MANUAL

SHARC-MN – version 1.2

Surface Hopping with Arbitrary Couplings – MN extension

Yinan Shu
University of Minnesota

Linyao Zhang
Harbin Institute of Technology and University of Minnesota

Donald G. Truhlar
University of Minnesota

SHARC-MN–v1.2 was finalized on February 10, 2022
The Web page of *SHARC-MN* is https://comp.chem.umn.edu/sharc-mn
This manual was last updated on February 10, 2022
SHARC-MN–v1.2 is built on *SHARC*–v2.1.
For recent versions of *SHARC*, see https://sharc-md.org

TABLE OF CONTENTS

Licensing

Acknowledgments

We requested permission from the authors of *SHARC*–v2.1 to distribute this modified version of the code, and we were given permission. We are grateful to the authors of *SHARC*–v2.1 for making their code available and for their cooperation every step of the way. Some of the modifications in *SHARC-MN* are scheduled to be included in a future version of *SHARC* itself.

Our work on the MN extension of *SHARC*–v2.1 was supported in part by the U. S. Department of Energy, Office of Basic Energy Sciences.

Citations

Users of *SHARC-MN*–1.2 are requested to cite both *SHARC* and *SHARC-MN*. Examples of appropriate referencing are as follows:

- Y. Shu, L. Zhang, and D. G. Truhlar, *SHARC-MN*–v1.2 (University of Minnesota, Minneapolis, 2022), https://comp.chem.umn.edu/sharc-mn
- 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-v2.1* (University of Vienna, Wien, 2019), https://sharc-md.org

See also the references for methods in Sections I.D and I.E.

I. Introduction

SHARC-MN is an extended version of *SHARC*. Both codes are used for direct dynamics calculations of electronically nonadiabatic processes 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. SHARC-MN also includes curvature-driven methods that do not require NACs.

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 coherences evolve in time are affected by more than one potential energy surface.

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 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 (at this time) in *SHARC*, and they are described in this manual. *SHARC-MN*-v1.2 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 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 in comparison to converged quantum dynamics. Therefore, although SHARC-MN contains all three methods, the one that is recommended is CSDM.

The nonadiabatic coupling vectors (NACs) that are usually used for direct dynamics 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 or the effective NACs used in some of the methods.

The **first set of major extensions** of *SHARC* that are included in *SHARC-MN*, versions 1.0 and later, are

- the inclusion of the SE, SCDM, and CSDM methods
- NAC projection
- the incorporation of adaptive time steps

The **second set of major extensions** (included in versions 1.0 and later) consists of adding the time-derivative versions of the the SE, SCDM, and CSDM methods. These are called tSE, tSCDM, and tCSDM. Among these, tCSDM is recommended. The time-derivative modification allows one to use time derivatives rather than NACs; this is more convenient and more efficient, and it allows more accurate integrations in regions where the NACs show sharp spikes.

The **third 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.

I.A. 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, PESs and NACs, or 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 wave function is 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 (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; 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.

!!

I.B. 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.

!!

I.C.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.

I.C.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.
- Auxiliary scripts:
  - **setup_traj.py** added the curvature based approximation option for coupling, the keyword is “etdc”.

I.C.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 difference of energy

    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 reflect momentum after a frustrated hop in directions of effective nonadiabatic coupling vector and projected effective nonadiabatic coupling vector (as expalined in Section II.B).
- Auxiliary scripts:
  - **setup_traj.py** changed the curvature based approximation option for coupling, the keyword is now changed to "ktdc".

I.D. 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.E for specific dynamical methods.

I.E. References: specific methods

- semiclassical Ehrenfest (SE): [8,9]
- coherent switching with decay of mixing (CSDM): [8]
- tSE and tCSDM: methods that use time-derivative couplings: [10]
- Projected NAC: [11]
- Curvature-driven methods, κCSDM, κSE, κTSH, and κTSH-EDC: [14]

I.F. Installation

Installation of *SHARC-MN* requires the system specific modification of **Makefile** that is located in the **source** folder. Installation is done by a single command from the **source** folder:

make

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

II. Input Keywords

II.A. List of keywords

Table II.1: Input keywords for sharc.x. The first column gives the name of the keyword, the second lists possible arguments, and the third line 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*. (The **overlap** keyword has not changed, but it is included for completeness.) Users should refer to Table 4.1 in 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.

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 $=ehrenfest will use a self-consistent potential; these two choices of keyword have the same meaning. Note that CSDM and SCDM are both combinations of Ehrenfest with decay of mixing and 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**	Quantities describing the nonadiabatic couplings. Uses vectorial nonadiabatic couplings $\langle\psi_I\|\partial/\partial\mathbf{R}\|\psi_J\rangle$. Uses temporal nonadiabatic couplings $\langle\psi_I\|\partial/\partial t\|\psi_J\rangle$. Uses the overlaps $\langle\psi_I(t_0)\|\psi_J(t)\rangle$ (local diabatization). Uses curvature-driven method for approximation of time derivative coupling.
ktdc_method	**string** $1=**energy** **(default for TSH)** $1=**gradient** **(default for SCP)**	Methods used to compute κTDC second order difference of energy (default for TSH) first order 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 correction for TSH. The TSH/EDC scheme is explained in Refs. [12,13] 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	**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	**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)
ekincorrect	**string** $1=none $1=**parallel_vel** $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. Velocity is rescaled to adjust kinetic energy. Only the velocity component in the direction of $\langle \psi_\alpha \vert \partial/\partial R \vert \psi_\beta \rangle$ 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	**string** $1=**none** $1=parallel_vel $1=parallel_nac $1=parallel_diff $1=parallel_pnac $1=parallel_enac $1=parallel_penac	Adjustment of the kinetic energy after a surface hop. No reflection. Full velocity vector is reflected. Only the velocity component in the direction of $\langle \psi_\alpha \vert \partial/\partial R \vert \psi_\beta \rangle$ 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.
integrator	**string** $1=**fvv** $1=avv $1=bsh	Will dynamics be calculated with adaptive time step? fixed time-step Velocity Verlet Integrator Adaptive time-step Velocity Verlet Integrator Bulirsch-Stoer Integrator

convthre	**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.
dtmin	**float** **0.0001 fs**	Minimum time step allowed in adaptive Velocity Verlet in fs.
nac_projection nonac_projection		Applies projected NAC (default) Applies original NAC

II.B. Detailed descriptions of the keywords

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 used in propagation of electronic coefficients

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

$$\sigma_{IJ}=\langle\psi_I|\partial/\partial t|\psi_J\rangle\approx\frac{1}{2}\left[\frac{\partial^2(V_I-V_J)}{\partial t^2}\frac{1}{V_I-V_J}\right]^{1/2}$$

ktdc_method

The **ktdc_method** keyword controls the way how κTDC is computed.

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

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

  Starting from fourth step, it is computed as,

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

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

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

integrator

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

- **fvv** for fixed time step Velocity Verlet
- **avv** for adaptive time step Velocity Verlet,
- **bsh** for Bulirsch-Stoer integrator

The **fvv** integrator is same as the default integrator in previous *SHARC* programs. We recommend using **fvv** or **avv**.

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.

dtimin
The **dtmin** keyword controls the minimum allowed time step for adaptive integrators.

decoherence_scheme
When performing self-consistent potential based methods, 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 [4,8], which is the default and is recommended.
- **switching_procedure scdm** turns on the older self-consistent decay of mixing [3,8]

neom
The **neom** keyword is used to control the form of the nonadiabatic coupling vector in the nuclear equations of motion; note that this is a separate choice from the form of the NAC 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 nuclear equation of motion follows original format for SE and CSDM, i.e.,

$$\left[\dot{\mathbf{P}}\right]_{\mathrm{SE}} = -\sum_{I} \mathrm{Re}\left(\rho_{II}\right)\frac{\partial V_I}{\partial \mathbf{R}} + \sum_{I}\sum_{J\neq I} \mathrm{Re}\left(\rho_{IJ}\right)\left(V_I - V_J\right)\mathbf{d}_{IJ}$$

$$\left[\dot{\mathbf{P}}\right]_{\mathrm{CSDM}} = \left[\dot{\mathbf{P}}\right]_{\mathrm{SE}} + \left[\dot{\mathbf{P}}\right]_{\mathrm{DM}}$$

$$\left[\dot{\mathbf{P}}\right]_{\mathrm{DM}} = \sum_{I\neq K}^{N} \frac{\rho_{II}\left(t\right)}{\tau_{IK}} \frac{\left(V_I - V_K\right)}{\left(\mathbf{s}_{IK}\bullet\mathbf{v}\right)}\mathbf{s}_{IK}$$

Alternatively, one can use an effective nonadiabatic coupling vector by setting **neom gdiff**. The effective nonadiabatic coupling vector is a vector defined in Section IV.C 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 the overlap of electronic wave functions at successive steps. See reference [10] for detailed description of effective nonadiabatic coupling

vector. Using the effective nonadiabatic coupling vector is the default option if one uses time-derivative coupling to propagate electronic coefficients, which is set by **coupling overlap**. If one uses **neom gdiff**, one must use **coupling overlap**, and if one uses **coupling overlap** one may use **neom nacdr**, **neom nacdr.** or **neom gdiff**. When using effective NAC in nuclear equation of motion, to conserve the energy of the system, the form has to be changed to,

$$\left[\dot{\mathbf{P}}\right]_{\mathrm{SE}} = -\sum_{I} \mathrm{Re}\left(\rho_{II}\right)\frac{\partial V_I}{\partial \mathbf{R}} + \sum_{I}\sum_{J\neq I}\left[\frac{\mathrm{Re}\left(\rho_{IJ}\right)\left(V_I - V_J\right)\sigma_{IJ}}{\mathbf{v}\bullet\mathbf{G}_{IJ}}\right]\bullet\mathbf{G}_{IJ}$$

where $\mathbf{G}_{IJ}$ is effective NAC, and σ_{IJ} is time derivative coupling between states *I* and *J*.

One does not need to set up **neom** keyword. In that situation, we will employ the default choice of nuclear EOM based on **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

nac_projection and **nonac_projection**
This two keywords control the use of projected nonadiabatic coupling vector or original nonadiabatic coupling vector. The projected 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.

ekincorrect
The **ekincorrect** keyword controls how the kinetic energy is adjusted after a surface hop to preserve total energy. The suggested methods are **ekincorrect parallel_pnac** for TSH with **coupling nacdr**, and **ekincorrect parallel_penac** for TSH with **coupling overlap** and **coupling ktdc**.

- **none** deactivates the adjustment, so that the total energy is not preserved after a hop. Using this option, jumps can never be frustrated and are always performed according to the hopping probabilities.
- **parallel_vel** (the default) the kinetic energy is adjusted by simply rescaling the nuclear velocities 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, angular momentum will NOT be conserved.
- **parallel_nac** the kinetic is adjusted by rescaling 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** 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.
- **parallel_diff** the kinetic energy is adjusted by rescaling 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 kinetic is adjusted by rescaling 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** 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** the kinetic 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_enac** 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** the kinetic 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 inverted after a frustrated hop. The suggested methods are **reflect_frustrated parallel_pnac** for TSH with **coupling nacdr**, and **reflect_frustrated parallel_penac** for TSH with **coupling overlap** and **coupling ktdc**.

- **none** (the default) after a frustrated hop, the velocity vector is not modified.
- **parallel_vel** the full velocity vector is inverted when a frustrated hop is encountered.
- **parallel_nac** only the velocity component parallel to the 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 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.

II.C. Combinations of keywords

Possible combinations of the keywords are shown in Table II.2.

Table II.2: Possible combinations of **coupling**, **neom,** and **nac_projectio**n keywords for running SE or CSDM calculations[a]

keywords			nuclear EOM	Notes
coupling	**neom**	**nac_projection**		
nardr(ddr)	nardr(ddr)	nac_projection	eq 21	The method used in refs. [8,11]
nardr(ddr)	nardr(ddr)	nonac_projection	eq 9	
overlap	nardr(ddr)	nac_projection	eq 21	recommended to be most accurate
overlap	nacdr(ddr)	nonac_projection	eq 9	
overlap	gdiff	nac_projection	eq 10	The method used in ref. [10]
overlap	gdiff	nonac_projection	eq 17	

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

III. 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 original *SHARC* manual for a detailed description of the **data_extractor.x** program. Here we describe additional options for this program.

Table III.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 III.2. Content of the files written by **data_extractor.x** corresponding to the additional options shown in Table III.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

IV. Theory

IV.A. Decoherence by decay of mixing

Decay of mixing is described in Refs. [2,3,4,8]. We refer the reader to Ref. [8] for details of implementing decay of mixing to be compatible with *SHARC*; the decoherent propagator is introduced in the MCH representation. The decay-of-mixing decoherence scheme is used by **decoherence_scheme dom**. The electronic coefficients are propagated with the following electronic equation of motion:

$$\mathbf{c}^{\mathrm{diag}}\left(t+\Delta t\right)=\mathbf{U}^{\dagger}\left(t+\Delta t\right)\left[\mathbf{P}_{\mathbf{C}}^{\mathrm{MCH}}\left(t+\Delta t,t\right)\mathbf{P}_{\mathbf{D}}^{\mathrm{MCH}}\left(t+\Delta t,t\right)\right]\mathbf{U}\left(t\right)\mathbf{c}^{\mathrm{diag}}\left(t\right) \tag{1}$$

The coherent propagator and decoherent propagator (which is diagonal) are

$$\mathbf{P}_{\mathbf{C}}{}^{\mathrm{MCH}}\left(t+\Delta t,t\right)=\exp\left(-\left(i\mathbf{H}^{\mathrm{MCH}}+\mathbf{v}\bullet\mathbf{d}^{\mathrm{MCH}}\right)\Delta t\right) \tag{2}$$

$$\left[P_D{}^{\mathrm{MCH}}\left(t+\Delta t,t\right)\right]_{II}=\begin{cases}-\dfrac{\Delta t}{2\tau_I\left(t\right)} & I\neq K\\ \displaystyle\sum_{J\neq K}^{N}\frac{\Delta t}{2\tau_J\left(t\right)}\frac{\rho_{JJ}{}^{\mathrm{MCH}}\left(t\right)}{\rho_{KK}{}^{\mathrm{MCH}}\left(t\right)} & I=K\end{cases} \tag{3a}$$

$$\left[P_D{}^{\mathrm{MCH}}\left(t+\Delta t,t\right)\right]_{II}=\begin{cases}-\dfrac{\Delta t}{2\tau_I\left(t+\Delta t\right)} & I\neq K\\ \displaystyle\sum_{J\neq K}^{N}\frac{\Delta t}{2\tau_J\left(t+\Delta t\right)}\frac{\rho_{JJ}{}^{\mathrm{MCH}}\left(t\right)}{\rho_{KK}{}^{\mathrm{MCH}}\left(t\right)} & I=K\end{cases} \tag{3b}$$

where eq (3a) is used in versions 1.0 and 1.1, and eq (3b) is used in version 1.2, τ_I is the decoherence time for state *I*:

$$\tau_I=\frac{\hbar}{\left|V_I-V_K\right|}\left(C+\frac{2E_0}{\displaystyle\sum_{i}^{N_{\mathrm{atoms}}}\frac{M_i}{2}\left|\mathbf{v}_i\bullet\hat{\mathbf{s}}_{i,IK}\right|^2}\right) \tag{4}$$

where V_I is the adiabatic energy of state *I*, and

$$\hat{\mathbf{s}}_{i,IK}=\frac{\mathbf{s}_{i,IK}}{\left|\mathbf{s}_{i,IK}\right|} \tag{5}$$

where *C* and E_0 are parameters, and M_i and $\mathbf{v}_i$ are the mass and velocity of atom *i*. The default values of the parameters are $C = 1$ (unitless) and $E_0 = 0.1$ hartree. The decoherent direction $\mathbf{s}_{IK}$ is given by,

$$\mathbf{s}_{IK}=\mathrm{Re}\left(\frac{a_0\mathbf{P}_{\mathrm{vib}}\bullet\mathbf{d}_{IK}}{\left|\mathbf{d}_{IK}\right|}\mathbf{d}_{IK}\right)+\mathbf{P}_{\mathrm{vib}} \tag{6}$$

where $a_0 \equiv 1$ bohr, $\mathbf{P}_{\mathrm{vib}}$ is the internal vibrational momentum (computed by removing the overall angular motion from the total momentum of the molecule), and $\mathbf{d}_{IK}$ is the nonadiabatic coupling vector between state *I* and decoherent state *K*. We use “decoherent state” and “pointer

state" as synonyms. Starting from *SHARC-MN* version 1.2, τ in eq 3 employs $\tau\left(t+\Delta t\right)$, in which,

$$\tau_I\left(t+\Delta t\right)=\frac{\hbar}{\left|V_I\left(t+\Delta t\right)-V_K\left(t+\Delta t\right)\right|}\left(C+\frac{2E_0}{\sum_i^{N_{\text{atoms}}}\frac{M_i}{2}\left|\mathbf{v}_i\left(t+\Delta t\right)\bullet\hat{\mathbf{s}}_{i,IK}\left(t+\Delta t\right)\right|^2}\right) \tag{7}$$

and since velocity $\mathbf{v}\left(t+\Delta t\right)$ is not available at the moment of computing decoherence time and decoherence direction, it is approximated by forward velocity-Verlet,

$$\mathbf{v}\left(t+\Delta t\right)\approx\mathbf{v}\left(t\right)+\mathbf{a}\left(t\right)\Delta t \tag{8}$$

In addition,

$$\mathbf{s}_{IK}\left(t+\Delta t\right)=\text{Re}\left(\frac{a_0\mathbf{P}_{\text{vib}}\left(t+\Delta t\right)\bullet\mathbf{d}_{IK}\left(t+\Delta t\right)}{\left|\mathbf{d}_{IK}\left(t+\Delta t\right)\right|}\mathbf{d}_{IK}\left(t+\Delta t\right)\right)+\mathbf{P}_{\text{vib}}\left(t+\Delta t\right) \tag{9}$$

IV.B. Pointer-state switching

The CSDM pointer-state switching probabilities between decoherent state *K* and another state *I* is,

$$P_{K\to I}=\left(1-\frac{\left|\left[c_K{}^{\text{diag}}\left(t+\Delta t\right)\right]_{\text{co}}\right|^2}{\left|\left[c_K{}^{\text{diag}}\left(t\right)\right]_{\text{co}}\right|^2}\right)\times\frac{\text{Re}\left[\left[c_I{}^{\text{diag}}\left(t+\Delta t\right)\right]_{\text{co}}\left(\mathbf{P_C}{}^{\text{diag}}\left(t+\Delta t,t\right)_{IK}\right)^*\left[c_K{}^{\text{diag}}\left(t\right)^*\right]_{\text{co}}\right]}{\left|\left[c_K{}^{\text{diag}}\left(t\right)\right]_{\text{co}}\right|^2-\text{Re}\left[\left[c_K{}^{\text{diag}}\left(t+\Delta t\right)\right]_{\text{co}}\left(\mathbf{P_C}{}^{\text{diag}}\left(t+\Delta t,t\right)_{KK}\right)^*\left[c_K{}^{\text{diag}}\left(t\right)^*\right]_{\text{co}}\right]} \tag{10}$$

where $\left[\mathbf{c}^{\text{diag}}\left(t\right)\right]_{\text{co}}$ are the coherent coefficients propagated fully coherently along the trajectory, and they are reset to $\mathbf{c}^{\text{diag}}\left(t\right)$ at each local minimum D_K. The coupling strength D_K is defined differently depend on which coupling is used. For nonadiabatic coupling vector,

$$D_K\left(t\right)=\sum_{I\neq K}^{N}\left|\mathbf{d}_{IK}\right| \tag{11}$$

For overlap,

$$D_K\left(t\right)=\sum_{I\neq K}^{N}\left|\sigma_{IK}\right| \tag{12}$$

Using of equation 9 is similar to the CSDM-C version in original CSDM reference[4]. The keyword coupling controls the use of equation 8 or 9 automatically, equation 8 corresponds to **coupling ddr** or **coupling nacdr**, and equation 9 corresponds to **coupling overlap**. The negative switching probabilities are set to zero and

$$P_{K\to K} = 1 - \sum_{I\neq K}^{N} P_{K\to I} \tag{13}$$

We refer the reader to reference [8] for details. This is used with **switching_procedure csdm**.

IV.C. Effective nonadiabatic coupling

The nuclear equation of motion in self-consistent potential methods in its original form requires full information of nonadiabatic couplings:

$$\dot{\mathbf{P}}(t) = -\sum_{I} \mathrm{Re}\left(\rho_{II}\right)\frac{\partial V_I}{\partial \mathbf{R}} + \sum_{I}\sum_{J\neq I} \mathrm{Re}\left(\rho_{IJ}\right)\left(V_I - V_J\right)\mathbf{d}_{IJ} \tag{14}$$

One can, however, employs an effective nonadiabatic couplings, $\mathbf{G}_{IJ}$ to replace $\mathbf{d}_{IJ}$. Using of $\mathbf{G}_{IJ}$ requires eq 14 to be re-written as,

$$\dot{\mathbf{P}} = -\sum_{I} \mathrm{Re}\left(\rho_{II}\right)\frac{\partial V_I}{\partial \mathbf{R}} + \sum_{I}\sum_{J\neq I}\left[\frac{\mathrm{Re}\left(\rho_{IJ}\right)\left(V_I - V_J\right)\sigma_{IJ}}{\mathbf{v}\bullet\mathbf{G}_{IJ}}\right]\mathbf{G}_{IJ} \tag{15}$$

The $\mathbf{G}_{IJ}$ is defined as,

$$\mathbf{G}_{IJ} = \mathbf{g}_{IJ} + \alpha\dot{\mathbf{R}} \tag{16}$$

where α is a parameter defined as

$$\alpha = \frac{\sigma_{IJ} - \dot{\mathbf{R}}\bullet\mathbf{g}_{IJ}}{\dot{\mathbf{R}}\bullet\dot{\mathbf{R}}} \tag{17}$$

A consequence of eq 17 is that

$$\dot{\mathbf{R}}(t)\bullet\mathbf{G}_{IJ}(t) = \sigma_{IJ}(t) \tag{18}$$

where,

$$\sigma_{IJ}(t) = \left\langle \phi_I\left(\mathbf{r};\mathbf{R}(t)\right)\middle|\partial\phi_J\left(\mathbf{r};\mathbf{R}(t)\right)/\partial t\right\rangle \tag{19}$$

is time-derivative coupling. This is related to a nonadiabatic coupling by,

$$\sigma_{IJ}(t) = \dot{\mathbf{R}}(t)\bullet\mathbf{d}_{IJ}(t) \tag{20}$$

And comparison of eq 18 to eq 20 justifies calling $\mathbf{G}_{IJ}$ an effective nonadiabatic coupling. To preserve the angular momentum, one should use a projected effective nonadiabatic coupling,

$$\mathbf{G}_{IJ}^{\mathbf{Q}} = \left(\mathbf{1}-\mathbf{Q}\right)\left(\mathbf{g}_{IJ} + \alpha\dot{\mathbf{R}}\right) \tag{21}$$

where **Q** is a projection operator that removes translational and rotational components. And hence using of a projected effective nonadiabatic coupling results in the nuclear equation of motion as,

$$\dot{\mathbf{P}} = -\sum_{I} \mathrm{Re}\left(\rho_{II}\right)\frac{\partial V_I}{\partial \mathbf{R}} + \sum_{I}\sum_{J\neq I}\left[\frac{\mathrm{Re}\left(\rho_{IJ}\right)\left(V_I - V_J\right)\sigma_{IJ}}{\mathbf{v}\bullet\mathbf{G}_{IJ}^{\mathbf{Q}}}\right]\mathbf{G}_{IJ}^{\mathbf{Q}} \tag{22}$$

Setting **neom ddr** and **neom nacdr** is using eq 14 to propagate the nuclear equation of motion, while setting **neom gdiff** is using eq 15 or 22 to propagate the nuclear equation of motion depends on whether one sets **nonac_projection** or **nac_projection**.

IV.D. Projection operator

To conserve both the nuclear angular momentum and the center of mass in nonadiabatic dynamics trajectories with TSH or self-consistent potential methods, one should use the projected nonadiabatic coupling or the projected effective nonadiabatic coupling.

The projection operator is a $3N\times 3N$ matrix with elements

$$Q_{i\gamma,i'\gamma'} = \frac{1}{N}\delta_{\gamma\gamma'} + \sum_{\alpha}\sum_{\beta}\sum_{\alpha'}\sum_{\beta'}\varepsilon_{\alpha\beta\gamma}R_{i\alpha}\left[\tilde{\mathbf{I}}^{-1}\right]_{\beta\beta'}\sum_{\alpha'}\sum_{\beta'}\varepsilon_{\alpha'\beta'\gamma'}R_{i'\alpha'} \tag{23}$$

where indices i and i' label the nuclei and vary from 1 to N, α, β, γ, α', β', and γ' take on the values x, y, and z, $\tilde{\mathbf{I}}^{-1}$ is the inverse of matrix $\tilde{\mathbf{I}}$, matrix $\tilde{\mathbf{I}}$ is same as moment of inertia matrix with all masses set to 1, and $\boldsymbol{\varepsilon}$ is the completely antisymmetric third-order unit pseudotensor, whose elements are the Levi-Civita symbol. The first term of the projection operator projects onto the three directions corresponding to overall translation, and the second term projects onto the three directions corresponding to overall rotation.

Use of projection operator on nonadiabatic coupling results in a projected nonadiabatic coupling,

$$\mathbf{d}_{IJ}^{\mathbf{Q}} = \left(\mathbf{1}-\mathbf{Q}\right)\mathbf{d}_{IJ} \tag{24}$$

For TSH, one uses the projected NAC when the momentum after a hop is adjusted,

$$\mathbf{P}_i\big|_{\text{post-hop}} = \mathbf{P}_i\big|_{\text{pre-hop}} - \alpha_{IJ}\mathbf{d}_{IJ,i}^{\mathbf{Q}} \tag{25}$$

For semiclassical Ehrenfest and CSDM dynamics, one employs a self-consistent potential (SCP), and this yields

$$\dot{\mathbf{P}} = -\sum_{I=1}^{n}\mathrm{Re}\left(\rho_{II}\right)\frac{\partial V_I}{\partial \mathbf{R}} + \sum_{I=1}^{n}\sum_{J\neq I}^{n}\left[\frac{\mathrm{Re}\left(\rho_{IJ}\right)\left(V_I - V_J\right)\left(\mathbf{v}\bullet\mathbf{d}_{IJ}\right)}{\mathbf{v}\bullet\mathbf{d}_{IJ}^{\mathbf{Q}}}\right]\mathbf{d}_{IJ}^{\mathbf{Q}} \tag{26}$$

For CSDM, the decoherent direction is,

$$\mathbf{s}_{AK} = \frac{a_0\mathbf{P}_{\text{vib}}\bullet\mathbf{d}_{AK}^{\mathbf{Q}}}{\left|\mathbf{d}_{AK}^{\mathbf{Q}}\right|}\mathbf{d}_{AK}^{\mathbf{Q}} + \mathbf{P}_{\text{vib}} \tag{27}$$

We refer users to reference [11] for more detailed information.

IV.D. Curvature-driven time derivative coupling

Selection the **coupling** keyword to **ktdc** allows one employs an approximation to time derivative coupling that is computed from electronic wave function overlap integrals of successive time steps. This new approximation of time derivative coupling is called curvature-driven time derivative coupling, κTDC. Methods that use κTDC in equation of motion are the series of κ methods that described above, namely, κSE, κCSDM, and κTSH. The κTDC writes,

$$\kappa\text{TDC} \equiv \sigma_{IJ}^{\kappa} = \left\langle\phi_I\left|d/dt\right|\phi_J\right\rangle \cong \frac{1}{2}\left[\frac{d^2\left(V_I - V_J\right)}{dt^2}\frac{1}{V_I - V_J}\right]^{1/2}, \quad \text{for } J > I \tag{28}$$

where $\left(V_I - V_J\right)$ is the local gap between adiabatic potential surfaces, and we use κ as a prefix and as a superscript to denote approximation based on the curvature of the gaps. Since the NAC

is skew-Hermitian, we also have

$$\sigma_{JI}^{\kappa} = -\sigma_{IJ}^{\kappa} \tag{29}$$

Eqs 28 and 29 set the basis for κTDC. In practical implementation, κTDC can be computed by first order difference of dot product of gradients and velocity vector

$$\sigma_{JI}^{\kappa}\left(t+\Delta t\right) \approx \frac{1}{2}\left[\frac{\Delta \dot{V}_{JI}\left(t+\Delta t\right)-\Delta \dot{V}_{JI}\left(t\right)}{\Delta t}\frac{1}{V_J - V_I}\right]^{1/2} \quad \text{for } J > I \tag{30}$$

where

$$\Delta \dot{V}_{JI}\left(t\right) = \frac{\partial V_J\left(t\right)}{\partial \mathbf{R}} \bullet \dot{\mathbf{R}}\left(t\right) - \frac{\partial V_I\left(t\right)}{\partial \mathbf{R}} \bullet \dot{\mathbf{R}}\left(t\right) \tag{31}$$

And because at the stage of evaluation $\sigma_{JI}^{\kappa}\left(t+\Delta t\right)$, $\dot{\mathbf{R}}\left(t+\Delta t\right)$ is not known. Therefore, it is approximated by forward propagation,

$$\dot{\mathbf{R}}\left(t+\Delta t\right) \approx \dot{\mathbf{R}}\left(t\right) + \mathbf{a}\left(t\right)\Delta t \tag{32}$$

where $\mathbf{a}\left(t\right)$ is acceleration vector at time *t*. Or, κTDC can be computed by second order difference of energies,

$$\sigma_{IJ}^{\kappa} = \frac{1}{2}\left[\frac{d\left(\frac{d\left(V_I - V_J\right)}{dt}\right)}{dt}\frac{1}{V_I - V_J}\right]^{1/2} \tag{33}$$

and

$$\frac{d\left(V_I - V_J\right)}{dt} \approx \frac{\Delta V_{IJ}\left(t+\Delta t\right) - \Delta V_{IJ}\left(t\right)}{\Delta t} \tag{34}$$

where

$$\Delta V_{IJ}\left(t\right) \equiv V_I\left(t\right) - V_J\left(t\right) \tag{35}$$

Therefore, for the third time step,

$$\frac{d^2\left(V_I - V_J\right)}{dt^2} \approx \frac{1}{\Delta t^2}\left[\Delta V_{IJ}\left(t\right) - 2\Delta V_{IJ}\left(t-\Delta t\right) + \Delta V_{IJ}\left(t-2\Delta t\right)\right] \tag{36}$$

and start from fourth time step,

$$\frac{d^2\left(V_I - V_J\right)}{dt^2} \approx \frac{1}{\Delta t^2}\left[2\Delta V_{IJ}(t) - 5\Delta V_{IJ}(t-\Delta t) + 4\Delta V_{IJ}(t-2\Delta t) - \Delta V_{IJ}(t-3\Delta t)\right] \quad (37)$$

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

[13] 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).

[14] Y. Shu, L. Zhang, X. Chen, S. Sun, Y. Huang, 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. XX, XXXX-XXXX (2022).

Appendix A. Complete license for *SHARC* – version 2.1

Terms of Use

If you want to download *SHARC,* you must accept the following Terms of Use.

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. Input Examples

B.1. Example input for tCSDM

```
printlevel   0

geomfile     "geom"
veloc        external
velocfile    "veloc"

nstates      3 0 0
actstates    3 0 0
state        2 diag
coeff        auto
rngseed      5895967

ezero        -78.054411630961
tmax         250.0
integrator   fvv
stepsize     0.1
nsubsteps    200

method       scp
surf         diagonal
coupling     overlap
neom         gdiff
nogradcorrect
decoherence_scheme dom
```

B.2. Example input for κCSDM

```
printlevel   0

geomfile     "geom"
veloc        external
velocfile    "veloc"

nstates      3 0 0
actstates    3 0 0
state        2 diag
coeff        auto
rngseed      5895967

ezero        -78.054411630961
tmax         250.0
integrator   fvv
stepsize     0.1
nsubsteps    200

method       scp
surf         diagonal
coupling     ktdc
neom         gdiff
nogradcorrect
decoherence_scheme dom
```

B.3. Example input for κTSH-EDC

```
printlevel   0
geomfile     "geom"
veloc        external
velocfile    "veloc"

nstates      3 0 0
actstates    3 0 0
state        2 diag
coeff        auto
rngseed      5895967

ezero        -78.054411630961
tmax         250.0
integrator   fvv
stepsize     0.1
nsubsteps    200

method       tsh
surf         diagonal
coupling     ktdc
ekincorrect  parallel_diff
decoherence_scheme edc
nogradcorrect

grad_sele
```