Introduction

This manual will introduce the method of simulating nonadiabatic dynamics implemented within the *Vienna Ab Initio Simulation Package* (VASP). With this tool, the nonradiative electronic relaxation can be investigated. After an initial excitation, the system is allowed to transition from one electronic state to another. The probability of a state transition is determined by the strength of the *nonadiabatic* (NA) coupling between two states. Relaxation is complete when the system reaches the ground state. In real systems, the evolution of the nuclear coordinates is governed by the potential energy surface (PES) of the currently occupied electronic state. One of the main assumptions made here is that the ground state dynamics provides a good approximation for the excited state dynamics.

In this approach, a *ground state* molecular dynamics simulation is first performed. The NA couplings, oscillator strengths, and Kohn-Sham energies are then computed. Once the NA coupling terms are determined, the hopping probability at each time step can be calculated and the system can transition to a different electronic state according to the Tully's fewest switches surface hopping (FSSH) prescription.

The computations have been divided into three steps. The initial geometry optimization is performed in Step 0. Since geometry relaxation is performed at T = 0K, the system must be up heated from 0K to some finite temperature. Temperature "ramping" is performed in Step 1a. The initial ground state molecular dynamics trajectory is performed in Step 1b. During this step, the nuclear coordinates are obtained for a trajectory of hundreds of femtoseconds to picoseconds. In Step 2, the NA couplings, oscillator strengths, and Kohn-Sham energies are computed. The NA couplings drive the relaxation, and the oscillator strengths are used to determine the initial conditions. Step 3 utilizes the NA couplings computed in Step 2 to calculate the actual hopping probabilities according to the FSSH method. The following sections provide a detailed description of each step.

VASP

The home page for VASP is http://cms.mpi.univie.ac.at/vasp/. The best way to become familiar with VASP is to go through the provided workshop pages, http://cms.mpi.univie.ac.at/vasp-workshop/slides/documentation.htm, and to read through the manual, http://cms.mpi.univie.ac.at/vasp/vasp/vasp.html.

Step 0: Geometry Optimization

Running geometry optimizations

Here is the INCAR for the first-round geometry optimization (INCAR1) starting from a poor initial guess with no wavefunction.

```
IBRION = 3          # damped dynamics for poor initial guess
ISMEAR = 0          # Gaussian smearing of electronic population
SIGMA = 0.025       # electronic temperature, here 300K
NELMDL = -15        # number of non-self consistent electronic steps
NELMIN = 4          # minimum number of electronic iterations
ISIF = 2            # relax ions but maintain constant cell volume and shape
ISYM = 0            # symmetry not taken into account
NSW = 25            # number of ionic steps
PREC = Accurate     # precision (low, med, accurate, high)
LREAL = A           # VASP sets up wavefunction projections
SMASS = -3          # microcanonical ensemble
POTIM = 0.1         # scaling constant for forces
LCHARG = .TRUE.     # write CHGCAR file
LWAVE = .TRUE.      # write WAVECAR file
```

The parameters to note here are the IBRION, NSW, and POTIM. IBRION specifies what kind of calculation is being done. In this case, IBRION = 3 specifies a geometry optimization with a damped relaxation algorithm. NSW is the number of ionic steps and POTIM is the scaling constant for forces. A small POTIM helps to damp the dynamics and, combined with IBRION = 3, helps ensure that the cell stays intact (sometimes a cell can “explode” during a first-round geometry optimization if the forces are too large). In this case, the NELMDL tag specifies 15 electronic iterations before applying the density and corresponding force to the ions. For any calculation where there is no initial wavefunction (WAVECAR), choose a larger 'delay' for starting the charge density update. Without setting a delay VASP will probably not converge or at least the convergence speed is slowed down.

Once the first optimization calculation has finished, there will be many output files. The most important are OUTCAR and CONTCAR. The OUTCAR file contains all of the information about the system and the minimization. The CONTCAR file lists the new (optimized) coordinates and has the same format as POSCAR. Before continuing the optimization, it is a good idea to save some important files.

```
$ cp OUTCAR out.1
```

```
$ cp POSCAR pos.0
```

Now update POSCAR to contain the new (optimized) coordinates

```
$ cp CONTCAR POSCAR
```

Before starting the second-round geometry optimization, some parameters in INCAR need to be changed to reflect the updated system (INCAR2).

⇨ `IBRION = 2          # conjugate gradient algorithm`

```
    ISMEAR = 0          # Gaussian smearing of electronic population
    SIGMA = 0.025       # electronic temperature, here 300K
⟹  NELMDL = -2          # number of non-self consistent electronic steps
    NELMIN = 4          # minimum number of electronic iterations
    ISIF = 2            # relax ions but maintain constant cell volume and shape
    ISYM = 0            # symmetry not taken into account
⟹  NSW = 40             # number of ionic steps
    PREC = Accurate     # precision (low, med, accurate, high)
    LREAL = A           # VASP sets up wavefunction projections
    SMASS = -3          # microcanonical ensemble
⟹  POTIM = 0.5          # scaling constant for forces
    LCHARG = .TRUE.     # write CHGCAR file
    LWAVE = .TRUE.      # write WAVECAR file
```

The changes in the IBRION and POTIM tags reflect the fact that our starting geometry should be much better than it was before the initial optimization. IBRION = 2 is a more efficient algorithm and a larger POTIM will help the system converge faster allowing a larger NSW to be set. NELMDL = -2 is much smaller since there is now a WAVECAR and CHGCAR, which supply good guesses for wavefunctions and charge densities.

Repeat the optimization at least one more time, (remember to save OUTCAR and POSCAR files and copy CONTCAR to POSCAR) this time setting POTIM = 1.0 and NSW = 100 (INCAR3). If the geometry still does not converge, set NSW to a larger number and run again. It is important to get the POTIM step to 1.0 (no damping of forces) before moving on to Step 1a to ensure that the geometry is fully relaxed.

Step 1a: Temperature Ramping

Now that the minimum energy structure has been obtained at T = 0K, thermal energy can be added to the system. First save the optimized structure at T = 0K and remember to copy CONTCAR to POSCAR

```
$ cp CONTCAR opt_0K.pos
```

Then copy the files needed for temperature ramping to a new directory (CONTCAR, POTCAR, KPOINTS, WAVECAR, CHG, CHGCAR). The WAVECAR, CHG, and CHGCAR files are fairly large, so it is recommended to delete these files from the original geometry optimization directory once they have been copied.

The INCAR for temperature ramping is as follows

```
IBRION = 0          # standard ab initio MD (Verlet algorithm)
ALGO = Fast
```

```
SMASS = -1          # velocities are scaled each NBLOCK step to the temperature
NBLOCK = 4          # number of ionic steps between kinetic energy scaling
TEBEG = 300         # beginning and ending temperatures for scaling
TEEND = 300
ISIF = 2            # calculate stress tensor; maintain constant cell volume and shape
LREAL = A
ISMEAR = 0          # Gaussian smearing of electronic population
ISYM = 0            # symmetry not taken into account
PREC = Low
NSW = 200           # number of ionic steps
POTIM = 1.0         # time step, fs
EDIFFG = -0.0001
```

This simulation is a type of MD calculation rather than a geometry optimization. The primary difference is that IBRION = 0 specifies the velocity verlet algorithm for molecular dynamics and for IBRION = 0, POTIM specifies the time step instead of the scaling factor. Here, the nuclear coordinates are calculated from the forces and electron density determined in the previous step. In contrast, the minimum energy was found at each step in the geometry optimization. The nuclear velocities should be scaled to the indicated temperature every NBLOCK ionic steps.

Physically, the system starts at T = 0K so the nuclei are not moving. In temperature ramping, the ions are randomly "kicked" with a thermal energy of 300K (TEBEG). The nuclei are then allowed to adjust by equilibrating the energy amongst the normal modes and other ions for NBLOCK steps. After NBLOCK steps, the nuclei are again "kicked".

The temperature of the system is printed in OSZICAR. Early in the calculation, the temperature fluctuates drastically. Eventually, it should start oscillating around 300K (TEEND). The system can be considered equilibrated once the fluctuations are less than ± 10% of the specified temperature.

To check the temperature

```
$ grep 'T=' OSZICAR | awk '{print $1 , $3 }' > temperature
```

This will create the file temperature containing the step and temperature. To plot this data open gnuplot plotting utility

```
$ gnuplot
        gnuplot> plot 'temperature' u 1:2 w l
```

It may require multiple iterations to reach the equilibrated temperature. As with the geometry optimization, copy CONTCAR to POSCAR before starting another calculation. If there is no hydrogen in the system, try using a larger time step (POTIM = 2 or 3). To ramp the temperature more slowly, choose a larger NBLOCK or try starting with a lower value of TEBEG. When doing

MD calculations, there is a finite temperature associated with the ions, with random fluctuations in the forces. Because of this, there is no need for a precise description of the electronic wavefunction. Thus, set PREC = Low instead of Accurate or High. This decreases the number of plane waves used to represent the electrons and increases the speed of the calculations.

Step 1b: Molecular Dynamics Trajectory

Once the heated simulation cell has been prepared, a molecular dynamics (MD) trajectory can be computed. Again, it is best to copy the necessary files from the previous temperature ramping calculation to a new directory. The CONTCAR, POTCAR, and KPOINTS from the previous run will be needed. If the WAVECAR, CHG, and CHGCAR files are present, copy these as well. If they are not present, simply set NELMDL to a value between -8 and -15 for the first MD calculation.

The nuclear positions calculated during MD will be used in the next step to calculate the wavefunctions and NA couplings between states. Therefore, the most important files generated from the MD trajectory are the OUTCARs, as they contain the exact nuclear coordinates at each time step.

The INCAR for MD calculation

```
IBRION = 0          # standard ab initio MD (Verlet algorithm)
SMASS = -3          # microcanonical ensemble (energy is conserved)
ALGO = Fast
NSIM = 4
NELMDL = -2         # number of non-self consistent electronic steps
TEBEG = 300         # beginning and ending temperatures for scaling
TEEND = 300
ISIF = 2            # relax ions but maintain constant cell volume and shape
LWAVE = .TRUE.      # write WAVECAR file
LCHARG = .TRUE.     # write CHGCAR file
LREAL = A
PREC = Low
ISMEAR = 0          # Gaussian smearing of electronic population
ISYM = 0            # symmetry not taken into account
POTIM = 1.0         # time step, fs
NSW = 1000          # number of ionic steps
EDIFFG = -0.0001    # break condition for the ionic relaxation loop, relaxation will stop if all
                      forces are smaller than |EDIFFG|
```

During the MD simulation the nuclear coordinates are calculated from the forces and electron density determined in the previous step. However, unlike the previous step, here SMASS = -3 to specify a microcanonical ensemble in which energy is conserved. The NELMDL tag should have

a higher value if the WAVECAR file is not present, but should be changed to -2 once the first calculation has been done. To determine the length of the trajectory, simply multiply NSW by POTIM. In this case, POTIM = 1.0 and NSW = 1000 specifies a 1 ps trajectory.

In many cases a trajectory longer than 1 ps is needed. In this case, the trajectory can be separated into multiple 1 ps trajectories. Once the first 1 ps has finished, save a copy of the OUTCAR file

$ cp OUTCAR out_1_1000fs

remember to copy CONTCAR to POSCAR

$ cp CONTCAR POSCAR

Start another 1 ps calculation and repeat this for as many times as necessary to achieve the desired length. Another option is that NSW can be changed to reflect the total desired time and the trajectory can then be computed in a single calculation. This would produce a single OUTCAR file for the entire MD simulation.

Step 2: NA Couplings and Oscillator Strengths

After the ground state MD trajectory has been computed, the NA couplings between states and the oscillator strengths are then computed at each time step. The first step of this process is to extract the nuclear positions at each time step from the OUTCAR files generated during the MD simulation. Copy all of the OUTCAR files to a new directory as well as the KPOINTS and POTCAR files. INCAR, POSCAR, and WAVECAR files are generated during the simulation so these files do not need to be present.

As an example, let's consider a short 2 ps ground state trajectory calculated in 3 segments to produce 2 separate OUTCAR files (out_1_1000fs and out_1001_2000fs). The nuclear geometries at each time step during the 2000 fs run must be extracted and used to generate a POSCAR file, this way a separate POSCAR file exists for each time step.

Regardless of how the calculation is organized, the process of extracting geometries and creating POSCAR files is the same

$ cp out_1_1000fs OUTCAR

$ outcar2poscar.pl 1

The script outcar2poscar.pl is used to create new POSCAR files from the geometries listed in the OUTCAR file. The number on the command line (1) is the initial time, for out_1001_2000fs the initial time would be 1001. This script generates a series of files "p####" where #### indicates

the time step. Each p file is a POSCAR that will be used to calculate energies and wavefunctions at each step.

There are two ways of organizing the calculation:

1. Create separate directory for each OUTCAR file (i.e. geometries extracted from out_1_1000fs will be put in the directory 1_1000, geometries extracted from out_1001_2000fs will be put in the directory 1001_2000… and so on). Step 2 can then be run simultaneously from each directory. Although this method helps to speed up the calculation, it does result in a problem with the sequential numbering of output files and creates a "time gap" in the output. At the beginning of each calculation the first time step is essentially thrown away because there is no previous time to calculate wavefunction overlap. This can be fixed by repeating one of the output files for the missing time step (in this example, 1001 and 2001 will be missing, use output for 1002 and 2002 in the place of the missing output). Rigorously, the last POSCAR file from the previous directory (p1000 for 1001_2000) should be duplicated to provide the starting wavefunction for overlap calculations.

OR

2. Extract all geometries from each OUTCAR file into a single directory and run one calculation. The disadvantage here is that the calculation will take much longer, however the problem with "time gap" for separate calculations is avoided.

To run the calculation, a copy of the script ST2_tammie.pl and run_STEP2 must be in the directory. In run_STEP2, the path to the working directory must be specified. Also, the range of states to be considered must be entered and the initial and final time for the simulation. A standard choice for the range of states is homo-5 to lumo+5.

Numerous scripts are involved during this step.

state-energy-extractor.pl : Gets energy of bands from OUTCAR and add correction alpha+bet. Prints the energy for each state in energy_by_bands. After each time step the energy_by_band is appended to the file "energy" which will be used in step 3.

td_energy_extract.pl : Gets energy of bands and occupation from OUTCAR for current step and prints to energy_pop to be read in ovlap_NORM_OS_imprv2.
ovlap_NORM_OS_imprv2 : Calculates overlap of wavefunctions (NA coupling), and normalized oscillator strengths. The nonadiabatic couplings between all pairs of states are printed for each time step to the files real####, and the normalized oscillator strengths from band n to band m are printed for each time to the file os_### where #### indicates the time. The os files also contain the band n and band m, oscillator strengths, ΔEnm, population of state n and m, and the dipole moments in x, y, and z directions.

At the end, it creates new directories “OS” and “REAL”. “OS” contains oscillator strength files, the same number of files as the number of p#### files. “REAL” contains non-adiabatic couplings, one fewer files than the number of p#### files.

The file “energy” contains band energies to be used in step 3.

Step 3: NAMD

In this step, the nonadiabatic couplings computed during the previous step are used to compute hopping probabilities and the state transitions are implemented. Surface hopping requires a statistical averaging over an ensemble of trajectories, so several hundred surface hopping trajectories must be computed using different initial geometries to provide adequate configuration sampling. Usually, 500 trajectories is a sufficient number.

The “real” files from the previous step must be copied to the directory for the surface hopping calculation. It is a good idea to compress these files into a single file

$ tar –czf realfiles.tgz real*

then copy this single compressed file to the new directory. The “energy” file from the previous step is also required for the new calculation. Two more files are needed for the surface hopping calculation where the calculation parameters will be specified. The “inputv6” file specifies some important parameters

```
2       first initial condition
1000    last initial condition
7       lower state     # these numbers correspond to the number in the energy file, not the
9       upper state     actual state number in the OUTCAR
1000    namdtime        # length of the NAMD trajectory in fs
0       sh_flag         # surface hopping flag
1000    num_sh_traj     # number of surface hopping trajectories
0       sh_cont_flag
0       sh_prob_flag
1       boltz_flag
0       debug_flag
300     temperature     # temperature
1       interp_flag
40      dephas_time     # dephasing time in fs (must be an integer)
```

and the file “icond.dat” specifies the initial state for each GS geometry. Ideally, the initial state should be determined from the oscillator strength data, however the current script requires the user to choose the initial state. The “icond.dat” file is created using the “recomb_icond.f” script.

To run NAMD, submit “run_STEP3”, which uses the executable “namdv6_prop”. This executable should be created by compiling files in the directory “STEP3_files/Namdv6/New_prop/” or “STEP3_files/Namdv6/” (this second version has no dephasing time). The “Makefile” in each directory contain compiling instructions. Also see the “READMEv6” file in the “STEP3_files/Namdv6” directory.

After the calculation is completed, an output file is generated for each NAMD trajectory. The output files must be averaged to produce statistically relevant results. To do this use the get_ET.pl

$ get_ET.pl <file prefix> <# states> <namd time> <last file #>