NEXMD: A software package for non-adiabatic and excited-state molecular dynamics

Manual by Dmitri Svetlov and Josiah Bjorgaard
Los Alamos National Laboratory

Last updated Oct. 23, 2019

CONTENTS

1

INTRODUCTION

This is a manual for the NEXMD package developed at Los Alamos National Laboratory under the direction of Sergei Tretiak. It simulates the photoinduced adiabatic and non-adiabatic ground- and excited-state molecular dynamics of organic chromophores. It uses the CEO (collective electronic oscillator) package with a variety of semiempirical methods from the SQM package. Tully's fewest-switches surface hopping approach to quantum transitions is employed, with instantaneous decoherence and a Min-Cost algorithm for the detection of trivial unavoided crossings. Several TD-SCF (time-dependent self-consistent field) QM/continuum models are available for including the effects of a solvent.

2

AUTHORS

J.A. Bjorgaard, K. A. Velizhanin, S. Falberti, S. Tretiak

3

TERMS OF USE

GPL

4

INSTALLATION

The distribution of NEXMD contains all code necessary to utilize the entire package (provided that effective compilers for Fortran 90/95 and ANSI C are available). Makefiles are provided for the benefit of those computing in a Unix-based environment; these have been used successfully by our group to compile the software on our own HPC clusters. However, we cannot guarantee that any makefile will successfully compile NEXMD on a given infrastructure with no changes needed.

To compile NEXMD, change into the top-level directory of the package and modify the provided Makefile to give the desired compilers (default gfortran or gcc) and provide compiled BLAS and LAPACK libraries. Then run

```
"make ic" or "make gnu" depending on the compiler available
```

Part I

METHODOLOGY AND PREPARATION OF INPUT FILES

This part is structured according to the general input file used by the entire package. For each section of input or group of closely related parameters, it briefly discusses the relevant underlying methodology and physics (assuming only a first course in quantum mechanics) and then describes how to configure the input for the system of interest. A link is included at the start of each section on methodology allowing readers who are familiar with it to skip ahead. Our hope is that this guide can thus effectively serve both mature researchers who are merely new to our software and beginning students with little to no background in the field of non-adiabatic molecular dynamics. References to the texts used and to additional resources are given in the bibliography.

5

ELECTRONIC / MOLECULAR STRUCTURE

Predicting the dynamical evolution of a chemical system requires first knowing the various states in which it can exist - here, these are its ground and excited electronic states. Ascertaining these is a task for computational quantum chemistry and one that cannot be completed with perfect accuracy for all but the simplest of molecular systems. Every practical methodology must therefore make some trade-off between computational tractability and chemical correctness. A broad discussion of the available techniques is beyond the scope of this manual; the coverage will be limited to Hartree-Fock theory, its extensions to excited-state calculations, and the inclusion of solvent models. (If you have not encountered the Born-Oppenheimer approximation or the electronic Hamiltonian, skip ahead briefly to 6 to familiarize yourself with the notation and rationale.)

5.1 HARTREE-FOCK THEORY: GROUND STATE CALCULATIONS

Skip ahead to 5.1.1

Recall from basic quantum mechanics that the hydrogen atom is the simplest system containing both a nucleus and an electron for which the Schrödinger equation can be solved exactly. Now consider a multi-electron atom. If the electrons did not interact with one another, the electronic repulsion energy $V_{ee}(\mathbf{r})$ would be zero, the Hamiltonian would be separable, and the total electronic wavefunction would be a simple, and symmetric, product of the wavefunctions of the individual electrons:

$$\Psi_{HF}(\mathbf{r}) = \psi(\mathbf{r}_1)\psi(\mathbf{r}_2)\cdots\psi(\mathbf{r}_n) \tag{1}$$

However, since electrons are fermions, electronic wavefunctions must be antisymmetric with respect to the exchange of any two electrons (and thus their space-spin coordinates). The easiest way to remedy this is to (1) replace the *spatial* wavefunctions/orbitals ψ above with *spin* orbitals χ, by multiplying them with either the α or β spin function; (2) changing thus from spatial coordinates $\mathbf{r}$ to spin coordinates $\mathbf{x}$ and (2) defining the total wavefunction as a normalized determinant of those spin functions:

$$\Psi_{HF} = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(\mathbf{x}_1) & \chi_2(\mathbf{x}_1) & \dots & \chi_n(\mathbf{x}_1) \\ \chi_1(\mathbf{x}_2) & \chi_2(\mathbf{x}_2) & \dots & \chi_n(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(\mathbf{x}_n) & \chi_2(\mathbf{x}_n) & \dots & \chi_n(\mathbf{x}_n) \end{vmatrix} \tag{2}$$

The exchange of any two electrons will change the sign of the total wavefunction, since exchanging any two rows or columns of a matrix negates the value of its determinant. This formulation also directly entails that electrons are indistinguishable and obey the Pauli exclusion principle.

To evaluate these eigenfunctions, we will rewrite the Hamiltonian by defining the *one-electron operator*

$$h(i) = -\frac{1}{2}\nabla_i^2 - \sum_A \frac{Z_A}{r_{iA}} \tag{3}$$

and the *two-electron operator*

$$v(i,j) = \frac{1}{r_{ij}}. \tag{4}$$

The electronic Hamiltonian then becomes

$$H_e = \sum_i h(i) + \sum_{i<j} v(i,j) + V_{NN}, \tag{5}$$

and since we are working in an adiabatic basis and thus fixing the set of nuclear coordinates $\{\mathbf{R}\}$, V_{NN} will be a constant that we can ignore without any change to the eigenfunctions. We can then employ a variational approach to find the electronic energy. In general, we have

$$E_e = \langle \Psi | H_e | \Psi \rangle \,. \tag{6}$$

It can be shown that the *Hartree-Fock energy* is a function of integrals of the one- and two-electron operators introduced above:

$$E_{HF} = \sum_i \langle i|h|i\rangle + \frac{1}{2}\sum_{ij} [ii|jj] - [ij|ji] \tag{7}$$

$$\langle i|h|j\rangle = \int d\mathbf{x}_1 \chi_i^*(\mathbf{x}_1) h(\mathbf{r}_1) \chi_j(\mathbf{x}_1) \tag{8}$$

$$[ij|kl] = \int d\mathbf{x}_1 d\mathbf{x}_2 \chi_i^*(\mathbf{x}_1)\chi_j(\mathbf{x}_1)\frac{1}{r_{12}}\chi_k^*(\mathbf{x}_2)\chi_l(\mathbf{x}_2) \tag{9}$$

The variational principle tells us that the solution to this equation (i.e., its eigenvalue) is an upper bound on the actual ground-state energy of the system; in particular, we have obviously neglected the energy resulting from electron-electron repulsion, so we cannot obtain the true minimum energy.

Computationally, these operators are represented in matrix form and the eigenvalue equations that result are solved via matrix diagonalization. However, the matrices depend on their own solutions (which will be linear combinations of the orbital bases we choose). Therefore, the equation can only be solved iteratively: starting from a reasonable initial guess of orbitals, the electron density and total energy (including nuclear repulsions) are computed. These are then used as updated guesses, and new densities and energies are computed. This process is repeated until a set number of iterations have been completed or until the energies and densities *converge* (i.e., the change in each from one iteration to the next is below a target value). Because of these aspects, this is known as a *self-consistent field* method.

In addition to purely *ab initio* methods such as these, there are also *semiempirical* methods that employ system-specific parameters derived from experimental data to formulate approximations of the true Hamiltonian that sacrifice chemical accuracy but are easier to solve. The earliest and perhaps most influential is the Hückel model for conjugated π systems, which uses a "tight-binding" approach where each carbon atom has a single p orbital, overlap integrals between non-identical atoms are taken to be zero, and only two parameters, the Coulomb integral and resonance/bond integral, are required. These parameters have been established by experiment, and the model has yielded qualitatively - and sometimes quantitatively - useful insights.

For example, `NEXMD` uses the method of *collective* (or *coupled*) *electronic oscillators* developed by Mukamel and colleagues in the calculation of electronic structures. This combines the Pariser-Parr-Pople (PPP) tight-binding Hamiltonian and the single-electron density matrix in the ground state with time-dependent Hartree-Fock theory to arrive at a set of electronic "normal modes" that are in many ways analogous to their nuclear counterparts. The advantage of this approach is that in many applications - such as the nonlinear optical responses of conjugated polyenes - only a few such oscillators tend to be important. Thus we are still able to capture many important phenomena and trends, but with significant computational savings.

The latest version of `NEXMD` interfaces with the `sqm` package to provide a variety of semiempirical models, listed in the table of input parameters below.

Finally, the package provides three methods for dealing with the computationally intensive problem of SCF iteration at each time step of the molecular dynamics simulation. In order to ensure adiabatic evolution of the trajectory, forces must always be calculated at the electronic ground state. Earlier approaches "propagated" combinations of the ground-state configurations from previous time steps as initial guesses to subsequent SCF calculations. While this reduces the number of iterations necessary, it also introduces systematic energy drifts, because electronic evolution is irreversible until SCF convergence is attained.

5.1.1 *Configuring the Input*

qm_theory	'MNDOD'	The approach to computing the HF integrals; consult additional references for more on each method and selecting the one that is optimal for your simulation. NOTE: Currently only the AM1 has been thoroughly tested in this package
	'AM1'	
	'AM1D'	
	'PM3'	
	'PM3CARB1'	
	'PM3ZNB'	
	'PM3MAIS'	
	'PM6'	
	'PDDGPM3'	
	'PDDGPM3_08'	
	'PDDGMNDO'	
	'RM1'	
	'DFTB'	
scfconv	X.Yd-Z	The convergence criterion for the ground-state SCF calculation, equal to $X.Y \cdot 10^{-Z}$ in units of electron-volts
verbosity	INTEGER	An integer between 0 (least output) and 5 (most) [Increasing the verbosity slows down the simulation and writes more data to disk. Consider a few short test calculations to determine the minimum level of verbosity needed for your purpose.]
printcharges	1	Prints Mulliken charges for the ground-state configuration
	0	Does not print Mulliken charges
printdipole	2	Prints unrelaxed/relaxed transitions plus total molecular
	1	Prints unrelaxed transitions plus total molecular
	0	Prints unrelaxed transitions

`density_predict`	0	Standard HF-SCF where required
`itrmax`	INTEGER	Maximum number of SCF iterations for the ground-state calculations [A negative integer signifies that convergence will be ignored.]

5.2 EXCITED-STATE CALCULATIONS

Skip ahead to 5.2.1

As Hartree-Fock theory is the simplest *ab initio* approach to ground-state electronic structure calculations, its extensions are the simplest ones to solving the structures of excited states. The package implements the two most basic: configuration interaction singles (CIS) and time-dependent Hartree-Fock theory (TD-HF), which is also known as the random-phase approximation (RPA).

CIS uses the ground-state HF wavefunction as a starting point for creating singly-excited determinants ("singles"); this is done by exchanging the occupancies of an unoccupied spin orbital with an occupied one. Excited-state wavefunctions are then expressed as linear combinations of these determinants. No additional information is needed, since this can be achieved using the Fock matrix elements and two-electron integrals used in solving for the ground-state wavefunction. A combinatorial challenge is introduced because multiple spin configurations will necessarily be available to the excited molecule. Therefore, if we are only interested in, for instance, triplet excited states, we can "spin-adapt" CIS to utilize only spatial rather than spin orbitals, which leads to a Hamiltonian matrix that is four times smaller and eight times faster to diagonalize.

RPA is similar to CIS in that it also uses only singles, and indeed, the CIS matrix itself, but it involves additional excitation and de-excitation operators. In a naïve implementation, this yields an eigenvalue problem with twice the dimensions (and thus four times the storage size) of an equivalent CIS; however, a creative rearrangement reduces the problem to one of identical size.

Each of these approaches faces the challenge of very large matrices that must be diagonalized; specifically, ones with dimensions at least as large as the product of the number of occupied orbitals and the number of unoccupied ones. For all but the smallest molecules, such matrices will not fit in core memory, and thus algorithms like the one by Ernest Davidson, or its extension by Bowen Liu, must be used. The algorithm works as follows:

1. Select reasonable "guess vectors", at least one for each desired eigenvalue (root).

2. Construct a "subspace" Hamiltonian within the space of guess vectors.

3. Build a set of "correction vectors," which depend on the diagonal terms of the Hamiltonian and on "residual vectors," which are defined such that they are zero if their corresponding guess vectors are equal to the true eigenvectors. At this point check the norms of the residuals and the changes in the corresponding eigenvalues for convergence.

4. Normalize each correction vector and orthogonalize it against the set of guess vectors. Check its norm against a minimum threshold; discard it if it does not reach that threshold.

   Return to Step 2.

5.2.1 *Configuring the Input*

`exst_method`	1	Configuration interaction singles (CIS)
	2	Random-phase approximation (RPA/TD-HF)
`dav_guess`	1	Previous configurations will be used as starting guesses
	0	Diagonalization will start from scratch every time
`ftol0`	X.Yd-Z	Acceptance energy tolerance $(e_{min} - e_{old})$ for diagonalization, equal to X.Y $\cdot 10^{-Z}$ in units of electron-volts
`dav_maxcyc`	INTEGER	Maximum number of Davidson diagonalization cycles for the excited-state calculations [A negative integer signifies that a fixed number (its arithmetic inverse) will be performed.]
`calcxdens`	`.true.`	Calculate cross-density dipole moments/oscillator strengths
	`.false.`	Do not calculate cross-density dipole moments/oscillator strengths

5.3 GEOMETRY OPTIMIZATION

Skip ahead to 5.3.1

Geometry optimization will attempt to alter the arrangement of the nuclei in order to minimize the total energy. Depending on your simulation and the stage within it, this optimal geometry may be desired, or it may not. For example, geometry optimization should not (be forced to) occur during a state transition.

5.3.1 *Configuring the Input*

`maxcyc`	INTEGER	Number of cycles of geometry optimization that will be performed; if `0` is used, the system will be "frozen" in the provided coordinates for electronic-structure calculations.
`ntpr`	INTEGER	Results will be printed after every `ntpr` cycle(s).
`grms_tol`	DOUBLE	Tolerance of energy derivatives, in units of electron-volts per angstrom, to be used for optimization

5.4 SOLVENT MODELS

Skip ahead to 5.4.1

The presence of a solvent can have significant effects on photodynamics. It is not computationally feasible to treat both a solute and a sufficient environment of solvent molecules quantum mechanically; therefore, the solute is treated quantum mechanically and the solvent is treated either molecular mechanically (a QM/MM approach) or as a polarizable continuum (QM/continuum) forming cavities in which solutes are enclosed. The latest version of `NEXMD` implements a QM/continuum treatment within the linear response (LR), vertical excitation (VE), and state-specific (SS) models. For more details, refer to our publications on "Solvent Effects in TD-SCF."

5.4.1 *Configuring the Input*

`ceps`	INTEGER	Dielectric constant
`cosmo_scf_ftol`	X.Yd-Z	SCF tolerance for SS model, given as $X.Y \cdot 10^{-Z}$ in units of electron-volts
`linmixparam`	INTEGER	Linear mixing parameter for state-specific solvent iterations
`doZ`	`.true.`	Calculate relaxed electron density in COSMO of SS model
	`.false.`	Calculate unrelaxed electron density in COSMO of SS model
`EF`	1	Include electric field for ground and excited states
	0	Include electric field only for excited states
`Ex`	INTEGER	x-component of electric field vector
`Ey`	INTEGER	y-component of electric field vector
`Ez`	INTEGER	z-component of electric field vector
`onsager_radius`	INTEGER	Onsager cavity radius
`solvent_model`	3	SS model
	2	VE model
	1	LR model
	0	No solvent (gas phase)
`potential_type`	1	COSMO
	2	Onsager

6

MOLECULAR DYNAMICS

Skip ahead to 6.1

Molecular dynamics (MD) simulations are used to predict how a given collection of atoms and molecules will behave under certain conditions (e.g. temperatures, pressures, electric fields). Classical physics holds that Newton's three laws of motion completely describe all mechanics. These involve only the mass, position, velocity, and acceleration of each particle and all interactive forces among the particles, which also affect their accelerations. A classical MD simulation therefore reduces to solving a system of differential equations, each one second-order in time, given a set of initial conditions. Using this solution, we can calculate any property of the system as an explicit function of time and its average value over a period of time.

Quantum MD rests on different postulates; namely, that all material systems are wavelike in nature, that the wavefunctions characterizing them are all solutions to the Schrödinger equation, and that all properties of a system can be calculated if its wavefunction is exactly known. A quantum MD simulation thus involves solving a single differential equation (the time-dependent Schrödinger equation), likewise second-order in time, to obtain the wavefunction describing the entire system at each point in time. The fundamental challenge in computational chemistry is that no analytical solutions for molecular systems exist, and numerical solution methods require trade-offs between accuracy and tractability.

A common first step in such simulations, and one that often yields tremendous reductions in computational cost, is the separation of electronic motions from nuclear ones. Consider the Hamiltonian operator for a general system described by electronic coordinates **r** and nuclear coordinates **R**:

$$H(\mathbf{r},\mathbf{R}) = T_N(\mathbf{R}) + \underbrace{T_e(\mathbf{r}) + V_{ee}(\mathbf{r})}_{H_e(\mathbf{r})} + V_{eN}(\mathbf{r},\mathbf{R}) \tag{10}$$

Here, T_e and T_N are the electronic and nuclear kinetic-energy operators, respectively, V_{eN} is the potential-energy operator including all electron-nuclear and nuclear-nuclear interactions, and H_e is the "electronic Hamiltonian" composed of purely electronic contributions to the energy: the kinetic energy of all electrons and the potential energy due to all electron-electron repulsions.

Now suppose that all nuclei were fixed in space. We can then exclude T_N from the Hamiltonian and write the Schrodinger equation as

$$[H_e(\mathbf{r}) + V_{eN}(\mathbf{r};\mathbf{R})]\,\phi_\alpha(\mathbf{r};\mathbf{R}) = \epsilon_\alpha(\mathbf{R})\phi_\alpha(\mathbf{r};\mathbf{R}) \tag{11}$$

where the eigenfunctions $\phi_\alpha(\mathbf{r};\mathbf{R})$ depend parametrically on the fixed **R**, have eigenvalues $\epsilon_\alpha(\mathbf{R})$, and are denoted as the *adiabatic basis functions*.

Allowing all possible values of **R**, the set of (orthonormal) adiabatic basis functions forms a complete basis, and we can write the total time-dependent molecular wavefunction as

$$\Psi(\mathbf{r},\mathbf{R},t) = \sum_\alpha c_\alpha(t)\phi_\alpha(\mathbf{r};\mathbf{R}(t)) \tag{12}$$

where $c_\alpha(t)$ are the time-dependent expansion coefficients. The equation of motion for these coefficients then becomes

$$i\hbar\dot{c}_\alpha(t) = c_\alpha(t)E_\alpha(\mathbf{R}) - i\hbar\sum_\beta c_\beta(t)\dot{\mathbf{R}}\cdot\mathbf{d}_{\alpha\beta}. \tag{13}$$

Here, $\mathbf{d}_{\alpha\beta} = \left\langle \phi_\alpha(\mathbf{r};R)\,\middle|\nabla_R\phi_\beta(\mathbf{r};R)\right\rangle$ is the NA coupling vector (NACV) and $\dot{\mathbf{R}}\cdot\mathbf{d}_{\alpha\beta} = \left\langle\phi_\alpha(\mathbf{r};R)\,\middle|\partial\phi_\beta(\mathbf{r};R)/\partial t\right\rangle$ is the NA coupling scalar term (NACT).

6.1 GENERAL PARAMETERS

Skip ahead to 6.1.1 or return to 6

These first parameters in the molecular-dynamics section of the input determine the most fundamental aspects of the simulation: whether state transitions will be allowed (non-adiabatic, or non-Born-Oppenheimer, dynamics) or not (adiabatic; Born-Oppenheimer), the initial state of the system, and which states will be accessible to the system over the course of the simulation. Also, since state transitions in non-adiabatic dynamics are made stochastically, an integer seed is provided for the pseudorandom number generator used.

6.1.1 *Configuring the Input*

`natoms`	INTEGER	This is the maximum number of atoms that can be used in the simulation. Setting to the exact number of atoms used in the simulation gives the best memory performance.
`rnd_seed`	INTEGER	An integer seed for the random number generator used [this value is irrelevant for single-point calculations].
`bo_dynamics_flag`	0	non-Born-Oppenheimer/non-adiabatic dynamics [state transitions allowed].
	1	Born-Oppenheimer/adiabatic dynamics [state transitions forbidden].
`exc_state_init`	INTEGER	Initial state: 0 denotes the ground state, 1 the first excited state, etc.
`n_exc_states_propagate`	INTEGER	Number of excited states to include in the calculations [This number is *critical* for all simulations except for adiabatic dynamics. Excited states above this number do not exist as far as the program is concerned.]

6.2 DYNAMICS PARAMETERS

Skip ahead to 6.2.1

The system evolves over a window time discretized into both classical and quantum time points. These are separated by constant *time steps* Δt (classical) and δt (quantum), with $\delta t < \Delta t$ because electronic dynamics are much faster than those of nuclei; large quantum steps are not sufficient for resolving NACT peaks that are strongly localized, leading to underestimated state transition probabilities.

At each intermediate δt the energies of all excited states and NACTs between all pairs thereof are calculated. These are also determined at each classical time step, but subsequently, the electronic state coefficients are updated according to the equation of motion above. For non-adiabatic dynamics, switching probabilities $g_{\alpha\beta}$ are also evaluated, as a summation over all $N_q = \frac{\Delta t}{\delta t}$ quantum steps made since the previous classical time step:

$$g_{\alpha\beta} = \frac{\sum_{j=1} N_q b_{\beta\alpha}(j)\delta t}{a_{\alpha\alpha}} \tag{14}$$

These terms a and b are explained in the following subsection on non-adiabatic parameters.

6.2.1 *Configuring the Input*

`time_init`	X.dY	The initial simulation time, given as $X \cdot 10^{Y}$ femtoseconds; this can be set to a non-zero time to restart a previous simulation at the corresponding time step.
`time_step`	DOUBLE	Time step, in units of femtoseconds
`n_class_steps`	INTEGER	Number of classical simulation steps, leading to a total simulation time of `n_class_steps` × `time_step` femtoseconds.
`n_quant_steps`	INTEGER	Number of quantum simulation steps per classical step
`moldyn_deriv_flag`	2	Derivatives calculated by finite-difference
	1	Derivatives calculated analytically
	0	Derivatives are set to zero
`num_deriv_step`	X.Yd-Z	Displacement used in calculating numerical derivatives, in units of angstroms
`rk_tolerance`	X.d-Y	Tolerance for the Runge-Kutta propagator.

6.3 non-adiabatic and decoherence parameters

Skip ahead to 6.3.1

In non-adiabatic dynamics, quantum transitions are permissible. `NEXMD` uses John Tully's *fewest-switches surface hopping* (FSSH) algorithm, whose validity has been demonstrated across a variety of systems. In this approach, a system evolves along the potential energy surface (PES) of a single electronic state, with nuclei treated classically and electrons quantum-mechanically; this continues until a "hop" (quantum transition) between states occurs. A single trajectory can illustrate mechanistic aspects of the underlying photophysics; a statistical ensemble (or "swarm") of many independent trajectories can yield observables such as branching ratios, non-radiative relaxation rates, quantum yields, and so on.

The probability of such a hop from a current state α to another state during a given classical time step Δt is given by

$$\dot{a}_{\alpha\alpha}(t) = \sum_{\beta \neq \alpha} b_{\alpha\beta} \tag{15}$$

where

$$b_{\alpha\beta}(t) = -2\mathrm{Re}\left(a^*_{\alpha\beta}\dot{\mathbf{R}} \cdot \mathbf{d}_{\alpha\beta}\right). \tag{16}$$

Hops are accepted or rejected stochastically; if the nuclear kinetic energy is insufficient to allow a particular hop to a higher state, that hop is classically forbidden and rejected. If the hop is accepted, nuclei evolve on the new PES, and conservation of energy is achieved by rescaling nuclear velocities in the direction of the NACV and by the appropriate amounts.

The FSSH method has the drawback that because it propagates quantum coefficients coherently on each trajectory, electronic coherence cannot be dissipated in any single-trajectory simulation. As a result, the classical and quantum "occupancies" of a given state will diverge from one another over the course of the simulation. To resolve this our code utilizes the *instantaneous decoherence* approach, in which the quantum population of the current state is reset to unity at each classical time step (where hops are attempted). The rationale is that wavepackets on different surfaces should instantaneously separate in phase space and evolve independently; no single trajectory should thus have any quantum amplitude in two states at once. Our code can also reinitiialize the qunatum populations after frustrated, rejected hops to add additional decoherence.

One final point concerns the problem of *unavoided crossings*. These events can significantly effect photochemistry and can occur readily in conjugated molecules, even between noninteracting states localized in spatially separated portions of the molecule. The latter are called *trivial* unavoided crossings and present a particular numerical challenge, since their NACTs are analogous to strong peaks appearing at the exact intersection and vanishing everywhere else. The numerical propagation of nuclear motion in finite time steps can cause such crossings to go undetected, leading to artifacts in state populations.

NEXMD uses a Min-Cost algorithm (discussed in details in one of our Relevant Publications) to ensure that such trivial crossings are detected and handled properly. This involves the occasional reduction of the quantum time step within individual classical time steps, in order to evaluate more couplings between states that could be relevant to a possible hop, and thus in essence "resolve" the NACT peak. A parameter is used to indicate how many times δt should be reduced in each such instance.

6.3.1 *Configuring the Input*

`decoher_type`	2 1 0	Quantum coefficients are reinitialized after successful and frustrated hops Coefficients are reinitialized after a successful hop Coefficients are never reinitialized
`quant_step_reduction_factor`	X.YdZ	Quantum time step reduction factor for detecting trivial avoided crossings, given as $1/(\text{X.Y} \cdot 10^{\text{Z}})$
`dotrivial`	1 0	Track trivial unavoided crossings Do not track trivial unavoided crossings

6.4 thermostat parameters

Skip ahead to 6.4.1

Classical molecular dynamics simulations, integrating Newton's equations of motion, obey the law of energy conservation. Given a fixed system volume and set of particles, these simulations thus reproduce the average thermodynamic properties of a *microcanonical*, or NVE, ensemble (fixed particle number N, volume V, and total energy E).

However, in many situations, we are interested in a system equilibrated such that some thermodynamic variable (e.g. temperature) is held constant at a given value. This can be accomplished by creating an NVE

supersystem where the surroundings of the (sub)system of interest serve as a bath. NEXMD provides one method for simulating dynamics under a *thermostat*, i.e. an NVT, or *canonical* ensemble.

The Langevin thermostat combines Langevin dynamics with velocity Verlet integration. A stochastic force **A** depends on the bath temperature and a friction coefficient γ, which also impacts the nuclear velocities in a direct, linear fashion. The excited-state gradients that determine nuclear motion are determined analytically. Due to the presence of the friction coefficient, various fluid-mechanical properties of the system's surroundings can be directly introduced, particularly in simulations where a solute molecule can be treated as the system and a collection of (relatively small) solvent molecules as the supersystem/surroundings.

6.4.1 *Configuring the Input*

`therm_type`	1 0	A Langevin thermostat is used No thermostat is used (a constant-energy simulation is performed)
`therm_temperature`	XYZ.dN	Thermostat temperature, as $XYZ \cdot 10^N$ K [This is only relevant for NVT ensembles.]
`therm_friction`	XY.dZ	Thermostat friction coefficient, as $XY \cdot 10^Z$ in units of inverse picoseconds [This is only relevant for Langevin thermostats.]

6.5 OUTPUT AND LOG PARAMETERS

As with the electronic structure calculations, the software allows for a variety of output levels and frequencies in the molecular dynamics simulations, and care should be chosen in setting these values so that simulations are not slowed down needlessly by printing excessive output.

6.5.1 *Configuring the Input*

`verbosity`	INTEGER	An integer between 0 (indicating minimal) and 3 (highest) verbosity of output.
`out_count_init`	INTEGER	The initial count for output files [Note: Analogous files in the working directory bearing corresponding count numbers will be overwritten.]
`out_data_steps`	INTEGER	Data will be written to files after every `out_data_steps` time step(s).
`out_coords_steps`	INTEGER	Coordinate files (which are sufficient to restart a simulation) will be written after every `out_coords_steps` time step(s).
`out_data_cube`	1	View-files to generate cubes will be written
	0	View-files will not be written

6.6 COORDINATES, VELOCITIES, AND STATE COEFFICIENTS

Coordinates are specified in a standard `XYZ` format as

Z_1	x_1	y_1	z_1
Z_2	x_2	y_2	z_2
Z_3	x_3	y_3	z_3
...	...	...	...

where Z_i denotes the atomic number and x_i, y_i, z_i the coordinates (in angstroms) of atom i. In preparing initial conditions for a single-point calculation or ground-state adiabatic trajectory, these coordinates can be obtained from a molecular builder, such as Avogadro (`http://avogadro.org/`) or IQmol (`http://iqmol.org/`), applying an optimization procedure, if needed or desired, to obtain a reasonable initial geometry.

Velocities are specified analogously, in units of angstroms per atomic time units, for each Cartesian coordinate of each atom:

$\dot{x}_1$	$\dot{y}_1$	$\dot{z}_1$
$\dot{x}_2$	$\dot{y}_2$	$\dot{z}_2$
$\dot{x}_3$	$\dot{y}_3$	$\dot{z}_3$
...	...	...

In the initial conditions, these may be fully specified; if they are not, a value of zero will be used for each component of each velocity that is omitted.

The state coefficients indicate the population p of each excited state at the start of the simulation. Populations should be provided for each excited state included in the calculations; if excess lines are provided, they will be ignored.

p_{S_1}	0.0
p_{S_2}	0.0
p_{S_3}	0.0
...	...

A

SAMPLE WORKFLOW AND HELPER SCRIPTS

Each input file for NEXMD specifies only one single-point calculation and/or molecular dynamics trajectory. This offers several advantages, such as the fact that a swarm of independent trajectories can be handled by a separate call to NEXMD, executed on a separate processor, for each individual trajectory. However, performing such simulations efficiently and in an organized fashion requires the construction of a workflow that involves helper scripts to prepare input for NEXMD and to process and present its outputs. In particular, it is beneficial to make use of *header files* that contain only the simulation parameters and omit all coordinates, velocities, and coefficients. Scripts can then be used to extract the coordinates and velocities from the output of previous runs and write a matrix of coefficients to select the initial state of the system. To aid with calculations we provided the getexcited.py script. This script, written in python 2.7, is capable of aiding NEXMD calculations in many ways including, but not limited to, preparing a swarm of trajectories for non-adiabatic simulation, preparing any number of single-point calculations, preparing restart files for unfinished non-adiabatic trajectories, and performing a geometry analysis of completed dynamics calculations. For a full list of the getexcited.py script's functions and an in-depth overview of how to use to the script see the documentation.pdf file provided inside the NEXMD package.

B

REFERENCES AND RESOURCES

B.1 RELEVANT PUBLICATIONS

Start with our review of `NEXMD`: `http://dx.doi.org/10.1021/ar400263p`
These publications have used `NEXMD` to study various phenomena:
State-specific vibrations in dendrimers: `http://dx.doi.org/10.1021/jp301293e`
Photoinduced dynamics of conjugated molecules: `http://dx.doi.org/10.1021/jp109522g`
Excited-state vibrational normal modes: `http://dx.doi.org/10.1021/jp503350k`
Energy transfer in weakly coupled dimers: `http://dx.doi.org/10.1021/jp510557f`
For more on trivial unavoided crossings: `http://dx.doi.org/10.1016/j.cplett.2013.10.052` and `http://dx.doi.org/10.1063/1.4732536`
For more on parameters and numerical testing of convergence: `http://dx.doi.org/10.1063/1.3680565`
For more on electronic decoherence as implemented in `NEXMD`: `http://dx.doi.org/10.1063/1.4809568`

B.2 ELECTRONIC STRUCTURE

B.2.1 *General Subjects*

The text by Atilla Szabo and Neil Ostlund (`http://store.doverpublications.com/0486691861.html`) introduces the theoretical underpinnings of mod-

ern *ab initio* electronic structure calculations, with an especially thorough treatment of Hartree-Fock (HF) theory.
David Cook's handbook (http://store.doverpublications.com/0486443078.html) provides additional instruction on practical implementations of the standard theories.
An article by Anders Odell and others (http://dx.doi.org/10.1063/1.3268338) describes time-reversible Born-Oppenheimer molecular dynamics (TR-BOMD) and its extended Lagrangian formulation (XL-BOMD).
Daniel Crawford's group has written a variety of tutorials in programming electronic structure calculations (http://sirius.chem.vt.edu/wiki/doku.php?id=crawdad:programming) that proceed step-by-step through the relevant physics; the discussions on excited-state HF methods here have followed theirs.
David Shirrell also provides a variety of helpful notes on quantum chemistry (http://vergil.chemistry.gatech.edu/notes/index.html) and programming resources (http://vergil.chemistry.gatech.edu/resources/programming/index.html). The discussion on ground-state HF here uses his notation.

B.2.2 *Davidson diagonalization*

Ernest Davidson's original article: http://dx.doi.org/10.1016/0021-9991(75)90065-0
Bowen Liu's original article is B. Liu, "The simultaneous expansion method for the iterative solution of several of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices," Technical Report LBL-8158, Lawrence Berkeley Laboratory, University of California, Berkeley, 1978.

B.2.3 *Solvent Effects in TD-SCF*

Our group's theoretical publications:

"Optical response calculations": http://dx.doi.org/10.1063/1.4905828

"Variational formulations and analytical gradients": http://dx.doi.org/10.1063/1.4927167

B.3 MOLECULAR DYNAMICS

The introductory text on classical molecular simulations by Daan Frenkel and Berend Smit (`http://store.elsevier.com/Understanding-Molecular-Simulation/Daan-Frenkel/isbn-9780122673511/`) provides a broad overview of the basic integration schemes and statistical ensembles and the relevant thermodynamics.
The original article by John Tully on fewest-switches surface hopping: `http://dx.doi.org/10.1063/1.459170`

B.3.1 *Decoherence*

The original article on Persico-Granucci decoherence: `http://dx.doi.org/10.1063/1.3489004`
The original article on Truhlar decoherence: `http://dx.doi.org/10.1063/1.1793991`

B.3.2 *Thermostats*

An overview of the Langevin, Berendsen, and several other thermostats is given by Yanxiang Zhao: `http://www.math.ucsd.edu/~y1zhao/ResearchNotes/ResearchNote007Thermostat.pdf`