April 14, 2023

INTRODUCTION TO *SHARC-MN 2.0*

Yinan Shu
University of Minnesota

Linyao Zhang
Harbin Institute of Technology and University of Minnesota

and Donald G. Truhlar
University of Minnesota

SHARC-MN is an extended version of *SHARC*. It includes all features of *SHARC 2.1* plus enhancements. Most of these enhancements are also in *SHARC 3.0*. 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, $\mathbf{R}$ 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 $\mathbf{R}$. The electronic Hamiltonian in a diabatic representation is called the diabatic potential energy matrix (DPEM, $\mathbf{U}$); 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 1.2*, 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 2.1* 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.
- Both *SHARC-MN 2.0* and *SHARC 3.0* contain an interface to multi-state PDFT methods, which is in the interface script SHARC_MOLCAS.py