 4** for mass-weighted, orthogonal normal modes.

4.1.4 Non-default masses

This usage of **state_selected.py** is adapted from **wigner.py**. When the **-m** option is used, the script will ask the user to interactively modify the atom masses. The user can provide a mass for each atom, which is defined by the atom index in a *Molden* file. Note that when using non-default masses, the frequency calculation should be consistent with these non-default masses.

4.1.5 Output

Similar as **wigner.py**, **state_selected.py** will generate a single output file, by default called **initconds**. All information of sampled initial conditions is stored in this file.
The **initconds** file format is specified in section 7.5.5 of the *SHARC* manual.
When the **-x** option is given, additionally the script will produce a file named **initconds.xyz**, which contains the sampled geometries in standard *xyz* format.

Table 4.1 provides the complete list of options can be used with **state_selected.py**

Table 4.1 Command-line options for script **state_selected.py**

Option	Description	Default
-h	Display help message and quit	-
-n INTEGER	Number of initial conditions to generate	3
--vibselect INTEGER	Controls the method used in vibrational state selection	6
--vibdist INTEGER	Controls the type of phase space distribution	0
--vibene FLOAT, FLOAT, …	A list of vibrational energy assigned to each mode	0.0 eV
--vibstate INTEGER, INTEGER, …	A list of vibrational level assigned to each mode	0
--method INTEGER	Method to compute potential energies	0
--template FILENAME	A user-written script to evaluate ab initio potential energy (see section 4.1.2)	-
-m	Modify atom masses (starts interactive dialog)	Most common isotopes
-s FLOAT	Scaling factor for the frequencies	1.0
-t FLOAT	The temperature used in Boltzmann-weighted distribution	0.0K
-T	Discard very high vibrational state at high temperatures	Don't discard, but warn
-L FLOAT	Discard frequencies bellow the given one (in cm^{-1})	10.0
-o FILENAME	Output filename	**initconds**
-x	Creates an xyz file with the sampled geometries	**initconds.xyz**
-r INTEGER	Seed for random number generator	16661
-f F	Type of normal modes read (0=detect automatically, 1-4=see section 4.1.3)	0
--kee_trans_rot	Do not remove translations and rotations from velocity vector	
--use_eq_geom	Sampling only velocities, but keep equilibrium geometry	Sample normally
--use_zero_veloc	Sampling only geometries, but keep velocities to zero	Sample normally

Appendix A. GNU General Public License v3

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those

that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.
The precise terms and conditions for copying, distribution and modification follow.
TERMS AND CONDITIONS
0. Definitions.
“This License” refers to version 3 of the GNU General Public License.
“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.
“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.
To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.
A “covered work” means either the unmodified Program or a work based on the Program.
To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.
To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.
An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.
1. Source Code.
The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.
A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.
The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.
The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System

Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities,

provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program,

that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.
When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.
Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.
If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.
Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).
However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally

terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.
Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.
Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.
An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.
You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.
A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.
Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.
If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.
If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.
A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.
Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

Appendix B. CC-BY-4.0 License

Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. **Adapter's License** means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.

i. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal

protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

a. **License grant**.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:
    A. reproduce and Share the Licensed Material, in whole or in part; and
    B. produce, reproduce, and Share Adapted Material.
2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.
3. Term. The term of this Public License is specified in Section 6(a).
4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.
5. Downstream recipients.
    A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.
    B. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.
6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. **Other rights**.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. **Attribution**.
    1. If You Share the Licensed Material (including in modified form), You must:
        A. retain the following if it is supplied by the Licensor with the Licensed Material:
            i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
            ii. a copyright notice;
            iii. a notice that refers to this Public License;
            iv. a notice that refers to the disclaimer of warranties;
            v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
        B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
        C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.
    2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
    3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.
    4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;
b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and
c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

a. **Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.**
b. **To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.**
c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.
b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:
    1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or
    2. upon express reinstatement by the Licensor.

    For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.
c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.
d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.
b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.
b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable.

If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.
d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

V. References

[1] R. Crespo-Otero, and M. Barbatti, "Recent Advances and Perspectives on Nonadiabatic Mixed Quantum-Classical Dynamics". *Chem. Rev.* **118**, 7026-7068 (2018)

[2] M. D. Hack, and D. G. Truhlar, "A Natural Decay of Mixing Algorithm for Non-Born-Oppenheimer Trajectories". *J. Chem. Phys.* **114**, 9305-9314 (2001)

[3] C. Zhu, A. W. Jasper, D. G. Truhlar, "Non-Born-Oppenheimer Trajectories with Self-Consistent Decay of Mixing". *J. Chem. Phys.* **120**, 5543-5557 (2004)

[4] C. Zhu, S. Nangia, A. W. Jasper, and D. G. Truhlar, "Coherent Switching with Decay of Mixing: An Improved Treatment of Electronic Coherence for Non-Born-Oppenheimer Trajectories". *J. Chem. Phys.* **121**, 7658-7670 (2004)

[5]W. H. Zurek, "Pointer Basis of a Quantum Apparatus: Into What Mixture Does the Wavepacket Collapse? ". *Phys. Rev. D* **24**, 1516-1525 (1981)

[6] W. H. Zurek, "Decoherence, Einselection, and the Quantum Origins of the Classical". *Rev. Mod. Phys.* **75**, 715-775 (2003)

[7] C. Zhu, A. W. Jasper, and D. G. Truhlar, "Non-Born-Oppenheimer Liouville-von Neumann Dynamics. Evolution of a Subsystem Controlled by Linear and Population-Driven Decay of Mixing with Decoherent and Coherent Switching". *J. Chem. Theory Comput.* **1**, 527-540 (2005).

[8] M. S. Topaler, T. C. Allison, D. W. Schwenke, and D. G. Truhlar, "What is the Best Semiclassical Method for Photochemical Dynamics in Systems with Conical Intersections?" *J. Chem. Phys.* **109**, 3321-3345 (1998).

[9] Y. Shu, L. Zhang, S. Mai, S. Sun, L. González, and D. G. Truhlar, "Implementation of Coherent Switching with Decay of Mixing in the SHARC program". *J. Chem. Theory Comput.*, **16**, 3464-3475 (2020)

[10] Y. Shu, L. Zhang, S. Sun, and D. G. Truhlar, "Time-Derivative Couplings in Electronically Nonadiabatic Dynamics Based on a Self-Consistent Potential". *J. Chem. Theory Comput.*, **16**, 4098-4106 (2020)

[11] Y. Shu, L. Zhang, Z. Varga, K. A. Parker, S. Kanchanakungwankul, S. Sun, and D. G. Truhlar, "Conservation of Angular Momentum in Direct Nonadiabatic Dynamics". *J. Phys. Chem. Lett.* **11**, 1135-1140 (2020).

[12] Y. Shu, L. Zhang, X. Chen, S. Sun, Y. Huang, and D. G. Truhlar, "Nonadiabatic Dynamics Algorithms with Only Potential Energies and Gradients: Curvature-Driven Coherent Switching with Decay of Mixing and Curvature-Driven Trajectory Surface Hopping". *J. Chem. Theory Comput.* **18**, 1320-1328 (2022).

[13] A. W. Jasper, and D. G. Truhlar, "Fewest-Switches with Time Uncertainty: A Modified Trajectory Surface-Hopping Algorithm with Better Accuracy for Classically Forbidden Electronic Transitions". J. Chem. Phys. **116**, 5424-5431 (2001).

[14] X. Zhao, Y. Shu, L. Zhang, X. Xu, and D. G. Truhlar, "Direct Nonadiabatic Dynamics of Ammonia with Curvature-Driven Coherent Switching with Decay of Mixing and with Fewest Switches with Time Uncertainty: An Illustration of Population Leaking in Trajectory Surface Hopping Due to Frustrated Hops". J. Chem. Theory Comput. **23**, 1672-1685 (2023).

[15] Y. Shu, L. Zhang, D. Wu, X. Chen, S. Sun, and D. G. Truhlar, "New Gradient Correction Scheme for Electronically Nonadiabatic Dynamics Involving Multiple Spin States". Accepted by J. Chem. Theory Comput.

[16] G. Granucci, M. Persico, and A. Zoccante, "Including Quantum Decoherence in Surface Hopping", J. Chem. Phys. **133**, 134111 (2010).

[17] F. Plasser, S. Mai, M. Fumanal, E. Gindensperger, C. Daniel, and L. González, "Strong Influence of Decoherence Corrections and Momentum Rescaling in Surface Hopping Dynamics of

Transition Metal Complexes" , J. Chem. Theory Comput. **15**, 5031-5045 (2017).

[18] G. A. Meek, and B. G. Levine, "Evaluation of the Time-Derivative Coupling for Accurate Electronic State Transition Probabilities from Numerical Simulations", J. Phys. Chem. Lett. **5**, 2351-2356 (2014).

[19] A. W. Jasper, and D. G. Truhlar, "Improved Treatment of Momentum at Classically Forbidden Electronic Transitions in Trajectory Surface Hopping Calculations", Chem. Phys. Lett. **369**, 60-67 (2003).

[20] Y. Shu, L. Zhang, J. Zheng, Z.-H. Li, A. W. Jasper, D. A. Bonhommeau, R. Valero, R. Meana-Pañeda, S. Mielke, Z. Varga, and D. G. Truhlar, *ANT 2023*, University of Minnesota, Minneapolis, 2023. https://doi.org/10.5281/zenodo.7807406